

《2022 年 Android 平台开发技术》 大作业项目总结报告

题目：C. 航友——学习/活动伙伴匹配

2022 年 12 月

一、功能要求

本作业要求完成的功能有：

- 1、帮助用户通过各类活动组局的形式结识校园中的其他同学。
- 2、对于具有相同爱好需求的同学，进行交友匹配。
- 3、为用户提供交流校园生活的平台，分享多种形式的动态。
- 4、制定完善的社区规则与奖惩制度，保证航友 app 能够长久、健康、稳定的运行。

二、已完成任务

必做任务完成情况（20/20）

1、鉴权

- （1）用户登录：支持用户注册（设置用户名、密码、邮箱、手机号等）、登录。

2、组局

- （2）发起组局：支持用户添加主题、时间、地点、人数、tag 等必要信息发起组局。
- （3）显示已加入的局：支持用户查看所有已加入的局（包括历史组局），点击可查看详情（如必要信息、评论等）。支持退出所在局。
- （4）显示正在组的局：主页显示所有正在组的局，显示攒局进度条、局内成员性别比，点击可查看详情。
- （5）筛选组局：支持对组局进行 tag 筛选。
- （6）评价组局：用户可以对组局进行评论。

3、个人主页

- （7）显示个人信息：支持用户查看自身用户名、密码、手机号、性别等信息。
- （8）修改个人信息：支持修改个人信息。
- （9）用户搜索：支持用户根据用户名搜索其他用户。
- （10）用户爱好匹配：根据用户选择的爱好或以往加入的组局进行好友匹配
- （11）支持添加/删除好友：支持用户添加/删除好友，查看好友列表。

4、发帖

- （12）帖子编辑：支持用户编辑帖子内容。
- （13）点赞/评论指定帖子：支持用户对指定帖子点赞/评论，在帖子上显示点赞、评论数。
- （14）举报：支持用户对不合规的发帖内容进行举报。

- (15) 查看贴主个人主页：支持用户通过点击头像查看贴主个人主页。
- (16) 匿名动态分享：支持匿名发帖/评论/投诉/点赞（注意恶意发帖、评论、投诉）。
- (17) 收藏帖子：用户可以收藏帖子，并支持查看所有已收藏的帖子。

5、其他

- (18) 自定义设置：支持用户修改主题风格、消息通知形式。
- (19) 公告：支持组局页滚动显示公告。
- (20) 展示使用手册：为用户提供本应用程序的使用说明，详细介绍组局、发帖等功能。

选做任务完成情况（15/18）

1、组局

- (1) 关注与被关注：可以查看关注自己的人与自己关注的人。
- (2) 局收款显示支付的人与未支付的人，并展示头像。
- (3) 局收款：支持用户发起局收款，分摊活动费用。
- (4) 显示局中的人：以横向列表的形式显示局中用户的头像与用户名。
- (5) 浮动窗评价：局结束后可以浮动窗的形式对局进行评价。

2、发帖

- (6) 展示当前热帖：根据算法，综合考虑浏览、点赞、评论等数量展示当前热帖。
- (7) 发帖规则：支持用户在发帖页查看发帖规则、恶意行为的惩罚措施。

3、个人主页

- (8) 显示消息：显示用户消息列表：点赞/评论/聊天/组局提醒/系统提示。
- (9) 消息检索：支持用户在消息队列中进行关键字检索。
- (10) 关键词屏蔽：用户可以输入自定义短语来屏蔽部分组局和帖子。
- (11) 头像替换：用户可从本地图库中选择图片作为用户头像。
- (12) 在用户列表处显示用户称号。

4、其他

- (13) 反馈：对应用程序进行评价/反馈。
- (14) 帮助：提供常见问题的使用说明和帮助。
- (15) 根据用户活跃度/诚信度授予成就：根据用户的活跃度/诚信度（如一定时间内不可以退出组局，违规发贴次数等）授予成就。

三、总体设计方案

1、注册登录

- 实现逻辑：接入数据库的 user 表，进行交互。注册时填入所需基本信息插入 user 表中。登录时检查账号密码是否匹配，匹配成功则成功登录，否则返回相应错误信息。

- 核心代码：

```
private void register() throws SQLException {
    .....
    if(rs.next()) {
        System.out.println("已注册");
        Toast.makeText(this, "账号已被注册，请重新输入", Toast.LENGTH_SHORT).show();
        et_account.setText("");
    } else if(password.equals("")) {
        Toast.makeText(this, "请输入密码", Toast.LENGTH_SHORT).show();
    } else if (phone.equals("")) {
        Toast.makeText(this, "请输入手机号", Toast.LENGTH_SHORT).show();
    } else {
        flag1.set(false);
        new Thread() -> {
            try {
                Connection conn = MySqlConnection.getConnection();
                String s = "INSERT INTO user(account, username, description, phone, gender, password)
VALUES(?, ?, ?, ?, ?, ?)";
                PreparedStatement p = conn.prepareStatement(s);
                String description = "这个人很懒，什么都没有留下";
                p.setString(1, account);
                p.setString(2, username);
                p.setString(3, description);
                p.setString(4, phone);
                p.setString(5, gender);
                p.setString(6, password);
                p.executeUpdate();
                flag1.set(true);
            } catch (InterruptedException | SQLException e) {
                e.printStackTrace();
            }
        }.start();
    }
    .....
}
```

- 具体实现图：



2、查看正在组的组局并根据标签进行筛选

- 实现逻辑：默认显示所有可加入的组局，点击相应标签时筛选出对应类别的组局显示。
- 核心代码：以显示电影组局为例

```
data.forEach(i -> {
    if (i.get("groupType").equals("电影")) {
        cdata.add(i);
        idLists.add(Integer.parseInt(Objects.requireNonNull(i.get("id")).toString()));
    }
});
GroupCardAdapter adapter = new GroupCardAdapter(GroupActivity.this, cdata);
ListView groupCards = findViewById(R.id.group_cards);
groupCards.setAdapter(adapter);
adapter.notifyDataSetChanged();
groupCards.setOnItemClickListener(new AdapterView.OnItemClickListener() {

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Bundle bundle=new Bundle();
        bundle.putInt("id", idLists.get(position));
        Intent intent =new Intent(GroupActivity.this, GroupCardDetailActivity.class);
        intent.putExtras(bundle);
        startActivity(intent);
        finish();
    }
})
```

- 具体实现图：



3、组局卡片显示组局名称、组局发起人姓名、组局日期、组局进度条，方便用户快速了解组局概况

- 实现逻辑：为 listview 中的 item 设计一个 big_card.xml 文件来控制每一个 item 的外观和显示的信息，并通过一个继承 BaseAdapter 类的 GroupCardAdapter 类来进行后端控制。

- 核心代码：GroupCardAdapter 中的 getView() 方法

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    convertView = LayoutInflater.from(ctx).inflate(R.layout.big_card, parent, false);
    TextView tv_groupName = convertView.findViewById(R.id.big_card_groupName);
    tv_groupName.setText(Objects.requireNonNull(data.get(position).get("groupName")).toString());
    TextView tv_groupType = convertView.findViewById(R.id.big_card_groupType);
    tv_groupType.setText(Objects.requireNonNull(data.get(position).get("groupType")).toString());
    TextView tv_groupDate = convertView.findViewById(R.id.big_card_groupDate);
    tv_groupDate.setText(Objects.requireNonNull(data.get(position).get("groupDate")).toString());
    TextView tv_groupInitiator = convertView.findViewById(R.id.big_card_groupInitiator);

    tv_groupInitiator.setText(Objects.requireNonNull(data.get(position).get("groupInitiator")).toString());
    ProgressBar progressBar = convertView.findViewById(R.id.progress_bar);
    int groupMaleExpectedNum = (int)
    Objects.requireNonNull(data.get(position).get("groupMaleExpectedNum"));
    int groupMaleNowNum = (int) Objects.requireNonNull(data.get(position).get("groupMaleNowNum"));
    int groupFemaleExpectedNum = (int)
    Objects.requireNonNull(data.get(position).get("groupFemaleExpectedNum"));
    int groupFemaleNowNum = (int) Objects.requireNonNull(data.get(position).get("groupFemaleNowNum"));
    progressBar.setMax(groupMaleExpectedNum + groupFemaleExpectedNum);
    progressBar.setProgress(groupMaleNowNum);
    progressBar.setSecondaryProgress(groupMaleNowNum + groupFemaleNowNum);
    return convertView;
}
```

- 具体实现图：



4、加入/退出组局

- 实现逻辑：点击入局，当符合人数性别要求时，即可加入组局；再次点击退出即可退出该组局，作为局主即组局发起者不可以退出组局。

- 核心代码：

```
private void addOrExitGroup() throws SQLException {
    if(bt_add_or_exit.getText().equals("入局")) {
        flag1.set(false);
        new Thread() -> {
            try {
                Connection connection = MySqlConnection.getConnection();
                String sql = "insert into user_group_relation(userId, groupId) values (?, ?)";
                PreparedStatement ps = connection.prepareStatement(sql);
                ps.setString(1, userId);
                ps.setString(2, String.valueOf(groupId));
                ps.executeUpdate();
                flag1.set(true);
            } catch (InterruptedException | SQLException e) {
                e.printStackTrace();
            }
        }.start();
    }
    .....
}
```

```
if(resultSet.get().next()) {
    /* 局主 */
    Toast.makeText(this, "您作为局主不可以退出哦", Toast.LENGTH_SHORT).show();
    return;
} else {
    flag1.set(false);
    new Thread() -> {
        try {
            Connection connection = MySqlConnection.getConnection();
            String sql = "delete from user_group_relation where userId=? and groupId=?";
            PreparedStatement ps = connection.prepareStatement(sql);
            ps.setString(1, String.valueOf(userId));
            ps.setString(2, String.valueOf(groupId));
            resultSet.set(ps.executeQuery());
            flag1.set(true);
        } catch (InterruptedException | SQLException e) {
            e.printStackTrace();
        }
    }.start();
    while (!flag1.get());
    .....
}
}
```

- 具体实现图：



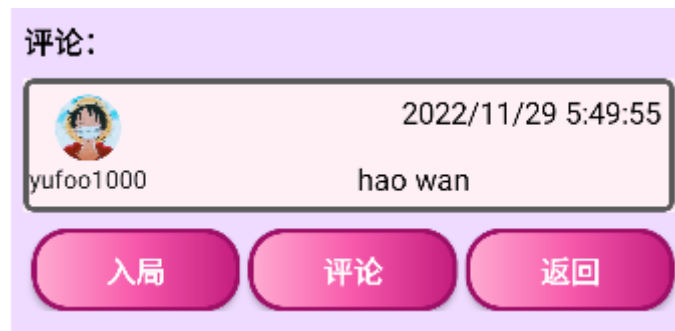
5、评论组局

- 实现逻辑：主页面传入 id 参数指定帖子，通过设置浮动窗获取用户评论并展示。
- 核心代码：

```
private void commit() {
    EditText et_comment = findViewById(R.id.group_comment_input);
    String comment = et_comment.getText().toString();
    new Thread(() -> {
        try {
            Connection conn = MySqlConnection.getConnection();
            String s = "insert into group_comment(groupId, userId, createdAt, comment) values(?, ?, ?, ?)";

            PreparedStatement p = conn.prepareStatement(s);
            p.setString(1, String.valueOf(groupId));
            p.setString(2, String.valueOf(userId));
            Calendar c = Calendar.getInstance();
            int year = c.get(Calendar.YEAR);
            int month = c.get(Calendar.MONTH);
            int date = c.get(Calendar.DATE);
            int hour = c.get(Calendar.HOUR_OF_DAY);
            int minute = c.get(Calendar.MINUTE);
            int second = c.get(Calendar.SECOND);
            p.setString(3, year + "/" + month + "/" + date + " " + hour + ":" + minute + ":" + second);
            p.setString(4, comment);
            p.executeUpdate();
        } catch (InterruptedException | SQLException e) {
            e.printStackTrace();
        }
    }).start();
    Toast.makeText(context, "评论成功", Toast.LENGTH_SHORT).show();
    onBackPressed();
}
```


- 具体实现图：



6、局主发起局收款

- 实现逻辑：连接数据库 group_money 表进行交互，获取组局号和支付金额后插入表中创建一个群收款记录。

- 核心代码：

```
private void sendGatherMoneyRequest() {
    System.out.println("发起局收款");
    EditText et_money = findViewById(R.id.group_card_detail_gather_money_money);
    String money = et_money.toString();
    if(money.equals("")) {
        Toast.makeText(this, "请输入每个人需要支付的金额", Toast.LENGTH_SHORT).show();
    } else {
        new Thread() -> {
            try {
                Connection connection = MySQLConnector.getConnection();
                String sql = "insert into group_money(groupId, money, createdAt) values(?, ?, ?)";
                PreparedStatement ps = connection.prepareStatement(sql);
                ps.setString(1, String.valueOf(groupId));
                ps.setString(2, money);
                Calendar c = Calendar.getInstance();
                int year = c.get(Calendar.YEAR);
                int month = c.get(Calendar.MONTH);
                int date = c.get(Calendar.DATE);
                int hour = c.get(Calendar.HOUR_OF_DAY);
                int minute = c.get(Calendar.MINUTE);
                int second = c.get(Calendar.SECOND);
                ps.setString(3, year + "/" + month + "/" + date + " " + hour + ":" + minute + ":" +
second);
                ps.executeUpdate();
            } catch (InterruptedException | SQLException e) {
                e.printStackTrace();
            }
        }.start();
        Toast.makeText(this, "成功发起群收款", Toast.LENGTH_SHORT).show();
        Intent intent = new Intent();
        intent.setClass(this, GroupActivity.class);
        startActivity(intent);
    }
}
```

- 具体实现图：





7、创建组局

- 实现逻辑：接入数据库的 group 表，进行交互。根据用户填写的信息插入到 group 表中创建一个新的组局。
- 核心代码：

```
ArrayList<String> types = new ArrayList<>(Arrays.asList("学习", "运动", "聚餐", "旅行", "拼单", "电影", "游戏", "其他"));
String groupType;
if (rb_type0.isChecked()) {
    groupType = types.get(0);
} else if (rb_type1.isChecked()) {
    groupType = types.get(1);
} else if (rb_type2.isChecked()) {
    groupType = types.get(2);
} else if (rb_type3.isChecked()) {
    groupType = types.get(3);
} else if (rb_type4.isChecked()) {
    groupType = types.get(4);
} else if (rb_type5.isChecked()) {
    groupType = types.get(5);
} else if (rb_type6.isChecked()) {
    groupType = types.get(6);
} else {
    groupType = types.get(7);
}

while (!flag1.get());
int cnt = 0;
while(resultSet.get().next()) {
    cnt++;
}
flag1.set(false);
int finalCnt = cnt;
```

```

new Thread(() -> {
    try {
        Connection connection = MysqlConnector.getConnection();
        String sql = "update user_group set groupFemaleNowNum=? where id=?";
        PreparedStatement ps = connection.prepareStatement(sql);
        ps.setString(1, String.valueOf(finalCnt));
        ps.setString(2, String.valueOf(groupId));
        ps.executeUpdate();
        flag1.set(true);
    } catch (InterruptedException | SQLException e) {
        e.printStackTrace();
    }
}).start();
}
System.out.println("新组添加成功");
Intent intent = new Intent();
intent.setClass(AddGroupPageActivity.this, GroupActivity.class);
startActivity(intent);

```

- 具体实现图：

组局名称
请简要输入组局名称，如玩狼人杀.....

组局详细内容
请输入组局详细内容，如【考完期末考试的晚上】在【新北负一楼】玩狼人杀

组局类型

☒ 学习 ☐ 运动 ☐ 聚餐 ☐ 旅行
☐ 拼单 ☐ 电影 ☐ 游戏 ☐ 其他

组局时间

1969	12	08
1970	1	09
1971	2	10

分别输入男生、女生人数

填好信息后点击确定即可创建组局

确定 返回

8、显示我加入的组局、我发起的组局

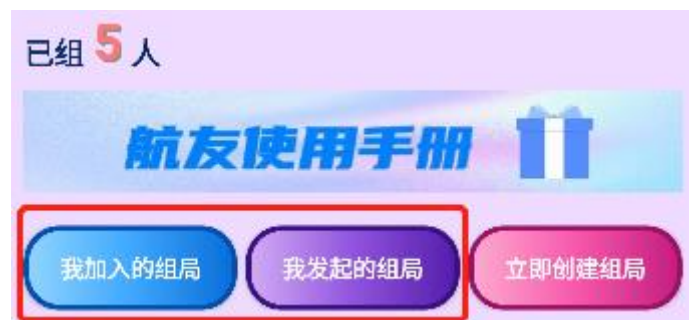
- 实现逻辑：我加入的组局可以通过 user_group_relation 中判断有无特定 userId 与 groupId；我发起的组局可以通过 use_group 中的 groupInitiator 判断。

- 核心代码:

```
new Thread(() -> {
    try {
        Connection connection = MySqlConnection.getConnection();
        String sql = "select * from user_group where groupInitiator=?";
        PreparedStatement ps = connection.prepareStatement(sql);
        SharedPreferences sp = getSharedPreferences("login", Context.MODE_PRIVATE);
        String account = sp.getString("account", "defaultValue");
        ps.setString(1, account);
        resultSet.set(ps.executeQuery());
        flag1.set(true);
    } catch (InterruptedException | SQLException e) {
        e.printStackTrace();
    }
}).start();

new Thread(() -> {
    try {
        Connection connection = MySqlConnection.getConnection();
        String sql = "select * from user_group, user, user_group_relation where user.account=?
and user.id=user_group_relation.userId and user_group.id=user_group_relation.groupId";
        PreparedStatement ps = connection.prepareStatement(sql);
        SharedPreferences sp = getSharedPreferences("login", Context.MODE_PRIVATE);
        String account = sp.getString("account", "defaultValue");
        ps.setString(1, account);
        resultSet.set(ps.executeQuery());
        flag1.set(true);
    } catch (InterruptedException | SQLException e) {
        e.printStackTrace();
    }
}).start();
```

- 具体实现图:



9、页面切换

- 实现逻辑: 设置固定按钮, 绑定响应事件, 同时特判是否原地跳转, 原地跳转无需处理。
- 核心代码:

```

        <ImageView
            android:layout_width="30dp"
            android:layout_height="30dp"
            android:layout_weight="1"
            android:layout_gravity="center"
            android:src="@drawable/group_icon">
        </ImageView>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:layout_gravity="center"
            android:text="组局">
        </TextView>
    </LinearLayout>

    private void jumpToGroup() {
        Intent intent = new Intent();
        intent.setClass(HomePageActivity.this, GroupActivity.class);
        startActivity(intent);
    }

```

- 具体实现图：



10、发帖和发帖规则

- 实现逻辑：连接数据的 post 表和 post_comment 进行交互。在树洞主页面会按时间顺序显示所有的帖子。点击发帖按钮会跳转到发帖界面。填写标题和内容之后点击发布，会在数据库的 post 表中新增一行记录。
- 核心代码：提交发帖的 commit() 方法

```

String userId = resultSet2.get().getString("id");

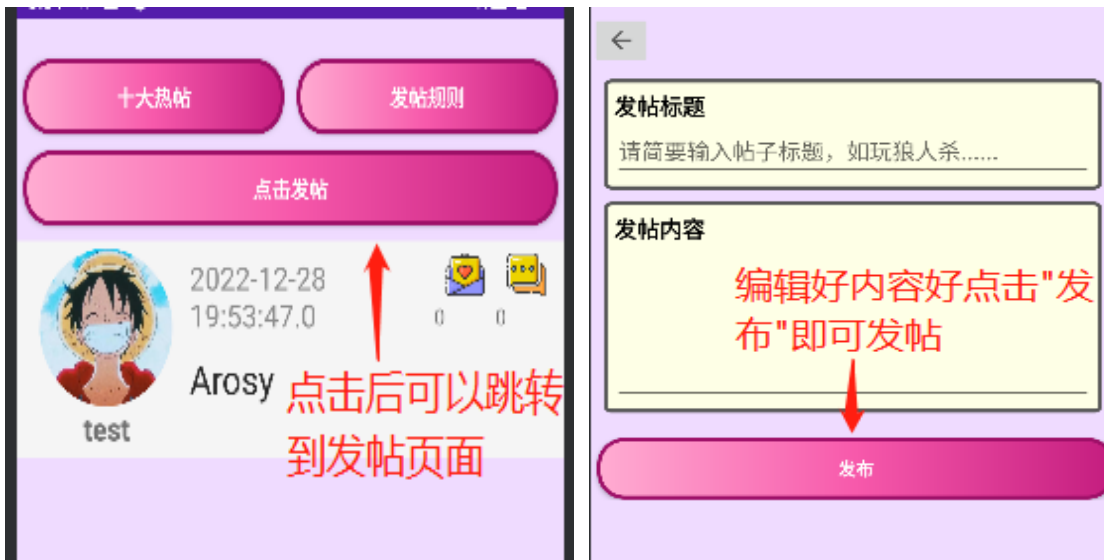
flag2.set(false);
new Thread(() -> {
    try {
        Connection connection = MySqlConnection.getConnection();
        String sql = "insert into post(postName, postText, userId, likeNum, " +
            "commentNum, reportNum) values (?, ?, ?, 0, 0, 0)";
        PreparedStatement ps = connection.prepareStatement(sql);
        ps.setString(1, postName);
        ps.setString(2, postText);
        ps.setString(3, userId);
        ps.executeUpdate();
        flag2.set(true);
    } catch (InterruptedException | SQLException e) {
        e.printStackTrace();
    }
}).start();

```

```
while (!flag2.get());

System.out.println("帖子发布成功");
Intent intent = new Intent();
intent.setClass(TreeHolePostCreate.this, TreeHoleActivity.class);
startActivity(intent);
```

- 具体实现图：



11、十大热帖

- 实现逻辑：连接数据库的 post 表和 post_comment 表根据热帖的计算标准对所有帖子进行排序后选取前十名进行展示。同时点击某一个帖子后也可直接进入该贴子的详情页。

- 核心代码：

```
new Thread(() -> {
    try {
        Connection connection = MySqlConnection.getConnection();
        String sql = "select * from post order by likeNum DESC,commentNum DESC";
        PreparedStatement ps = connection.prepareStatement(sql);
        resultSet.set(ps.executeQuery());
        flag1.set(true);
    } catch (InterruptedException | SQLException e) {
        e.printStackTrace();
    }
}).start();
while(!flag1.get());
```

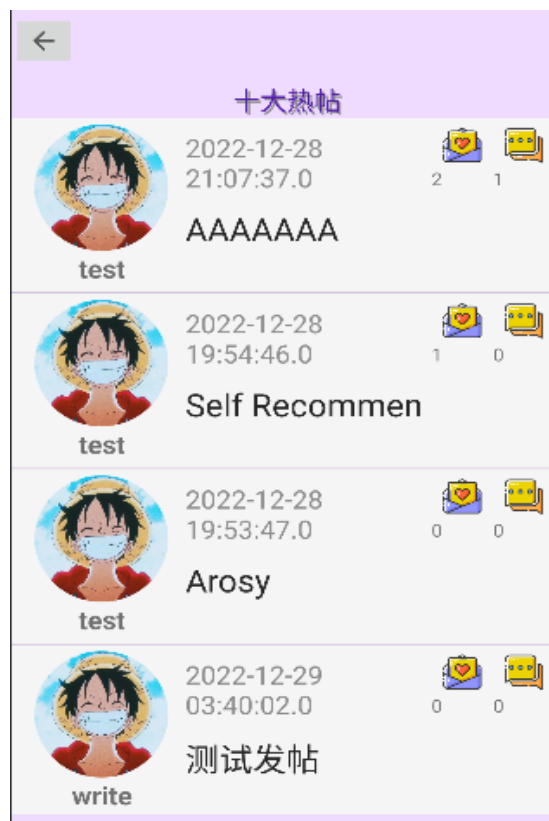
```

while(!flag2.get());
resultSet2.get().first();

item.put("postUser", resultSet2.get().getString("username"));
item.put("id", resultSet.get().getString("id"));
idLists.add(Integer.parseInt(resultSet.get().getString("id")));
data.add(item);
count++;
if(count==10) break;

```

- 具体实现图：



12、帖子缩略卡片

- 实现逻辑：为 listview 中的 item 设计一个 post_card.xml 文件来控制每一个 item 的外观和显示的信息，并通过一个继承 BaseAdapter 类的 PostCardAdapter 类来进行后端控制。

- 核心代码：PostCardAdapter 中的 getView() 方法

```

public View getView(int position, View convertView, ViewGroup parent) {
    convertView = LayoutInflater.from(ctx).inflate(R.layout.post_card, parent, false);
    TextView tv_postName = convertView.findViewById(R.id.postCardPostName);
    tv_postName.setText(Objects.requireNonNull(data.get(position).get("postName")).toString());
    TextView tv_postTime = convertView.findViewById(R.id.postCardPostTime);
    tv_postTime.setText(Objects.requireNonNull(data.get(position).get("postTime")).toString());
    TextView tv_postUser = convertView.findViewById(R.id.postCardUserName);
    tv_postUser.setText(Objects.requireNonNull(data.get(position).get("postUser")).toString());
    TextView tv_numOfLike = convertView.findViewById(R.id.num_of_like);
    tv_numOfLike.setText(Objects.requireNonNull(data.get(position).get("numOfLike")).toString());
    TextView tv_numOfComment = convertView.findViewById(R.id.num_of_comment);
    tv_numOfComment.setText(Objects.requireNonNull(data.get(position).get("numOfComment")).toString());

    return convertView;
}

```

- 具体实现图：



13、点赞、评论、举报帖子

- 实现逻辑：连接数据库的 post 表进行交互。对于点赞和举报，直接对数据库中 post 表的对应字段进行修改。同时控制了刷赞等不良行为，奇数次点赞是会将点赞数加一，偶数次点赞时会前一次的点赞取消。而对于评论行为，则跳转到新的页面填写评论。

- 核心代码：

```

private void reportPost() throws SQLException {
    AtomicReference<ResultSet> resultSet = new AtomicReference<>();
    AtomicBoolean flag1 = new AtomicBoolean(false);
    new Thread() -> {
        try {
            Connection connection = MySqlConnection.getConnection();
            String sql = "select * from post where id=?";
            PreparedStatement ps = connection.prepareStatement(sql);
            ps.setString(1, String.valueOf(id));
            resultSet.set(ps.executeQuery());
            flag1.set(true);
        } catch (InterruptedException | SQLException e) {
            e.printStackTrace();
        }
    }.start();
    while(!flag1.get());
}

```

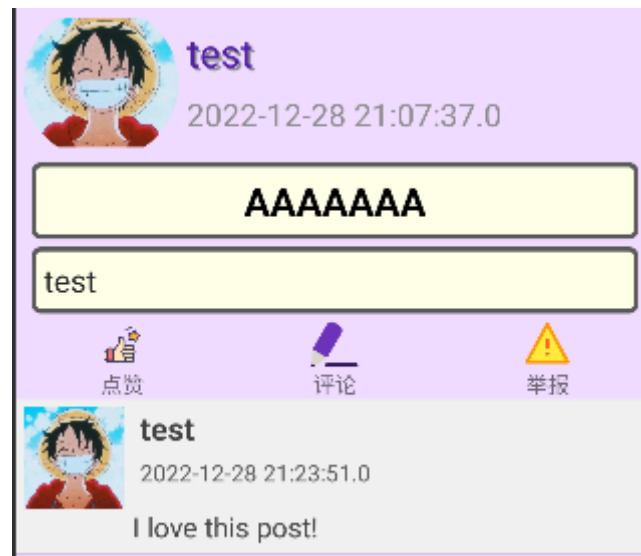


```

resultSet.get().first();
int reportNum = resultSet.get().getInt("reportNum");
if(!reportState){
    reportNum+=1;
    reportState = true;
} else {
    reportNum-=1;
    reportState = false;
}
int finalReportNum = reportNum;

```

- 具体实现图:



14、个人主页显示个人信息、修改个人信息

- 实现逻辑: 连接数据库的 user 表, 进行获取信息并展示即可。
- 核心代码:

```

private void initView() throws SQLException {
    SharedPreferences sp = getSharedPreferences("login", Context.MODE_PRIVATE);
    String account = sp.getString("account", "defaultValue");
    String headPortrait = sp.getString("head_portrait", "");
    AtomicBoolean flag1 = new AtomicBoolean(false);
    AtomicReference<ResultSet> resultSet = new AtomicReference<>();
    flag1.set(false);
    new Thread(() -> {
        try {
            Connection connection = MySqlConnection.getConnection();
            String sql = "select * from user where account=?";
            PreparedStatement ps = connection.prepareStatement(sql);
            ps.setString(1, account);
            resultSet.set(ps.executeQuery());
            flag1.set(true);
        } catch (InterruptedException | SQLException e) {
            e.printStackTrace();
        }
    }).start();
}

```

```
while (!flag1.get());
resultSet.get().next();
TextView username = findViewById(R.id.home_page_username);
username.setText(resultSet.get().getString("username"));
TextView description = findViewById(R.id.home_page_description);
description.setText(resultSet.get().getString("description"));
```

- 具体实现图：

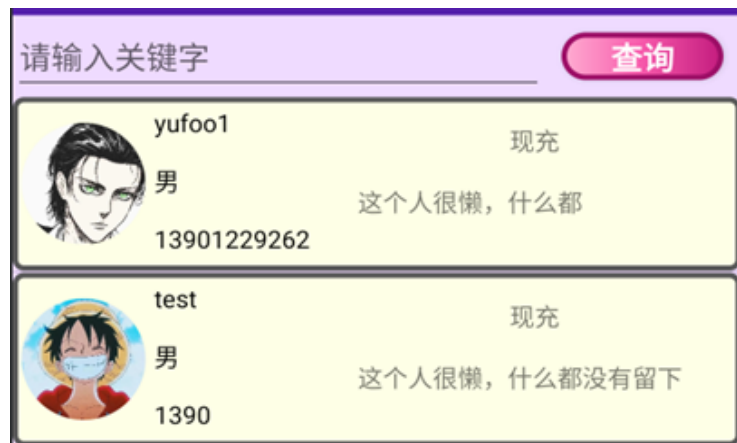


15、搜索用户

- 实现逻辑：连接数据库的 user 表进行交互。首先显示所有的用户，输入用户的用户名进行用户查找，来实现用户基本信息的查看。
- 核心代码：

```
if(type.equals("showAll")) {
    new Thread() -> {
        try {
            Connection connection = MySqlConnection.getConnection();
            String sql = "select * from user ,user_head_portrait where user.id=user_head_portrait.userId and user.username like '%" + keyword + "%'";
            PreparedStatement ps = connection.prepareStatement(sql);
            resultSet.set(ps.executeQuery());
            flag1.set(true);
        } catch (InterruptedException | SQLException e) {
            e.printStackTrace();
        }
    }.start();
} else if (type.equals("showFollowers")) {
    new Thread() -> {
        try {
            Connection connection = MySqlConnection.getConnection();
            String sql = "select * from user, user_head_portrait inner join follow on user.account=follow.followerAccount and user.account=? and user.id=user_head_portrait.userId and user.username like '%?%'";
            PreparedStatement ps = connection.prepareStatement(sql);
            ps.setString(1, account);
            ps.setString(2, keyword);
            resultSet.set(ps.executeQuery());
            flag1.set(true);
        } catch (InterruptedException | SQLException e) {
            e.printStackTrace();
        }
    }.start();
}
```

- 具体实现图：

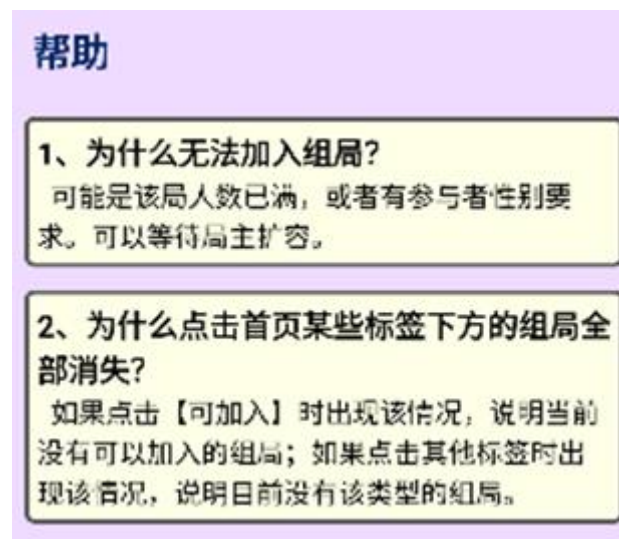


16、查看帮助

- 实现逻辑：点击“帮助”按钮，直接跳转到帮助页面展示帮助信息。
- 核心代码：

```
private void jumpToHelp() {  
    Intent intent = new Intent();  
    intent.setClass(this, HelpActivity.class);  
    startActivity(intent);  
}
```

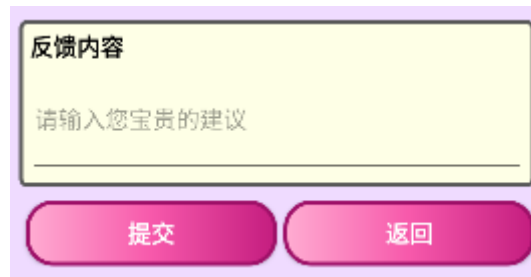
- 具体实现图：



17、提交反馈

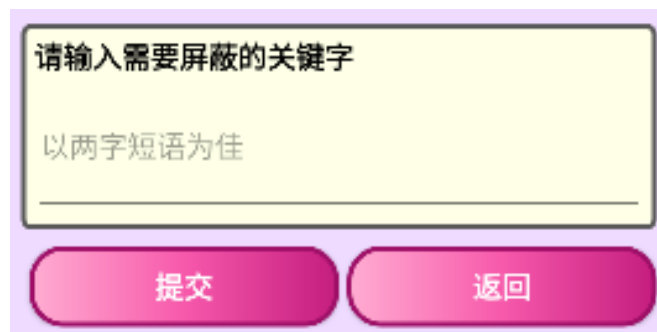
- 实现逻辑：点击“反馈”按钮跳转到反馈填写页面，填写反馈后点击提交，添加到数据库的 feedback 表中进行记录。

- 核心代码：代码原理与发评论相似，故不再赘述。
- 具体实现图：

A feedback form with a yellow background and a purple border. It has a title "反馈内容" (Feedback Content) and a placeholder text "请输入您宝贵的建议" (Please enter your valuable suggestions). Below the text input area are two pink buttons: "提交" (Submit) and "返回" (Return).

18、关键字屏蔽

- 实现逻辑：在索引条件中加入特判条件，规避掉具备相似关键字的局或帖子。
- 核心代码：代码原理与组局展示相似。
- 具体实现图：

A keyword blocking form with a yellow background and a purple border. It has a title "请输入需要屏蔽的关键字" (Please enter the keywords you want to block) and a placeholder text "以两字短语为佳" (Two-character phrases are best). Below the text input area are two pink buttons: "提交" (Submit) and "返回" (Return).

19、自定义设置更换主题

- 实现逻辑：我们设置了八种主题颜色可供切换，点击“更换主题”按钮跳转到更换主题界面，选择一个喜欢的主题确认即可完成更换。
- 核心代码：

```
private void commit() {  
    RadioButton rb_type0 = findViewById(R.id.change_theme_type_0);  
    RadioButton rb_type1 = findViewById(R.id.change_theme_type_1);  
    RadioButton rb_type2 = findViewById(R.id.change_theme_type_2);  
    RadioButton rb_type3 = findViewById(R.id.change_theme_type_3);  
    RadioButton rb_type4 = findViewById(R.id.change_theme_type_4);  
    RadioButton rb_type5 = findViewById(R.id.change_theme_type_5);  
    RadioButton rb_type6 = findViewById(R.id.change_theme_type_6);  
    RadioButton rb_type7 = findViewById(R.id.change_theme_type_7);  
    SharedPreferences sp = getSharedPreferences("theme", Context.MODE_PRIVATE);  
    SharedPreferences.Editor editor = sp.edit();  
    if(rb_type0.isChecked()) {  
        editor.putInt("theme", 0);  
    } else if(rb_type1.isChecked()) {  
        editor.putInt("theme", 1);  
    } else if(rb_type2.isChecked()) {  
        editor.putInt("theme", 2);  
    } else if(rb_type3.isChecked()) {  
        editor.putInt("theme", 3);  
    } else if(rb_type4.isChecked()) {  
        editor.putInt("theme", 4);  
    }  
}
```

```
    } else if(rb_type5.isChecked()) {
        editor.putInt("theme", 5);
    } else if(rb_type6.isChecked()) {
        editor.putInt("theme", 6);
    } else {
        editor.putInt("theme", 7);
    }
    editor.apply();
    Intent intent = new Intent();
    intent.setClass(this, HomePageActivity.class);
    startActivity(intent);
}
```

• 具体实现图：

☒ 默认

☐ 浅蓝

☐ 浅红

☐ 浅黄

☐ 深蓝

☐ 深红

☐ 深黄

☐ 灰色

提交

返回



20、系统消息

- 实现逻辑：点击“系统消息”按钮跳转到系统消息页面，可以查收并阅读所有的系统消息
- 核心代码：与其他展示类、无交互页面相似，故不再赘述。
- 具体实现图：



21、展示使用手册

- 实现逻辑：在组局主页面点击“航友使用手册”按钮跳转到航友使用手册页面，会详细展示航友使用手册。
- 核心代码：

```
private void jumpToGuide() {  
    Intent intent = new Intent();  
    intent.setClass(GroupActivity.this, GuideActivity.class);  
    startActivity(intent);  
}
```

- 具体实现图：

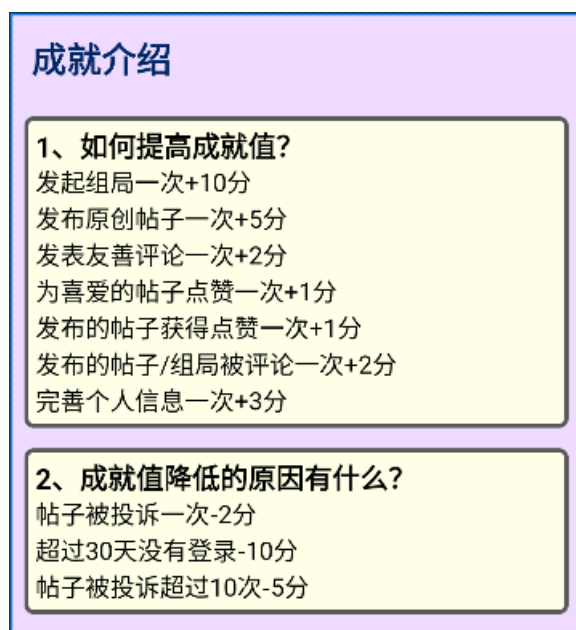


22、成就规则

- 实现逻辑：点击“成就介绍”按钮跳转到成就介绍页面，可以查看所有的成就，已经

达成相应成就所需的要求。

- 核心代码：与其他展示类、无交互页面相似，故不再赘述。
- 具体实现图：



四、创新之处

1、支持更换主题颜色

我们设置了八种主题颜色可供切换，点击“更换主题”按钮跳转到更换主题界面，选择一个喜欢的主题确认即可完成更换。自定义的设置能够给用户带来更好的使用体验。

```
switch (theme) {  
    case 0: findViewById(R.id.change_theme_page).setBackgroundResource(R.color.purple_2); break;  
    case 1: findViewById(R.id.change_theme_page).setBackgroundResource(R.color.blue_2); break;  
    case 2: findViewById(R.id.change_theme_page).setBackgroundResource(R.color.red_2); break;  
    case 3: findViewById(R.id.change_theme_page).setBackgroundResource(R.color.yellow_2); break;  
    case 4: findViewById(R.id.change_theme_page).setBackgroundResource(R.color.blue_6); break;  
    case 5: findViewById(R.id.change_theme_page).setBackgroundResource(R.color.red_4); break;  
    case 6: findViewById(R.id.change_theme_page).setBackgroundResource(R.color.yellow_6); break;  
    case 7: findViewById(R.id.change_theme_page).setBackgroundResource(R.color.gray_2); break;  
}
```

更详细的实现方案可以参见上方第三部分总体设计方案。

2、组局不止局限于基本信息，允许进行动态的评论和局收款

更详细的实现方案可以参见上方第三部分总体设计方案。

3、恰当的提示和指引

一款 APP 不仅要功能丰富优秀，还应该便于用户使用，我们的航友系统在用户登录、发起局收款等处使用悬浮窗的形式对用户当前的操作进行提示，如“登录成功”、“请输入每个人应该支付的金额”等，向用户提供了恰当的提示和指引，方便用户的使用


```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    this setContentView(R.layout.group_comment_edit);
    Window dialogWindow = this.getWindow();
    WindowManager m = context.getWindowManager();
    dialogWindow.setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));
    Display d = m.getDefaultDisplay();
    WindowManager.LayoutParams p = dialogWindow.getAttributes();
    p.height = (int) (d.getHeight() * 0.55);
    p.width = (int) (d.getWidth() * 0.8);
    dialogWindow.setAttributes(p);

    Button bt_commit = findViewById(R.id.group_comment_commit);
    bt_commit.setOnClickListener(v -> commit());
    Button bt_return = findViewById(R.id.group_comment_return);
    bt_return.setOnClickListener(v -> clockDialog());

    this.setCancelable(true);
}

```

浮动窗需要手动实现类继承 Dialog，定义尺寸，通过 show() 方法浮现优化点在于浮动窗动画的实现。

4、展示十大热帖，能够根据算法事实更新，引导用户关注焦点问题

5、向用户提供使用手册、发帖规则、帮助等说明

6、给用户提供了反馈的窗口

7、使用 mysql 数据库存储二进制图像文件并实现连接。

本组项目的图片没有存在阿里云 OSS 或者 MinIO 中，而是选择将图片文件转成 BitMap 类型，再将 BitMap 转化为 Byte 数组，最后从 Byte 数组转化为 String 类型存储于数据库当中。当需要取出图片并展示时，只需要执行倒序操作即可。将 OSS 存储于 Mysql 当中统一了查询，降低了数据库的维护难度。

五、项目总结

本次项目开发总体达到了预期，对于先前计划的需求基本都进行了实现或转型，最终实现了一个很精致的项目。同时，在开发过程中我深刻体会了自研 APP 与商用 APP 的查询优化的差距。本组数据库部署服务器云端，使用 Mysql，在查询过程中尽可能少地使用了函数等可能降低查询速度的语句，但实际上查询速度仍然较慢，用户体验稍有降低。经分析认为可能与服务器配、驱动版本、并发访问数据库加锁有关。就访问加锁，本组主要通过手动实现信号量，但将部分本可以并行的操作强制为串行，该处具有优化空间。

六、课程学习总结

1、课程收获和难点分析（是否有安卓基础，做完这个大作业自我感觉是否有提高等其他收获，本次项目感觉最困难的地方在哪里）

为了更好地实现团队协作，减少代码合并的难度和时间，我们在开发前便约定好了项目的大致结构、数据的存储方式等，并按照模块进行了分工。大家遵循高内聚、低耦合的原则，并写了适当的注释让其他人更容易看懂代码。

android 系统是目前运用最广泛的手机操作系统，历经多年的发展，仍然具有绝对的优势。本学期中我们在课堂上对于 android 开发的基础知识进行了学习，并在课后配以了相应的实践，加深了对 android 操作系统和 android 开发的基础。

我们团队的 4 位成员中，只有 1 位有过安卓开发基础，其余 3 位同学是首次接触安卓开发。在这次开发过程中，我们实际应用了课上所学的 xml 和 android 开发的基础知识，前端的同学学会了使用各种安卓组件，如 ListView、LinearLayout、MaterialCardView 等，并能够使用 drawable 对组件进行美化；后端的同学学习到了很多相关技术，最初的开发使用的是 sqlite 数据库，比较方便好用，但是仅限于本地，这与我们的设计理念是背道而驰的，因此再开发的中后期我们将数据库改为了部署在远端服务器上的 mysql，迁移的过程相当曲折，代码的修改量也较大。同时我们也充分运用了安卓开发的各种知识与理念，比如各类的 UI 控件与后端的绑定，以及 activity、service、content provider、broadcast receiver 这四大组件的应用。尤其是 activity 组件在我们的开发中被大量使用，对它生命周期等各类特性有了更深刻的理解。

总的来说，我们在面对问题时通过查阅资料、相互讨论、共同研究等方式解决问题，实际提高了我们的开发能力和解决实际问题满足用户需求的能力。感觉这次开发实践意义深远。

2、教师授课评价（老师上课过程的一些建议，以及希望老师之后能够介绍一些什么东西）

在课堂上老师对安卓的历史，以及安卓开发的整个流程进行了详细的介绍，为我们完成安卓开发大作业指明了方向，提供了强有力的引导。希望能够在课程中穿插一些小项目的自愿实现练习，可以让我们循序渐进的了解安卓开发的方式，实际提高自己的能力。

3、助教评价

本门课程的助教认真负责，能够及时解答同学们的疑惑，及时发布任务，安排好答

辩、实践项目、成品展示等环节。

4、当前课程教授内容评价与课程进一步改进建议

可以增加一些小项目的自由练习，让我们能够循序渐进一部分一部分地进行练习，提高自己的开发能力。

七、资料

[1] 安卓开发常用的框架

https://m.baidu.com/sf?pd=topone_trustpaper&resource_id=55858&word=%E5%AE%89%E5%8D%93%E5%BC%80%E5%8F%91%E5%B8%B8%E7%94%A8%E7%9A%84%E6%A1%86%E6%9E%B6&ext=%7B%22url%22%3A%22https%3A%5C%2F%5C%2Fwww.apicloud.com%5C%2Fm%5C%2FblogDetails%5C%2F8579%22%2C%22srcid%22%3A%2228420%22%2C%22jumptype%22%3A%22text%22%2C%22urlsign%22%3A%227284024676449629996%22%2C%22sitesign%22%3A%223550531655276660426%22%2C%22query%22%3A%22%5Cu5b89%5Cu5353%5Cu5f00%5Cu53d1%5Cu6709%5Cu54ea%5Cu4e9b%5Cu6846%5Cu67b6%22%2C%22token%22%3A%2214bd42%22%7D&top={%22sfhs%22:1}&atn=index&lid=9731067898737315493

[2] android 实现 switchbutton 选择器效果

https://blog.51cto.com/u_15359293/3909141

[3] Android 开发者必备的 15 款 UI 库

https://blog.51cto.com/u_15065849/2604320

[4] Android 经常使用开源组件汇总

https://blog.51cto.com/u_15127619/4661841?articleABtest=0

[5] Android 最全 UI 库合集

https://blog.csdn.net/sinat_33585352/article/details/88825720

[6] 【Android】ScrollView 实现滚动条效果

https://blog.csdn.net/qq_44833724/article/details/124022493

[7] AndroidStudio 中 Project 下的 build.gradle 没有 buildscript 和 allprojects

https://blog.csdn.net/Deep_rooted/article/details/124764731

[8] gradle7 代理配置

<https://www.jianshu.com/p/c53d47733da9>

[9] Duplicate class android.support.v4.app.INotificationSideChannel found in modules classes?

https://blog.csdn.net/Ryan_Captain/article/details/121098116

[10]Android 开发 JCenter 仓库替换华为云或阿里云镜像仓库

<https://www.jianshu.com/p/a414cc2c1137>