

深圳大学

新闻管理 APP

项目实施过程与使用说明书

罗艺菲 2022280131

2024-12-4

目录

一、开发环境搭建	2
二、功能实现细节	2
1、APP 标题及图标	2
2、新闻列表界面	2
3、新闻详情界面	3
4、新闻编辑界面	4
5、新闻增加界面	4
6、新闻删除功能	5
三、安装指南	6
1、环境准备	6
2、项目导入	7
3、运行应用	7
四、使用说明	8
1、查看新闻	8
2、编辑新闻	8
3、增加新闻	9
4、删除新闻	9
五、常见问题解答	10
六、结语	10

一、开发环境搭建

- 1、 编程语言：Java/Kotlin
- 2、 开发工具：Android Studio
- 3、 依赖管理：Gradle

二、功能实现细节

1、APP 标题及图标

在 AndroidManifest.xml 中编写代码，将 APP 命名为“新闻管理”，并修改 APP 的图标。

```
5  <?xml version="1.0" encoding="utf-8"?>
6  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
7  android:allowBackup="true"
8  android:fullBackupContent="@xml/backup_rules"
9  android:icon="@mipmap/news"
10 android:label="新闻管理"
11 android:roundIcon="@mipmap/news"
12 android:supportsRtl="true"
13 android:theme="@style/Theme.NewRoomProject"
14 tools:targetApi="31">
```

图 1 设置 APP 标题及 icon

2、新闻列表界面

使用 RecyclerView 控件实现新闻条目的展示。

```
public NewsAdapter(List<News> newsList, Context context, NewsDatabase database, Activity activity, OnNewsClickListener listener) {
    this.newsList = newsList;
    this.context = context;
    this.database = database;
    this.activity = activity;
    this.listener = listener;
}

@NonNull
@Override
public NewsViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View itemView = LayoutInflater.from(context).inflate(R.layout.news_item, parent, attachToRoot: false);
    return new NewsViewHolder(itemView);
}

@Override
public void onBindViewHolder(@NonNull NewsViewHolder holder, int position) {
    News news = newsList.get(position);
    holder.titleTextView.setText(news.getTitle());
    holder.contentTextView.setText(news.getContent());
    holder.reporterTextView.setText(news.getReporterName());

    // 设置点击事件 - 编辑按钮
    holder.editButton.setOnClickListener(v -> editNews(news));

    // 设置点击事件 - 删除按钮
    holder.deleteButton.setOnClickListener(v -> deleteNews(news));
}
```

```

// 点击标题或者内容跳转到新闻详情
holder.titleTextView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (listener != null) {
            listener.onNewsClick(news);
        }
    }
});
holder.contentTextView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (listener != null) {
            listener.onNewsClick(news);
        }
    }
});
}

@Override
public int getItemCount() { return newsList.size(); }
}

```

图 2 新闻列表的实现代码

3、新闻详情界面

展示单条新闻的详细信息，包括标题、内容等。

```

public class NewsDetailFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_news_detail, container, attachToRoot: false);

        titleTextView = rootView.findViewById(R.id.news_detail_title);
        contentTextView = rootView.findViewById(R.id.news_detail_content);
        reporterTextView = rootView.findViewById(R.id.news_detail_reporter);
        backButton = rootView.findViewById(R.id.news_detail_back_button);

        // 获取数据并展示
        if (getArguments() != null) {
            String title = getArguments().getString(key: "title");
            String content = getArguments().getString(key: "content");
            String reporter = getArguments().getString(key: "reporter");

            titleTextView.setText(title);
            contentTextView.setText(content);
            reporterTextView.setText(reporter);
        }

        // 返回
        backButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (getParentFragmentManager().getBackStackEntryCount() > 0) {
                    // 将数据展示 删除fragment
                    requireActivity().findViewById(R.id.recyclerView).setVisibility(ViewGroup.VISIBLE);
                    getParentFragmentManager().popBackStack();
                }
            }
        });
    }
}

```

图 3 新闻详情界面的实现代码

4、新闻编辑界面

提供文本输入框以编辑新闻标题和内容，实现保存功能。

```
// 编辑新闻的函数
private void editNews(News news) { 1 usage
    // 创建一个对话框用于编辑新闻
    AlertDialog.Builder builder = new AlertDialog.Builder(context);
    builder.setTitle("编辑新闻");

    // 创建一个布局并绑定到对话框
    View view = LayoutInflater.from(context).inflate(R.layout.dialog_add_news, root: null);
    builder.setView(view);

    // 获取视图中的控件
    EditText titleEditText = view.findViewById(R.id.edit_title);
    EditText contentEditText = view.findViewById(R.id.edit_content);
    EditText reporterEditText = view.findViewById(R.id.edit_reporter);

    // 预填充已有的数据
    titleEditText.setText(news.getTitle());
    contentEditText.setText(news.getContent());
    reporterEditText.setText(news.getReporterName());

    builder.setPositiveButton(text: "保存", (dialog, which) -> {
        // 获取用户输入的内容
        String title = titleEditText.getText().toString();
        String content = contentEditText.getText().toString();
        String reporterName = reporterEditText.getText().toString();

        // 更新新闻对象
        news.setTitle(title);
        news.setContent(content);
        news.setReporterName(reporterName);

        // 在后台线程中执行更新操作
        new Thread(() -> {
            database.newsDao().updateNews(news); // 这里使用 insertNews 方法进行更新
            activity.runOnUiThread(() -> {
                // 刷新列表
                notifyDataSetChanged();
            });
        }).start();
    });

    builder.setNegativeButton(text: "取消", listener: null);
    builder.show();
}
```

图 4 新闻编辑界面的实现代码

5、新闻增加界面

提供界面以输入新增新闻的标题和内容，并实现增加功能。

```
// 添加新闻 弹出框
private void showAddNewsDialog() { 1 usage
    View dialogView = LayoutInflater.from( context: this).inflate(R.layout.dialog_add_news, root: null);
    EditText editTitle = dialogView.findViewById(R.id.edit_title);
    EditText editContent = dialogView.findViewById(R.id.edit_content);
    EditText editReporter = dialogView.findViewById(R.id.edit_reporter);

    new AlertDialog.Builder( context: this)
        .setTitle("添加新闻")

        .setView(dialogView)
        .setPositiveButton( text: "添加", (dialog, which) -> {
            String title = editTitle.getText().toString();
            String content = editContent.getText().toString();
            String reporter = editReporter.getText().toString();
            if (!title.isEmpty() && !content.isEmpty() && !reporter.isEmpty()) {
                addNews(new News(title, content, reporter));
            } else {
                Toast.makeText( context: this, text: "请填写全部内容!", Toast.LENGTH_SHORT).show();
            }
        })
        .setNegativeButton( text: "取消", listener: null)
        .show();
}

// 添加新闻数据
private void addNews(News news) { 1 usage
    new Thread() -> {
        database.newsDao().insertNews(news);
        loadNews();
    }.start();
}

@Override 2 usages
public void onNewsClick(News news) { showNewsDetailFragment(news); // 点击新闻时显示详情页 }
```

图 5 新闻增加界面的实现代码

6、新闻删除功能

在新闻列表中提供删除按钮，实现删除指定新闻条目。

```

132 // 删除新闻
133 private void deleteNews(Newsp news) { 1 usage
134     new Thread(() -> {
135         database.newsDao().deleteNews(news);
136         activity.runOnUiThread(() -> {
137             // 刷新列表
138             newList.remove(news);
139             notifyDataSetChanged();
140         });
141     }).start();
142 }
143
144 public static class ViewHolder extends RecyclerView.ViewHolder { 4 usages
145
146     TextView titleTextView; 3 usages
147     TextView contentTextView; 3 usages
148     TextView reporterTextView; 2 usages
149     Button editButton; 2 usages
150     Button deleteButton; 2 usages
151
152     public ViewHolder(View itemView) { 1 usage
153         super(itemView);
154
155         titleTextView = itemView.findViewById(R.id.text_title);
156         contentTextView = itemView.findViewById(R.id.text_content);
157         reporterTextView = itemView.findViewById(R.id.text_reporter);
158         editButton = itemView.findViewById(R.id.btn_edit);
159         deleteButton = itemView.findViewById(R.id.btn_delete);
160     }

```

图 6 新闻删除功能的实现代码

三、安装指南

1、环境准备

在开始使用新闻管理应用之前，请确保您的开发环境已经准备妥当。具体步骤如下：

- 1) 安装 Java 开发工具包 (JDK):
 - a) 访问 Oracle 官网或其他 JDK 提供商，下载并安装适合您操作系统的 Java 开发工具包。
 - b) 安装完成后，配置环境变量，确保 JAVA_HOME 指向 JDK 安装目录，并且 java 和 javac 命令可以在命令行中直接执行。
- 2) 安装 Android Studio:
 - a) 访问 Android Studio 官网，下载最新版本的 Android Studio。
 - b) 按照安装向导完成安装，并在首次启动时允许 Android Studio 下载和安装必要的组件，包括 Android SDK 和 Emulator。
- 3) 配置 Android SDK:
 - a) 在 Android Studio 中，打开 SDK Manager，选择并安装所需的 Android

SDK 版本。

- b) 确保您安装了最新版本的 Android SDK Platform 和相应的 Build-Tools。

2、项目导入

完成环境搭建后，您可以开始导入新闻管理应用项目：

1) 打开 Android Studio：

启动 Android Studio，等待其完成初始化。

2) 导入项目：

- a) 点击“Open an existing Android Studio project”选项，然后浏览到包含新闻管理应用的项目文件夹。
- b) 选择项目根目录（包含 build.gradle 文件的目录），然后点击“OK”完成项目导入。

3) 同步项目：

导入项目后，点击 Android Studio 工具栏中的“Sync Project with Gradle Files”按钮，以确保所有依赖项都被正确下载和配置。

3、运行应用

在项目成功导入并同步后，您可以开始运行应用：

1) 选择运行设备：

- a) 连接您的 Android 设备到计算机，或者在 Android Studio 中配置一个 Android 虚拟设备（AVD）。
- b) 在 Android Studio 工具栏中，选择您要运行应用的设备或模拟器。

2) 运行应用：

- a) 点击 Android Studio 工具栏中的“Run”按钮（绿色三角形图标），开始构建并运行应用。
- b) 首次构建可能需要一些时间，因为 Gradle 需要编译代码并生成 APK 文件。

3) 调试和测试：

- a) 应用启动后，您可以在设备或模拟器上查看应用界面，进行调试和测试。
- b) 使用 Android Studio 的 Logcat 和 Debug 功能来监控应用运行情况，并解决可能出现的问题。

按照以上步骤，您应该能够顺利地安装和运行新闻管理应用。如果在安装或运行过程中遇到任何问题，请参考 Android Studio 的帮助文档或搜索相关错误信息以

获得解决方案。

四、使用说明

1、查看新闻

打开应用，即可在主界面看到新闻列表，在列表中点击新闻，即可在详情界面查看新闻的详细内容。



图 7 查看新闻列表与详情界面

2、编辑新闻

点击列表中任一新闻条目，进入详情界面，点击编辑按钮进行编辑。

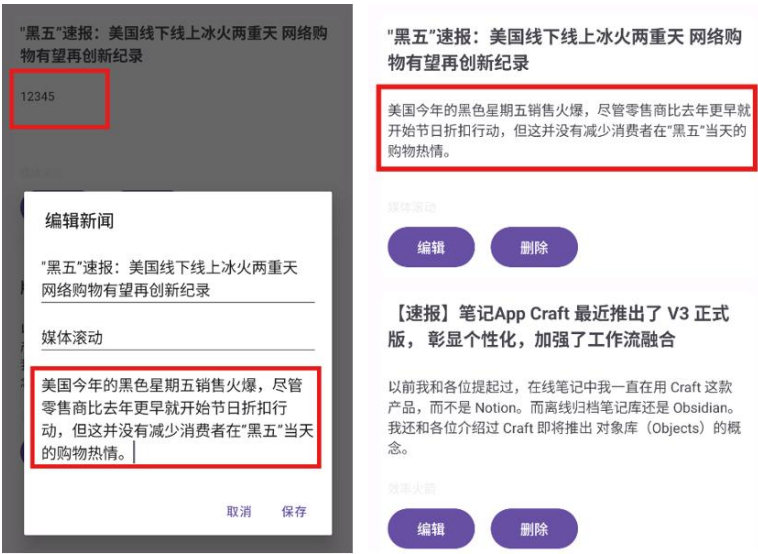


图 8 编辑新闻

3、增加新闻

点击主界面的添加按钮，输入新闻标题和内容，保存后新闻将显示在列表中。



图 9 增加新闻

4、删除新闻

在新闻列表中，点击新闻条目下方的删除按钮即可删除。



图 10 删除新闻

五、常见问题解答

Q1: 应用无法运行？

A1: 确保设备兼容且 Android Studio 环境配置正确。

Q2: 如何更新新闻内容？

A2: 选择新闻条目，点击编辑，修改后保存即可。

六、结语

感谢您选择并使用我们的新闻管理应用，希望这个应用能够成为您日常工作中的得力助手，帮助您高效、便捷地管理新闻内容。

如果您在使用过程中遇到任何问题，或者有任何宝贵的意见和建议，我们都非常欢迎您的反馈。您的每一次互动都是我们不断进步的动力。我们期待与您一起，让新闻管理变得更加简单、高效。

再次感谢您的信任和支持，让我们携手共创更美好的明天。