

**2023微信菁英班结课作业**

***——基于Firebase的安卓聊天app的实现***

|  |  |
| --- | --- |
| 专 业： | 物联网 |
| 班 级： | IOT2101 |
| 学 号： | U202115621 |
| 姓 名： | 温宇鹏 |

2023 年 10 月 31 日

目 录

[1 产品功能介绍 1](#_Toc5717)

[1.1项目开发意义 1](#_Toc16818)

[1.2产品功能 1](#_Toc8303)

[2 程序概要设计 2](#_Toc14760)

[2.1 开发实现 2](#_Toc1501)

[2.2 Firebase部署 2](#_Toc28946)

[2.3 Android应用程序 7](#_Toc14763)

[2.4 使用的依赖 12](#_Toc5159)

[3 软件架构图 13](#_Toc12044)

[4 技术亮点及其实现原理 17](#_Toc18885)

[5 发展空间和课程总结 19](#_Toc14352)

# 1 产品功能介绍

## 1.1项目开发意义

制作了一个简单的聊天app，内容包括发送手机验证码进行登录注册，更新个人资料，更换头像，添加其他用户并建立聊天室等。能够促进人们之间的沟通和交流，让用户更轻松地进行互动和社交，与他人实时交流，从而加强社交关系并促进信息传递，提高工作效率。

采用的语言是开源免费的JAVA，使用的开发工具是Google免费的Android开发编译器 AndroidStudio。

此外，该项目有助于提高开发者积累前端后端协调开发的经验。可以让开发者能够学习和应用现代化的用户界面设计、实时消息传递、响应式设计、服务器端开发、数据存储和管理等多方面技术，提高自身开发能力和经验。

## 1.2产品功能

### 1.2.1用户手机号登录并输入验证码和用户名进入app：

用户使用手机号进行验证码登录，之后输入用户名，作为用户在app中的唯一标识，若重名则无法同时使用app。登录后，可建立或访问与其他用户的聊天室获取数据库中的全部聊天信息并参与实时聊天。

### 1.2.2实时聊天：

用户可以在聊天室中发送和接收实时消息。用户可以通过文本输入框输入消息，并点击发送按钮发送到聊天室中。聊天消息将被广播给聊天室中的用户，以便其能够实时查看和回复。

### 1.2.3聊天历史记录：

Firebase中的Firestore Database连接的数据库可以保存聊天室的ID、最新消息、最新消息的法案宋哲ID、记录时间以及用户ID、用户手机号、用户名、用户创建时间等。用户可以在需要时查看以前的聊天消息，以回顾过去的对话和信息。

### 1.2.4 其他功能：

还包括输入栏随软键盘呼出相应提升等功能，可在提交文件夹的演示视频中查看。

# 2 程序概要设计

## 2.1 开发实现

### 2.1.1编程语言：

Java

### 2.1.2编程环境：

Firebase

Android Studio Giraffe | 2022.3.1 Patch 2

## 2.2 Firebase部署

### 2.2.1 Firebase概览

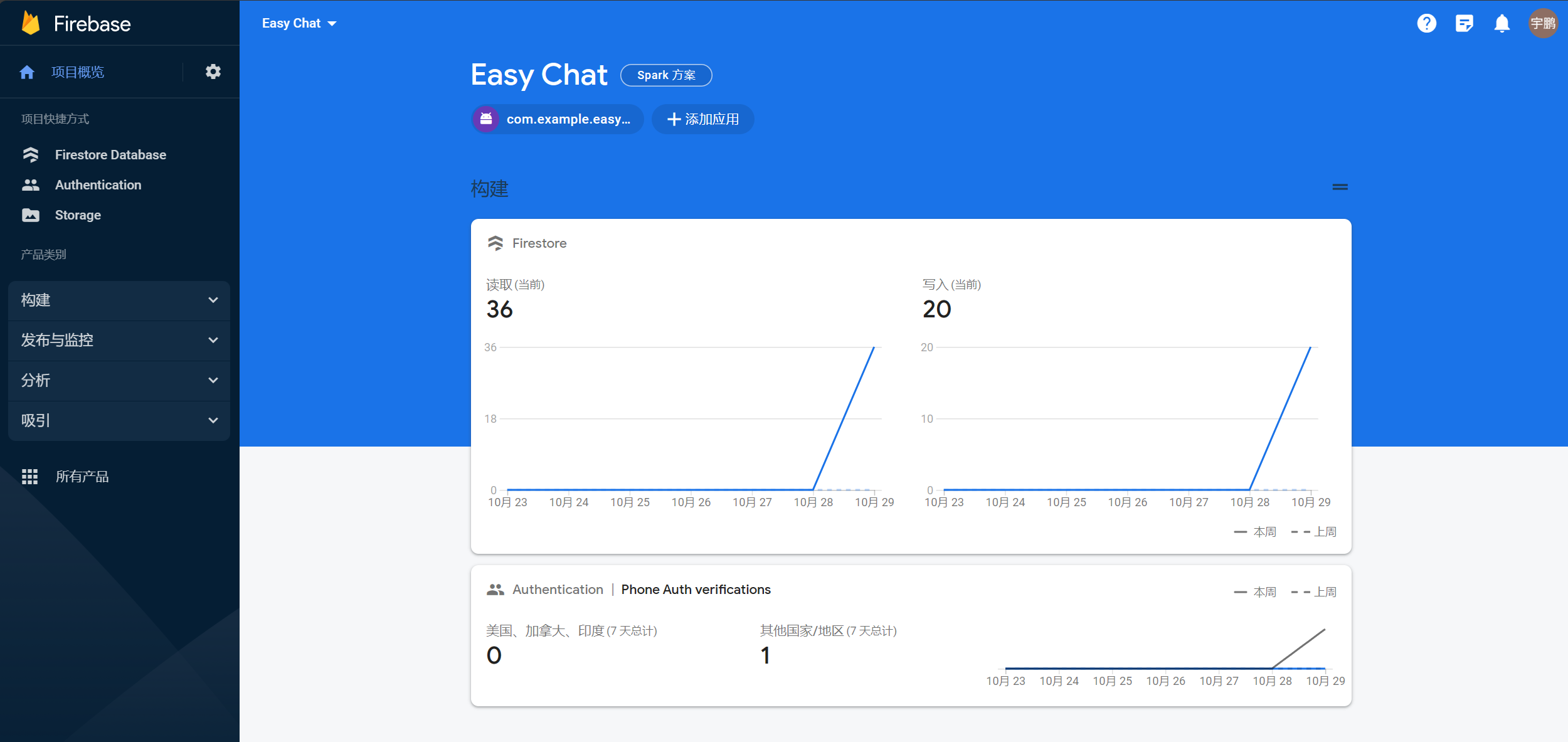


图 2.2.1 后端Firebase项目结构

Firebase中，我们用到了Authentication、Firestore Database以及Storage等项目，其中Authentication用于用户的手机验证码登录，Firestore Database作为后端数据库存储用户、聊天室信息等，Storage则用于存储用户的图片信息（头像）。

### 2.2.2 结构

1.Firestore Database

我们在users收集表里记录了每个用户的注册记录，内容包括创建时间、fcm令牌、手机号、用户Id以及用户名。如下图所示：

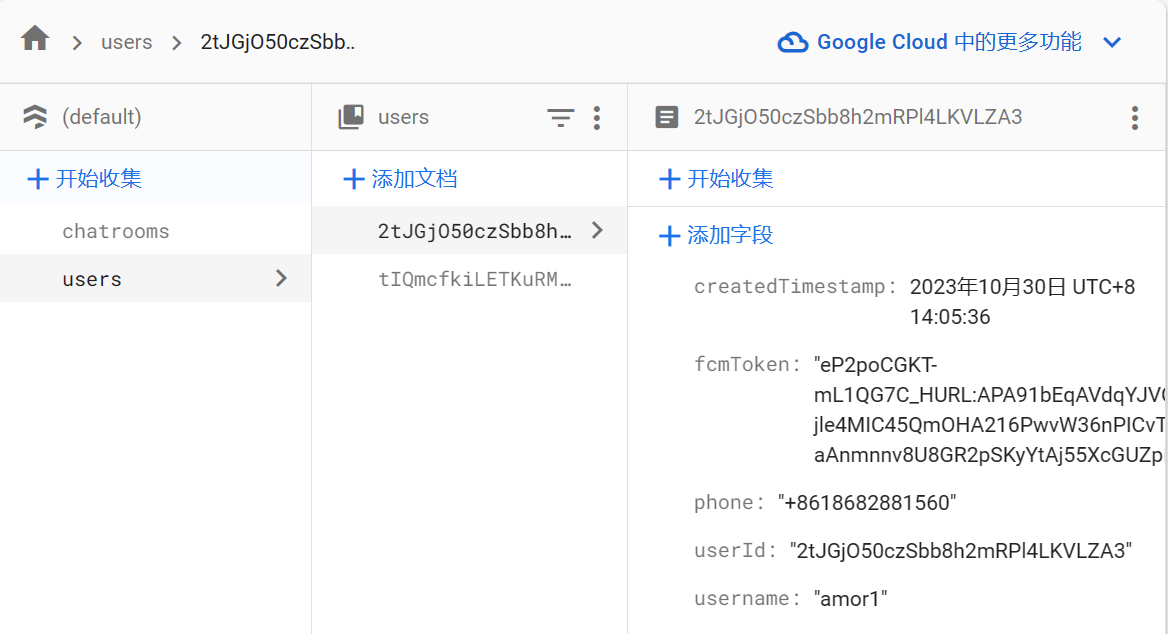


图 2.2.2用户收集表

在chatrooms收集表里记录了每个聊天室Id、聊天室中双方用户的Id、上一条消息、上一条消息的发送者Id以及时间。如下图所示：

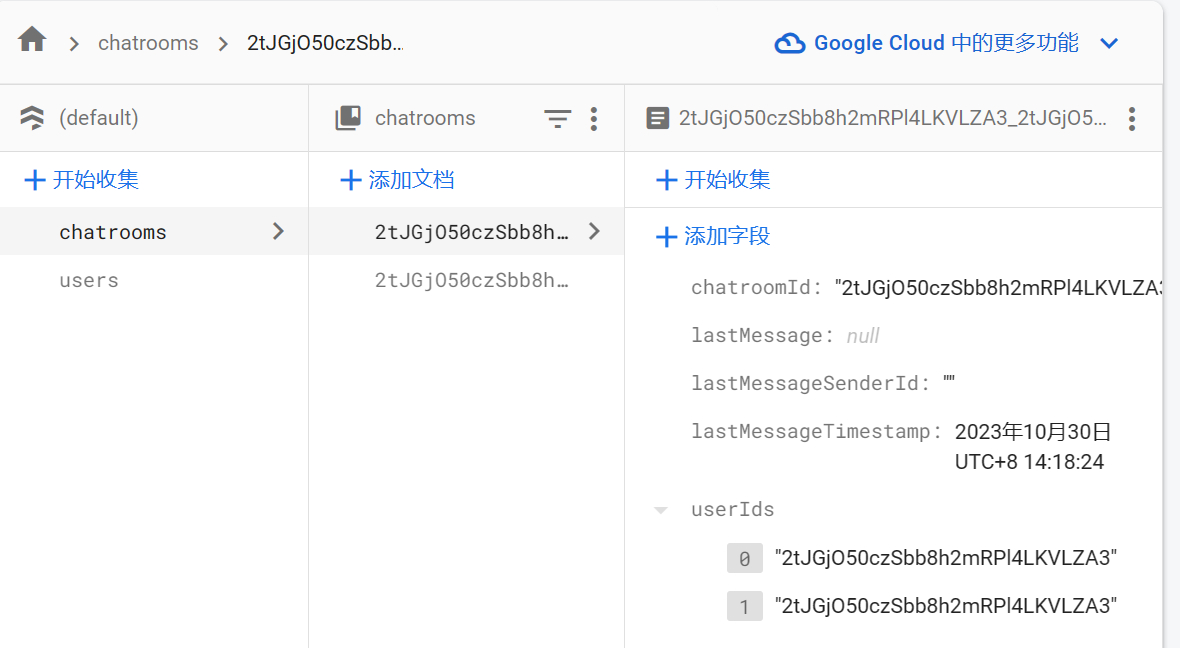


图 2.2.3聊天室收集表

1. Authentication

该项目中用于向输入的手机号发送验证码进行登录注册，其中记录的用户的注册记录，包括手机号、创建时间、上次登录日期以及用户的UID等。如下图所示：



图 2.2.4 Authentication界面

1. Storage

该项目用于存储用户的图片信息，在本app中即用户的头像。存储格式如下图所示：

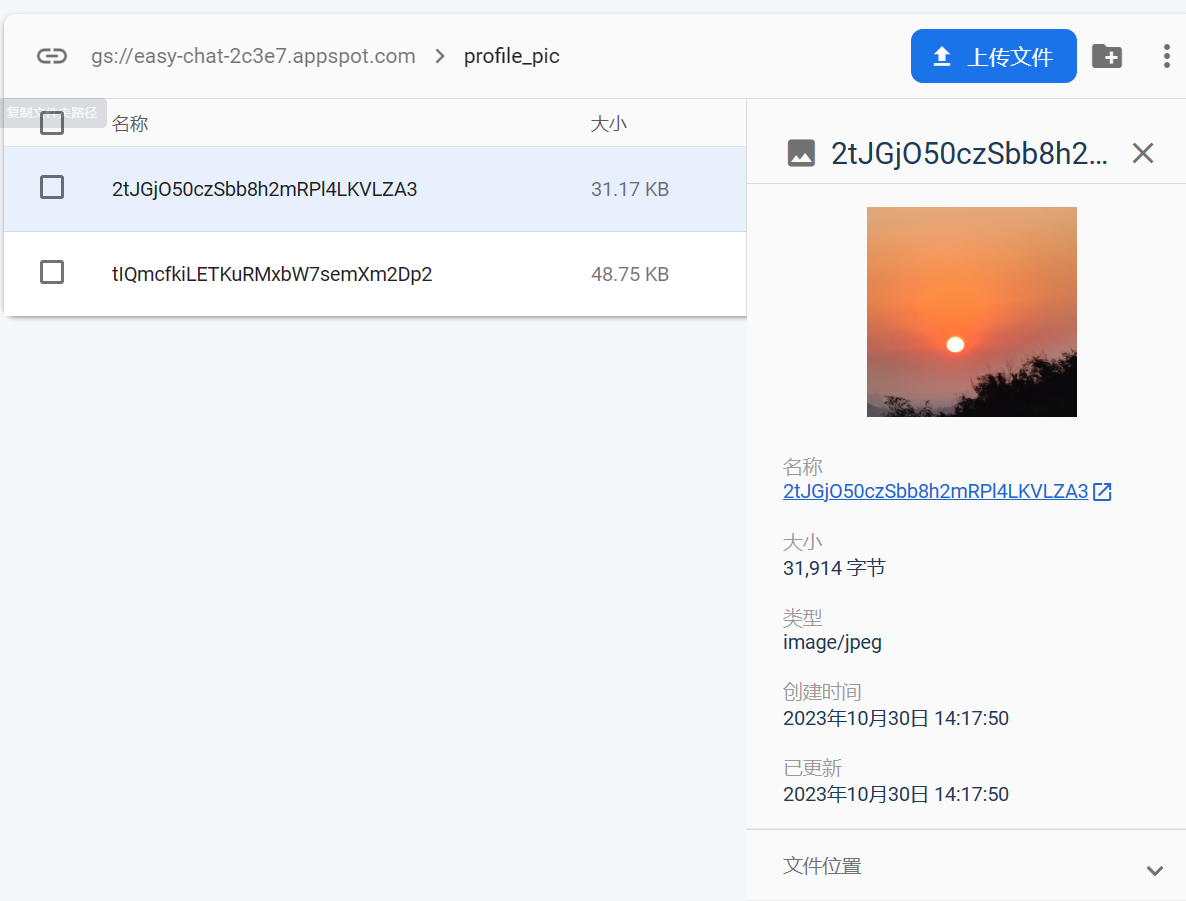
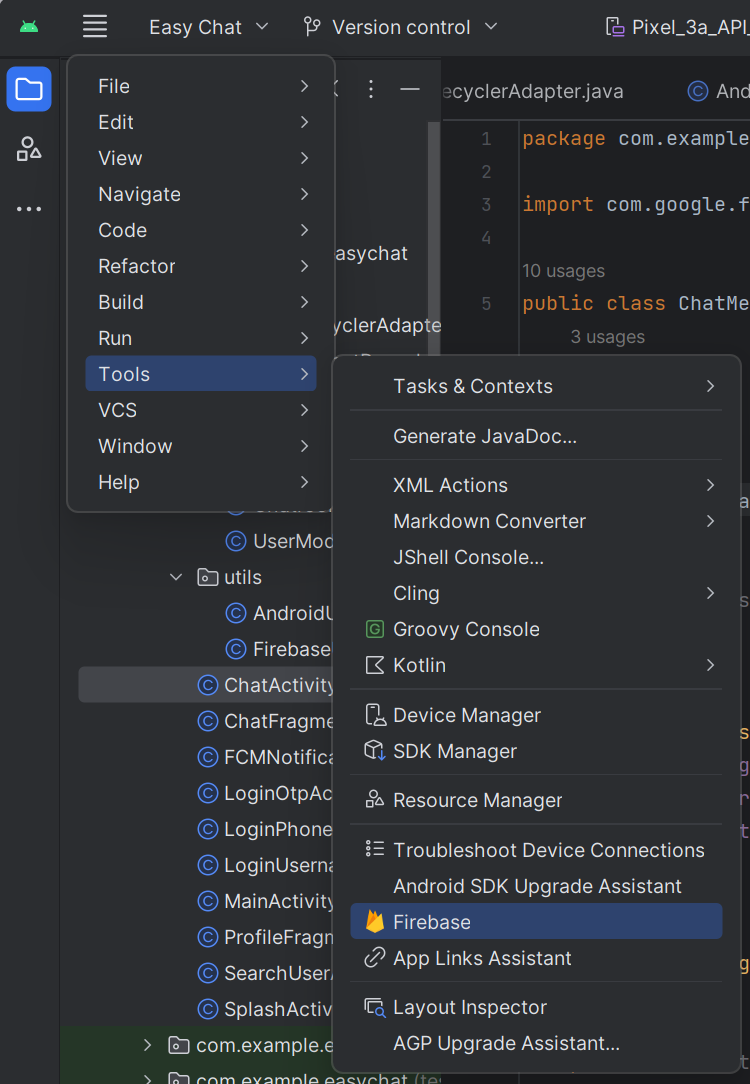
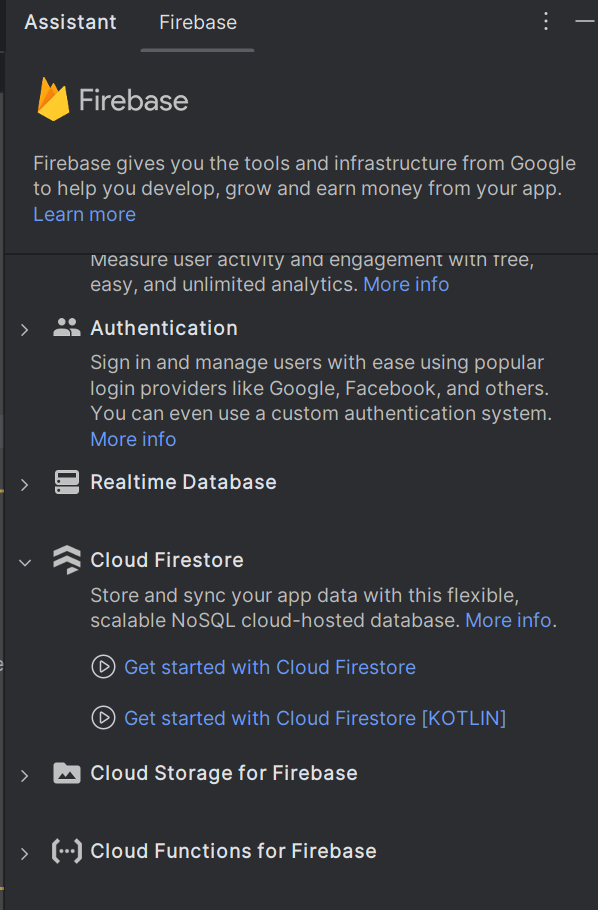


图 2.2.5 Storage存储图像格式

### 2.2.3 使用方法

打开Android Studio中的Tools一栏，选择其中的Firebase，随后在Assistant中选择Cloud Firestore，随后点击Get started with Cloud Firestore，按照连接提示，一步一步连接到自己的Firebase项目中。





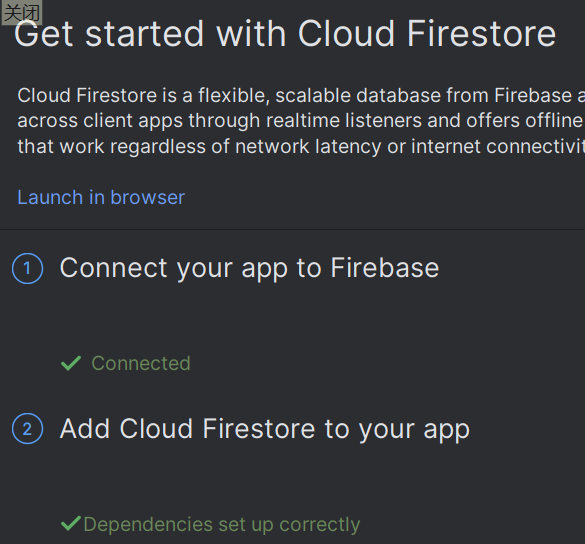


图 3.2.2连接流程

### 2.2.4 测试

通过一下代码可以判断是否连接成功：

Map<String, String> data = new HashMap<>();

FirebaseFirestore.getInstance().collection(“test”).add(data);

如下图课件，在数据库中发现了test，表面我们连接成功。

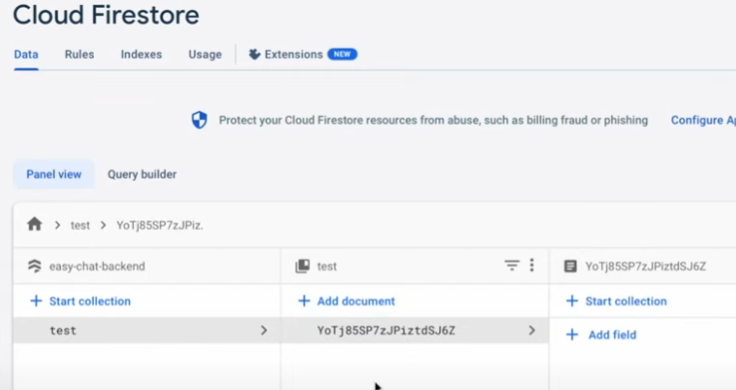


图 2.2.4-1 Firebase连接测试结果

## 2.3 Android应用程序

### 2.3.1 项目结构和概览

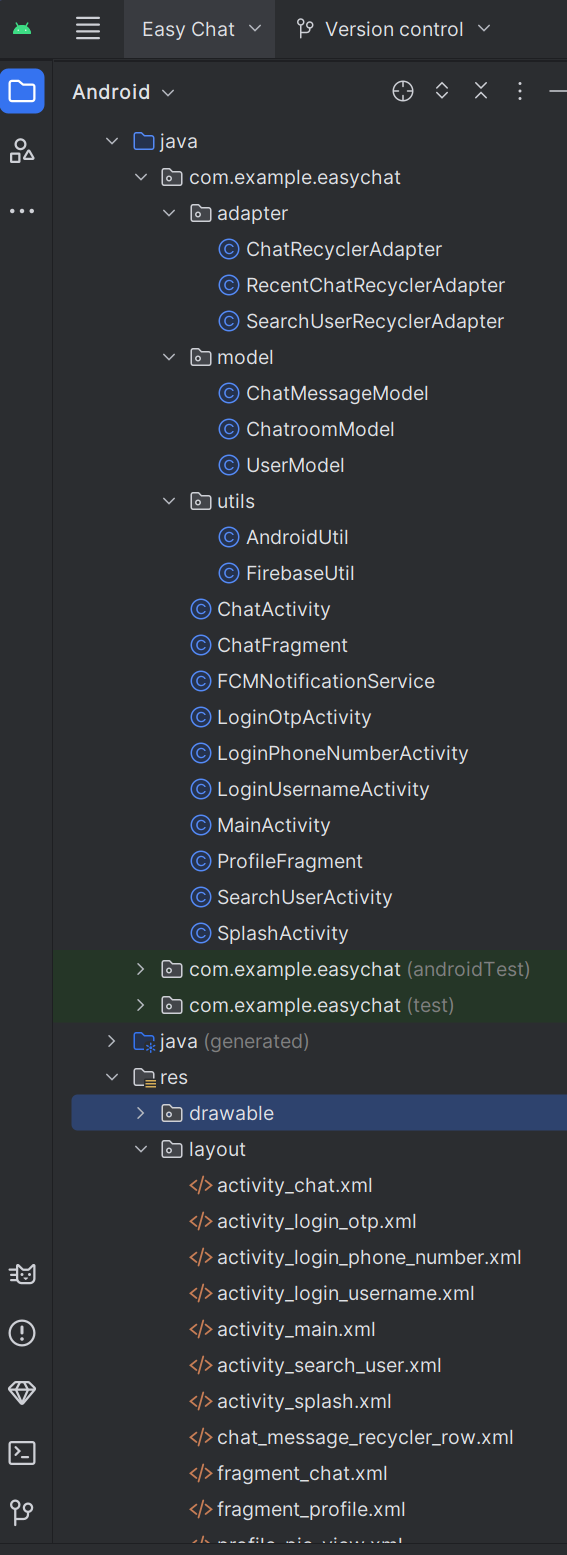


图 2.3.1-1 Android应用程序项目结构

Gradle JDK ：jbr-17

MinSdk 26

Android应用程序的代码主要负责与Firebase进行通信（例如发送请求、处理响应），以及实现应用程序的业务逻辑、界面交互和数据持久化等功能。通过编写这些代码，可以实现Android应用程序与Firebase的连接和交互，提供完整的应用程序体验。

### 2.3.2 数据结构

用户模型：

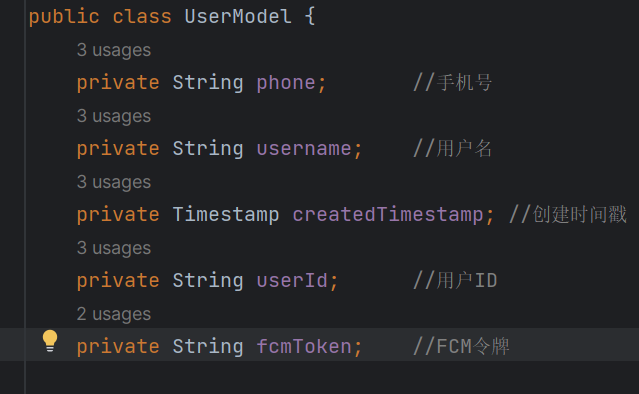


图 4.2.2 用户模型

聊天室模型：



图 2.2.3 聊天室模型

聊天消息模型：

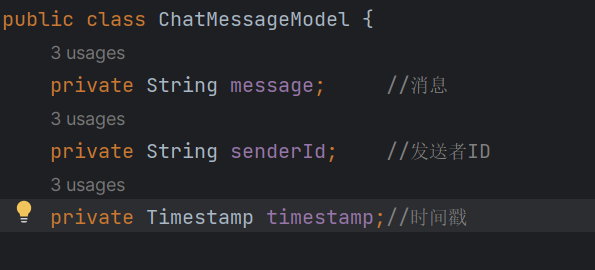


图 2.2.4 聊天消息模型

### 2.3.3 主要组件功能介绍

***2.3.3.1* SearchUserRecyclerAdapter**

用于搜索用户并显示在RecyclerView中的适配器类。适配器继承自FirestoreRecyclerAdapter，用于从Firestore数据库获取用户模型数据，并将其绑定到RecyclerView的视图上。在构造函数中接收FirestoreRecyclerOptions和Context参数，并将其传递给父类的构造函数。

在onBindViewHolder方法中，将用户模型的数据绑定到ViewHolder的视图上。具体来说，设置用户名和电话号码的文本显示，并根据当前用户判断是否为自己的数据，在用户名后面添加"(我)"标识。

使用FirebaseUtil.getOtherProfilePicStorageRef方法获取其他用户的个人资料图片的下载URL。当下载URL获取成功时，使用AndroidUtil.setProfilePic方法将图片设置到ImageView中。

为每个RecyclerView的项设置点击事件。点击项时，创建一个跳转到ChatActivity的Intent，并传递选中用户的模型数据作为Intent的附加信息。然后启动该Intent，打开聊天界面。

在onCreateViewHolder方法中，通过LayoutInflater从布局文件中实例化视图，并返回一个新的ViewHolder对象。

***2.3.3.2* RecentChatRecyclerAdapter**

该适配器是用于在RecyclerView中显示最近聊天记录的列表。它继承了FirestoreRecyclerAdapter，用于从Firestore数据库获取聊天记录数据，然后将其绑定到RecyclerView的视图上。

在构造函数中接收FirestoreRecyclerOptions和Context参数，并将其传递给父类的构造函数。

在onBindViewHolder方法中，从ChatroomModel中取出与当前用户对话的另一个用户的userModel，并更新其用户名、最后一条消息内容、时间戳等信息到ViewHolder的视图上。

使用FirebaseUtil.getOtherProfilePicStorageRef方法获取其他用户的个人资料图片的下载URL。当下载URL获取成功时，使用AndroidUtil.setProfilePic方法将图片设置到ImageView中。

为每个RecyclerView的项设置点击事件。点击项时，创建一个跳转到ChatActivity的Intent，并传递选中用户的模型数据作为Intent的附加信息。然后启动该Intent，打开聊天界面。

在onCreateViewHolder方法中，通过LayoutInflater从布局文件中实例化视图，并返回一个新的ViewHolder对象。

***2.3.3.3*** ChatRecyclerAdapter

该适配器是用于在RecyclerView中显示聊天记录的列表。它继承了FirestoreRecyclerAdapter，用于从Firestore数据库获取聊天记录数据，然后将其绑定到RecyclerView的视图上。

在构造函数中接收FirestoreRecyclerOptions和Context参数，并将其传递给父类的构造函数。

在onBindViewHolder方法中，根据每个消息的发送者ID判断该消息是自己发送的还是对方发送的，并将相应的聊天布局设置为可见，另一个布局设置为不可见。然后将消息内容设置到聊天布局对应的TextView中。

在onCreateViewHolder方法中，通过LayoutInflater从布局文件中实例化聊天布局，并返回一个新的ViewHolder对象。

***2.3.3.4*** LoginOtpActivity

这段代码实现了一个登录页面，用于用户通过手机号码和验证码进行登录。

在 onCreate 方法中，首先获取从上一个页面传递过来的手机号码，并调用 sendOtp 方法发送验证码。

sendOtp 方法通过 PhoneAuthProvider 发送验证码短信，设置了回调函数来处理验证码认证的状态变化。

当验证码发送成功后，在回调函数的 onCodeSent 方法中，保存验证码和重发令牌，并显示一个 Toast 提示验证码发送成功。

用户输入验证码后，点击 "下一步" 按钮触发 nextBtn 的点击事件，会调用 signIn 方法进行验证码认证。

signIn 方法使用 PhoneAuthCredential 进行用户认证，如果认证成功，则跳转到下一个页面；如果认证失败，则显示一个 Toast 提示验证码认证失败。

setInProgress 方法根据参数来设置进度条和按钮的可见性，用于表示当前是否正在进行验证码认证的过程。

startResendTimer 方法用于启动一个定时器，在60秒内每隔1秒更新倒计时并禁用重新发送按钮。当倒计时结束后，重新启用按钮。

***2.3.3.5 MainActivity:***

这段代码实现了一个主界面，包含底部导航栏和两个片段（ChatFragment 和 ProfileFragment）。

在 onCreate 方法中，首先初始化 chatFragment 和 profileFragment 的实例。

初始化底部导航栏 bottomNavigationView 和搜索按钮 searchButton。

当点击搜索按钮时，通过 startActivity 方法跳转到搜索用户界面。

设置底部导航栏的选择监听器，当选择聊天菜单项时，使用 getSupportFragmentManager().beginTransaction().replace() 方法替换主界面的内容为 chatFragment；当选择个人资料菜单项时，替换内容为 profileFragment。

默认选中聊天菜单项。

getFCMToken 方法用于获取 Firebase Cloud Messaging (FCM) 的设备令牌。在 FirebaseMessaging.getInstance().getToken() 完成后，将令牌更新到当前用户的详细信息中。

总体来说，这段代码完成了一个具有底部导航栏和两个片段的主界面，并提供了相应的用户界面和交互逻辑。用户可以通过底部导航栏的菜单项切换不同的功能页面，并能够进行用户搜索和获取 FCM 设备令牌等操作。

### 2.3.4 测试

启用Android Studio连接真机进行真机调试，将手机设置为开发者模式，允许USB调试。测试细节可见测试视频。

## 2.4 使用的依赖

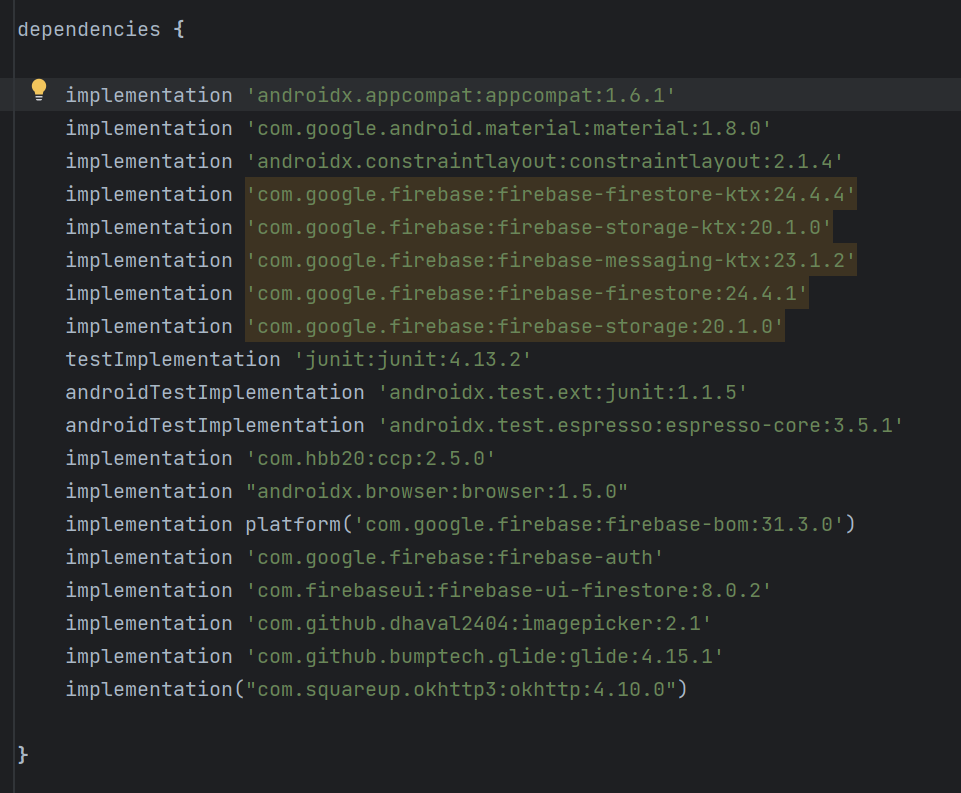


图 2.4 所使用的app依赖库

# 3 软件架构图

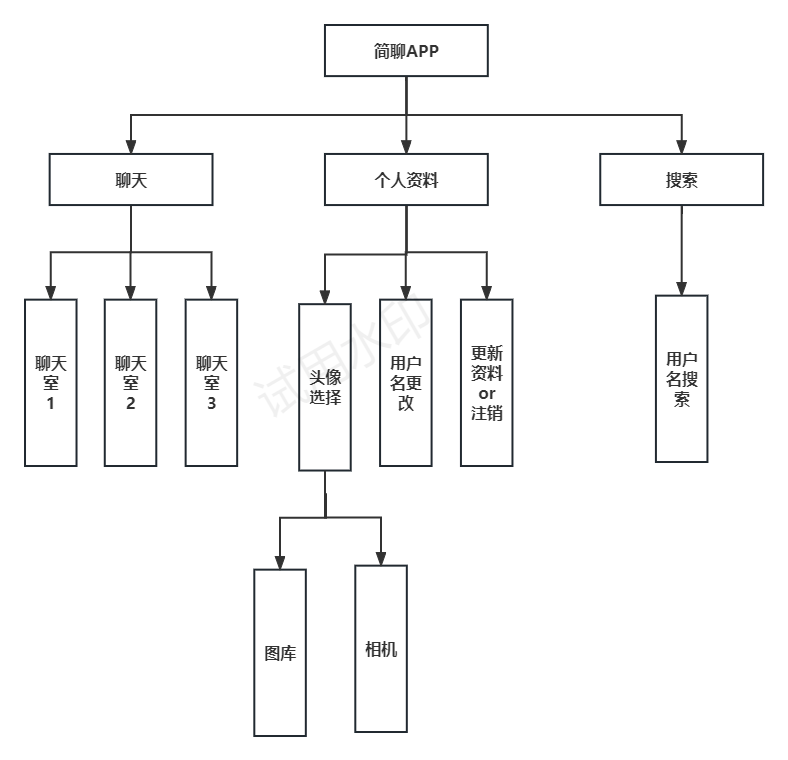


图 3.1 软件架构图

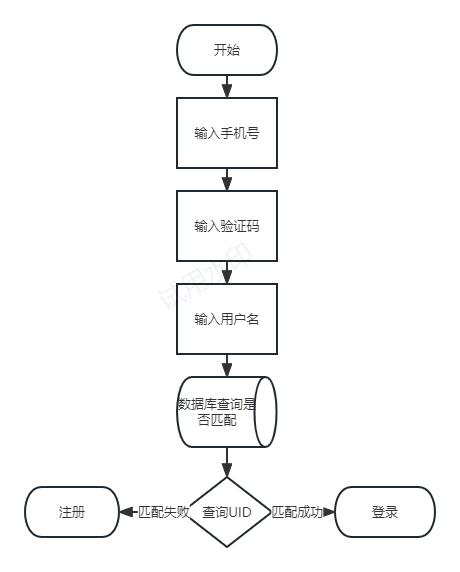


图 3.2 登录/注册流程图

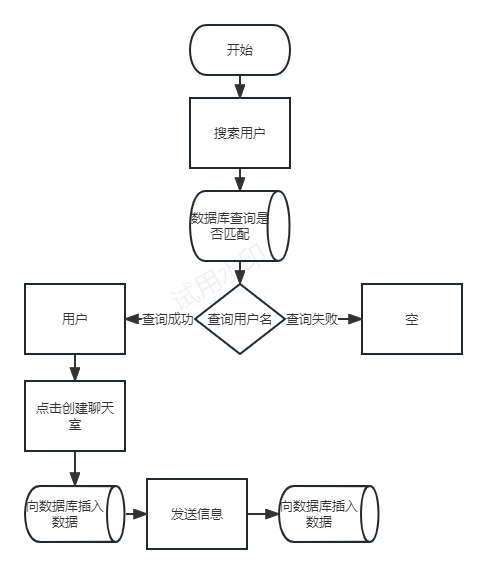


图 3.3 聊天流程图

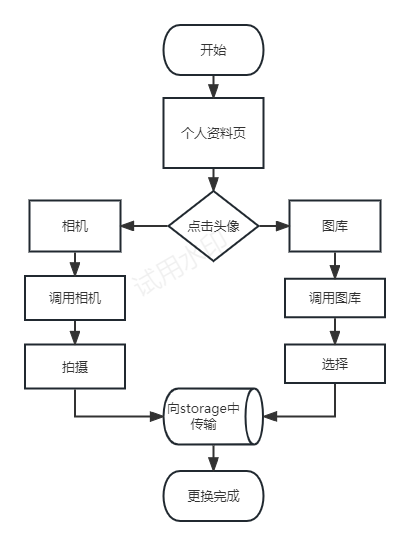


图 3.4 头像更换流程图

# 4 技术亮点及其实现原理

### 4.1引用各种依赖

引用这些依赖库的好处是可以快速构建具有现代设计和功能的 Android 应用程序，而不必从头开始编写和实现所有组件和功能：

AndroidX AppCompat 库，提供了与旧版 Android 系统版本兼容的支持。

Google Material Design 库，提供了实现 Material Design 设计风格的组件和样式。

ConstraintLayout 库，用于创建用户界面布局。

Firebase Cloud Messaging (FCM) 的 Kotlin 扩展库，用于在 Android 应用中发送和接收推送通知。

Firebase Firestore 库，用于在 Android 应用中使用云端数据库。

Firebase Storage 库，用于在 Android 应用中存储和获取文件。

JUnit 测试框架库，用于编写和执行单元测试。

AndroidX 扩展测试库，提供了扩展的 JUnit 功能。

Espresso UI 测试库，用于编写和执行 Android 用户界面的自动化测试。

Country Code Picker 库，用于选择国家/地区电话号码代码。

AndroidX 浏览器库，用于在应用中打开网页链接。

Firebase BOM（Bill of Materials）库，用于简化 Firebase 依赖库的管理。

Firebase Authentication 库，用于在 Android 应用中实现用户身份验证。

Firebase-UI Firestore 库，提供了在 Android 应用中快速构建用户界面的工具，特别是与 Firestore 数据库相关的部分。

ImagePicker 库，用于从设备的相册或相机中选择图像。

Glide 图像加载库，用于在 Android 应用中加载和显示图像。

OkHttp 库，用于进行网络请求和数据传输。

### 4.2 优化的用户体验

***1.简约背景，应用图标导入：***

给用户带来美观的视觉体验，同时也能够提高应用程序的品质和专业度。将相关的文件存储在res文件包中，图片存储在drawable中，midmap等文件中放着软件的图标。

***2.输入框位置调整：***

输入框会随着手机软键盘的呼出自动提高到软键盘上，并在软键盘消失之后回到底部，以优化用户体验。

通过使用isKeyboardVisible来跟踪键盘的可见性，并在键盘显示时将输入框向上偏移。

# 5 发展空间和课程总结

在移动互联网的蓬勃发展下，智能手机成为人们生活中不可或缺的一部分。在这个背景下，移动端开发成为软件开发领域的重要分支，而聊天类应用程序更是其中的关键应用之一。

掌握各种现有工具和库的使用，可以显著缩短开发周期，并能够打造出界面美观、功能丰富的应用程序。腾讯菁英班移动端开发课程提供了全面而完整的应用程序开发课程，帮助我们深入了解移动端开发的流程和技术要点。通过实践项目，我们得以提升自己的开发能力和实战经验。

这门课程在教学方式上注重实践，让我们亲身经历了一个完整的应用程序开发过程。通过理论与实践相结合，我对于安卓端开发的知识体系有了更加深入的理解。这不仅对于我个人的职业发展至关重要，还能使我在未来的工作中更好地应对挑战，成为一名优秀的移动端应用程序开发者。

综上所述，腾讯菁英班移动端开发课程为我提供了全面的移动端开发知识和实践经验，在短时间内快速掌握了开发技能。我期待今后继续深入学习并在安卓端开发领域不断成长。