北京理工大学计算机学院

《Android技术开发基础》课程设计模板

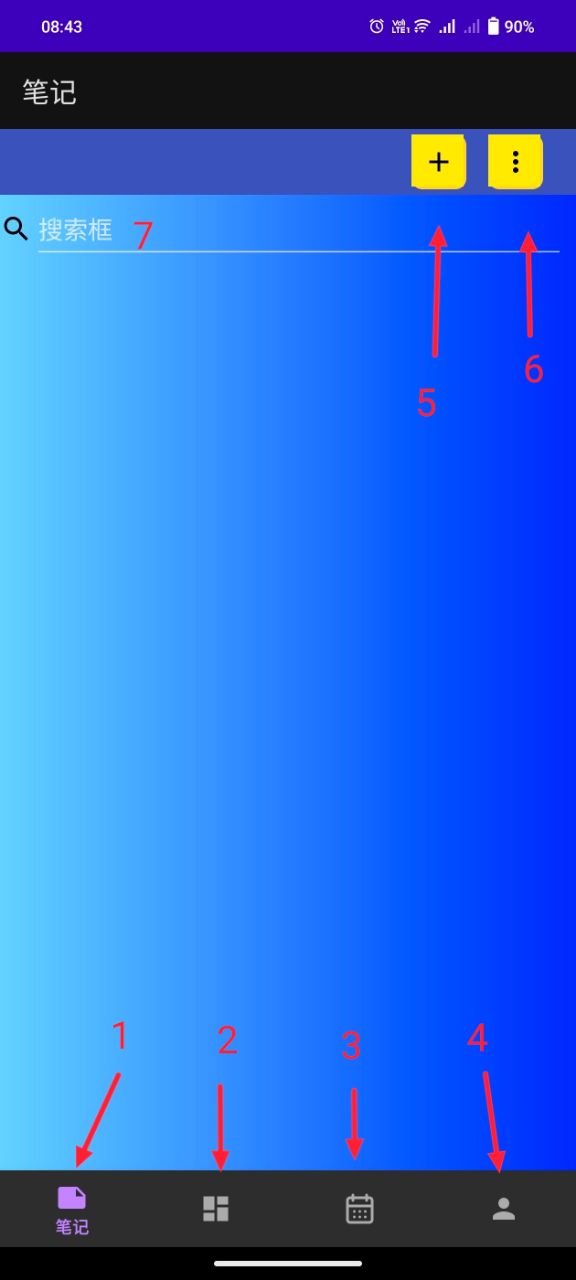
班级 07812201 学号 1820221050 姓名 丘绎楦

班级 07812201 学号 1820221053 姓名 曾泇睷

#### 1 App的运行与开发环境

1. **运行环境**： 6.0以上版本Android的Android手机
2. **部署方法：** 直接安装notebook.apk即可
3. **开发环境：** Android Studio 3.1.1
4. **手写代码行数：**手机端约2000行（Kotlin）

#### 2 App功能说明：

以下是app的页面展示：

如右图所示，这个APK有四个分页，分别为【笔记】（1），【事件】（2），【日历】（3），以及【我的】（4）。

功能：

* 笔记的注册登录

在用户页点击【Login】，弹出登录或者注册页面，没有账户点击【REGISTER】进行注册，输入用户名和密码，然后再通过此用户名和密码登录。

* 对笔记的增删改查

在笔记页点击右上角【+】（5）则会进入添加页面，里面输入笔记标题以及内容，点击【完成】就能成功保存笔记。

单击笔记可以进入编辑功能页面，数据会回显，修改后点击【完成】即可完成编辑。

长按笔记则会弹出删除笔记提示，点击【ok】则会删除笔记本。

* 对清单的增删改查

在事件页点击右下角【+】则会进入添加页面，里面输入笔记标题以及选择事件的截止时间，点击【add】就能成功保存清单。

单击清单可以编辑清单，长按清单会弹出清单提示框，点击【ok】删除清单。

已经添加的清单会在日历页面上显示出来，日历会显示本月的所有任务。

* 更换背景

在用户页面点击【SET BACKGROUND】可以选择本地图片作为笔记本全局背景。

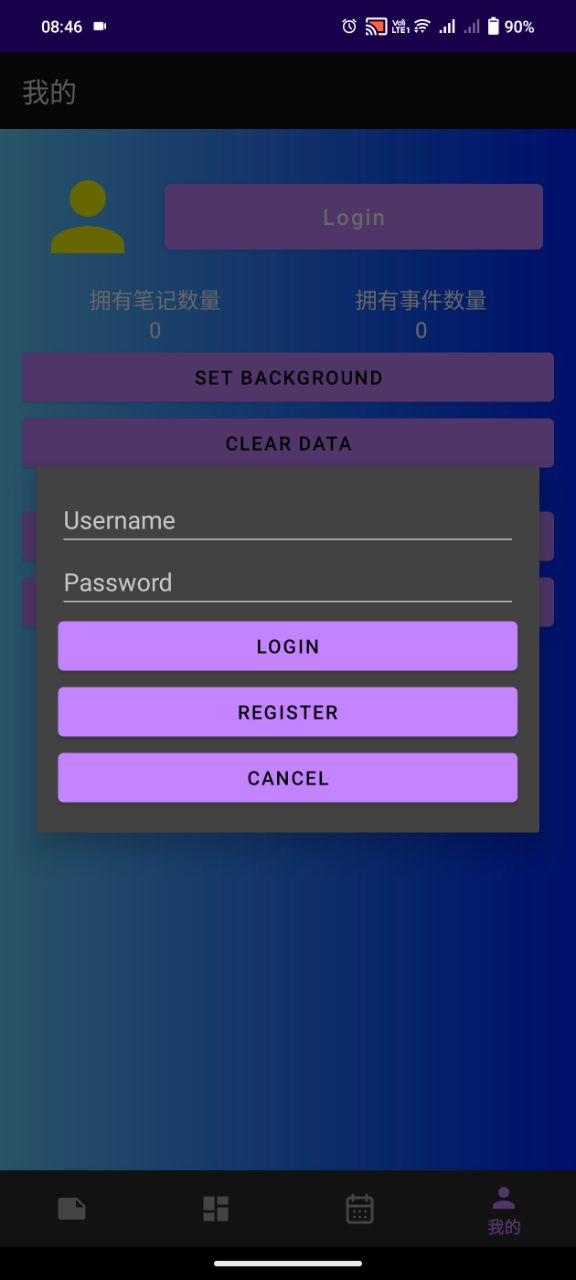
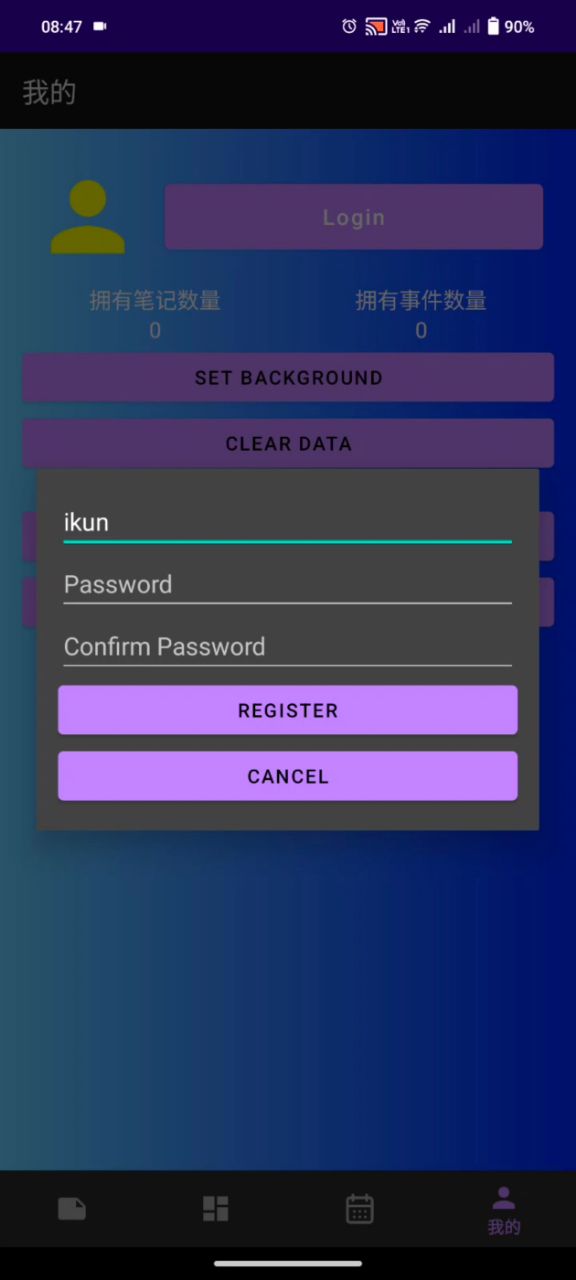
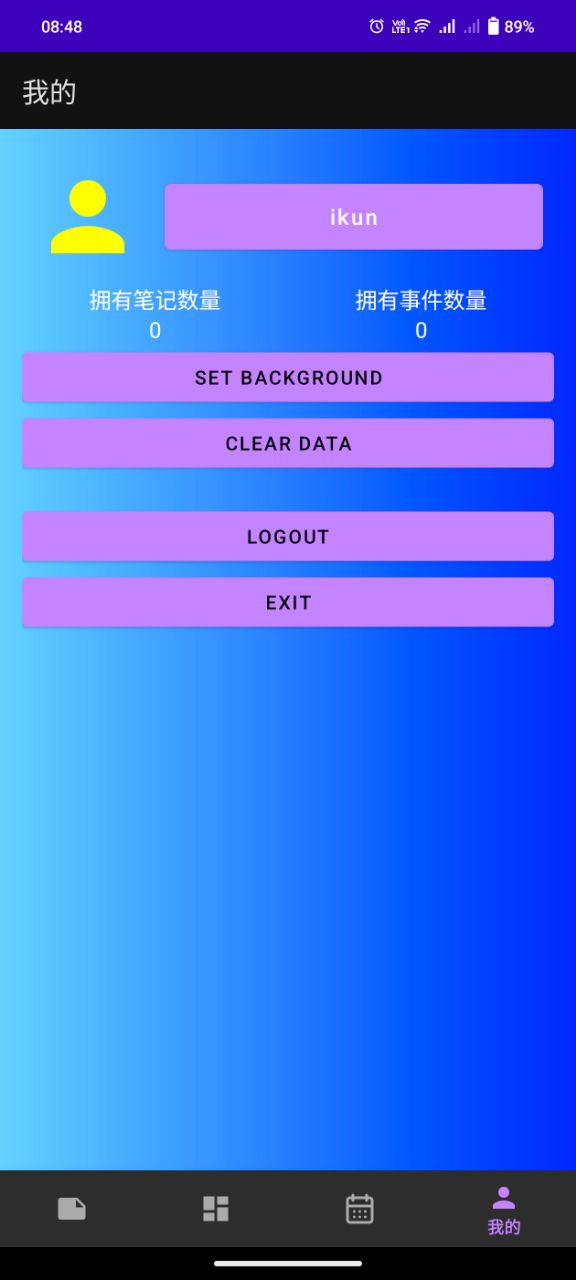
* 更换头像

在用户页面点击头像可以选择本地图片作为头像。

* 其他功能

在笔记页（或事件页）点击右上角三个点（6），会有一个小窗口弹出，包含【Clear】以及【Refresh】。

点击【Clear】可以清除所有笔记（或清单），点击【Refresh】刷新页面。

以下是功能的基本使用说明图片：

注册和登录功能展示图

左图为新增笔记页面

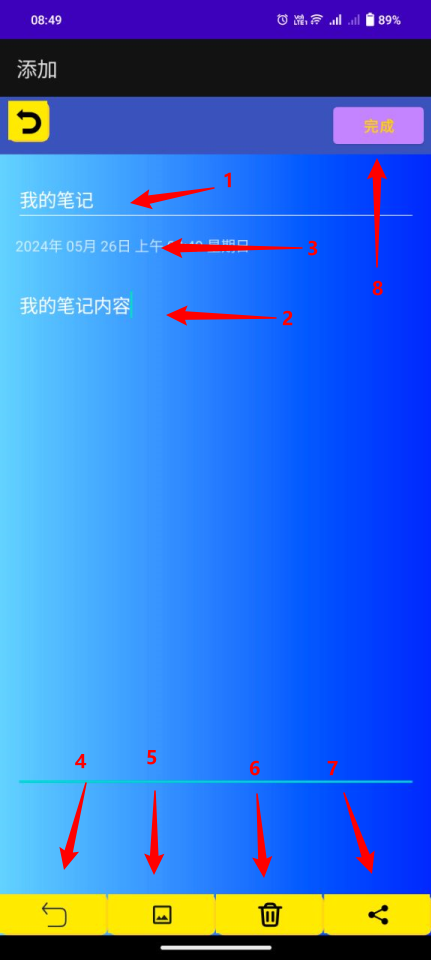
1是笔记标题

2是笔记内容

3是创建笔记的时间

4是撤销背景图

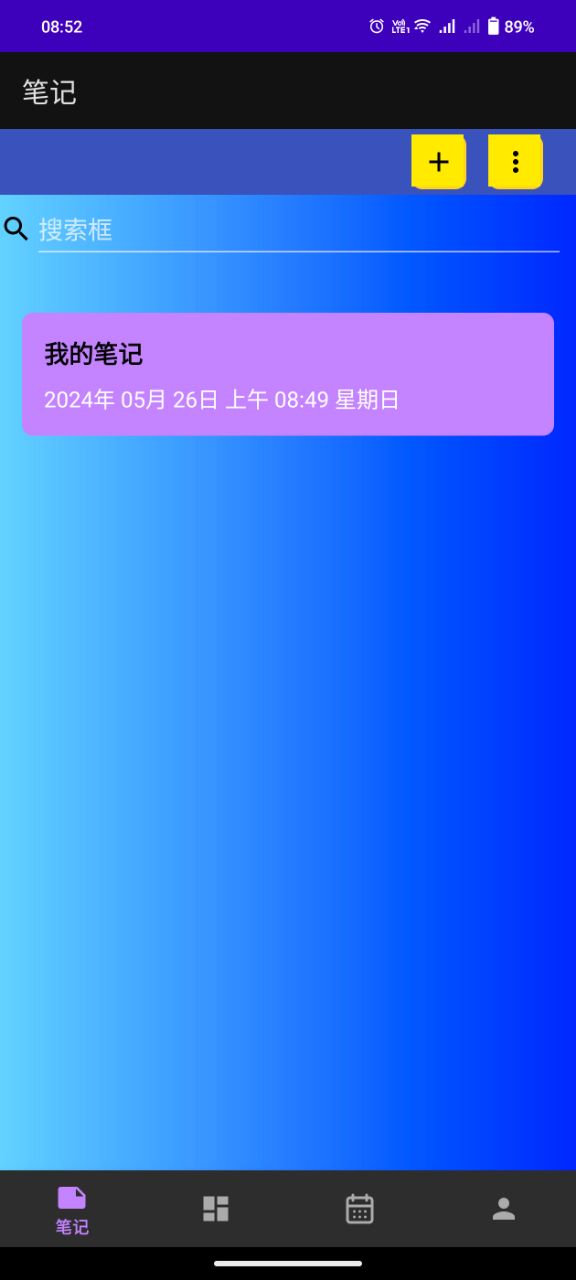
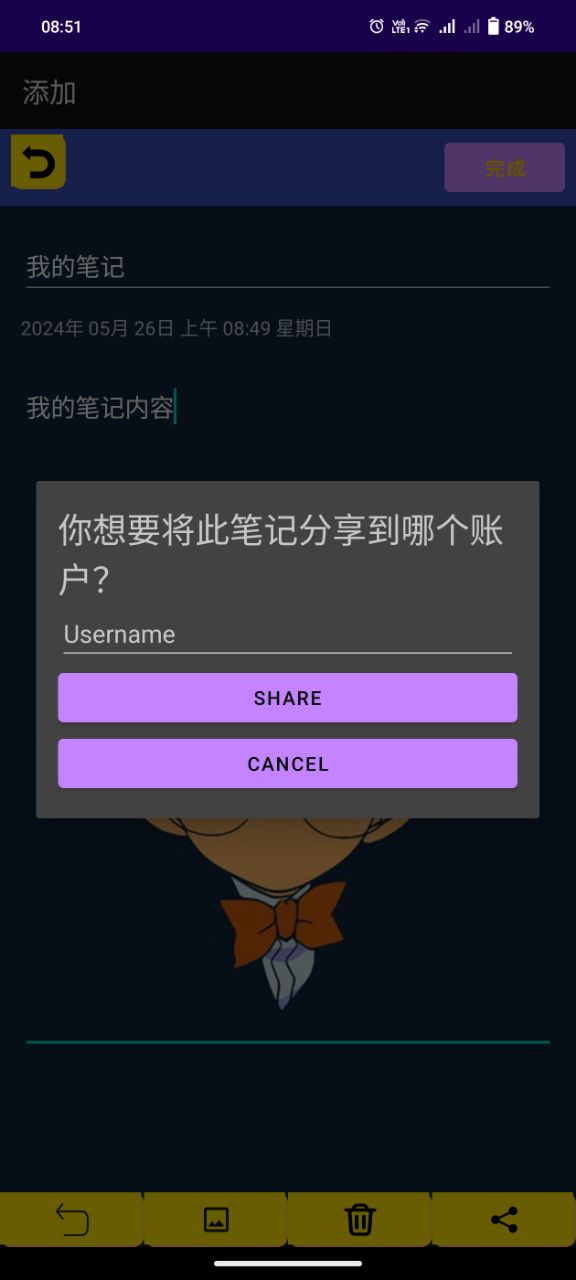
5是更换笔记背景图（仅限于此笔记）

6是清除笔记内容

7是共享当前笔记给其他用户

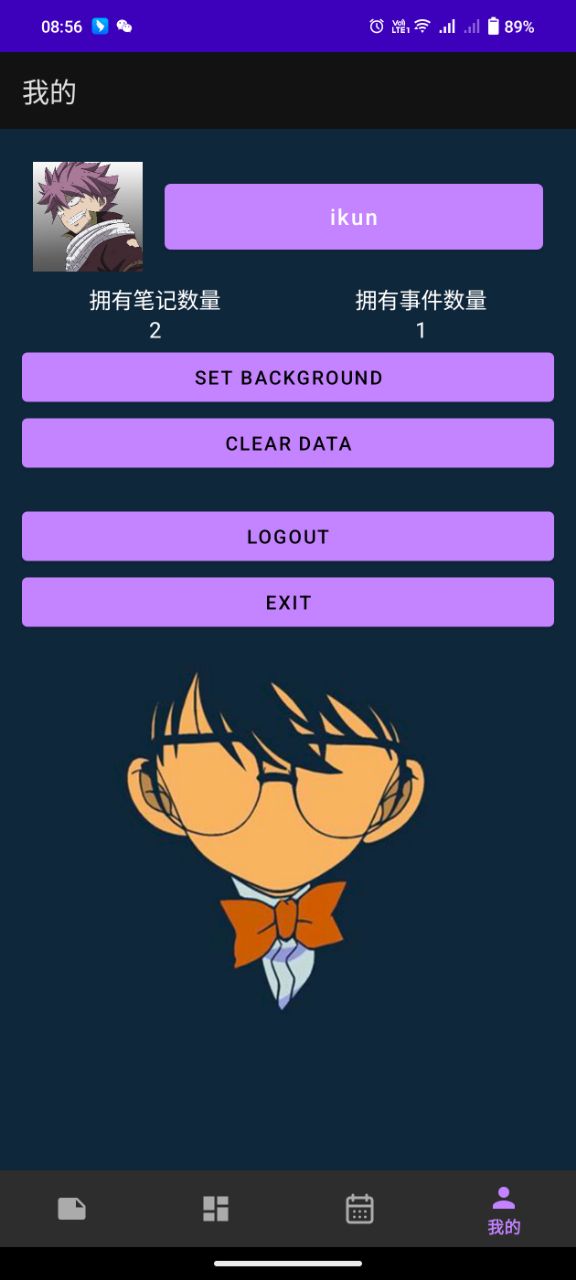
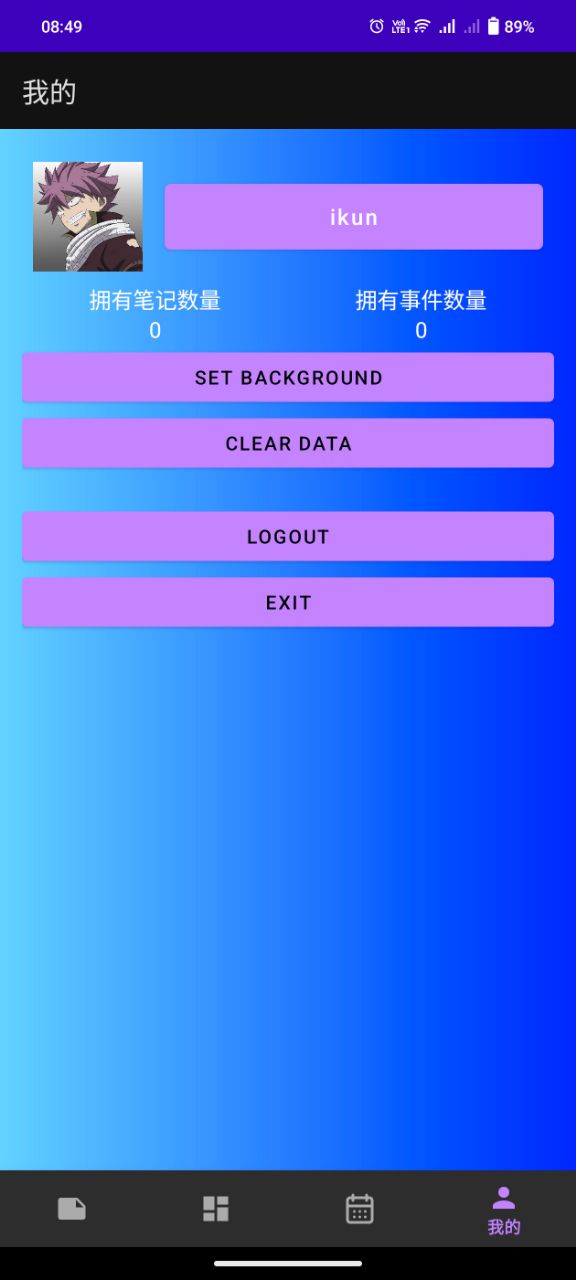
8是保存笔记

点击【完成】后，笔记就算创建成功，如下图所示





上面是清单的创建以及关联到的日历图 下面是笔记的删除以及搜索功能图

 添加头像以及全局背景设置

#### 3 App架构设计及技术实现方案

此应用的架构设计分为数据库结构、应用入口、笔记活动结构、笔记编辑活动结构、事件活动结构、日历活动结构、我的活动结构。

**数据库（Room）：**

在此应用中创建了三个数据库（NotesDatabase, EventsDatabase, UserDatabase）分别管理三个实体集（Note, Event, User）并分别给予对应的DAO接口（NoteDao, EventDao, UserDao）来进行数据库操作。而数据库的使用，在这里创建了一个DatabaseViewModel类来进行管理使用，并且类中包含了当前登入的用户名成员变量。

**应用主入口（MainActivity）：**

在该模块中，设置了应用打开时的布局。并且利用了SharedPreferences类获取上一次退出应用时，应用是否处于登入状态，用户是否设置了背景，从而恢复上次的状态。并设置了底部导航栏监听器，让用户点击导航栏切换到相应页面。

**笔记主页（HomeFragment、NotesAdapter、EditNoteFragment）：**

笔记主页的设计主要涉及数据的显示和用户交互，这包括从数据库读取数据并在界面上显示，以及处理用户的输入（如添加、编辑、删除笔记）。下面概述算法设计和技术实现：

1. **数据读取和显示**

从数据库中获取当前用户所拥有的笔记数据，通过编写笔记容器放入RecycleView中显示笔记。

1. **笔记的添加/修改**

算法设计介绍，设置添加按钮点击监听器（添加笔记）和RecyecleView的Item点击监听器（修改笔记），触发时跳转到笔记编辑页面（EditNoteFragment），如果是点击已存在笔记进入编辑页面，则在进入前使用Bundle将点击的笔记信息也传送到编辑页面中，编辑页面布局初始化先判断是否存在保存的Bundle信息，是则设定笔记内容，否则初始化为新页面，编辑页面算法设计有设置笔记背景、撤销背景、清空内容、分享笔记、保存笔记、返回

**设置笔记背景实现：**应用请求读取相册权限，获得图片Uri并用MediaStore.Images.Media.getBitmap获得Bitmap再转换成Drawable设定背景。

**撤销背景实现：**先检查用户之前是否设置了应用背景，是则采用用户所设置背景，否则采用应用初始背景。

**清空内容实现：**将内容EditText设置为空

**分享笔记：**让用户输入分享到的用户名，检查用户数据库是否存在该用户，否则拒绝分享，是则将当前笔记保存一份至所填入的用户数据库中。

**保存笔记：**获取当前页面标题文本、时间文本、内容文本以及当前登入的用户名，保存到数据库中，完成后退出EditNoteFragment，回到HomeFragment。

**返回：**回到HomeFragment

1. **笔记的删除**

对每个RecycleView的Item设置长按监听器（setOnLongClickListener），然后弹出确认操作框，再对数据进行删除。

1. **笔记标题搜索**

在搜索栏设置文本改变监听器（addTextChangedListener），然后对当前用户的数据库进行关键字筛选，具体语句为SELECT \* FROM notes WHERE user = :user AND title LIKE '%' || :keyword || '%' 。当中keyword为关键字，user为当前用户。

1. **笔记页清除数据**

笔记也设置了更多选项按钮，并设置点击监听，触发时利用PopupMenu方法弹出更多选项框，当中包含了清除数据，点击后弹出确认框、刷新页面，点击刷新页面。

**事件主页（EventsFragment、EventsAdapter）：**

事件主页主要设计了事件的添加、修改、删除功能，以及显示事件并提供倒数日期。

在EventsFragment中主要用于管理事件的添加修改删除，点击新增按钮弹出添加时间窗口，窗口包含标题栏和日历供用户设定标题和选择事件发生日期，点击确认后保存事件到数据库中。在页面中点击事件，事件包含三个监听器（完成勾选框监听、点击监听、长按监听），勾选框改变会触发更新事件状态并更新到数据库中，点击监听出发，会弹出编辑事件窗口，标题栏显示原先标题，用户可重新编辑并选择日期进行保存。长按事件将会弹出删除事件确认框，点击确认便删除当前用户名下的数据库对应的事件。

在EventsAdapter中主要用于事件的显示样式、日期倒数计算功能。当中绑定了itemview的布局绑定。获得事件时，设定好显示的标题、倒数日期。日期倒数采用了对Date转换到LocalDate，toInstant().atZone(ZoneId.systemDefault()).toLocalDate()，并使用ChronoUnit.DAYS.between()进行天数差。

**日历主页（NotificationsFragment、NotificationsAdapter）：**

在NotificationsFragment中，主要算法设计包含，UI初始化、事件获取和展示、日历查看、日历回到今天。通过Android组件Calendarview 提供用户查看日历，并设置了日期改变监听器，每当日期改变时，获取当前日期月份，从数据库中筛选该月份内的所有事件并显示。当中还设置了一个按钮，监听点击事件，触发时利用Calendarview.date设置回今天日期。

在 NotificationsAdapter中，主要算法设计包含，RecycleView的插入绑定、并设定事件显示的样式。当中使用了SimpleDateFormat将事件的时间转换为用户可读日期的样式。

**我的主页（ProfileFragment）：**

在 ProfileFragment 中，主要用于显示和管理用户个人资料。当中包含了用户的登录、数据清除、注册功能，并允许用户设置个人头像和背景图片。下面简要介绍其中的算法设计和技术实现。

1. 用户登录和注册

用户点击登录按钮后，弹出一个登录对话框。用户输入用户名和密码并提交后，进行数据库查询验证。如果验证成功，则更新用户状态为已登录，并刷新界面显示用户信息。

用户点击注册按钮后，弹出一个注册对话框。用户输入用户名、密码和确认密码，并提交。系统检查用户名是否已存在以及密码是否一致。如果验证通过，则在数据库中插入新用户。

用户点击登出按钮后，登出账号并刷新当前应用的登入状态。

2. 图像选择和保存

用户点击头像或背景按钮后，检查权限（如读取存储权限）。如果权限已授予，启动图片选择器；否则，请求权限。

用户选择图像后，保存图像的 URI 到数据库（对于头像）或 SharedPreferences（对于背景）。更新用户界面以显示新选择的图像。

3. 数据获取

进入页面后对数据库进行操作，找出当前用户的所有笔记和事件数量，并显示到页面中。

4. 数据清除

点击数据清除按钮后，对数据库操作，清除所有当前用户名下的所有笔记和事件。



#### 4 技术亮点、技术难点及其解决方案

**技术亮点：**

1. **多数据库管理**

通过使用多个数据库（NotesDatabase, EventsDatabase, UserDatabase），实现了对笔记、事件和用户数据的独立管理，提高了数据的分离性和模块化设计的灵活性。每个数据库都有独立的DAO接口，简化了数据库操作，使代码更清晰可维护。

使用Room数据库进行数据持久化，确保数据的安全性和持久性。使用协程（Coroutines）进行异步数据库操作，避免阻塞主线程，提高应用的响应速度和用户体验。

1. **用户交互体验**

通过RecyclerView和自定义Adapter，提供了高效的笔记和事件列表展示，支持快速滑动和交互。

使用AlertDialog和DatePicker等组件，提供直观的添加和编辑界面。

实现了事件倒计时功能，通过日期计算展示事件的剩余或超时天数，增加用户的计划性。集成了CalendarView，允许用户方便地查看和管理每月的事件。

1. **个性化设置**

支持用户头像和背景图片的自定义设置，提升应用的个性化和用户粘性。

**技术难点及其解决方案：**

1. **难点：**管理多个数据库时，确保它们之间的数据一致性和同步性是一个挑战。

**解决方案：**使用ViewModel集中管理数据库操作，通过ViewModel中的方法进行统一调用，确保各模块的数据操作一致性。

1. **难点：**处理异步数据库操作时，避免内存泄漏和确保UI更新的及时性是一个挑战。

**解决方案：**使用lifecycleScope启动协程，确保协程在Fragment或Activity的生命周期范围内执行。通过withContext(Dispatchers.Main)在主线程中更新UI。

1. **难点：**在RecyclerView中展示大量数据时，保持滑动流畅和高效的数据更新是一个挑战。

**解决方案：**通过ViewHolder模式实现View的复用，减少不必要的View创建。使用notifyDataSetChanged等方法高效更新数据。

1. **难点：**在笔记和事件的添加、编辑、删除过程中，确保用户操作的正确性和系统状态的一致性是一个挑战。

**解决方案：**通过AlertDialog提供确认对话框，确保用户操作的确定性。使用Bundle传递数据，确保在Fragment之间传递数据的完整性和准确性。

1. **难点：**处理用户头像和背景图片的选择和保存时，确保权限的正确管理和图像的高效处理是一个挑战。

**解决方案：**使用ActivityCompat.requestPermissions方法请求权限，确保权限的安全管理。通过MediaStore.Images.Media.getBitmap获取Bitmap对象，并转换为Drawable设置背景，确保图像处理的高效性和正确性。

1. **难点：**准确计算事件的剩余天数或超时天数，并实时更新展示是一个挑战。

**解决方案：**使用Java 8的时间API，通过

toInstant().atZone(ZoneId.systemDefault()).toLocalDate()方法将Date对象转换为LocalDate对象，并使用ChronoUnit.DAYS.between方法计算日期差异。

**与其他类似App的区别：**

1. **多功能集成：**相比单一功能的笔记或事件管理App，此应用集成了笔记、事件和日历管理功能，提供了全面的时间管理解决方案。
2. **用户个性化设置：**支持用户自定义头像和背景图片，增加了个性化和用户参与度。
3. **数据共享功能：**支持笔记的跨用户分享，提升了协作和交流的便利性。

#### 5 简要开发过程

4月27日 搭建了项目框架

4月28日 对框架修改了格式

5月1日 初步数据库连接

5月5日 完成将笔记保存进数据库功能

5月6日 页面渲染功能

5月10日 新增了日期模块

5月19日 日期模块页面增删改查功能

5月20日 日历功能显示

5月22日 新增修改背景，修改头像功能

5月23日 新增注册登录功能

5月24日 页面美化，对程序进行集成测试

5月25日 程序开发工作完毕

5月26日 撰写文档

#### 6 学习感悟及对本课程的建议（可选）

经过一个学期的理论与技术学习和亲自动手开发，你有何感悟？对本课程的教学有何建议？

经过一个学期的学习和实践，我对Android开发有了更深入的理解，尤其是在使用Kotlin语言进行笔记本应用开发的过程中，我们体会到了现代编程语言的简洁性和强大功能。从项目框架的搭建到功能的逐步完善，我们学会了如何将理论知识应用到实际开发中，同时也认识到了持续学习新技术的重要性。在开发过程中，我们深刻感受到了代码结构和设计模式对于维护和扩展应用的重要性。此外，团队合作的经验让我认识到了协作的力量，它不仅提升了项目的效率，还让我学会了如何与他人沟通和解决冲突，这对于任何软件开发项目都是至关重要的。我们都很喜欢本课程的实践环节，比起被动接受，这种主动学习是远远更有收获的。