

总成绩

《移动平台开发技术》

课程结业大作业软件设计说明书



任课教师：王 峥

指导教师：王 峥

班 级：1611101

姓 名：李洋，臧博浩，姚越，
侯禹驰，李旭，张立才

计算机科学与技术学院软件学院

哈尔滨工业大学课程结业大作业任务书

姓 名： 李洋	院（系）： 软件工程
专 业： 软件工程	班 号： 1611101
任务起至日期： 2018 年 4 月 28 日至 2018 年 5 月 9 日	
课程设计题目：“研途”软件设计与实现	
<p>已知技术参数和设计要求：</p> <p>根据已经学过的 Android 的知识和技能，设计并实现一个基于 Android 平台下应用程序。基本要求如下：</p> <ol style="list-style-type: none"> 1.软件可以是手机小型应用工具(计算器、记事本等)、小型数据库管理系统(个人理财系统、个人课程管理、通信录管理系统等)和图形数字化游戏(挖地雷、消消乐等)。 2.对软件进行详细描述。包括软件的所有功能、性能、软件的界面。 3.进行详细规范的需求分析和软件设计。可使用面向对象或面向过程的分析、设计及描述方法，对程序进行分析和设计。分析要详细，图表要规范。 4.在 Android Studio3.2 For Windows 环境下用 Java 语言完成软件的编程。 5.对完成的程序进行测试，并记录测试过程。 6.软件以组为单位完成，完成的软件要完整，工作量 200 行代码/人。要明确标注出来。 	
工作量：2 周。	
<p>工作计划安排：</p> <ol style="list-style-type: none"> 1.第 9 周周四 7-8 节：布置任务，开始检索资料。 2.第一次检查：第 10 周周一晚 24：00 前将包括：软件简介：包括软件的详细介绍和各项指标的 报告提交至蓝墨云班课：本次检查为大作业成绩的 10%。 3.第二次检查：第 10 周周五晚 24：00 前将包括需求分析和总体设计的报告提交至蓝墨云班课， 本次检查占总成绩的 40%。 <ol style="list-style-type: none"> (1) 需求分析：包括功能分析、数据分析和基本流程分析，并绘制出相应的图表。 (2) 软件总体设计：软件模块的划分；每个模块的功能设计；包括每模块基本算法设计，要求绘制相应的图表。 4.结题检查：第 11 周周三晚 24：00 前将完成的程序工程压缩提和完成的大作业报告提交至蓝墨云班课。按时提交结题文档点大作业成绩的 20%。 5.结题验收答辩：第 11 周周四 5-10 节：携带打印出的纸制作业报告和装有作业程序的电脑，按结题答辩顺序，在 507 实验室进行答辩，结题检查占作业成绩的 30%。 	
<p>同组设计者及分工</p> <p>161110119 姚越：负责主界面。包括考研倒计时、显示今日习惯、打卡功能。</p> <p>161110118 侯禹驰：负责日常习惯界面。包括所有日常习惯的显示。</p> <p>161110114 臧博浩：负责日常习惯详细信息的显示。包括显示打卡日历、打卡日志。</p> <p>161110117 李洋：负责编辑习惯界面。包括日常习惯的添加、修改功能。</p> <p>161110108 李旭：负责番茄钟界面。包括番茄钟计时功能。</p> <p>160700108 张立才：负责设置界面、启动界面。包括软件的设置，启动软件欢迎界面的显示。</p>	
<p>指导教师签字_____ 年 月 日</p> <p>教研室主任意见：</p> <p>教研室主任签字_____ 年 月 日</p>	

*注：此任务书由课程设计指导教师填写。

一、软件简介

1) 软件概述

名称：研途

详细说明：

“研途”是一个面向考研同学的简单的习惯追踪、管理自己时间的应用。是一种习惯养成类型的应用。

“研途”可以进行的功能有：每日打卡、展示所有自己想养成的好习惯、自定义添加自己想养成的好习惯、查看自己的打卡记录、为自己的习惯设定番茄钟。

2) 软件的背景、现状和核心技术：

国内外不乏这类软件，例如国内的“天天”“番茄打卡”等。这些应用并不是为考研同学量身定制的，因此缺少了考研同学需要的一些功能。我们准备以现有的软件为参照，进行新的改进，将习惯养成、考研倒计时与“番茄钟”联系在一起。相比于其他同类型软件，该软件功能更完备、操作更简便。软件运用了 Android 开发的基础知识。

二、需求分析与软件设计

1) 需求分析

软件功能叙述：

- ① 显示日常习惯。可以查看习惯名称与习惯坚持天数。
- ② 添加新的习惯。可添加信息有：名称、习惯图标、时间（一天中进行习惯的时间，分为任意时间、上午习惯、下午习惯、晚间习惯）、番茄钟周期数。
- ③ 查看日常习惯详情。可以查看当月日历，“打卡日”会高亮显示。可以查看打卡日志，是对“打卡日”的统计，有“总共坚持 x 天”“连续坚持 x 天”。
- ④ 修改习惯。
- ⑤ 删除习惯。
- ⑥ 开启番茄钟。在日常习惯页面，在点击每个习惯右侧的按钮即可以开启番茄钟。
- ⑦ 显示今日习惯。今日习惯页面包括考研倒计时、今日习惯打卡。今日习惯与日常习惯的种类一致，仅负责打卡功能，点一下习惯图标就可以进行打卡。
- ⑧ 设置。可以设置是否显示通知、是否开启振动等。

核心数据结构的分析叙述：

- ① 在今日习惯页面（TodayHobbyActivity 类），用到的核心数据结构有 ArrayList<Hobby>，HashSet<Integer>。

TodayHobbyActivity 类的主要功能是处理打卡逻辑。

a. ArrayList<Hobby>：读取数据库中所有的 Hobby，存入 Hobby 列表中。

b. HashSet<Integer>：读取每个 Hobby 是否已经打卡，需要查找 ClockIn 表，获得当日打卡信息。将已打卡的 Hobby Id 存入集合中。

进行显示时，需要遍历 ArrayList<Hobby>中的每一个 Hobby，如果 Hobby 的 Id 在集合中，需要将该 Hobby 对象中的图片路径更改为加入签到标识的图片路径。

- ② 在习惯详情页面（HobbyDetailActivity 类），用到的核心数据结构有 ArrayList<String>。

显示当月打卡日历需要存入已打卡的日期，因此需要存入“年-月-日”。

用例图：

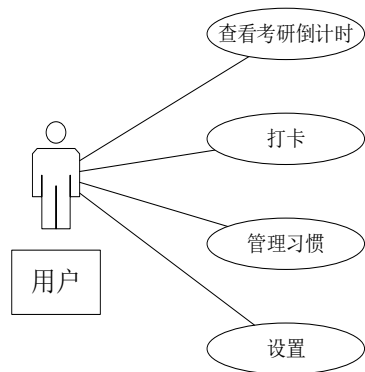


图-1 用例图

描述：

用例 1：查看考研倒计时（基本路径）

参与者：用户

表-1 查看考研倒计时（基本路径）用例的步骤

步骤	事件
1	打开应用
2	查看界面上面显示的倒计时

用例 2：打卡（基本路径）

参与者：用户

表-2 打卡（基本路径）用例的步骤

步骤	事件
1	打开应用
2	选择要打卡的习惯
3	点击习惯
4	软件提示“打卡成功”

用例 2.1：打卡，重复打卡（可选路径）

参与者：用户

表-3 打卡（可选路径）用例的步骤

步骤	事件
1	打开应用
2	选择要打卡的习惯
3	点击习惯
4	软件提示“请勿重复打卡”

用例 3：管理习惯（基本路径）

参与者：用户

表-4 管理习惯主界面（基本路径）用例的步骤

步骤	事件
1	打开应用
2	点击底部导航栏“日常习惯”，进入日常习惯页面

表-5 查看习惯详情（可选路径）用例的步骤

3A	点击想要查看的习惯，进入习惯详情页面，查看习惯的各种信息
----	------------------------------

表-6 添加习惯（可选路径）用例的步骤

3B	点击日常习惯页面右上角添加按钮进行新的习惯的添加
----	--------------------------

表-7 修改习惯（可选路径）用例的步骤

3C	在习惯详情页面点击右上角的修改对习惯进行修改
----	------------------------

表-8 删除习惯（可选路径）用例的步骤

3D	在习惯详情页面点击右上角的删除习惯进行删除
----	-----------------------

用例 4：设置（基本路径）

参与者：用户

表-9 设置（基本路径）用例的步骤

步骤	事件
1	打开应用
2	点击右上角的设置按钮
3	选择要设置的选项
4	点击右上角返回主页面

领域模型图：

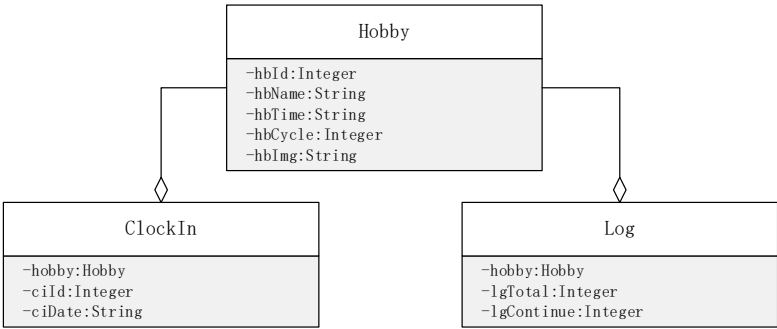


图-2 领域模型图

描述：

在这个软件系统中有三个实体类，分别是习惯类 **Hobby**、打卡类 **ClockIn**、打卡记录类 **Log**。其中习惯类中包含 4 条属性分别是习惯的 id、习惯的名称、习惯的常用时间、习惯的番茄钟周期。打卡类中包含 3 条属性，分别是习惯类的对象、打卡的 id、打卡的日期。打卡记录类包含三条属性，分别是习惯类的对象、习惯的总打卡天数、习惯的连续打卡天数。

功能模块图：

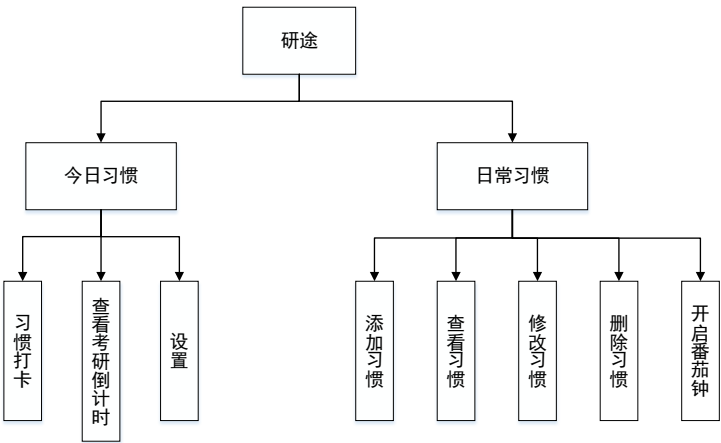


图-3 功能模块图

描述：

“研途”软件分为两个模块，分别是今日习惯模块和日常习惯模块。

今日习惯模块包括显示考研倒计时、点击自己设定的习惯进行打卡、进行系统的相关设置功能。

日常习惯模块包括习惯的增删改查。另外，每个设定了番茄钟周期的习惯均有开启番茄钟的功能。

开发的硬件（配置）、软件(版本)环境及工具(版本)：

硬件配置：CPU: intel Core i5 及以上；内存：4G 及以上；硬盘 100G 及以上；系统：Windows 10。

软件环境：Android 5.0 及以上。

开发工具：Android Studio 3.2 For Windows。

2) 软件设计

分层用例图：

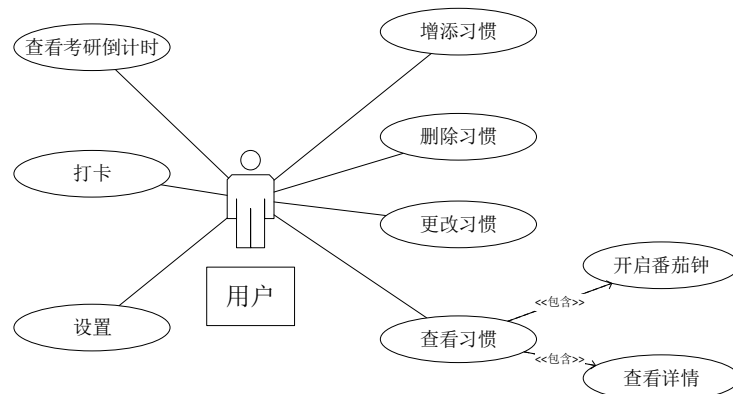


图-4 分层用例图

用例 1：增添习惯（基本路径）

参与者：用户

表-10 增添习惯（基本路径）用例的步骤

步骤	事件
1	打开应用
2	点击底部导航栏“日常习惯”，进入日常习惯页面
3	点击右上角添加按钮
4	填写新建的习惯的各种信息
5	点击添加按钮，完成添加，返回主页面

用例 2：删除习惯（基本路径）

参与者：用户

表-11 删除习惯（基本路径）用例的步骤

步骤	事件
1	打开应用
2	点击底部导航栏“日常习惯”，进入日常习惯页面
3	点击想要删除的习惯
4	进入习惯的详情右上角的删除按钮
5	确认删除
6	完成习惯的删除，返回主页面

用例 3：更改习惯（基本路径）

参与者：用户

表-12 更改习惯（基本路径）用例的步骤

步骤	事件
1	打开应用
2	点击底部导航栏“日常习惯”，进入日常习惯页面
3	点击想要更改的习惯
4	进入习惯的详细信息页面，点击右上角的编辑按钮
5	点击修改按钮，完成修改，返回主页面

用例 4：查看习惯（基本路径）

参与者：用户

表-13 查看习惯（基本路径）用例的步骤

步骤	事件
1	打开应用
2	点击底部导航栏“日常习惯”，进入日常习惯页面
3	点击想要查看的习惯
4	进入习惯的详细信息页面，查看习惯

用例 5：开启番茄钟（基本路径）

参与者：用户

表-14 开启番茄钟（基本路径）用例的步骤

步骤	事件
1	打开应用
2	点击底部导航栏“日常习惯”，进入日常习惯页面
3	点击想要开启的番茄钟
4	进入番茄钟页面，点击开始，开启番茄钟。
5	番茄钟周期结束，番茄钟停止。

用例 5.1：开启番茄钟，番茄钟运行中退出程序（异常路径）

参与者：用户

表-15 开启番茄钟（异常路径）用例的步骤

步骤	事件
4	进入番茄钟页面，点击开始，开启番茄钟。
5	番茄钟运行过程中退出程序，番茄钟在后台运行
6	打开应用，番茄钟倒计时继续

用例 5.2：开启番茄钟，番茄钟运行中杀死程序（异常路径）

参与者：用户

表-16 开启番茄钟（异常路径）用例的步骤

步骤	事件
4	进入番茄钟页面，点击开始，开启番茄钟。
5	番茄钟运行过程杀死程序
6	番茄钟停止运行，程序完全关闭

完整类图：

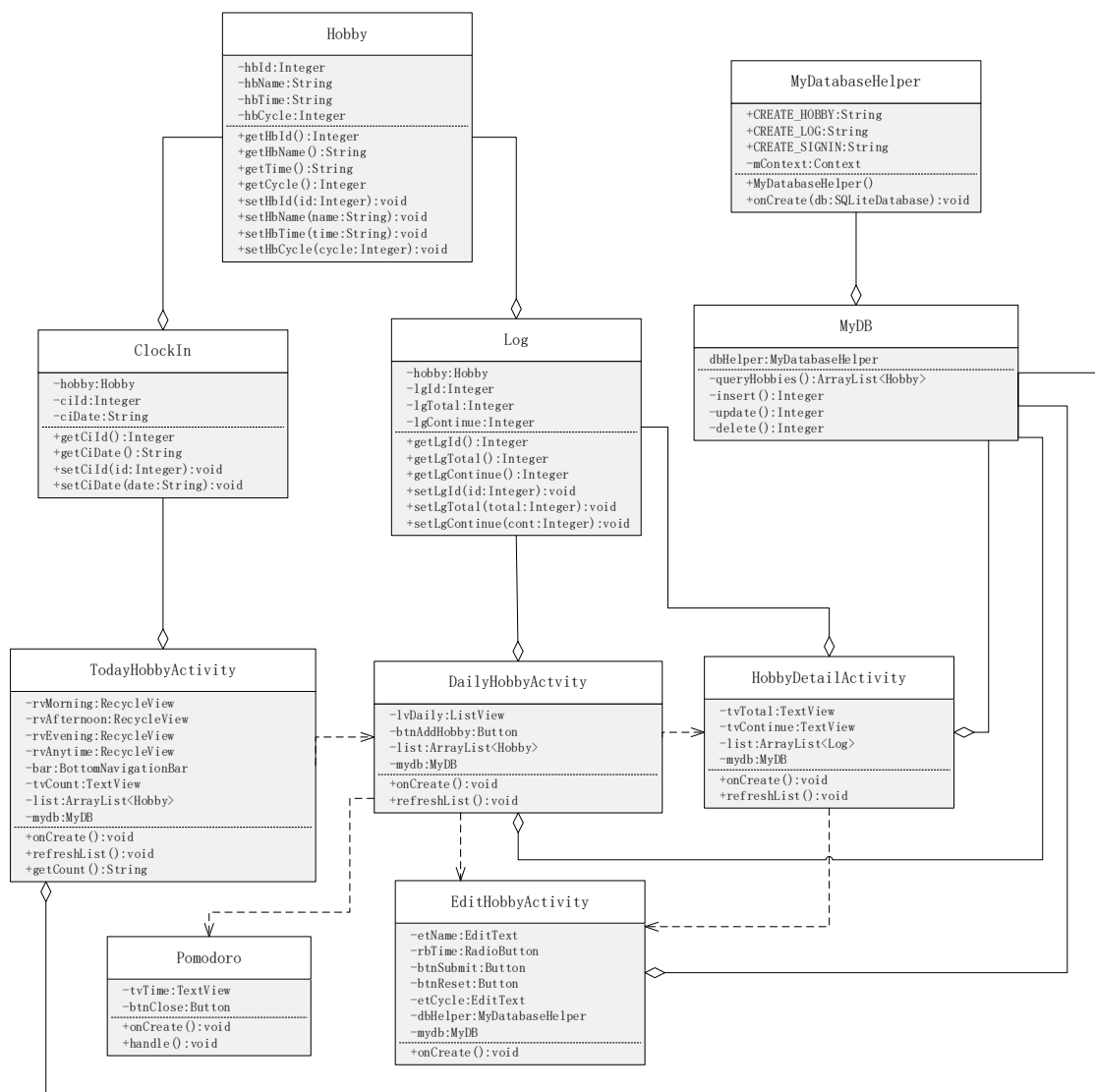


图-5 类图

实体类：

Hobby（习惯类），**ClockIn**（打卡类），**Log**（打卡日志类）。每个实体类的属性与数据库的字段相对应，每个属性都有 `get`、`set` 方法。

界面与逻辑类：

- ① **ToadyHobbyActivity**（今日习惯类）。处理打卡、显示考研倒计时的逻辑。与日常习惯界面、设置界面进行交互。函数主题逻辑在 `onCreate()` 函数内。该类包含了 `ClockIn` 和 `MyDB` 的对象，因此和 `ClockIn` 类、`MyDB` 是聚合关系。该类与 `DailyHobbyActivity` 类有界面交互，存在依赖关系。
- ② **DailyHobbyActivity**（日常习惯类）。处理显示所有习惯、开启番茄钟的逻辑。与习惯详情界面、习惯编辑界面进行交互。同上，存在聚合关系与依赖关系。
- ③ **HobbyDetailActivity**（习惯详情界面）。处理显示打卡日历、打卡日志的逻辑。与习惯编辑界面进行交互。
- ④ **EditHobbyActivity**（编辑习惯界面）。处理添加习惯、删除习惯的逻辑。
- ⑤ **PomodoroActivity**（番茄钟界面）。处理番茄钟的开启、关闭、计时。

数据库操作类：

MyDatabaseHelper 类：继承了 DatabaseHelper 类。

MyDB 类：封装了所有数据库操作。

时序图：

① 查看考研时间

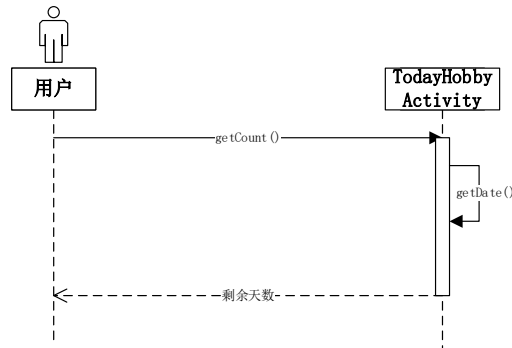


图-6 查看考研时间时序图

用户打开应用，通过 TodayHobby 类中的 getCount()函数获取到考研剩余的天数，然后显示在主页上。

② 编辑习惯

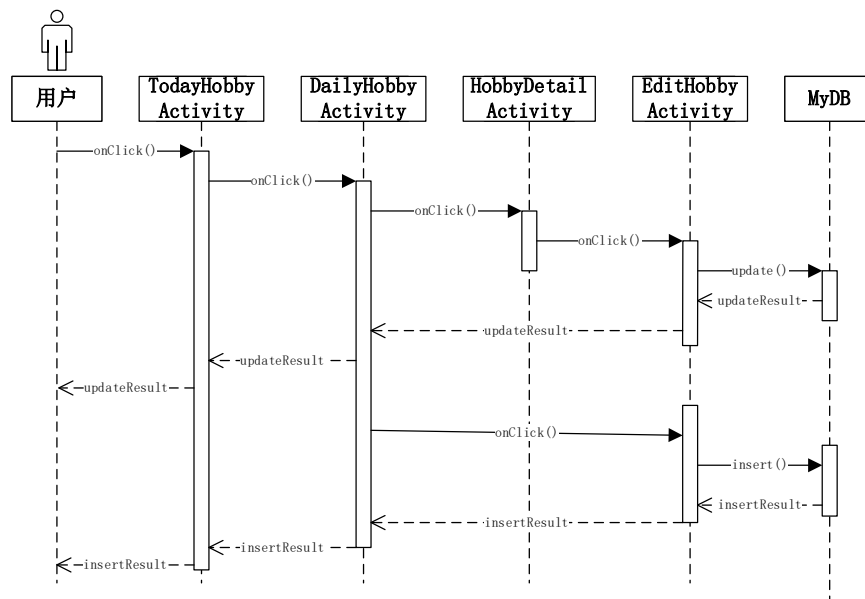


图-7 编辑习惯时序图

用户首先进入应用看到 TodayHobbyActivity 页面，点击按钮进入 DailyHobbyActivity 页面，再点击按钮进入 HobbyDetailActivity 页面展示习惯的详细信息，然后再点击右上角的按钮进入 EditHobbyActivity 习惯的编辑页面，在编辑页面编辑习惯的详细信息。可以是更改现有的习惯的一些信息，也可以是增添新的习惯，加入或更新到数据库。然后从数据库返回到最初的页面上，显示现有的习惯。

③ 查看习惯

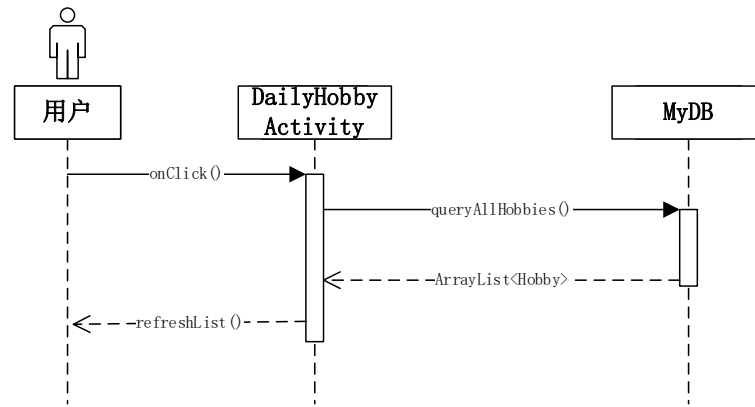


图-8 查看习惯时序图

用户进入应用，点击日常习惯进入 **DailyHobbyActivity** 页面，在 **DailyHobbyActivity** 中通过 **queryAllHobbies()**方法从数据库中读取习惯的信息，然后存到数组中，返回到页面，通过 **refreshList**刷新页面。

④ 查看详细信息

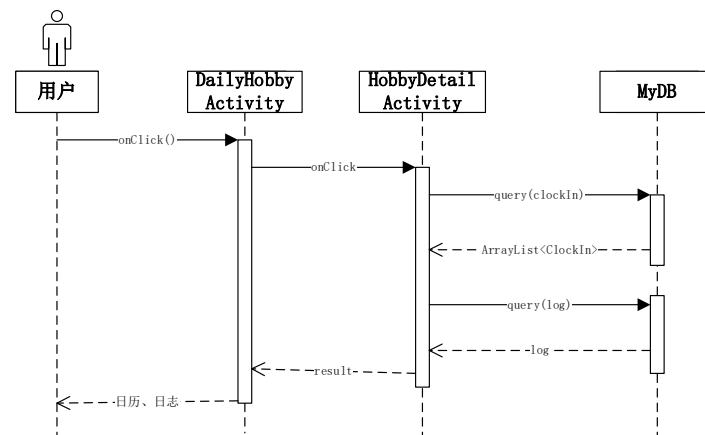


图-9 查看详细信息时序图

用户进入应用，点击进入 **DailyHobbyActivity** 页面在根据想要查看的习惯的详细信息，点击想查看的习惯进入相应的详细信息的页面 **HobbyDetailActivity**，在此页面通过 **query()**函数从数据库中查询相应的信息，将返回的数据存到一个数组中，再在页面上显示出来。

状态图:

① 打卡状态

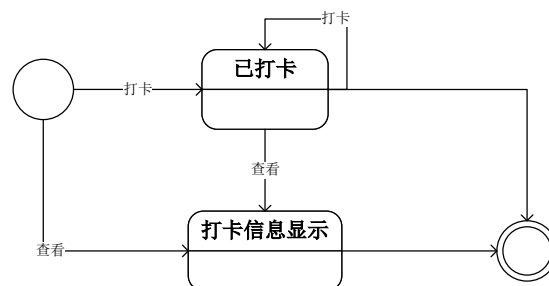


图-10 打卡状态图

当用户打卡时，状态从未打卡变成已打卡；当用户查看打卡信息时，给用户显示打卡的历史信息。

② 习惯状态

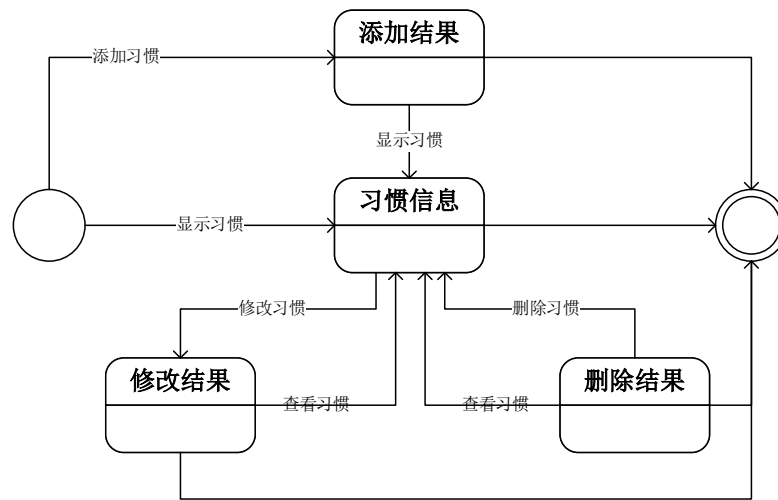


图-11 习惯状态图

习惯的增删改查拥有添加成功、添加失败、修改成功、修改失败、删除成功、删除失败、信息显示状态。

③ 番茄钟状态

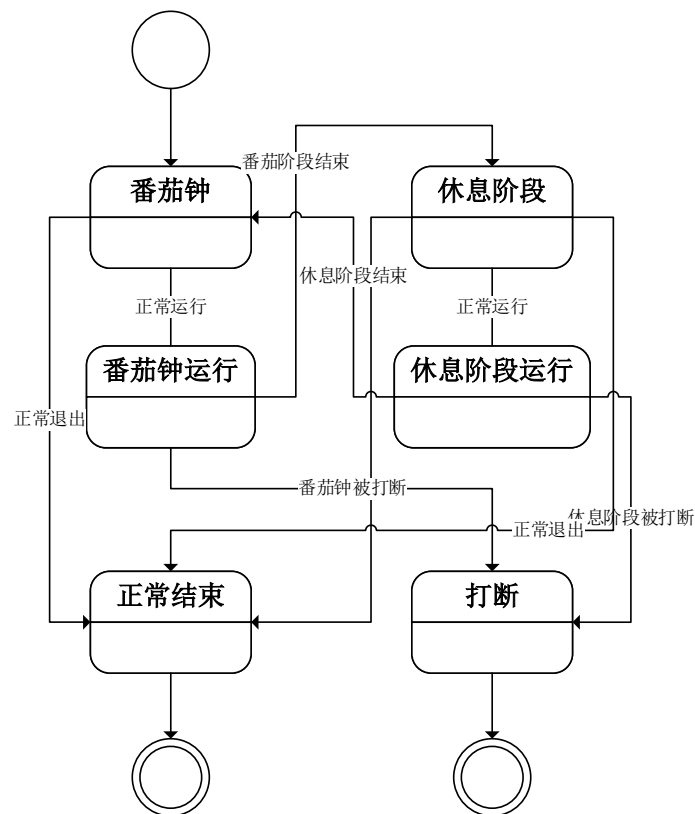


图-12 番茄钟状态图

番茄钟总计有四个状态：番茄周期准备、番茄周期运行、休息周期准备以及休息周期运行。再正常运行状态下，准备状态之后紧接着就是运行状态，而番茄周期和休息周期来回切换。如果在运行周期被打断，则进入打断状态，并提示用户退出；否则，则进入正常结束状态。

活动图：

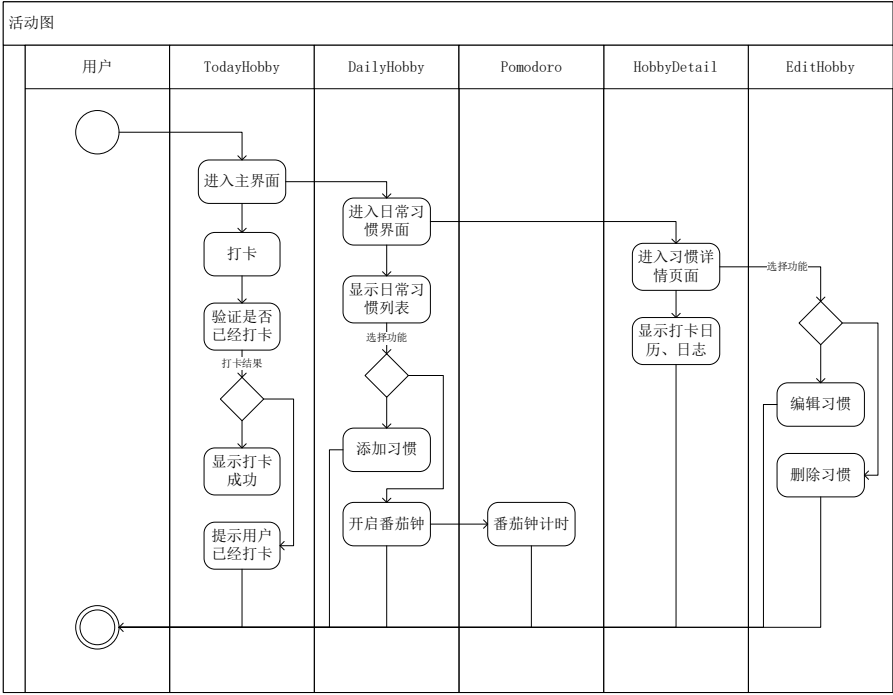


图-13 活动图

描述：

用户首先进入主页面，可以在点击相应的习惯图标进行每日打卡，在用户点击之后要进行验证用户今天是否已经打卡。如果用户已打卡则提示用户已经打卡，否则显示打卡成功。

用户可以点击下方的日常习惯的按钮进入日常习惯的页面，在此页面用户可以看到自己的日常习惯，用户可以点击右上角的按钮进行新的习惯的添加。同时，用户可以点击习惯右方的番茄钟的按钮选择开启番茄钟。开启番茄钟后进入番茄钟计时页面进行计时。

用户在日常习惯页面点击相应的习惯可以进入到此习惯的详细信息页面，在此页面用户可以查看自己的打卡日期和连续打卡天数。同时，用户在此页面也可以进行习惯的编辑和删除。

三、详细设计

流程图及描述：

① 习惯详细信息模块——臧博浩，161110114

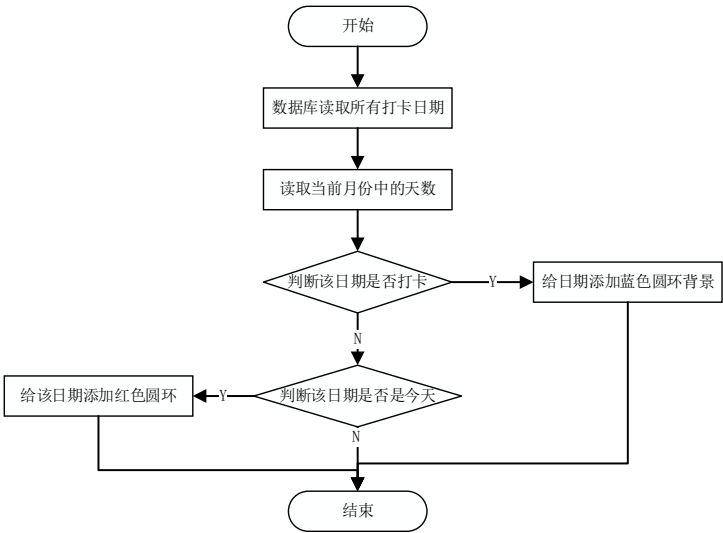


图-14 展示习惯的详细打卡信息

描述：

本页面主要实现的是展示习惯的详细打卡信息。首先，先初始化所要用到的布局文件中的控件，再利用 `getIntent()` 获取从日常习惯页面给我们传递的 `hobby` 对象。通过 `hobby` 对象的 `id` 查询数据库中的 `Log` 表单来查找该习惯的连续签到天数和总共的签到天数，将查询好的数值填写到相应的控件中。之后再查找 `Clockin` 表单查询该习惯的所有签到的日期。利用所签到的日期给日历控件上的相应日期画上蓝色的圆环表示。同样也在日历控件上给今天的日期画上红色的圆环。

② 今日习惯模块——姚越，161110119

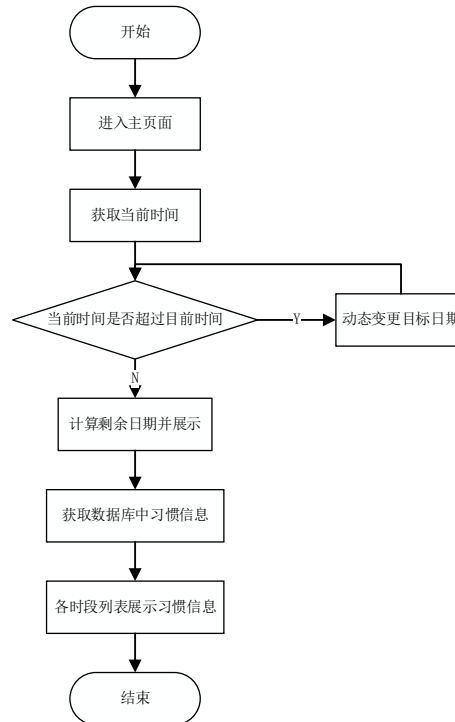


图-15 展示功能流程图

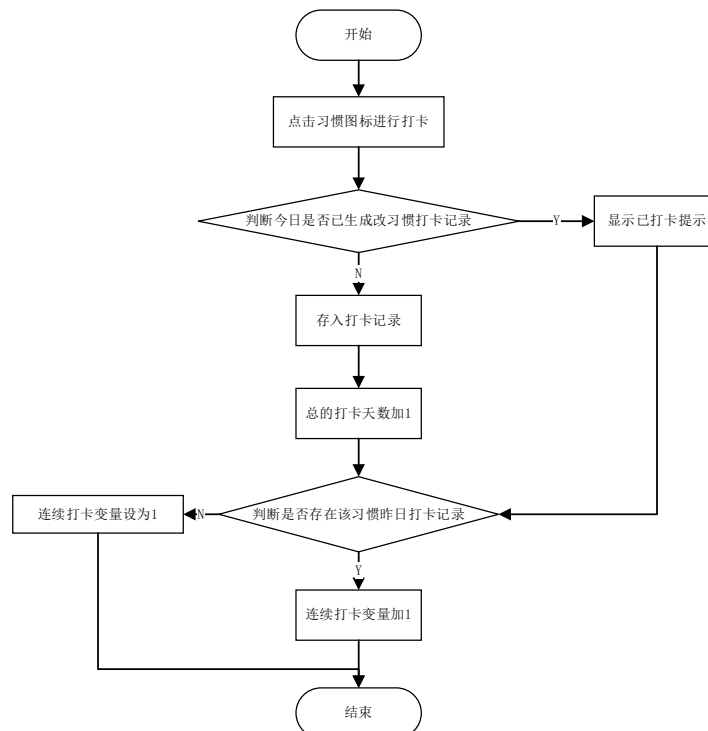


图-16 打卡功能流程图

描述：

主界面展示及打卡模块设计：本部分包含程序主界面内容展示及基本操作，主要包含三个子模块，分别是倒计时获取模块、习惯读取展示模块、还有打卡操作模块。

进入主界面时，程序会获取系统当前时间，与预设的 2019 年考研日期进行比对，计算出当前距离考研日期的天数显示在主界面头部。当获取的当前日期超过了预设目标日期时，程序会进行计算动态改变目标日期再次计算剩余天数，使每一次显示的剩余天数为距离下一次考研时间的倒计时。

主界面的主显示区按照时间划分为四个时段，分别在四个时段各自区域读取数据库中已保存于该时段的若干习惯，每个时段默认展示三个习惯，超过三个习惯可通过左右滑动列表进行查看。在完成每日计划的习惯后可以点击习惯图标进行打卡操作，点击确认打卡后，习惯图标将标记红色对勾，每天只能对每个习惯打卡一次，多次打卡不会重复记录，同时在打卡后还会判断连续打卡天数，存入打卡记录，可在程序其他界面的打卡日历中显示该活动的所有打卡记录。

③ 编辑习惯模块——李洋，161110117

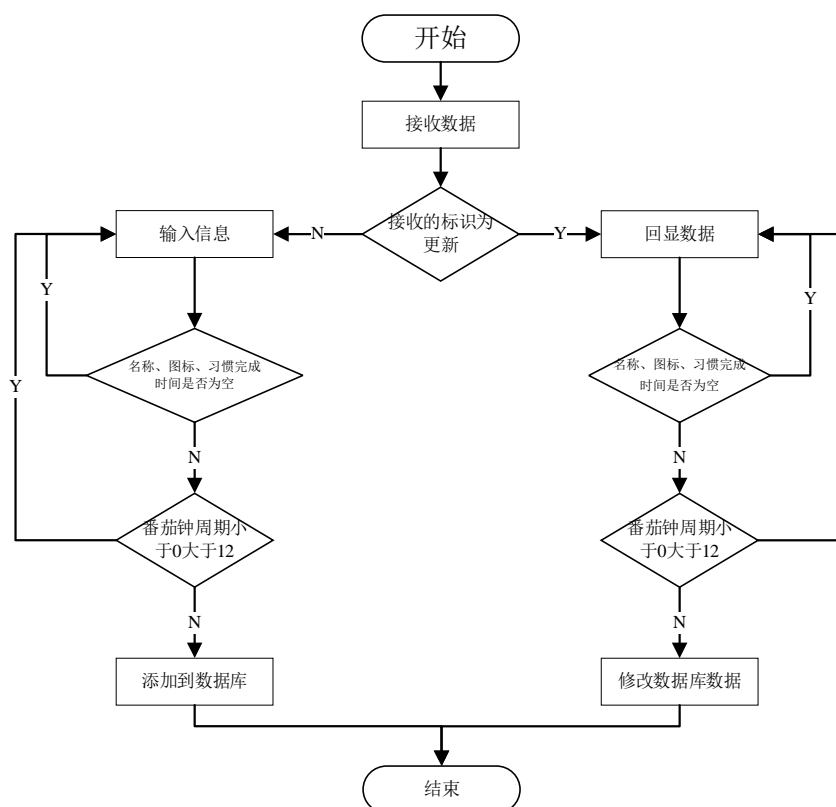


图-17 编辑习惯流程图

描述：

编辑习惯页面将添加与编辑合二为一，通过判断由上一个页面传来的数据判定进行添加操作还是修改操作。接下来修改界面，如果是添加操作，将标题、按钮改为添加；如果是修改，将标题、按钮改为修改。用户输入数据，系统依次进行为空判断、数据合法性判断。如果都准确无误则添加到数据库或修改数据库信息。

④ 设置模块——张立才，160700108

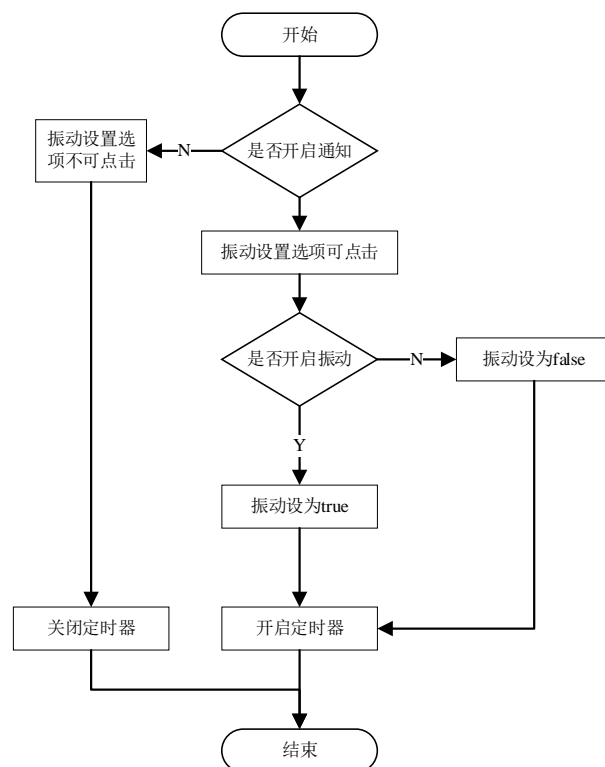


图-18 设置

描述：

本界面可以进行通知、振动的设置。如果开启了通知，那么振动选项可以点击。如果关闭了通知，那么振动选项将不可点击。默认振动选项是关闭的。如果打开了振动，则会重置定时器。

⑤ 日常习惯模块——侯禹驰，161110118

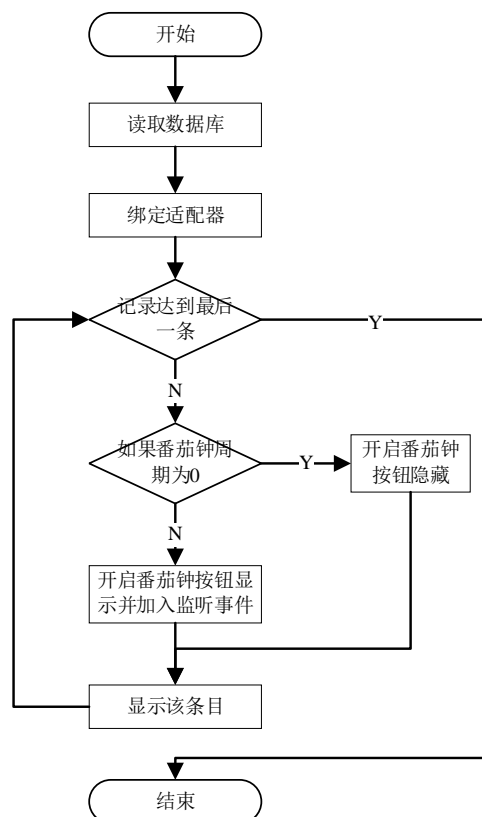


图-19 展示日常习惯列表

描述：

首先在进入该界面时读取数据库，将 RecyclerView 控件与 Adapter 绑定，向 Adapter 传入从数据库读取的习惯列表，Adapter 会依次读取 list<Hobby>每一项，将文本、按钮显示到屏幕上。如果 Hobby 中的番茄钟周期为 0，则不需要显示按钮，因此将其隐藏。如果 Hobby 中的番茄钟周期不为 0，则需要添加点击事件。

四、编码实现

编程记录：

李洋 161110117:

日期	编程记录
2019.04.27	创建工程
2019.04.28	建立整个软件的框架，包括包内的文件夹的创建，以及 Activity 的命名
2019.05.02	编写编辑习惯界面，剩余数据库操作尚未完成
2019.05.05	编辑习惯界面数据库操作完成，但是图片仍然是写死的
2019.05.06	编辑习惯界面添加 RecyclerView，编写 Adapter
2019.05.07	编辑习惯界面完成，并且与其他界面进行交互
2019.05.09	为了配合底部的导航栏，将今日习惯页面、日常习惯页面、设置页面提取到 Fragment 中
2019.05.10	编写定时器
2019.05.11	解决定时器部分 bug，重写返回键
2019.05.12-2019.05.14	综合调试

臧博浩 161110114:

日期	编程记录
2019.05.02	使用 Android studio 自带日历控件，出现无法标记日期问题，尝试寻找开源日历控件
2019.05.03	调试开源日历控件，给特定日期添加背景圆特效。
2019.05.04	寻找图标，绘制详细信息页面，编写实体类
2019.05.05	发现超过特定时间，日历显示出现问题。调试后更改对比项，修复 bug。
2019.05.07-2019.05.08	调试详细习惯页面

姚越 161110119:

日期	编程记录
2019.05.03	进行主界面样式设计，列表窗口布局设计
2019.05.04	进行底部菜单控件设计和功能实现
2019.05.05	添加顶部考研倒计时模块逻辑并显示
2019.05.06	添加横向滑动列表，并对已添加的习惯进行读取显示
2019.05.07	修改图标文件，添加打卡功能并将每日打卡和连续打卡记录写入数据库
2019.05.08	主界面功能调试

侯禹驰 161110118:

日期	编程记录
2019.05.06	完成了日常习惯的基本界面，初步试验了 recycleView 的用法。

2019.05.07	完善了 recycleView 所需的 Adapter 及 recycleView 的格式，试验了在 Adapter 中编写点击事件及数据库操作。
2019.05.08	将对数据库操作封装到指定的文件并编写 Hobby 实体类来存储信息，暂时没找到解决点击事件同时需要 Adapter 及界面信息的方法。
2019.05.09	整合代码，添加了添加习惯按钮，点击事件问题尚未解决。
2019.05.10	解决了点击事件问题，采用了网上查到的注册点击事件方法，在 Adapter 中编写了接口，在 Fragment 中实现接口并回传。完成日常习惯应有功能。

张立才 160700108:

日期	编程记录
2019.05.07	编写设置界面
2019.05.09	编写启动页
2019.05.10	将设置页面调整到 Fragment 中
2019.05.12	进行综合调试

李旭 161110108:

日期	编程记录
2019.04.30	开始整体设计，番茄钟后端调研
2019.05.01	对比 Timer、CountDownTimer 以及 ScheduledExecutorService 方案
2019.05.02- 2019.05.03	完成 ScheduledExecutorService 的 Timer 封装 2019.05.04 开始 TomatoClockActivity 的 UI 设计
2019.05.05 -2019.05.06	完成 TomatoClockActivity 的主逻辑编写
2019.05.09 -2019.05.12	对 TomatoClockActivity 进行完善以及联调

问题与解决方案:

李洋 161110117:

- ①定时推送通知功能在编程时遇到了很多问题。定时功能需要用到 Android 的系统服务之定时服务——AlarmManager，以及特殊的 Intent——PendingIntent。PendingIntent 用来发送广播，AlarmManager 用于定时执行 PendingIntent。代码编写完后出现了安卓版本不兼容的问题，模拟器（Android 5.0）可以成功运行但是真机（Android 8.0，Android9.0）不能运行。考虑到课上 Notification 功能从 Android8.0 开始使用渠道，我想到高版本 Android 一定也对 AlarmManager 进行了限制。由于软件需要的是一个每天定时执行的任务，所以我想采用 AlarmManager 的重复定时函数 setRepeating()，但是查阅资料发现该函数高版本不能使用。因此我只能使用单一定时函数 set()，采取当收到一次广播时重置这个定时器的方法，做到重复执行的效果。PendingIntent 在高版本同样有限制，必须显式声明发送广播的类与接收广播的类。解决了这两个问题，高版本的 Android 也可以定时推送通知了。
- ②底部导航栏切换时不流畅问题。安卓切换 Activity 时会有自底向上弹出的特效，导致底部导航栏每次切换都会弹出一。我们预想的结果是底部导航栏与标题栏不动，只有上面刷新。因此我们使用了 Fragment 嵌入在标题栏与底部导航栏之间，通过 MainActivity 控制 Fragment 的显示。
- ③Activity 返回栈问题。进行如下操作：在日常习惯界面，点击右上角“+”号添加习惯，添加完成后回到日常习惯界面，切换到今日习惯界面，然后点手机返回键。此时不是退出程序，而是返回到了日常习惯界面，并且刚刚添加的数据消失了，再切换到今日习惯界面，刚刚添加的数据同样也消失了。我猜想，添加习惯是由 MainActivity 至 EditHobbyActivity，此时 MainActivity 进

入返回栈，当 `EditHobbyActivity` 结束时，由于我想刷新整个页面，因此我使用了 `startActivity()` 重新创建了一个 `MainActivity`。当点击返回键时，刚才存入返回栈中的 `MainActivity` 重新显示，并且是未添加数据的状态。解决方案是将 `MainActivity` 设置为 `singleTask`，或及时让 `MainActivity` 执行 `finish()`。

臧博浩 161110114:

- ① 在本页面与日常习惯页面和编辑习惯页面之间进行传递的是 `hobby` 对象，由于 `hobby` 是自己自定义的实体类，所以无法利用 `intent` 进行传递。查阅资料之后决定将 `hobby` 实体类实现 `Serializable` 接口从而可以将 `hobby` 对象序列化进行传递。
- ② 本页面中要用到日历控件，`android studio` 中自带了 `CalendarView` 控件，但是想要实现对日历控件中的日期进行修饰，`CalendarView` 控件没有相应的方法。于是选择了 `GitHub` 上的一个开源日历控件 `MaterialCalendarView`，利用其中的 `addDecorators()` 方法进行特定日期的修饰。
- ③ 在日历控件选好之后就是对特定日期修饰时背景的绘制，在本次项目中是给特定的日期进行圆环的装饰，而画圆环又成了一个新的问题。查阅资料之后，写了一个函数调用 `LineBackgroundSpan` 接口，利用 `drawBackground()` 方法给日期添加背景。首先设置 `paint` 类型为圆环，消除锯齿、设置颜色。再利用 `canvas` 中的 `drawCircle` 方法绘制圆环。
- ④ 在日历上显示日期时 10 月和 10 日以下的日期可以正确显示，但是之后的就无法正确显示。在仔细查看代码之后，发现在用 `calendar.getInstance()` 获取目前的日期之后，转换成标准格式时比较的为 `calendar.month`，而这个参数的返回值永远为 2，所以就导致在每个月份前面都加了 0。解决方法时将比较条件进行更改，改成月份。

姚越 161110119:

- ① 当前日期超过预定考研日期。
在显示距离考研剩余天数时会预设目标日期为 2019 年 12 月 21 日，但是可能会遇到当前时间超过目标日期的情况。解决方案是首先会对当前年份和目标年份作对比，如果当前年份超过目标年份，则对目标年份循环增加直到与当前年份相同，再比较当前日期是否仍然超过目标日期，若超过则将目标年份加一。
- ② 同时段多个习惯的滑动展示由于不确定用户会在一个时段内添加多少习惯，所以采用滑动列表展示。在课堂上学习的 `ListView` 不能用于横向布局的列表，所以通过查找资料，我们使用了 `RecyclerView` 空间作为水平滑动列表用于读取展示数据库中已添加的习惯。
- ③ 连续打卡记录对前一天打卡情况的判断。
程序还设置了连续打卡天数记录，每次记录是否连续打卡时都要先获取昨天时间，再在 `ClickIn` 表中查找是否有标记昨天时间的打卡记录。问题就在于获取昨日时间，不能通过简单的日期减一操作，因为可能遇到每月初的情况，减一无法跨月。所以通过查找资料，可以使用安卓库函数里自带的日期减函数对当前日期进行减一操作，则可以实现任意日期的按日历加减操作。

张立才 160700108:

- ① 设置的存储。本程序中，设置都是只有 `true` 与 `false` 两种状态，存入数据库读取存储都感觉不是很方便。所以我想用到 `SharedPreferences` 来存储。`SharedPreferences` 本身就多用于存储应用程序的配置信息，而且采用键值对的形式存储，十分方便。

侯禹驰 161110118:

- ① 习惯点击事件的处理一开始把点击事件写在了 `DailyRecyclerViewAdapter` 类中，后来在进行界面间测试时发现没办法在 `DailyRecyclerViewAdapter` 类中通过点击事件跳转到其他界面，但是如果把点击事件只写在 `Fragment` 中会遇到无法判断触发点击事件的是哪个按钮。最后通过在网上查找，使用了在 `Adapter` 类中提供接口、在 `Fragment` 中实现接口并回传给 `Adapter` 类的方法，即 `Adapter`

类提供了模板，通过 `Fragment` 类注册的方法。

- ② 数据库信息传递最初是采用了直接在 `adapter` 或 `Fragment` 中进行数据库读取操作，并直接使用这些数据显示的方法，但是发现在两处直接读取信息都不方便，在 `adapter` 中读取信息要读到当前需要赋值的相应字段，非常麻烦且耗时间；在 `Fragment` 中可以进行读取但是信息的存储结构比较复杂，不如直接写一个实体类来存储信息，于是采取了在 `MyDB` 类中统一进行数据库操作，并创建 `Hobby` 实体类来存储对应的信息。

李旭 161110108:

- ① 在前期调研过程中，我们总结了好几个倒计时方案，如 `java.util.Timer` 或者 `Android` 自带的 `CountDownTimer`，但是这些组件都与我们的需求有一定出入，最后选择使用 `ScheduledExecutorService` 封装了一个线程安全的 `Timer` 组件，这个组件也可以在其他项目中使用。
- ② 最初设计中番茄钟的切换是直接通过 `Activity` 的切换来达成的，但是无论是对 `Activity` 栈的压力还是中间状态的切换都存在着一定难度。在多次尝试之后，换成了基于 `Eventloop` 的状态模式。
- ③ 在 `UI` 刷新过程中，出现了一部分操作无意中在其他线程中阻塞的情况形成了死锁，修改使用 `Activity.runOnUiThread` 来托管到主线程来处理得到解决。

五、软件的测试记录

注：进行测试的环境。

- ① 夜神模拟器（`Android 5.0`）
- ② `Huawei Nova2`（`Android 8.0`）
- ③ `OnePlus 3T`（`Android 9.0`）
- ④ `Android` 自带模拟器（`Android 9.0`）

黑盒测试：

表-17 黑盒测试用例

测试模块	测试用例	测试结果	测试人员
添加习惯	输入正确数据（背单词，1 图标，1，任意时间）	添加成功。测试通过。	李洋
添加习惯	不输入名称，输入其他正确数据	提示仍有信息未输入，无法添加。测试通过。	李洋
添加习惯	不选择图标，输入其他正确数据	添加成功，图标为默认值。测试通过。	李洋
添加习惯	不输入番茄钟周期，输入其他正确数据	提示仍有信息未输入，无法添加。测试通过。	李洋
添加习惯	不选择习惯完成时间，输入其他正确数据	添加成功，时间为任意时间（默认）。测试通过。	李洋
添加习惯	输入数据，点击重置	成功重置。测试通过。	李洋
修改习惯	点击习惯详情中修改按钮	数据成功回显。测试通过。	李洋
修改习惯	输入正确数据（背单词，1 图标，1，任意时间）	添加成功。测试通过。	李洋
修改习惯	不输入名称，输入其他正确数据	提示仍有信息未输入，无法修改。测试通过。	李洋
修改习惯	不选择图标，输入其他正确数据	修改成功，图标为默认值。测试通过。	李洋

修改习惯	不输入番茄钟周期,输入其他正确数据	提示仍有信息未输入,无法修。测试通过。改	李洋
修改习惯	不选择习惯完成时间,输入其他正确数据	修改成功,时间为任意时间。测试通过。	李洋
修改习惯	输入数据,点击重置	成功重置。测试通过。	李洋
考研倒计时	正常打开程序	显示距离2019年12月21日的正常倒计时。测试通过。	姚越
今日习惯	正常打开程序	列表正常加载显示所有习惯。测试通过。	姚越
今日习惯	添加新习惯后返回主界面	列表刷新显示添加后的所有习惯。测试通过。	姚越
今日习惯	删除习惯后返回主界面	列表刷新显示删除该习惯后的所有习惯。测试通过。	姚越
今日习惯	点击习惯图标	弹出确认是否打卡弹窗。测试通过。	姚越
今日习惯	打卡弹窗点击确认打卡	弹窗消失,后台完成打卡并弹出打卡成功提示信息,已打卡的习惯图标添加了红勾。测试通过。	姚越
今日习惯	打卡弹窗点击取消	弹窗消失。测试通过。	姚越
今日习惯	对已打卡的习惯再次点击确认打卡	弹窗消失并提示请勿重复打卡。测试通过。	姚越
日常习惯	点击导航栏进入日常习惯页面	显示日常习惯列表。测试通过。	侯禹驰
日常习惯	点击列表中的每一项	成功进入习惯详情页面。测试通过。	侯禹驰
日常习惯	点击列表中每一项的按钮	成功打开番茄钟。测试通过。	侯禹驰
日常习惯	从其他页面跳转到日常习惯	日常习惯页面刷新数据。测试通过。	侯禹驰
日常习惯	点击日常习惯右上角加号	成功进入添加习惯界面。测试通过。	侯禹驰
今日习惯 日常习惯 查看习惯	点击日常习惯添加按钮,添加习惯,返回日常习惯页面,点击今日习惯页面,按手机实体的返回键。(修改前)	添加习惯后日常习惯、今日习惯成功显示。但是按手机实体返回键后,日常习惯、今日习惯页面中刚添加的条目消失。测试未通过。	李洋
今日习惯 日常习惯 查看习惯	点击日常习惯添加按钮,添加习惯,返回日常习惯页面,点击今日习惯页面,按手机实体的返回键。(修改后)	添加习惯后日常习惯、今日习惯成功显示。按手机实体返回键后,日常习惯、今日习惯页面中刚添加的条目也正常显示,测试通过。	李洋
日常习惯 查看习惯	点击日常习惯列表的某一项,进入习惯详情界面,再点击返回,再进入这一项,再点击返回。进行数次。(修改前)	列表的间隙越来越大。测试未通过。	李洋
日常习惯 查看习惯	点击日常习惯列表的某一项,进入习惯详情界面,再点击返回,再进入这一项,再点击返回。进行数次。(修改后,及时 finish() 或 singleTask)	列表的间隙不变。测试通过	李洋
查看习惯	在日常习惯页面点击习惯	标题正确显示习惯名称。测试通过。	臧博浩
查看习惯	存入一组打卡日期	日历上对应打卡日期标上蓝色圆环背景。测试通过。	臧博浩
查看习惯	显示日历控件	日历控件当天标上红色圆环背景。测试	臧博浩

		通过。	
查看习惯	存入一组打卡日期	显示正确连续打卡天数。测试通过。	臧博浩
查看习惯	存入一组打卡日期	显示正确总共打卡天数。测试通过。	臧博浩
查看习惯	点击左上角返回按钮	成功进入日常习惯页面。测试通过。	臧博浩
查看习惯	点击右上角删除按钮	弹出是否确认删除对话框。测试通过。	臧博浩
查看习惯	是否删除对话框选是	删除该习惯，返回日常习惯页面。测试通过。	臧博浩
查看习惯	是否删除对话框选否	对话框消失。测试通过。	臧博浩
查看习惯	点击右上角编辑按钮	成功进入习惯编辑页面。测试通过。	臧博浩
设置	点击是否开启通知选项开启通知	在 7: 50, 11: 50, 19: 50 成功通知。测试通过。	张立才
设置	点击是否开启通知选项关闭通知	在 7: 50, 11: 50, 19: 50 均为收到通知。测试通过。	张立才
设置	点击是否开启振动选项开启振动	收到通知时振动，测试通过。	张立才
设置	点击是否开启振动选项关闭振动	收到通知时未振动，测试通过。	张立才
启动页	打开程序	进入启动页，2 秒后进入今日习惯页面。测试通过。	张立才
番茄钟	启动番茄钟	番茄钟正常启动，初始数据显示正常。测试通过。	李旭
番茄钟	运行番茄钟	番茄钟倒计时功能正常，结束后无法正常进入下一个阶段。测试未通过。	李旭
番茄钟	运行番茄钟	番茄钟倒计时功能正常，结束后正常进入下一个阶段。测试通过。	李旭
番茄钟	运行休息阶段	休息阶段倒计时功能正常，结束后正常进入下一个番茄周期，且周期数正常递增。测试通过。	李旭
番茄钟	打断	打断功能正常，正常显示提示信息，被打断后信息回传正常。测试通过。	李旭
番茄钟	运行至结束	正常运行至结束，进度条正常显示为满，震动功能正常。测试通过。	李旭
番茄钟	退出	多个阶段中均能正常退出。测试通过	李旭

白盒测试：

表-18 白盒测试用例

测试模块	测试用例	测试结果	测试人员
考研倒计时	系统时间设为 2019 年 12 月 30 日	显示距离 2020 年 12 月 21 日的正常倒计时。测试通过。	姚越
考研倒计时	系统时间设为 2020 年 1 月 30 日	显示距离 2020 年 12 月 21 日的正常倒计时。测试通过。	姚越
设置	系统时间改为 2019 年 5 月 13 日 7: 49	等待一分钟后，通知显示“研途—早上好!”。测试通过。	李洋
设置	系统时间改为 2019 年 5 月 13 日 14: 49	等待一分钟后，通知显示“研途—下午好!”。测试通过。	李洋
设置	系统时间改为 2019 年 5 月 13 日 19: 49	等待一分钟后，通知显示“研途—晚上好!”。测试通过。	李洋

设置	系统时间改为2019年5月14日7:49。（检测 5.13 日执行完后是否重置定时器时间为 5.14 7: 49）	等待一分钟后，通知显示“研途—早上好!”。测试通过。	李洋
设置	系统时间改为2019年5月14日7:49。（检测 5.13 日执行完后是否重置定时器时间为 5.14 14: 49）	等待一分钟后，通知显示“研途—下午好!”。测试通过。	李洋
设置	系统时间改为2019年5月14日7:49。（检测 5.13 日执行完后是否重置定时器时间为 5.14 19: 49）	等待一分钟后，通知显示“研途—晚上好!”。测试通过。	李洋
番茄钟	完整运行进行状态切换检查。	状态切换符合预期，未出现进入其他状态的情况。测试通过。	李旭
番茄钟	与主 Activity 数据交互检查	多种状态下数据交互均正常，数据均能成功到达番茄钟 Activity，主 Activity 能正常的接收到合理的数据。测试通过。	李旭

六、体会、感想与建议

李洋 161110117:

本次大作业，用到了课上讲过的组件、技术，比如 Broadcast Receiver、Fragment、Notification，还有一些新的控件比如 RecyclerView、AlarmManager 等。通过解决一系列问题，对安卓的生命周期理解的更加深入，已学过的知识更加熟悉。使用 Github 进行代码管理，这是一个新的尝试，避免了反复传递代码导致混乱的问题。有问题互相帮助，有经验共同分享，增进了团队的情感。

臧博浩 161110114:

通过本次的大作业，我获得了许多提升，首先知道了如何引用一些开源的控件来使自己的项目实现更加的简单。也对安卓中日期的获取和表示方法有了一定的了解。同时初步了解了利用 span、canvas、paint 画图的方法。在对数据库的调用中也对根据某一条件进行查询或删除的操作更加的熟悉。也实现了点击弹出对话框的操作，对 AlertDialog 的有了初步的了解。

姚越 161110119:

通过设计主界面及部分功能的完善，使我对课堂上学习到的安卓开发技术有了更好的熟悉，并且在解决一些之前没遇到过的问题时，在与小组成员讨论和上网查询资料的过程中也锻炼了我解决问题的能力。最终在我们共同的合作下完成了令我们满意的安卓小程序，也是一件很有成就感的事。

张立才 160700108:

本次安卓大作业作业，通过学习 Android 的控件、布局、Activity、Fragment 等一系列基础知识，对整个 Android 的开发有了大致的了解。例如：要的布局或者控件在学习界面中，我发现 Android 为我们提供了很好的类似反射机制，通过 Layout 文件下的配置文件，可以快速形成界面，在配置文件可以设置属性或者样式都很快捷方便。对比较特殊的界面也可以通过处理嵌入到指定的界面，同样你可以通过 JAVA 代码直接创建 View 进行添加。温习了 Fragment，学习了新的控件 Switch 等，当遇到新的控件可以举一反三。其次，本次大作业的小组模式，也让我深刻的意识到团队协作的重要性。

侯禹驰 161110118:

这次项目最大的收获是接触了自己 Android 方面了解最少的 ListView 类型控件—recycleView，之

前的实验中对 adapter 及 ListView 的概念很模糊，这次通过使用 recycleView 并手写其 Adapter 充分了解了两者的关系，Adapter 是将传入的数据对应 ListView 中的每一项进行适配，并返回适配后的结果以供 ListView 使用。

同时这也是我第一次接触 recycleView，相对于其他类型的 ListView，recycleView 的大部分功能需要手动实现，如刷新、点击事件的注册等，这也有助于我更一步了解 ListView 类型控件。

其他方面的收获还有比如项目的规范化，这次我们的项目全程是用 Git 进行管理，在项目实施中我还体会到职责分离的重要性，当职责分离之后，代码的逻辑会很清晰、易于编写。

对于我自身而言，我感觉已经在这次项目中完成了自己任务，唯一的不足的界面不怎么好看，代码方面由于负责的部分相对简单，实际上真正遇到的难点不多。

李旭 161110119:

这个项目的体量并不大，但是麻雀虽小五脏俱全，这个项目的整个架构和流程显得比较清晰，整个编写过程就很容易去做到比较好的追踪和管控。

在这次的小组作业中，我主要负责了番茄钟的设计。番茄钟说小不小，说大不大。由于 Android 平台的固有限制，UI 仅能由主线程来操作，而其余线程只能处理事务性的任务，这就对番茄钟的设计提出了挑战：如何正确处理好倒计时和进度条的 UI 刷新。为此，我特地看了 Java 多线程的部分内容，然后参考了一些其他的实现，最终封装了一个 Timer，不仅能作为番茄钟的计时后端，更能当做一个独立的组件进行应用。

UI 设计的方面我并不是很擅长，这一部分参考了部分同组人员的设计，他们也在我的设计过程中给我提供了很大的帮助。

做一个课设就应该学到一点东西，这次课设让我对 Java 多线程的东西有了一定了解，就算是达成了最初的目的。

七、参考文献

图书文献：

[1] 孙玉山，徐汉川。面向对象技术与建模[M]。北京：电子工业出版社，2015：159-162。

结课大作业修改说明

序号	问题描述（照片）	教师修改意见	学生修改说明
1			
2			
3			
4			
5			