

# MindMeet分析模型

## MindMeet分析模型

### 1. 介绍

#### 1.1 主要功能

#### 1.2 目前进展

### 2. 架构分析

#### 2.1 总体介绍

#### 2.2 界面层

#### 2.3 应用逻辑层

#### 2.4 数据持久层

#### 2.6 通用服务层

### 3. 分析模型

#### 3.1 用户认证

#### 3.2 专注模式

#### 3.4 悄悄话

#### 3.5 日程规划

### 4. 更新UI

#### 4.1 专注视图

#### 4.2 设置专注时间

#### 4.3 专注进行

#### 4.4 日程界面

#### 4.5 设置日程

### 5. 参考引用

### 6. 分工

## 1. 介绍

**MindMeet** 是一款时间管理和社交平台，旨在帮助用户更有效地管理时间和实现个人目标。用户可以创建自己的日程表，使用专注功能进行自律专注，打卡功能能够激励用户自律和持续努力，进度跟踪功能帮助用户提高效率 and 保持动力。用户还可以在社交网络中找到志同道合的朋友，分享自己的日程表，相互交流和监督，设定目标和挑战。

## 1.1 主要功能

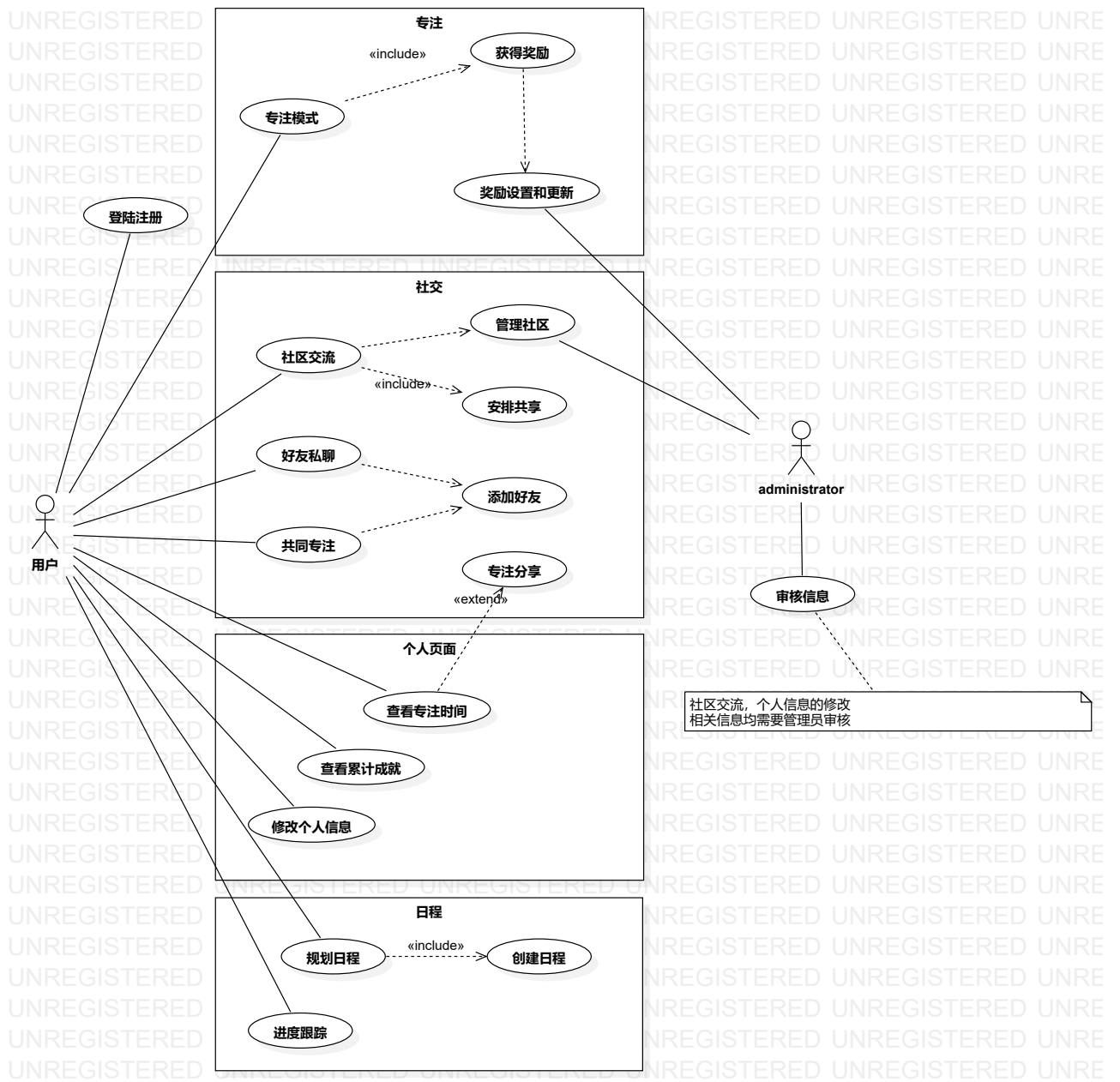
MindMeet 实现的主要功能有以下几点：

1. 专注模式：可以进行正计时专注与倒计时专注
2. 日程管理：可以添加日程，规划日程，日程提醒
3. 专注记录分享：可以分享自己一段时间内的专注记录
4. 日程分享：可以分享自己一段时间的日程
5. 计划借鉴：可以导入他人的日程计划，借鉴学习
6. 好友私聊：与好友相互沟通交流
7. 线上自习室：用户可以邀请一定数量好友定时专注，相互监督
8. 记录跟踪：软件可以跟踪用户的专注记录，并进行数据可视化展示

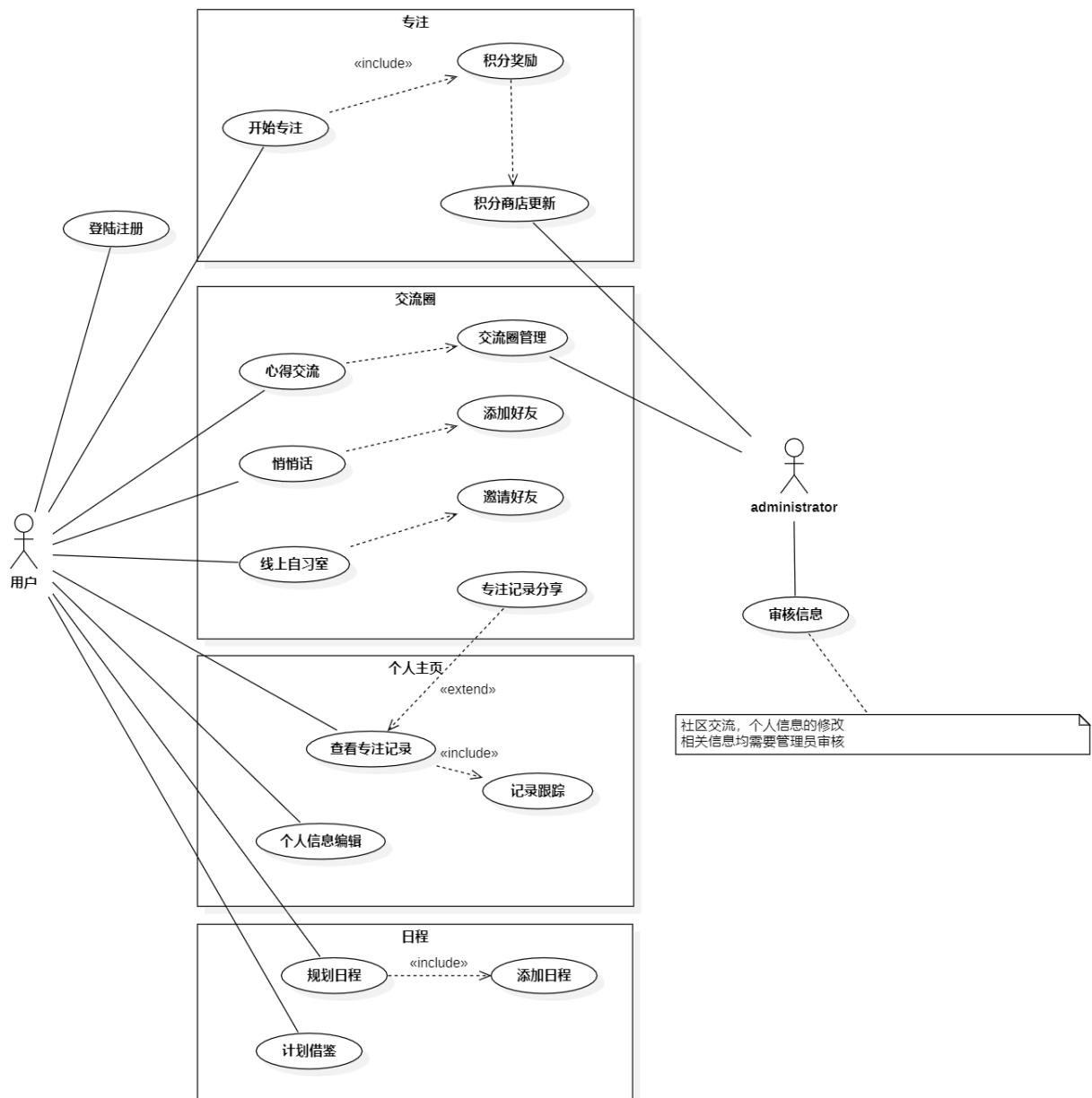
## 1.2 目前进展

从上次需求文档以来，我们取得了以下进展：

1. 对系统架构的分析与规划
2. 对总体用例图的细节更新
  - 原用例图



- 现在用例图



## ● 更改

### 1. 用例改名:

- 专注模式->开始专注
- 获得奖励->积分奖励
- 奖励设置和更新->积分商店更新
- 社交->交流圈
- 好友私聊->悄悄话
- 共同专注->线上自习室
- 查看专注时间->查看专注记录
- 进度跟踪->记录跟踪

更改后的用例名更加符合事实情况，且更加生动

2. 将进度跟踪移到个人主页板块

进度跟踪在个人主页板块展示，故将其移动到个人主页板块

3. 对专业术语的相关更新，以下是更新后的专业术语表：

| 术语       | 解释   |
|----------|--|
| MindMeet | 本软件的名称，意为促进思想交流和分享的时间管理平台                      |
| SRS      | 软件需求规格说明书（Software Requirements Specification） |
| UI       | 用户界面（User Interface）                           |
| API      | 应用程序接口（Application Programming Interface）      |
| 日程表      | 用来记录用户日程的对象                                    |
| 交流圈      | 用户可以在其中分享专注记录与日程表，同时对他人的分享进行评价                 |
| 自习室      | 用户可以创建、加入的虚拟自习空间，用于自习和共同学习                     |
| 专注力      | 集中注意力于特定任务或活动，而不受干扰或分心                         |
| 专注模式     | 用户使用软件进行专注的软件状态                                |
| 任务管理     | 通过记录、安排和跟踪任务来有效管理时间和工作                         |
| 日程安排     | 通过创建和管理日程表来安排和规划时间                             |
| 记录跟踪     | 记录专注记录用于评估时间使用情况和提高生产力                         |
| 悄悄话      | 好友间私聊，交流学习                                     |
| 计划借鉴     | 指用户将导入他人分享的日程表导入自己的日程，一般用于用户像有相关经验用户           |

4. 细化了部分用例的流程，如线上自习室的用例

5. 主要功能的类设计，与交流图设计

6. 更新了部分ui，将在最后一部分展示

2. 架构分析

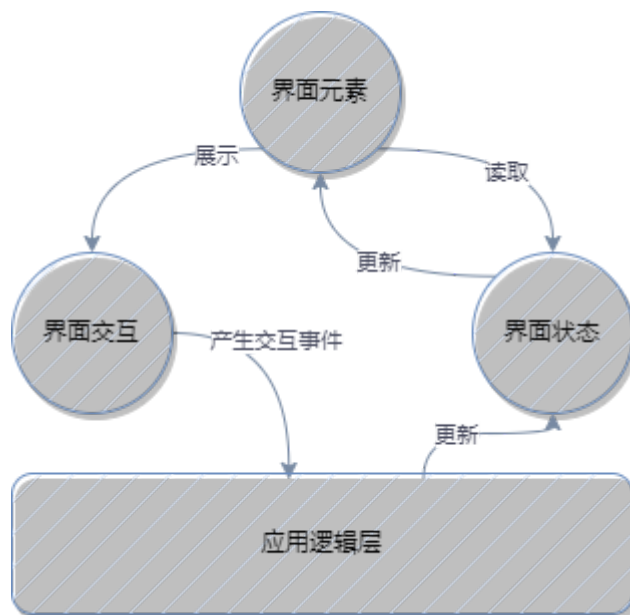
2.1 总体介绍

MindMeet 采用分层架构，如下图所示：



其中，界面层展示页面与进行交互，产生交互事件，并传递数据到应用逻辑层，应用逻辑层对业务进行处理，通过数据层对数据库进行读写查询，通用服务层提供一些通用的服务，如日志服务，网络服务等等。下面是更加细致的层次架构：



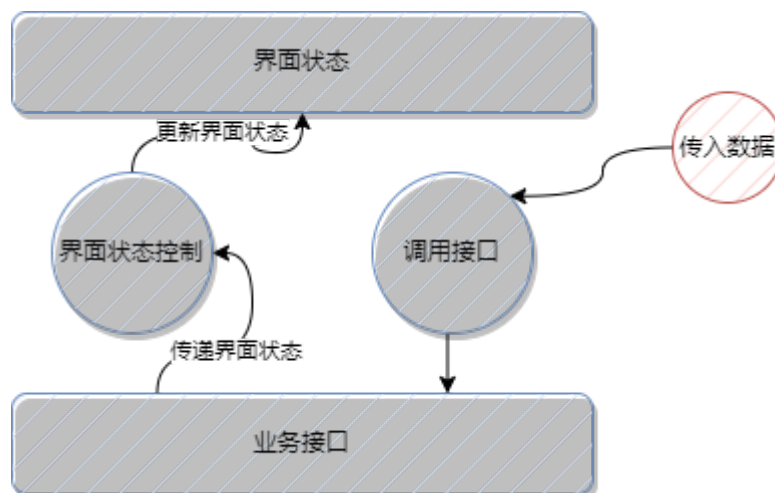


界面元素展示ui与界面交互，界面交互产生用户交互事件到应用逻辑层，经过应用逻辑层处理后，应用逻辑层更新界面状态（如界面元素的相关参数），界面元素不断读取界面状态后更新ui与界面交互。

## 2.3 应用逻辑层

应用逻辑层分为**控制层**与**业务层**。

- 控制层负责在联系界面层与业务层，控制层调用业务层实现的业务接口，将界面传入数据传入，经由业务层处理后获得更新的界面状态，通过界面状态控制组件更新界面状态。具体工作流程如下：



- 业务层实现具体业务逻辑，分为三个子系统，分别是：

- 用户管理子系统

用户管理子系统实现用户认证与好友管理功能点。负责用户的注册登录与好友的添加与删除。

- 时间管理子系统



时间管理子系统负责应用核心功能，实现日程管理，专注功能与记录跟踪功能。

#### ■ 交流圈子系统

交流圈子系统实现专注记录分享，日程表分享功能，负责应用社交主要功能实现。

其中用户管理子系统与时间管理子系统共同实现**线上自习室**功能，用户可以发起固定时间线上自习室，邀请好友共同专注，相互监督；时间管理子系统与交流圈子系统共同使用**数据可视化**功能，时间管理子系统通过数据可视化展示专注记录跟踪，交流圈子系统可以将经过数据可视化后的专注记录数据与日程表分享出去。

## 2.4 数据持久层

数据持久层完成一些与数据库交互等相关操作，如数据库连接，数据库查询，增删改查等操作。

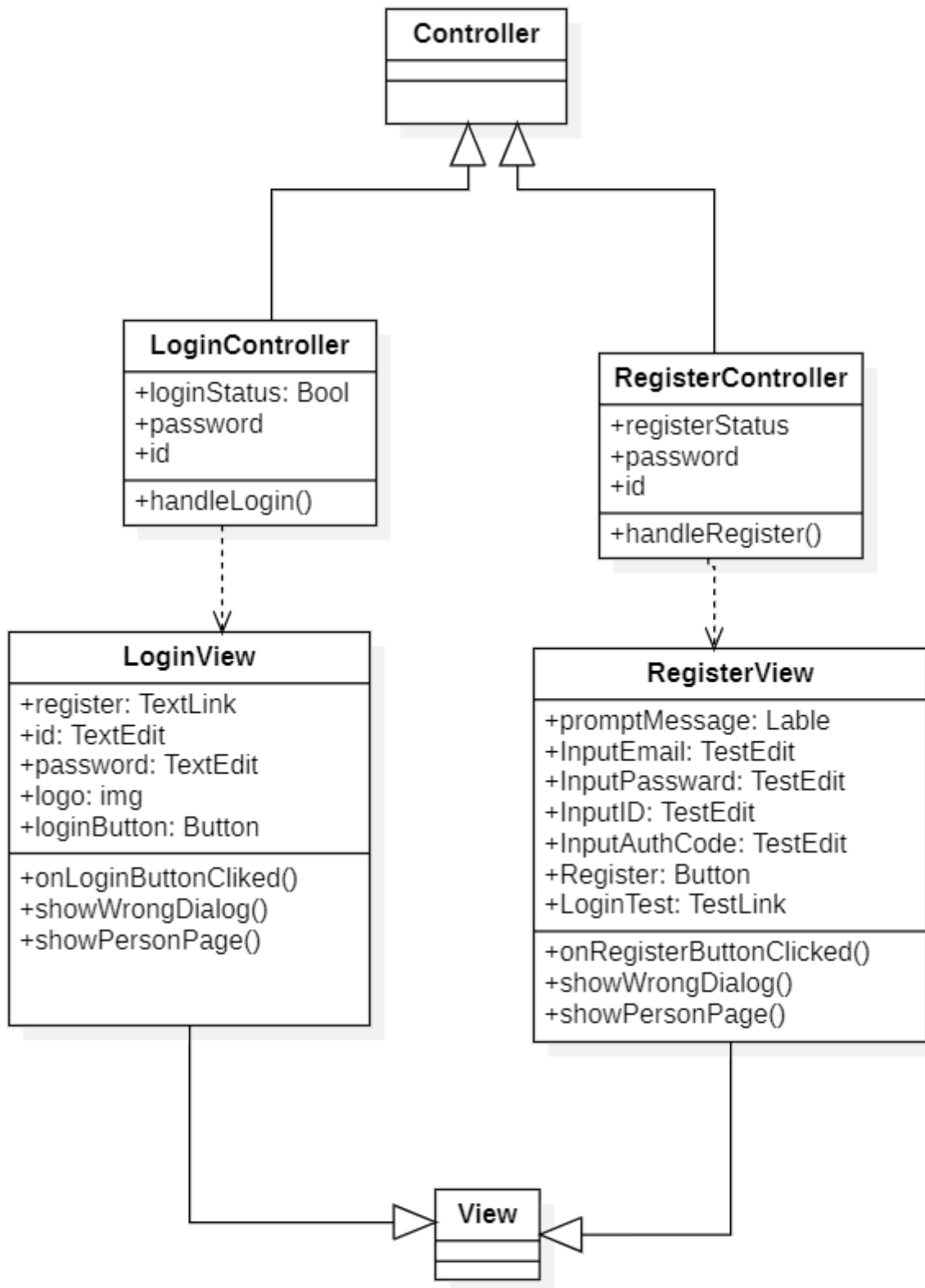
## 2.6 通用服务层

通用服务层提供一些多层次所共需的相关服务，如日志服务，缓存服务，网络服务与加密服务等等。

# 3. 分析模型

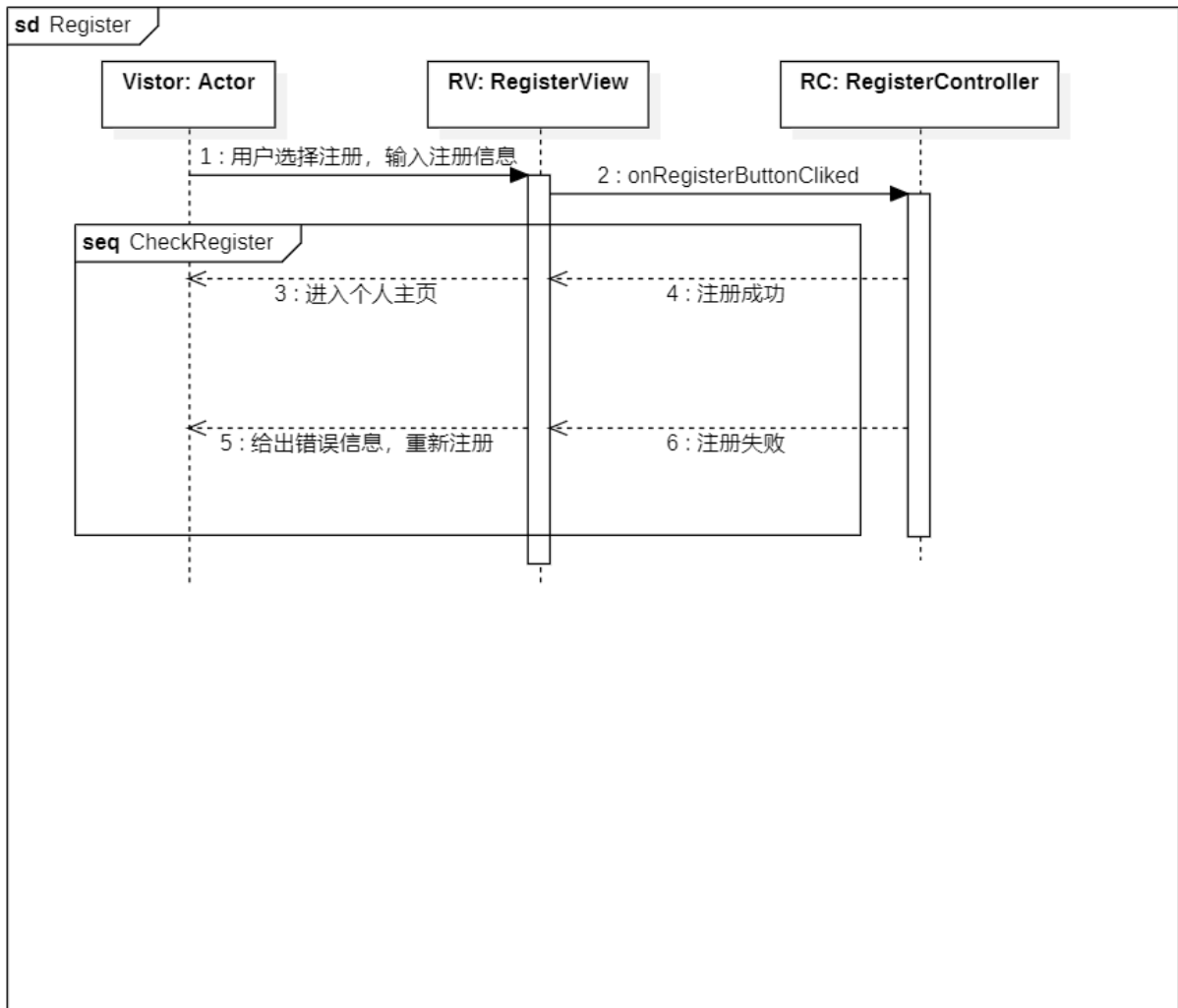
## 3.1 用户认证

- 类图

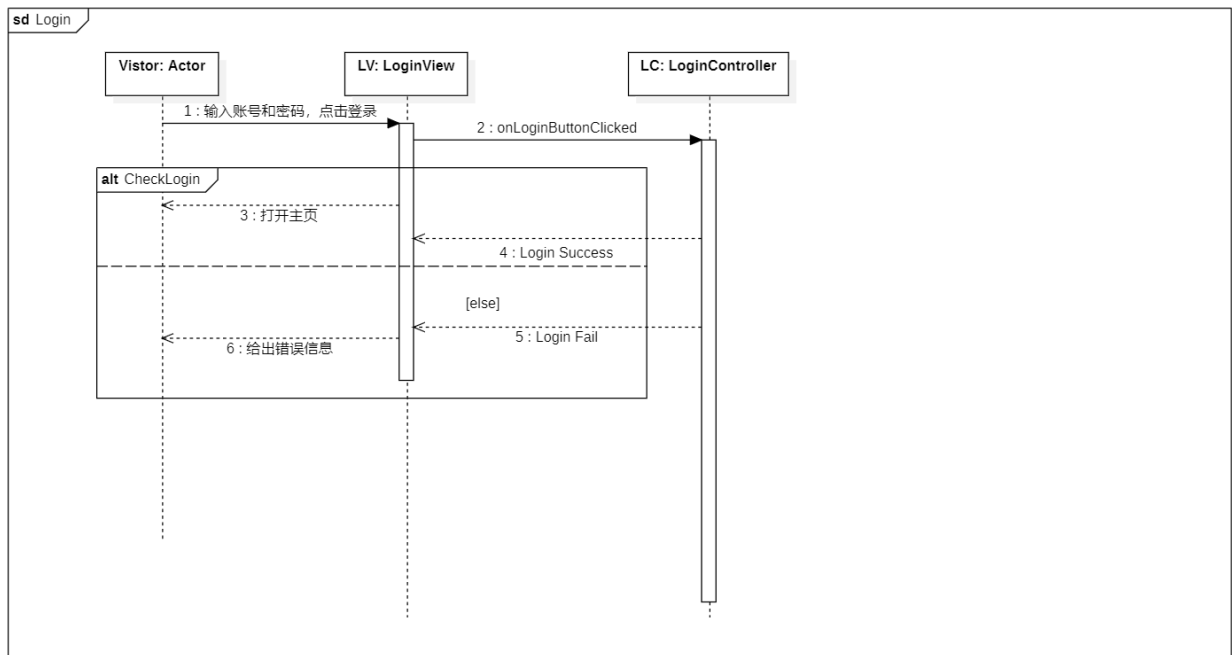


- 序列图

- 1. 注册



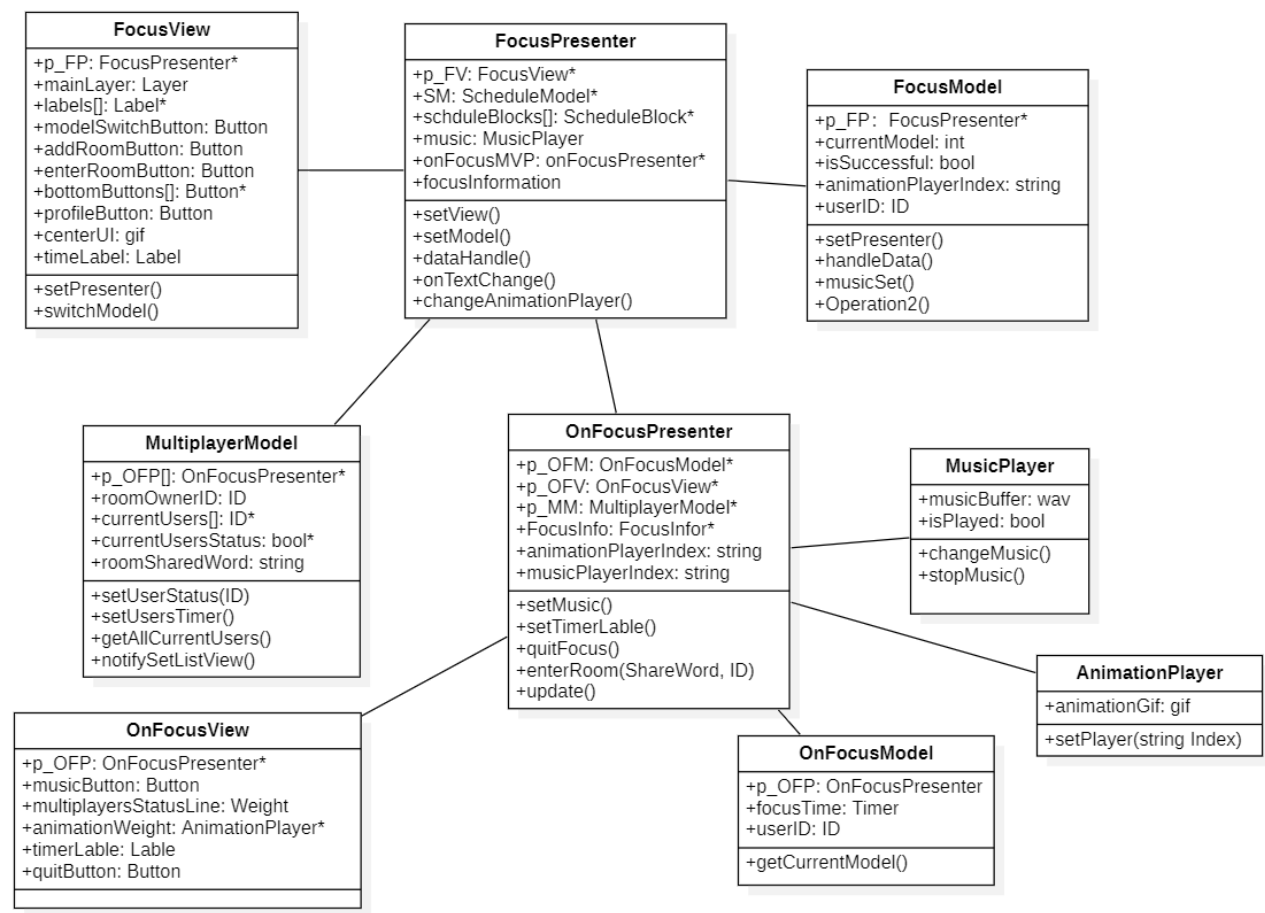
## 2. 登录



## 3.2 专注模式

- 类图

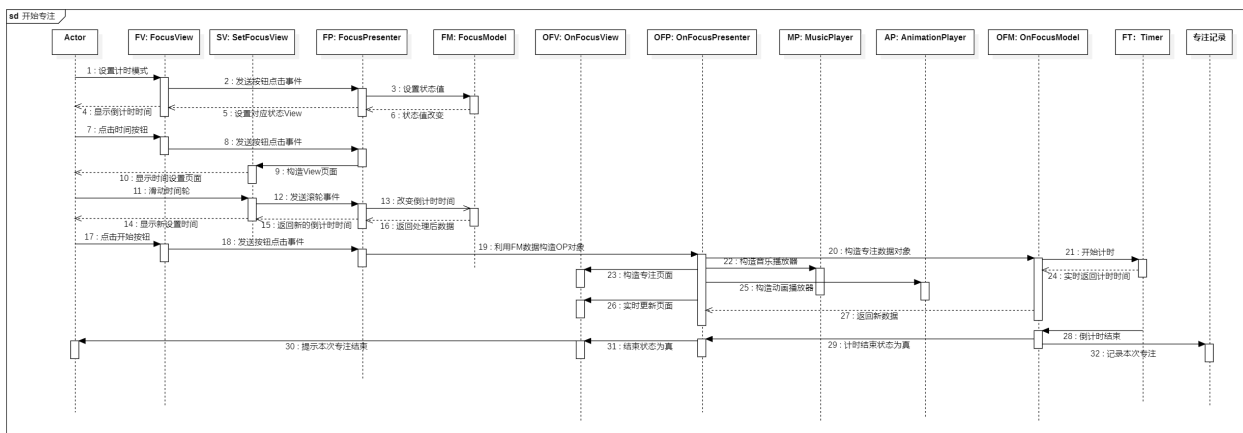
专注模式的整个过程包括两个方面，一个是专注设置，用户交互包括时间设置、动画设置、计时设置，另一个是专注计时，主要包括音乐播放的设置，退出按钮和计时时间的显示。因此在MVP架构下我们选择了两层Presenter的类设计，一层负责专注设置交互展示逻辑，另一层负责个人或自习室的专注计时的交互展示逻辑。而上层类负责构造下层类并传入用户相应设置好的数据。



- 序列图

### 1. 开始专注

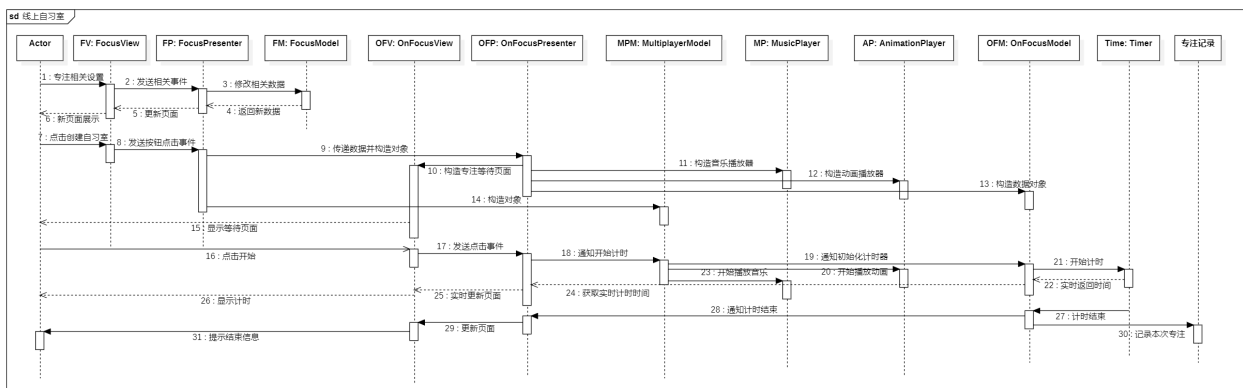
个人专注模式中，首先在第一层MVP架构上进行用户的专注设置相关交互，并将设置的数据存入该层的Model对象。当用户点击开始按钮时，该层Presenter则构造并初始化下一层的Presenter，而下层的Presenter进一步构造初始化该层所需的相关对象，例如音乐播放器对象，该层Model对象等。当用户点击退出，或者倒计时专注结束，负责计时的Timer对象会发出通知，Presenter收到通知后首先将本次专注记录，通过调用接口层函数传入给业务逻辑层，进而进行进一步的包装处理并记录本次专注。然后Presenter会更新页面信息，提醒用户本次专注结束。



## 2. 线上自习室

线上自习室在页面上表现为当用户选择倒计时模式时，系统会渲染出“创建自习室”和“加入自习室”两个按钮供用户点击。而当用户选择“创建自习室”时，用户会进入原本的“专注计时”页面，然而此时计时器并不会计时，且页面上方会多出一个用户头像显示条，下方会多出一个“开始”按钮，表示用户已经在线上自习室中，用户可以点击头像显示条的加号来选择邀请好友，邀请后会通过“悄悄话”的私聊界面发送包含房间密匙的邀请信息。当用户点击开始按钮时，则自习室中每个用户的计时器开始计时，同时退出按钮也向每个用户显示，此后的展示互动逻辑同个人的专注模式。

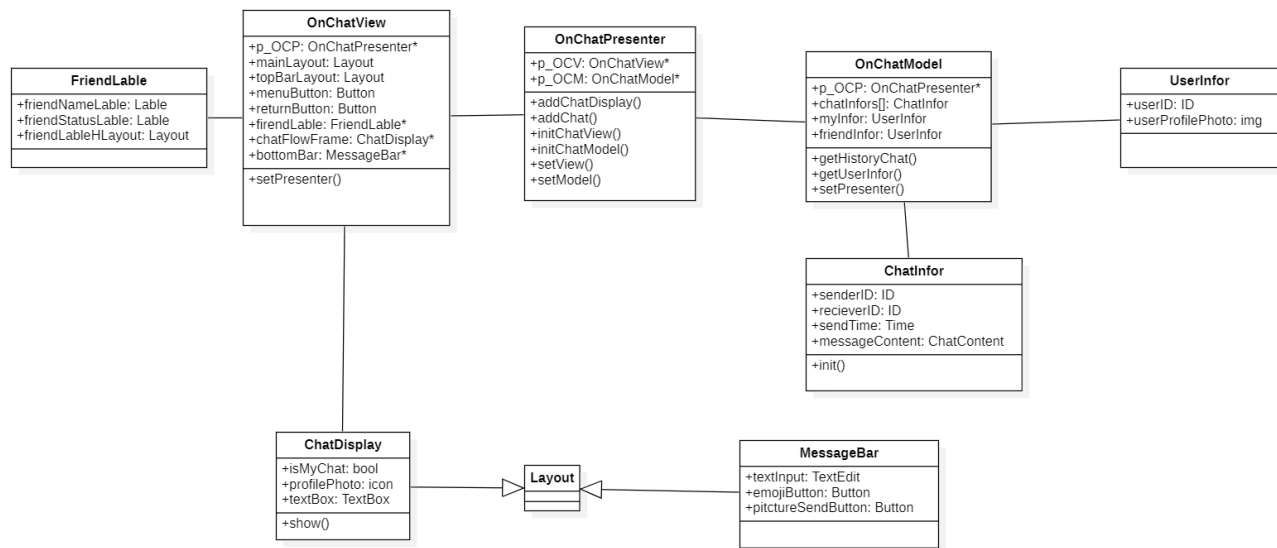
因此，为了满足在“专注计时”页面的“等待好友加入”，以及的显示交互逻辑，我们加入了MultiplayModel这一额外的类来负责统一通知自习室各用户的计时器开始计时，并且实时更新当前自习室各用户的状态（已退出/专注中）。当某用户创建线上自习室时，该用户的Presenter会构造初始化一个MultiplayModel对象并上传服务器，当某一好友通过房间密匙想要进入该自习室时，其Presenter会搜索对应MultiplayModel对象，若找到则构造初始化其他相关对象并在该对象中进行“注册”。当房主点击开始时，该MultiplayModel对象会通知各用户的下层各对象进入“计时”的状态。在专注过程中，当存在用户退出专注，该对象会进行相应“注销”，并通知剩下各用户的Presenter，进而更新用户头像条，提示用户有人退出。



### 3.4 悄悄话

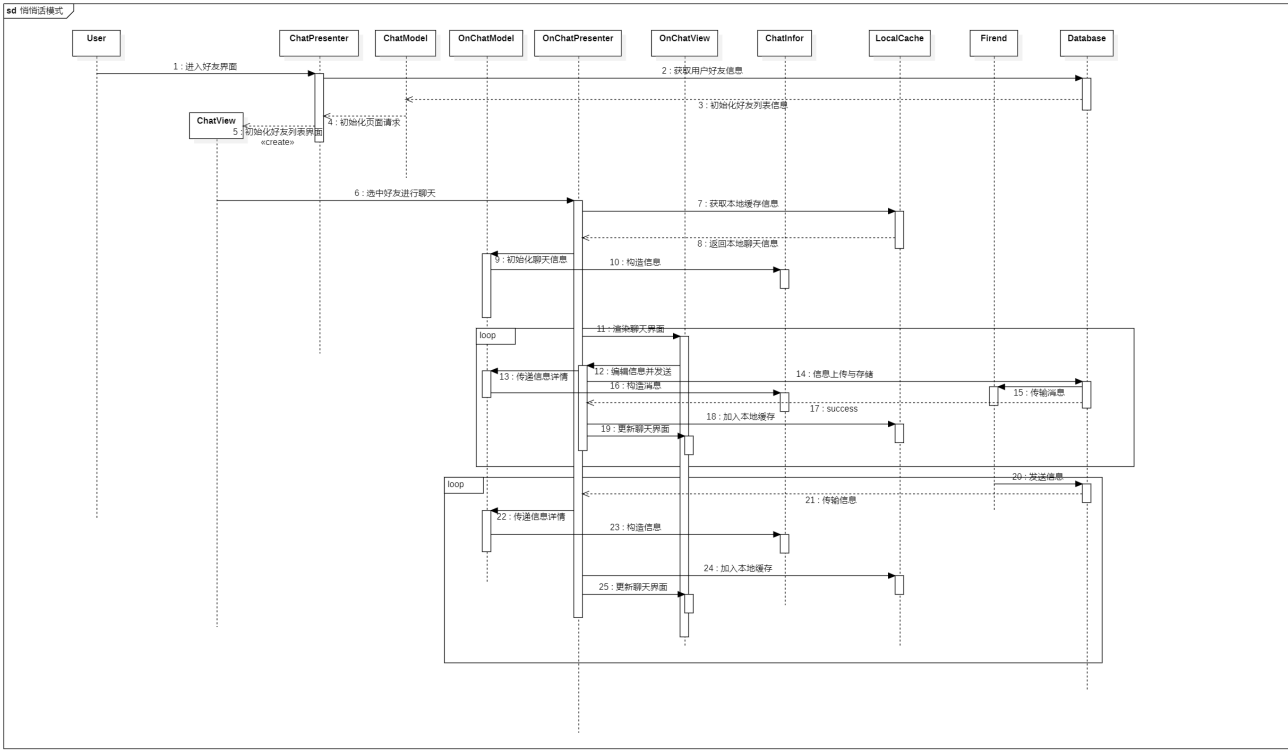
- 类图

好友模块一共包含两个界面，一个是好友列表界面，另一个则是好友私聊(OnChat)界面。在进入私聊界面后，可以进行简单的信息交互，发送图片、表情、文本消息。一条消息除了本身信息外，还需要包含双方信息、发送时间等故在此封装为一个单独的类。悄悄话界面也是使用的MVP架构，其中由OnChatModel来存储交流信息。界面元素由三层构成，上层显示好友基本状态，中层为交流信息的chatFlowFrame，底部为输入框。



- 序列图

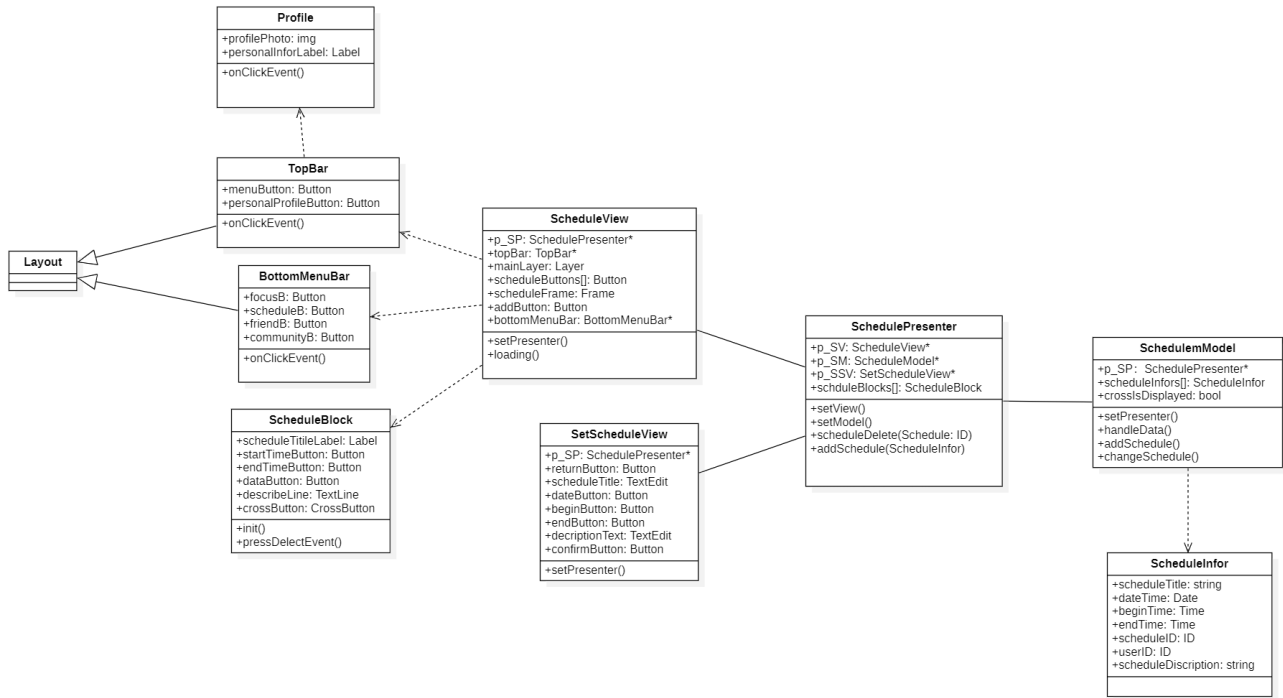
当进入好友界面首先会获取数据库中自己的好友信息，以渲染ChatView(好友列表视图)，如若选中好友进行聊天，此时会获取本地缓存的交流信息以初始化OnChatView(聊天界面)。在此模式下，共有两个主要事件，发送消息和接受消息。在此采用多线程模式，使得发送和接受消息能同时运行。当编辑完消息并发送后，数据会存储在数据库中，本地会构造一个ChatInfor类存储在OnChatModel中，并告知OnChatPrensenter去更新OnChatView。



### 3.5 日程规划

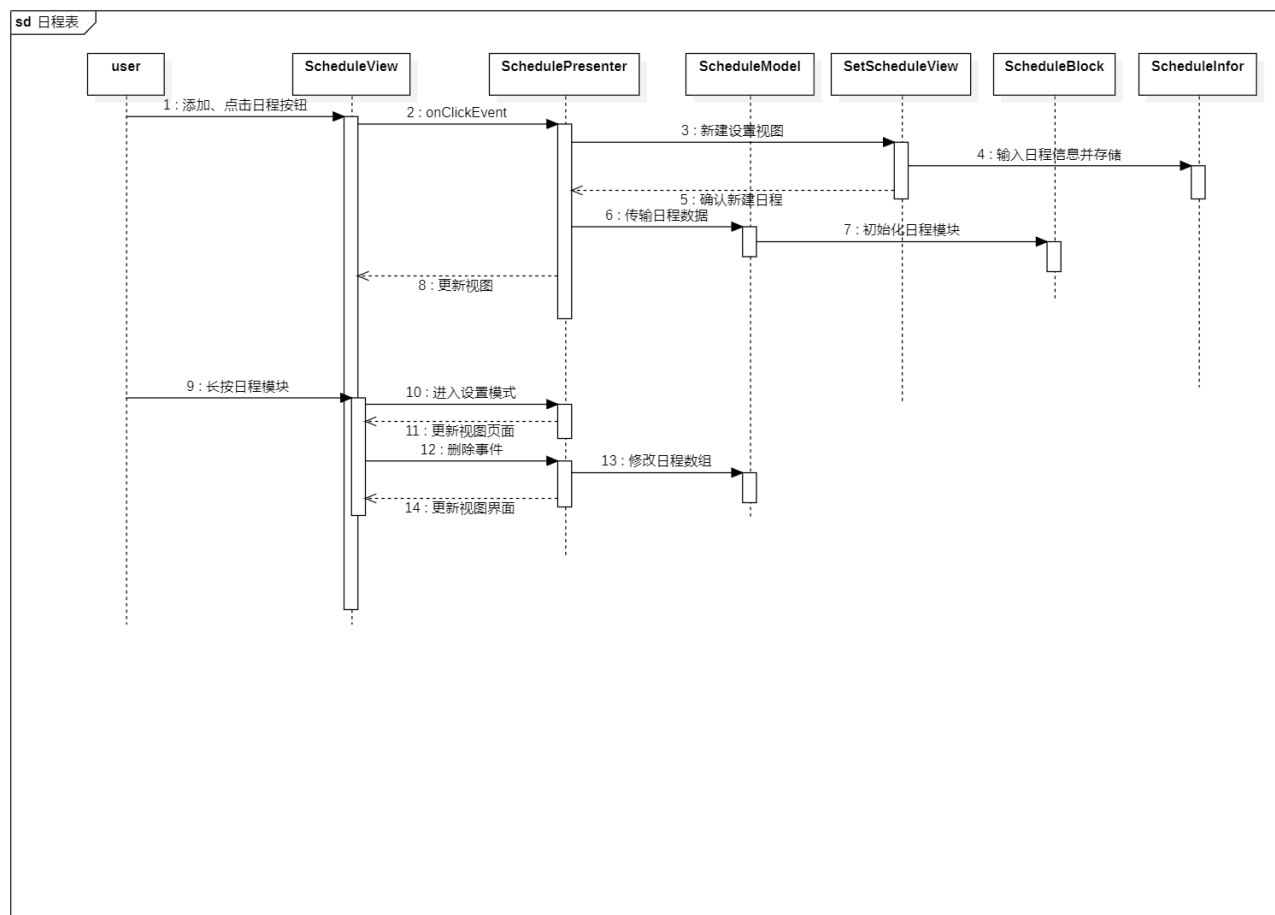
- 类图

MindMeet提供简单的日程规划模式，在日程规划模式下用户能够预览、新建、修改、删除日程。该模式也采用MVP架构，同时采用双层View(ScheduleView、SetScheduleView)以满足新建日程和预览所有日程的需求。而ScheduleInfor包含日程信息该类存储于ScheduleModel中。



- 序列图

日程表模式下的事件交互较为简单。当用户选中添加日程按钮或者某日程后，会新建一个SetScheduleView，在该界面下用户进行日程的信息的相关设置，设置的信息存储在ScheduleInfor这个类中，当确认创建后，SchedulePresenter则会控制ScheduleModel进行信息存储并更新SheduleView。当用户长按日程模块时会进入到设置模式，设置模式下用户可以删除已完成的任務，同时SchedulePresenter会控制ScheduleModel和SheduleView的修改。



## 4. 更新UI

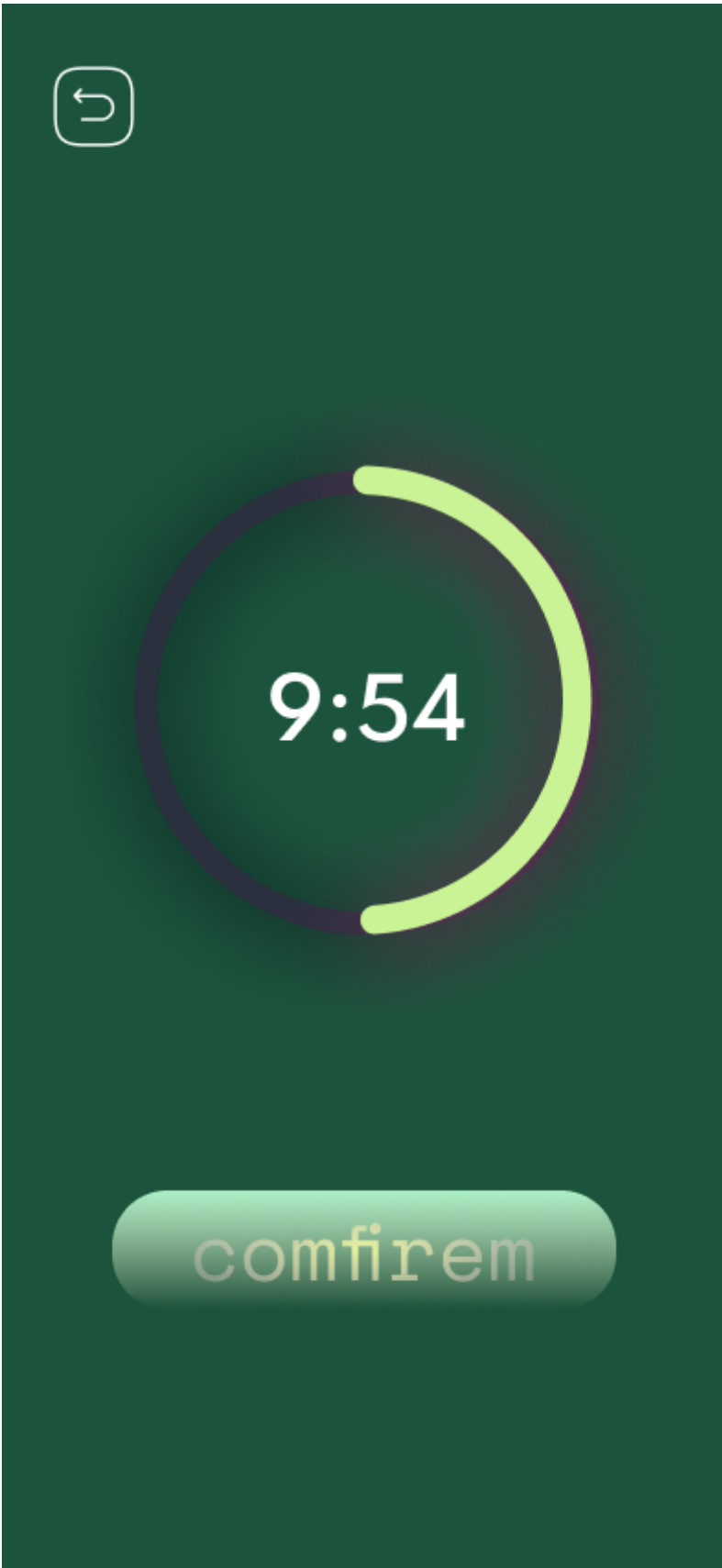
### 4.1 专注视图





与上次作业相比，该板块UI添加了正计时和倒计时模式的选择。用户在此界面选择正计时或倒计时专注。

## 4.2 设置专注时间



新增设置专注时间界面，用户可在此界面设置专注时间。

4.3 专注进行



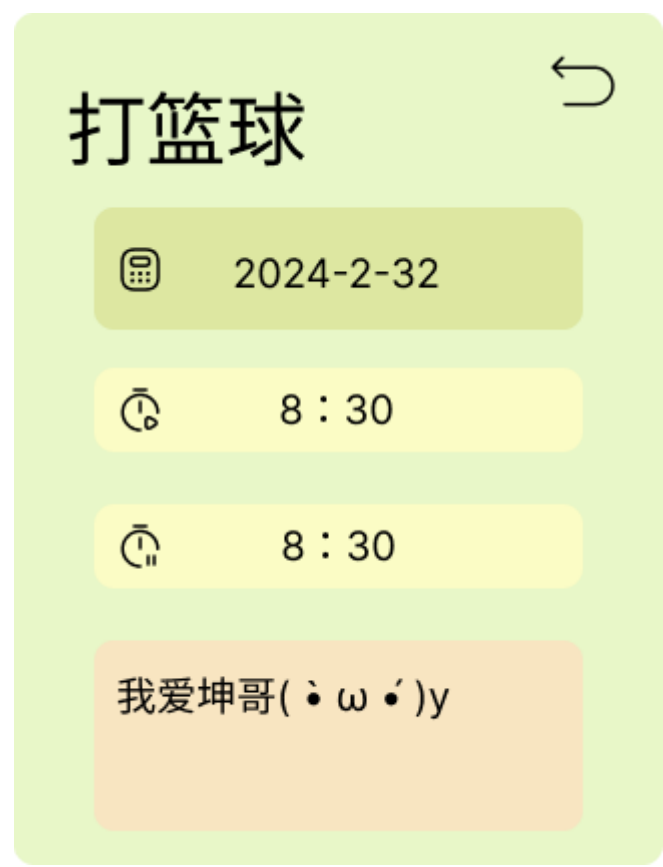
新增专注进行的UI界面，图示正计时模式，可以按下停止按钮停止专注。

4.4 日程界面



添加了新建日程的按钮区域，用户可点击“+”号添加日程。

## 4.5 设置日程



新增新建日程界面，用户在此界面新建日程。

## 5. 参考引用

### 1. [应用架构指南 | Android 开发者 | Android Developers \(google.cn\)](#)

此网页介绍了Android应用开发常见的应用架构，向开发者介绍一系列常见的架构原则、推荐的应用架构、常见的最佳做法以及一些真实的应用实例。本项目所采用分层架构以及界面展示部分均参考此设计准则设计。

### 2. [大象：ThinkinginUML\\_百度百科 \(baidu.com\)](#)

《大象：ThinkinginUML》是2009年中国水利水电出版社出版的图书，以UML为载体，将面向对象的分析设计思想巧妙地融入建模过程中，通过贯穿全书的实例将软件系统开发过程中方方面面的知识有机地结合在一起，用生动的语言和精彩的事例将复杂枯燥的软件过程讲解得津津有味。本项目参考该书设计了此次的类图以及序列图

## 6. 分工

- 2152057 杨瑞华
  - 文档编写
  - 系统架构设计及绘图
- 2152831 陈峥海
  - 文档编写
  - 类图，序列图设计
- 2153691 邓岳衡
  - 文档编写
  - 类图，序列图设计
  - UI更新