



**签到SoEasy**

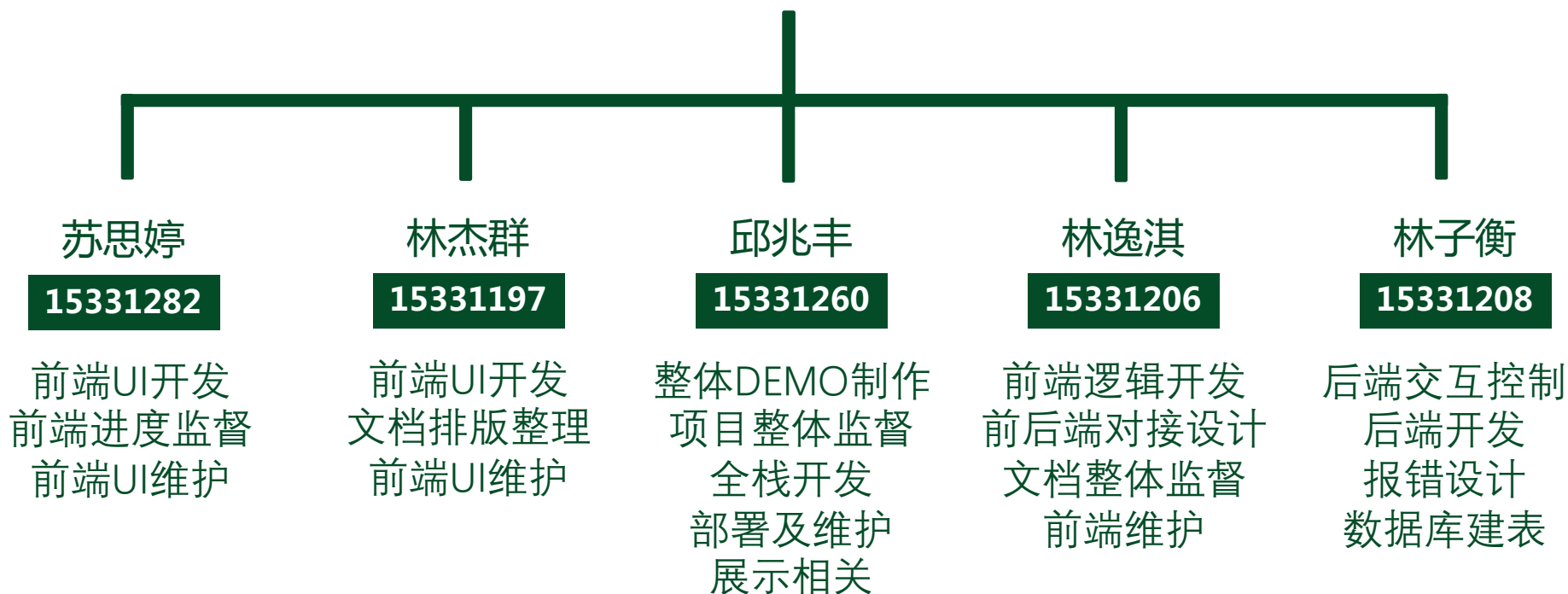
# 系统分析与设计例案分享

基于微信小程序的课程签到系统

# 成员分工及管理

指导老师：衣杨

## 成员及分工股管理



小组所有成员均参与到项目文档及制图的各个阶段的编写！



Search or jump to...

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



[mgsweet](#) / [Lesson-Attendance-WechatApp](#) Private

[Unwatch](#) 3

[Star](#) 0

[Fork](#) 0

[Code](#)

[Issues](#) 0

[Pull requests](#) 0

[Projects](#) 0

[Wiki](#)

[Insights](#)

[Settings](#)

A lesson sign in system base on Wechat.

[Edit](#)

[Add topics](#)

266 commits

7 branches

0 releases

5 contributors

MIT

Branch: dev

[New pull request](#)

[Create new file](#)

[Upload files](#)

[Find file](#)

[Clone or download](#)

Switch branches/tags

Find or create a branch...

Branches

Tags

✓ dev

ljq

lyq

lzh

master

mgsweet

test

良好分工  
良好的分支管理  
大家一起维护

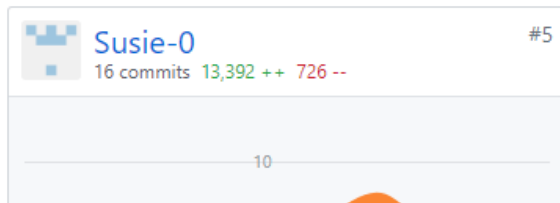
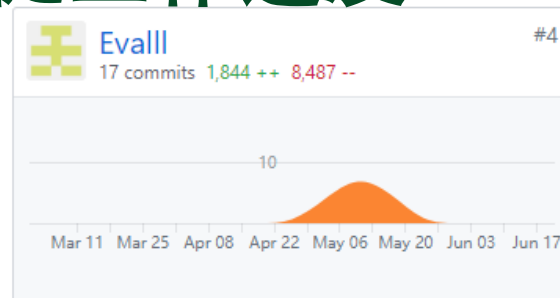
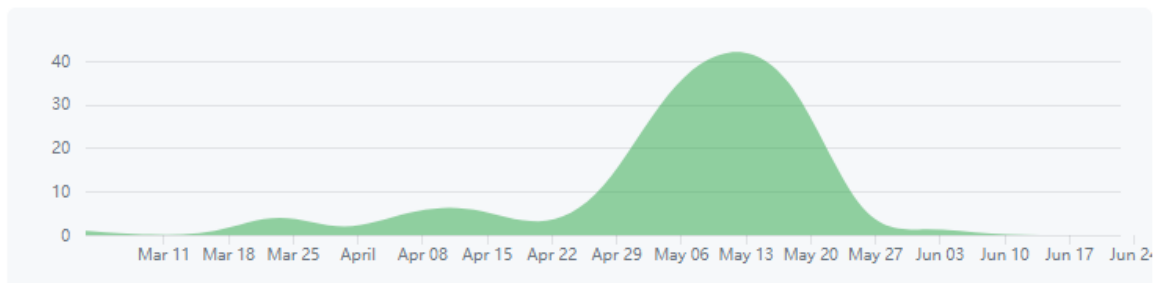
## Lesson-Attendance-WechatApp

A lesson check system. 中山大学微信小程序签到系统

该签到微信小程序是为了完成教务处的考勤要求，方便老师统计学生本学期课程出勤率，督促学生参与到课堂而开发的基于微信平台的签到系统。

规范

Contributions to master, excluding merge commits



通过贡献图  
了解并督促工作进度



Search or jump to...



Pull requests

Issues

Marketplace

Explore



🔒 mgsweet / Lesson-Attendance-WechatApp Private

👁 Unwatch ▾

3

★ Star

0

Fork

0

↔ Code

🔔 Issues 0

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

📊 Insights

⚙ Settings

Branch: dev ▾

Lesson-Attendance-WechatApp / doc /

Create new file

Upload files

Find file

History

This branch is 68 commits ahead, 4 commits behind master.

🔗 Pull request

🔗 Compare



823046544 收尾

Latest commit 18cce87 9 days ago

..

📁 00 总文档	字体统一	14 days ago
📁 01 文档规范参考	fix doc prob	a month ago
📁 02 文档_邱兆丰	add doc	14 days ago
📁 03 文档_苏思婷	用例析取	14 days ago
📁 04 文档_林逸淇	总文档&合并类图	14 days ago
📁 05 文档_林子衡	收尾	9 days ago
📁 06 文档_林杰群	查看个人签到记录&登录系统-用例析取	15 days ago

文档进度监督





# 项目文档分享

解析各个知识点



# 简述

我们以我们的项目为例，针对各个章节分享我们系统分析与设计的过程，包括需求分析、架构设计、用例分析、子系统及其接口设计、部件设计等内容，分享我们对可视化建模过程和相应的制品的理解

# 1.需求分析

# 1.1 问题陈述

## 介绍问题现状、设计和开发本系统的原因及背景

为了完成教务处的考勤要求，方便教师统计学生本学期课程出勤率，督促学生参与到课堂而开发了一个基于微信平台的签到系统。

## 介绍参与使用系统的成员角色

系统内教师和学生以课程为单位进行发起签到和完成签到以及对课程进行相应的管理。本签到系统需要教师和学生通过绑定微信并上传其真实姓名及学/工号成为注册用户，而实名信息由系统管理员维护。

# 1.1 问题陈述

简要介绍核心的功能和使用流程  
说明将开发一个什么系统。

在每门课程开始前，可以由用户创建课程，备注基本课程信息。在成功创建课程后，系统会自动生成唯一识别该课程的课程ID以及二维码。系统允许用户在自己创建的课程内进行删除课程和删除已加入课程的用户的操作，以及把课程管理权限分配加入课程的用户如助教、教务人员等，协同管理课程签到。

用户通过搜索相应的课程ID或扫描二维码加入该课程，如果课程结束或者误加课程，用户可以选择退出课程。

当需要签到时，具有管理权限的用户可以发起签到任务，并自定义签到时限和附加签到验证。若无任何验证，在签到时限内用户进入系统签到界面即可完成该次签到并显示签到成功。如果附加PIN码验证，发起签到的用户会收到系统随机生成标识该次签到的PIN码，在场的用户通过输入签到发起者公布的PIN码可完成PIN码验证。如果附加地理验证，加入课程用户在进入签到系统后，系统会尝试获取用户的地理位置，如果其地理位置与签到发起者的地理位置在合理范围内，则会满足地理验证。在签到时限内满足所有验证后，用户进入系统签到界面即可完成该次签到并显示签到成功。

具有管理权限的用户能够随时查看或导出所有已加入课程用户的签到记录，而加入课程的用户可以随时查看自己的历史签到记录。

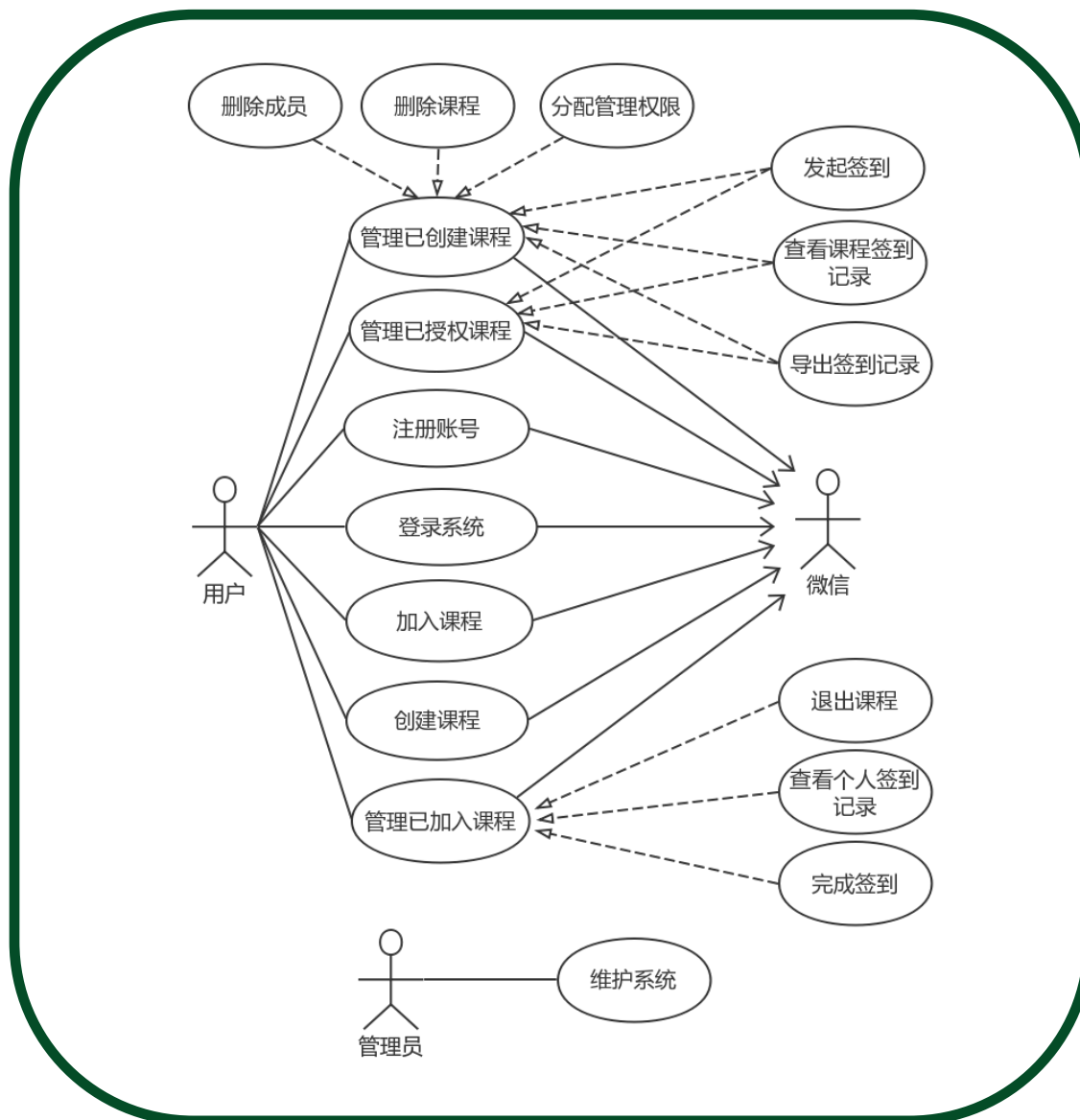
# 1.1 问题陈述

## 最后简述程序实现平台和使用语言

本系统基于微信平台的小程序实现，采用JavaScript语言，主要面向移动平台。

## 1.2 用例析取

该部分对本系统的用例进行了析取，得到本系统的用例图



## 1.3 用例规约

该部分需要对本系统的用例进行详细的描述，其中包括名称、简要说明、参与者、事件流、特殊需求、前置条件、后置条件、活动图等。

# 1.3 用例规约

## 1.3.13 退出课程

用例名称	退出课程
简要说明	用户可以退出之前已经加入的课程。
参与者	用户
事件流	1) 用户从加入的课程列表中选择课程 2) 系统显示该课程详细内容 3) 用户确认退出课程
特殊需求	无
前置条件	当前用户已经在后台管理成功登陆并且已经加入到对应的课程
后置条件	用户退出课程后，系统会有确认反馈，而且后台数据库会取消关联
活动图	<pre>graph TD; Start(( )) --&gt; A[打开已加入课程列表]; A --&gt; B[选择欲退出课程]; B --&gt; C[退出课程]; C --&gt; End((( )));</pre>

用词要准确，不能出现二义性

成双出现的符号要注意，例如同步条和开始结束符号

图要有名字，文字大小不能太小

图 1-14 退出课程活动图



## 1.4 补充规约

该部分描述了整个系统全局性的非功能需求，如可用性、可靠性、可支持性、安全性等。

- 1.目标
- 2.范围
- 3.参考
- 4.可靠性
- 5.安全性
- 6.可用性
- 7.性能
- 8.可支持性
- 9.可扩展性
- 10.设计约束

# 1.5 术语表

对一些文本中的特殊词汇进行解释，格式如下

术语	定义和信息
用户	所有注册该系统的用户。
成员	在一门课程中除了课程创建者外的所有用户。
课程	用户发起和完成签到的单位。
管理权限	加入课程的用户允许在该课程中发起签到、查看课程签到记录、导出签到结果。
已授权课程	课程创建者对用户加入某一课程后授予管理权限。
课程 ID	在用户创建一门课程后，系统自动对该课程生成唯一的标识码 <u>做为该</u> 课程的课程 ID。
Pin 码	由系统生成的 4-6 位的 <u>短数字码</u> ，用于判断签到的到场证明之一。
<u>二维码</u>	由系统生成的二维码，允许用户通过 <u>扫二维码加入</u> 课程或注册系统。
地理位置	通过获取移动端的地理定位信息来判断用户当前地理位置。
导出签到记录	允许用户以某种文件格式将签到记录导出到本地。

## 2.架构设计

本部分首先说明本系统使用了什么架构，然后给出相应的架构图和关键抽象

## 框架描述

### 2.1 架构描述

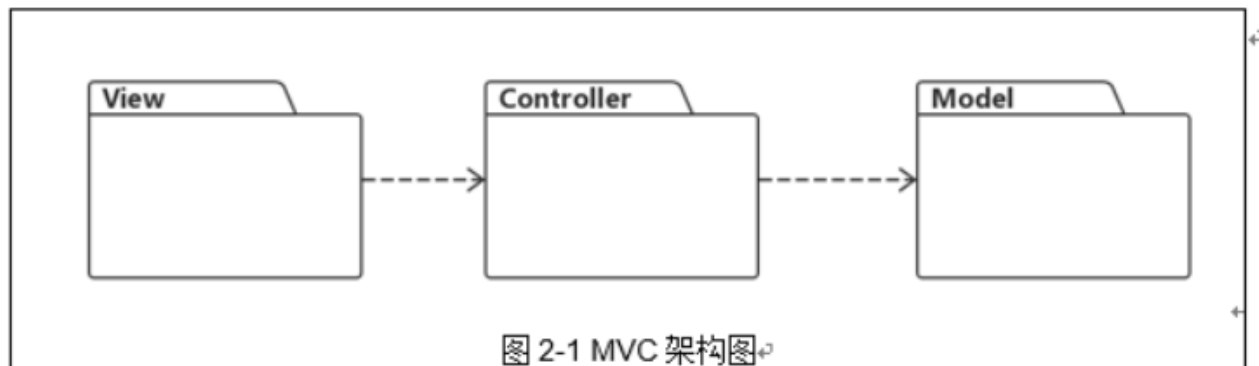
本系统采用 MVC 架构实现。

MVC 架构由三个层次组成：模型层 (Model)、视图层 (View) 和控制器层 (Controller)。视图层即图形界面，为用户提供交互界面；控制器层负责转发和处理请求，控制业务逻辑；模型层则负责封装与调用的底层数据。

# 架构图

## 2.2 架构图

根据2.1的描述，图文结合，画出架构图



# 关键抽象

## 2.3 关键抽象

良好的关键抽象有助于数据库建表



# 3.类的设计

## 3.1 类的析取

在这一节中，需要找出每个用例的三种类型（边界类、控制类、实体类），确定类的属性和操作，画出对应的用例析取图和时序图。

具体流程：

- 1.找出三种类型，对三种类型进行简单的描述
- 2.画出用例析取图
- 3.画出时序图与协作图（确定类的操作）
- 4.对时序图及协作图进行简单的描述

## 下面以发起签到为例分析

### 找出三种类

边界类：SignInSubmitFrom。SignInSubmitFrom为发布签到页面。↵

控制类：SignInSubmitControler。SignInSubmitControler控制类负责处理关于发布签到的相关操作，包括查看正在进行签到的信息、发布新的签到。↵

实体类：RecordModel。RecordModel实体类表示签到记录实体，包含课程编号、签到设置、签到时间、是否签到、签到者编号。↵



## 画出用例析取图

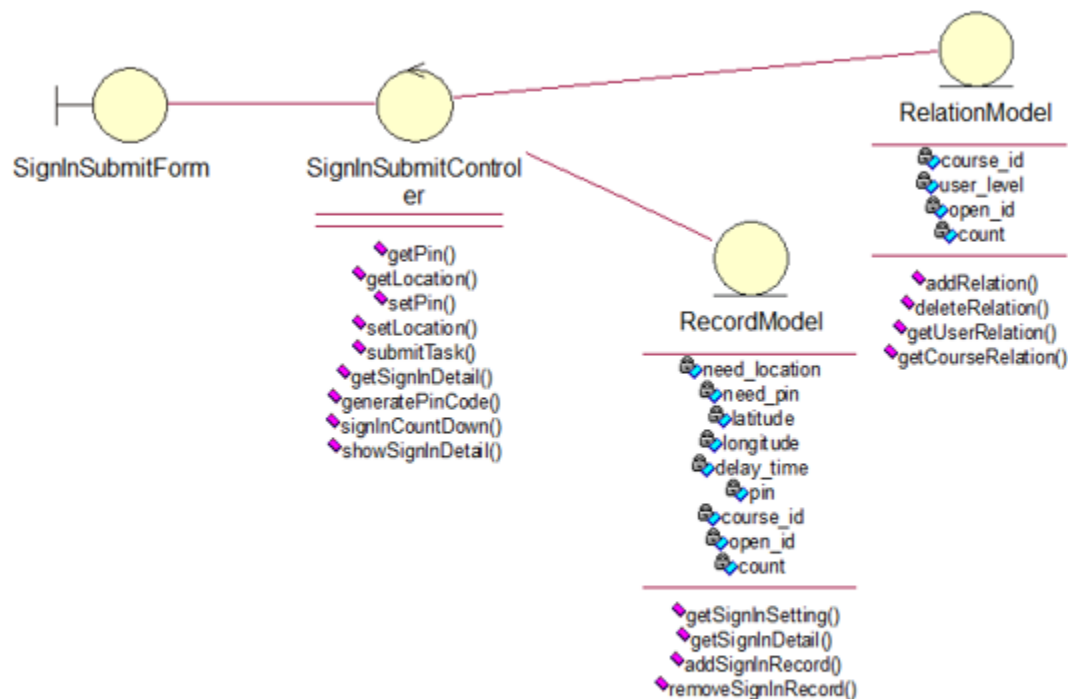


图 3-9 发布签到用例析取类图

Tips: 将所有图统一到Rational Rose 有助于快速拖动创建

## 画出时序图和协作图

### 时序图

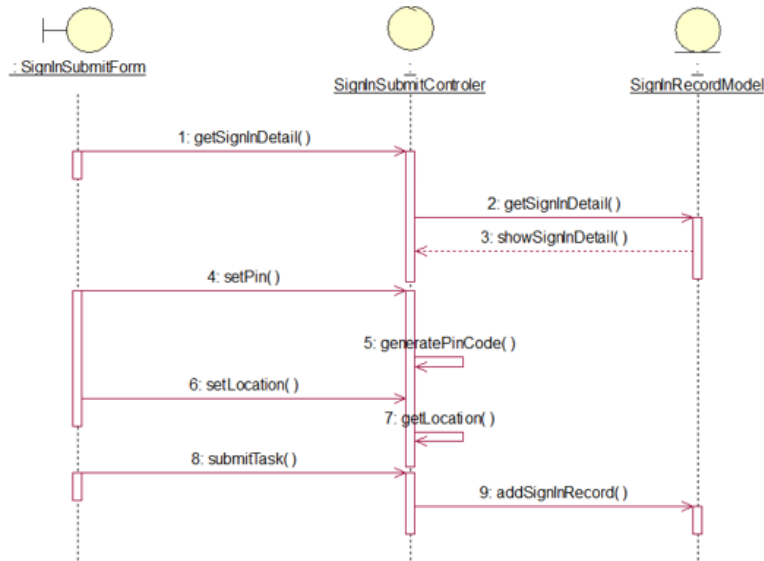


图 3-10 发起签到用例析取时序图

### 协作图

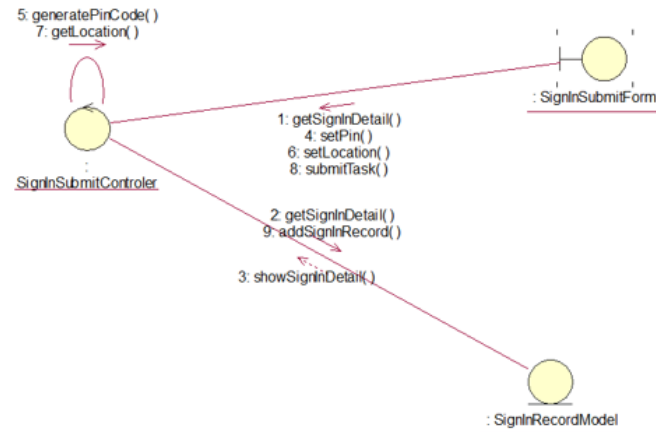


图 3-11 发起签到用例析取协作图

Tips: 利用Rational Rose 的F5快捷键可以实现一键转化时序图为协作图，减少任务量

## 简单描述

用户进入页面后自动调用 `getSignInDetail()` 获取数据库中签到状态, 如果已经发起了签到, 则不会显示签到设置界面。进入发布签到界面后用户点击 PIN 和地理验证的按钮后系统会分别调用 `setPin()` 和 `setLocation()` 更新设置。当用户选择发布按钮后, 系统调用 `submitTask()` 会将本地设置发送到服务器。服务器将本次签到记录到数据库中, 从而完成签到发布。↵

## 3.2 系统分析机制

本节根据补充规约得到上述边界类、控制类、数据类需要满足的非功能性需求，列出系统的分析机制表，如下表所示。

### 3.2 系统分析机制

分析类 (analysis classes)	分析机制(analysis mechanism)
CourseModel	持久性、安全性
UserModel	持久性、安全性
SignInRecordModel	持久性、安全性

表 3-1 微信签到系统分析机制表

## 3.3 合并分析类

本节将析取出来的边界类、控制类、数据类进行合并整理，得到系统的合并类图。

3.3 系统总体类的设计

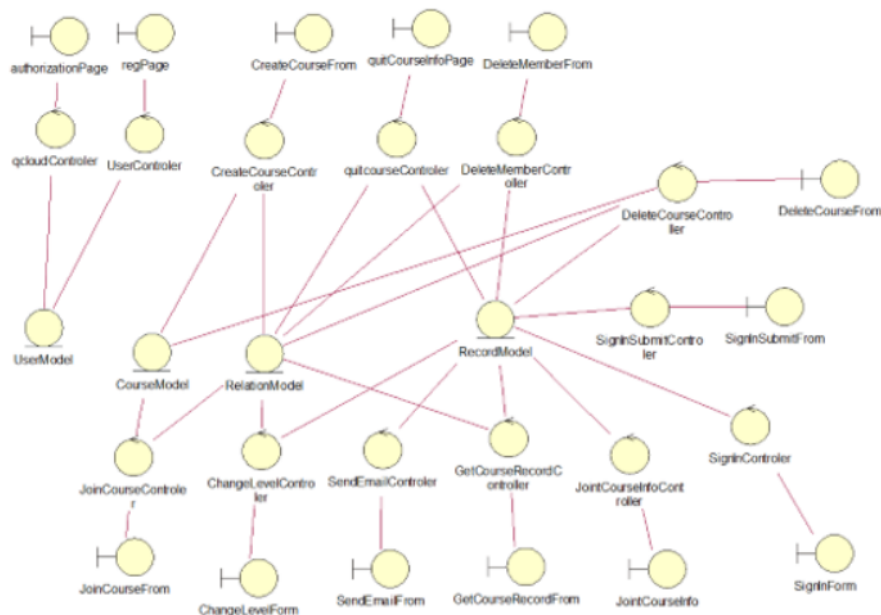


图 3-26 系统总体合并类图

# 4.子系统及其接口设计

但由于我们的系统比较简单，没有引入子系统的缘故，所以在该章节上没有做太多文章，只在章节中如下表示

#### 4. 子系统及其接口设计↵

##### 4.1 确定设计类↵

本项目组对第三章的分析类进行了分析与检查，以确定其是否能成为设计类。经过分析发现，第三章中所有分析类均为单逻辑，不需要进行类的分解或合并，因此不做修改，设计类如第三章所示。↵

##### 4.2.定义子系统↵

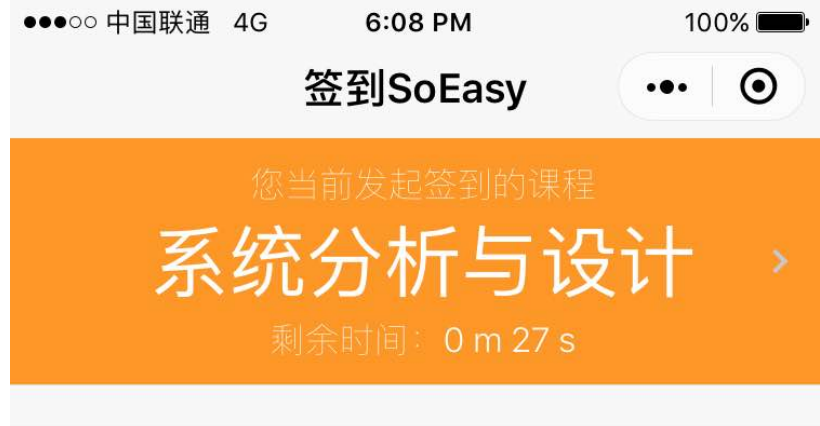
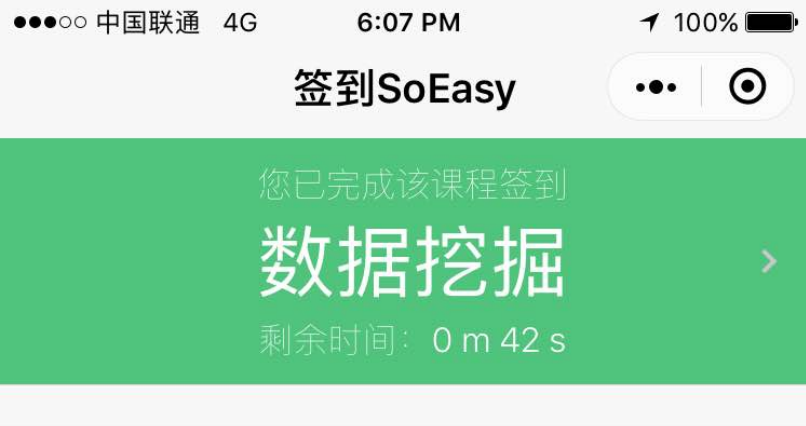
经过本项目组分析，本系统无需进行子系统设计及其接口设计。↵

# 5.部件设计



# 系统展示

SHOW TIME



简介清晰的界面  
多种状态分色显示

课程id:6

# 系统分析与设计 嵌入式

已发起1次签到

发起签到 >

签到统计 >

管理成员 >

显示班级二维码 >

修改课程信息 >

导出签到统计到邮箱 >

删除课程

丰富健全的签到班  
级管理功能

# 可视美观的签到统计记录

●●●○○ 中国联通 4G

6:10 PM

99%

<

个人签到记录

⋮ | 🎯

数据挖掘 从入门到放弃

2/3

●

●

●

签到纪录

第1节课签到成功2018/06/24 18:07:30

第2节课签到成功2018/06/24 18:09:02

第3节课缺勤

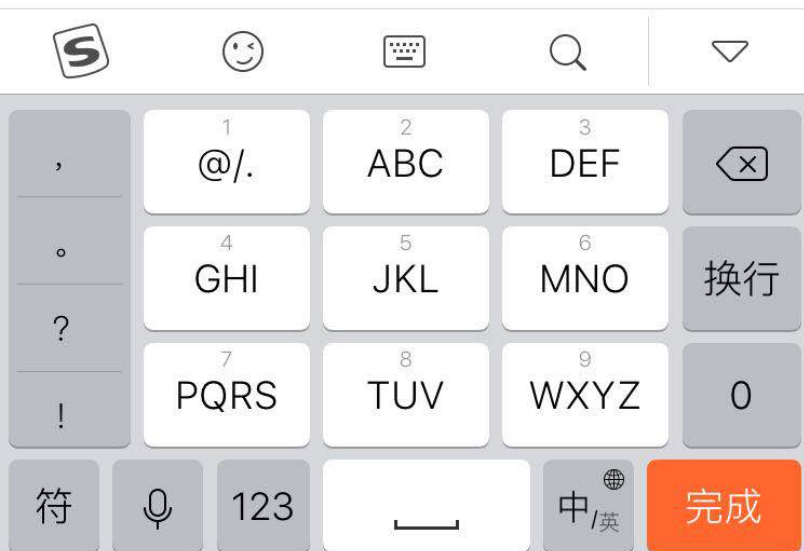
退出课程

邮箱地址

请输入您的邮箱地址...

注意：邮件可能会被当作垃圾邮件

确认发送



你甚至可以导出签到记录到邮箱

**What's more?**

课程ID: 6

## 系统分析与设计

BYE~!



BYE~!

可通过扫描本二维码进入班级