



中山大学  
数据科学与计算机学院  
软件工程  
电子政务方向

## 课室在线查询预约系统-WeClass

---

课程信息：《软件设计综合实验》

指导老师：王青

姓名：庞景龙 administrator 15331250

莫疑 developer 15331244

杨金辉 developer 15331357

杨玉楠 developer 15331368

卢健彬 developer 15331225

王子城 developer 15331306

日期	版本号	描述	编制者
2018-03-10	V 0.1	问题陈述	王子城
2018-03-14	V 0.2	用例规约：登录、查询、关键字	杨金辉
2018-03-15	V 0.3	用例规约：查看申请列表、审批、修改	卢健彬
2018-03-15	V 0.4	用例规约：查看预约状态、个人信息管理、取消	莫疑
2018-03-15	V 0.5	用例规约：预约课室；排版	庞景龙
2018-03-15	V 0.6	术语表	王子城
2018-03-18	V 1.0	第一次作业版本	全体
2018-03-19	V 1.1	修改用例规约的表达，修改登录系统、修改个人信息为修改密码和预约查询及删除的用例规约内容	全体
2018-03-20	V 1.2	修改文档整体的表达，统一用例规约的格式及排版，修改预约信息修改用例规约	莫疑
2018-03-21	V 1.3	添加补充规约和术语表	杨玉楠、王子城
2018-03-22	V 2.0	第二次作业版本	全体
2018-03-24	V 2.1	删除补充规约，添加系统框架、系统关键抽象	杨玉楠
2018-04-04	V 2.2	更新补充规约、系统框架描述	卢健彬
2018-04-06	V 3.0	第三次作业版本	全体
2018-04-07	V 3.1	补充术语表、添加系统架构图、系统关键抽象	全体
2018-04-21	V 3.3	添加三个用例分析	庞景龙
2018-04-30	V 3.4	完善用例分析描述、整理用例分析类格式	庞景龙
2018-05-11	V 3.6	修改用例分析、添加描述分析机制	杨玉楠、王子城
2018-05-11	V 3.7	添加用例的类图	王子城
2018-05-11	V 4.0	第四次作业版本	全体
2018-05-15	V 4.1	全面更新用例时序图、类图以及系统类图	王子城
2018-05-17	V 4.2	更新补充规约、分析机制、系统关键抽象	杨玉楠
2016-05-18	V 5.0	第五次作业版本	全体

WeClass 项目组

---

2018-06-12	V 5.1	添加组件设计	全体
2018-06-14	V 5.2	确定 App 图标，完成文档	全体
2018-06-14	V 6.0	第六次作业版本	全体

目录

引言.....	8
文档用途.....	8
阅读对象.....	8
中山大学课室查询预订 App-WeClass.....	9
1 需求分析.....	9
1.1 问题陈述.....	9
1.2 用例析取.....	11
1.3 用例规约.....	11
1.3.1 登录用例规约.....	11
1.3.2 查询课室规约.....	13
1.3.3 审批申请用例规约.....	15
1.3.4 管理预约状态用例规约.....	18
1.3.5 修改密码用例规约.....	20
1.3.6 预约课室用例规约.....	23

1.4 补充规约	25
1.4.1 可靠性	25
1.4.2 安全性	26
1.4.3 可用性	26
1.4.4 性能	26
1.4.5 可支持性	26
1.4.6 可扩展性	26
1.4.7 实时性	26
1.4.8 防止冲突	26
1.5 术语表	26
2 架构设计	28
2.1 系统架构描述	28
2.2 系统架构图	28
2.3 系统关键抽象	28
3 用例分析	29

3.1 用例析取	29
3.1.1 登录用例析取	29
3.1.2 审批申请用例析取	31
3.1.3 管理预约状态用例析取	33
3.1.4 密码修改用例析取	34
3.1.5 预约课室用例析取	36
3.2 分析机制	38
Table 课室申请系统的类分析机制	38
3.3 补充用例规约	38
4 组件设计	39
4.1 分析并发需求	39
1) 申请人数超过现有空闲课室数目	39
4.2 针对需求的设计方案	39
4.2.1 针对申请人数超过现有空间课室数目	39
4.3 映射到现实系统	39

# 引言

## 文档用途

本文档的编写目的是让读者对中山大学在线查询预约课室系统的需求和设计有全局性、总体方面的了解。

## 阅读对象

此文档将适合以下人员阅读：

- 本项目组成员
- 对本系统感兴趣的研究人员或教务工作者



# 中山大学课室查询预订 App-WeClass

## 1 需求分析

### 1.1 问题陈述

对于一名在校大学生而言，每天的学习时间占了日常活动的大部。因此，一个良好的学习空间是每一位大学生都十分需要的，然而由于校内图书馆的座位资源有限，所以很多同学会选择去教室进行自习。同时，为了增强大学班级的凝聚力，常常需要在课室内举办班会或者一些班级聚会。所以，教室资源利用效率的提高对于高等教育的发展和学校办学效益都具有重要意义。

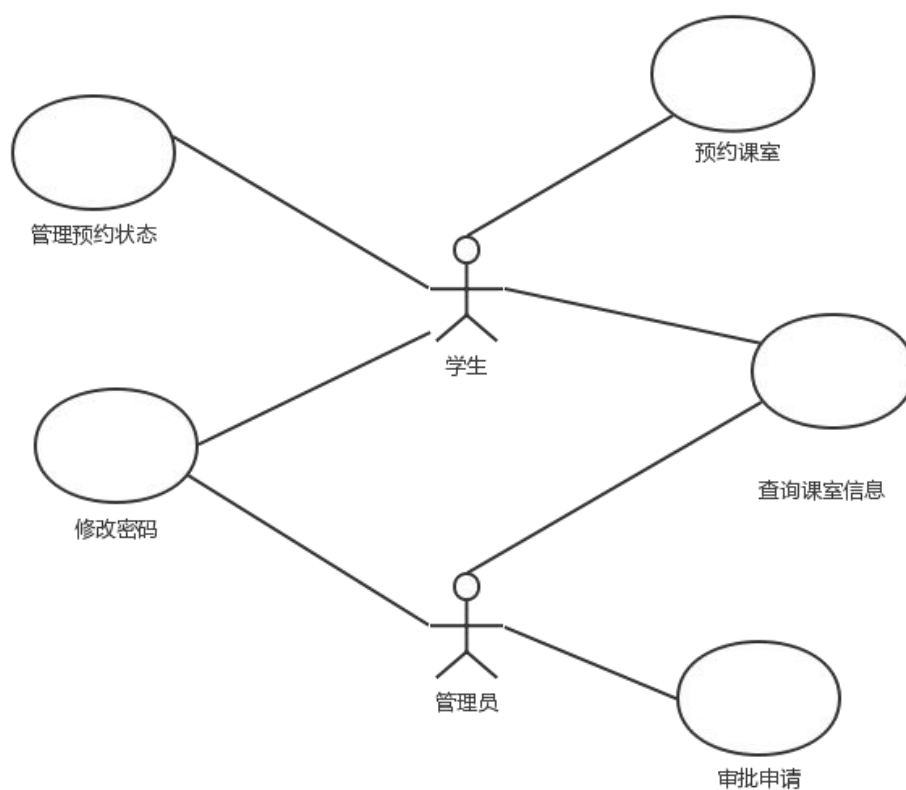
通过实地考察，我们分析了本校的教室资源利用率情况、各类型教室的分布、教室座位数量和设备情况、上课课时安排。之后进行对一些学生和社团负责人的走访询问，发现目前校内学生申请课室的流程较为复杂，手续麻烦，申请办理时长长，空闲教室资源查询操作复杂等问题。因此，学生和审批的老师都缺少一个集合度高的，条理清晰，功能明确的课室资源查询及申请平台，通过这一平台，实现对本校的教室资源的优化配置。

基于此，本项目组将设计和开发一个课室资源查询及申请管理系统。该系统使用 ThinkAndroid 开发框架，基于 MVC 模块三层架构，集管理、查询和申请等功能于一身，为学生用户提供了查询空闲课室和申请空闲时间段课室的平台。该系统设定两个角色，分别为管理员和学生用户。

管理员功能较为单一,仅负责审核上传的课室申请条目并作出反馈,审核结果为“通过”或“不通过”,未审核信息显示为“待审核”。同时管理员可以查看和修改课室状态,课室状态为“空闲”或“已申请”,已预约但暂未通过的为“待审批”。

学生用户能够登录系统、修改个人信息、查看当前空闲课室及空闲时间段申请、查看某课室的信息和预约情况、通过课室名字,大小,位置等进行信息筛选以及对空闲课室进行预约和取消已预约的课室申请等。

## 1.2 用例析取



## 1.3 用例规约

### 1.3.1 登录用例规约

#### 1.3.1.1 简要说明

本用例描述了学生和管理员登录的过程

#### 1.3.1.2 参与者

用户：学生和管理员

#### 1.3.1.3 事件流

##### I.基本事件流

A: 系统将登录所需填写的内容列表显示给用户，内容包括学工号、密码和身份；

B: 用户点击选择身份：学生或者是管理员；

C: 用户按照列表内容填写信息；

D: 用户点击登录；

D1: “学工号” 未填写；

D2: “密码” 未填写；

D3: “学工号” 与 “密码” 不匹配；

D4: “学工号” 与 “密码” 匹配，与身份不匹配；

E: 登录成功，进入到用户主界面。

## II. 后备事件流

D1: “学工号” 未填写

当用户未填写学工号而点击“登录”时，系统提示“学工号不能为空”；用户返回基本事件流第三步。

D2: “密码” 未填写

当用户未填写密码而点击“登录”时，系统提示“密码不能为空”；用户返回基本事件流第三步。

D3: “学工号” 与 “密码” 不匹配

当用户填写的学工号和密码不匹配而点击“登录”时，系统提示“学工号或密码错误”；用户返回基本事件流第三步。

D4: “学工号” 与 “密码” 匹配，与身份不匹配

当用户填写的学工号和密码匹配但是和身份不匹配而点击“登录”时，系统提示“学工号或密码错误”；用户返回基本事件流第二步。

### 1.3.1.4 特殊需求

无

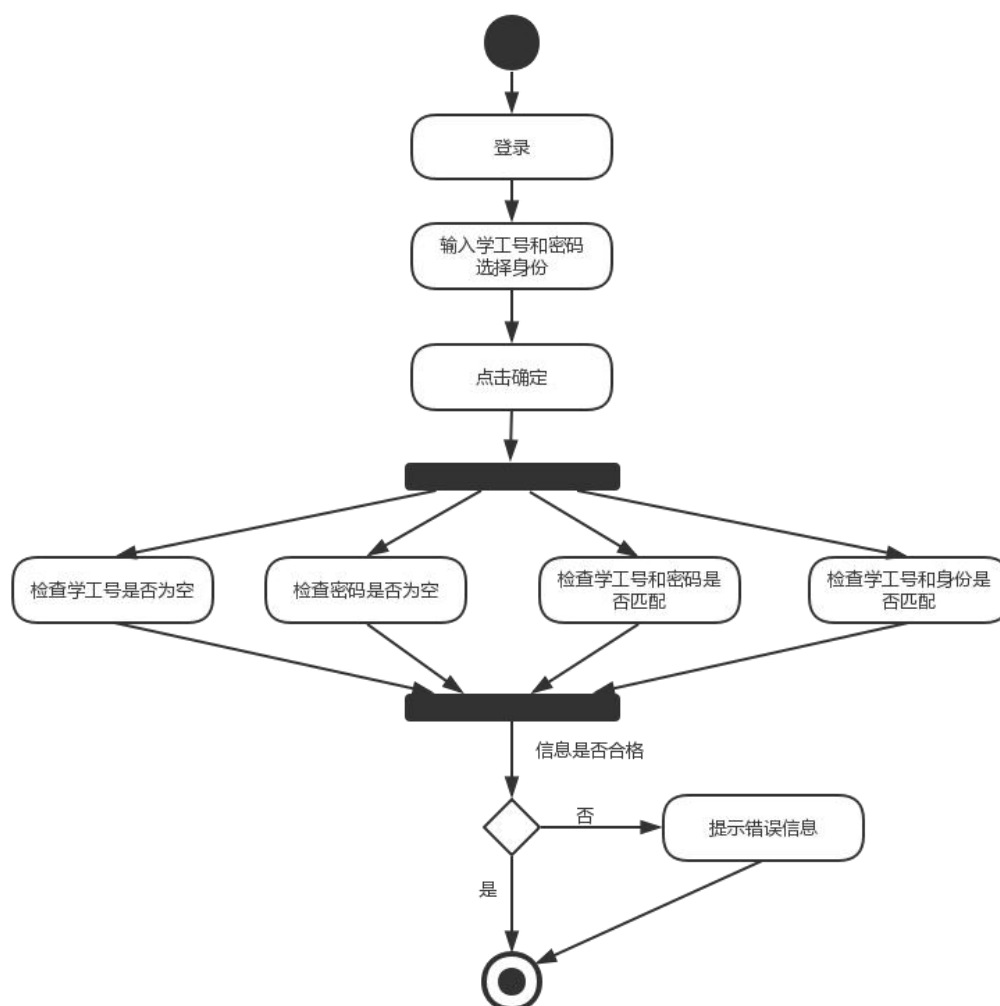
### 1.3.1.5 前置条件

无

### 1.3.1.6 后置条件

登录成功，进入到用户主界面

### 1.3.1.7 活动图



## 1.3.2 查询课室规约

### 1.3.2.1 简要说明

本用例描述了学生和管理员查询课室信息的过程

#### 1.3.2.2 参与者

用户：学生和学校管理员

#### 1.3.2.3 事件流

##### I.基本事件流

用例开始于用户已登录系统，并且点击“查看课室信息”选项。

A. 系统将所有教室条目列在界面，用户通过上下翻页浏览课室信息。

B. 点击教室名字，进入该教室具体使用情况的页面。

#### 1.3.2.4 特殊需求

无

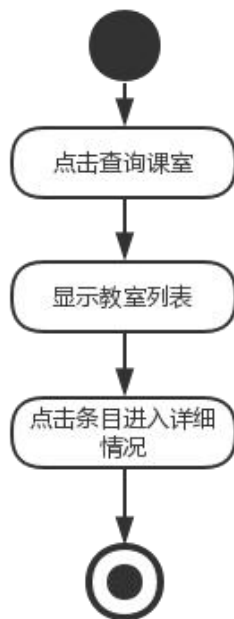
#### 1.3.2.5 前置条件

本用例开始前学生/管理员已经登录系统并且点击“课室信息”

#### 1.3.2.6 后置条件

系统将所有教室条目列在界面

#### 1.3.2.7 活动图



### 1.3.3 审批申请用例规约

#### 1.3.3.1 简要说明

本例描述了管理员用户查看、审批申请的过程。

#### 1.3.3.2 参与者

学校管理员

#### 1.3.3.3 事件流

##### 1.3.3.3.1 基本事件流

用例开始于管理员用户已登录系统，查看申请条目并在某条目的详细信息页面点击"通过"或者"拒绝"进行审批。

A. 系统将申请列表的简要内容显示给管理员用户。内容包括：申请日期、预约课室的时间、课室编号、预约申请人的姓名、学号和联系方式。

B. 学校管理员点击条目进入详细申请信息页面，一旦选择需要操作，  
以下的一条子事件流将被执行：

选择通过，“审批通过”子事件流被执行；

选择拒绝，“审批拒绝”子事件流被执行。

C.审批通过

A) 管理员点击通过

C1：该申请与被申请课室的状态冲突

B) 系统提示“审批成功”，将申请状态更新到数据库，并返回  
申请列表

D.审批拒绝

A) 管理员点击拒绝

B) 系统弹窗提示管理员输入拒绝理由，输入完成点击确认

C2：未输入拒绝理由就点击确认

C3：点击取消

C) 系统提示“审批成功”，将申请状态更新到数据库，并返回  
申请列表

1.3.3.3.2 后备事件流

C1: 该申请与被申请课室的状态冲突

当该申请与现有的活动有冲突(如该课室在该时段已被预约)时，  
系统提示“该申请与现有活动有冲突，请选择拒绝。”管理员回到  
基本事件流的第二步。

C2: 未输入拒绝理由就点击确认。



当管理员选择“拒绝”选项但未填写不通过理由时,系统提示“不通过理由不能为空!” 管理员回到基本事件流的第二步。

C3: 点击取消

管理员回到基本事件流第二步。

#### 1.3.3.4 特殊要求

无

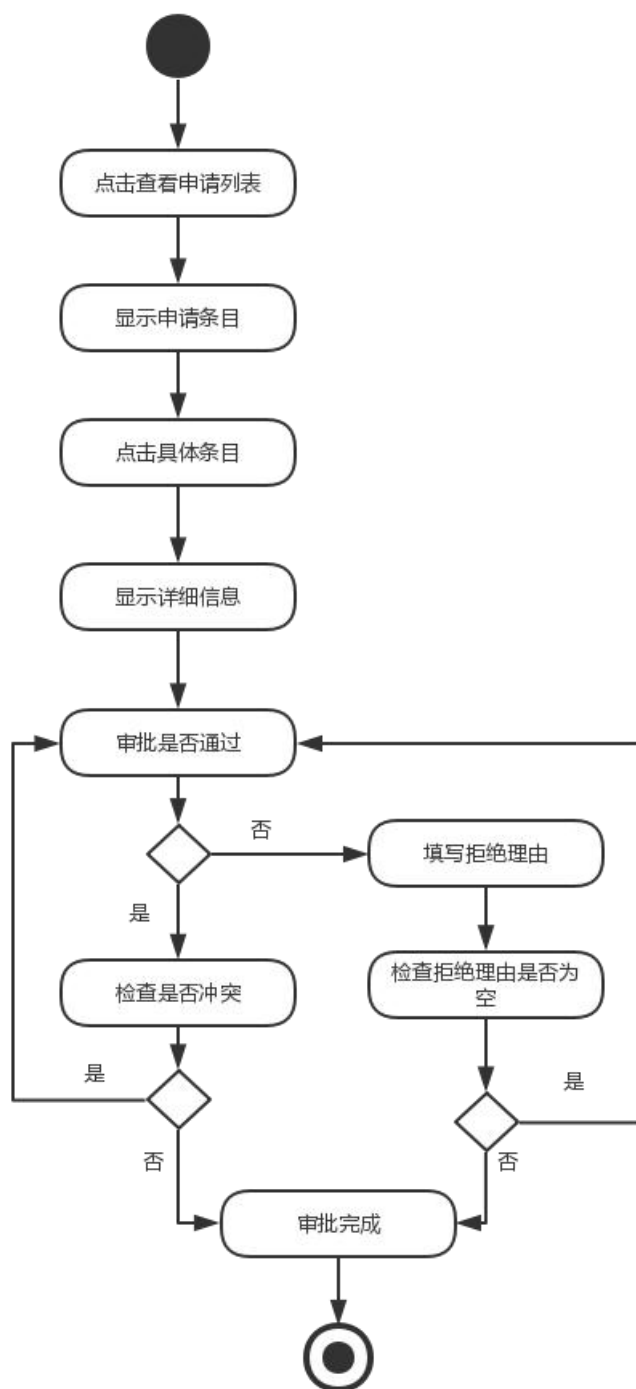
#### 1.3.3.5 前置条件

本用例开始前管理员用户以登录系统并点击查看申请列表。

#### 1.3.3.6 后置条件

如果用例成功,若通过申请,系统向申请人发送申请成功的消息;若不通过申请,系统向申请人发送申请失败的消息,并提示失败原因。

#### 1.3.3.7 活动图



### 1.3.4 管理预约状态用例规约

#### 1.3.4.1 简要说明

本用例描述了学生管理预约状态的过程。

#### 1.3.4.2 参与者

学生

#### 1.3.4.3 事件流

##### 1.3.4.3.1 基本事件流

用例开始于用户已登录系统，并且点击“预约状态”选项。

A. 系统将用户已预约的所有教室条目列在界面，用户通过上下翻页浏览课室信息，若无预约，则界面显示为空。

B. 点击课室条目查看详情后，通过点击“取消预约”取消课室申请预约，系统提示是否确认取消预约

C. 点击确认，系统提示“成功取消预约”，将数据更新到数据库并返回预约列表

C1. 点击取消

##### 1.3.4.3.2 后备事件流

C1. 点击取消

回到基本事件流第二步

#### 1.3.4.4 特殊条件

无

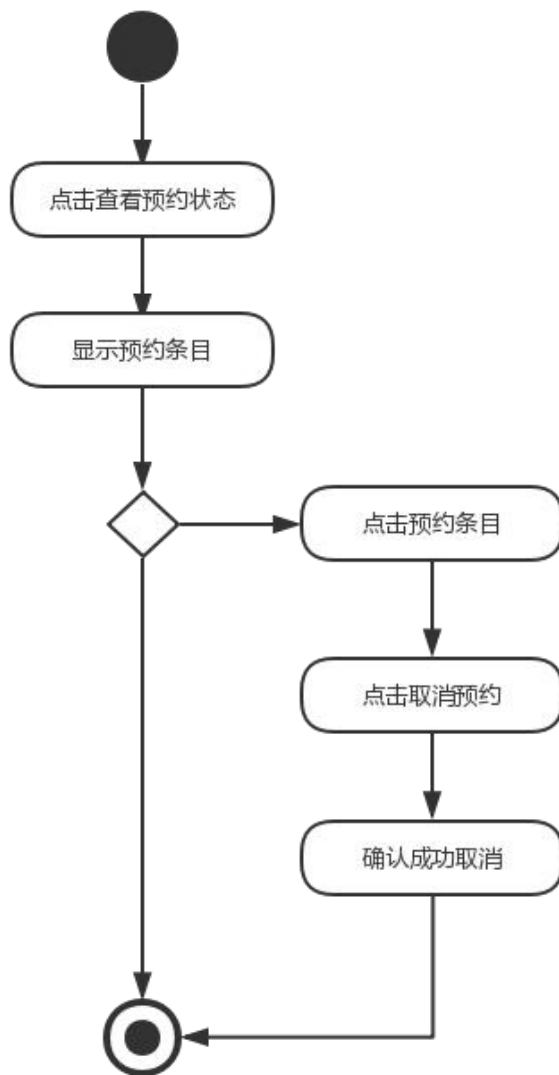
#### 1.3.4.5 前置条件

本用例开始前学生已经登录系统并且点击“预约状态”。

#### 1.3.4.6 后置条件

如果本用例成功，系统将更改后的课室预约信息更新到数据库；若失败，系统状态不改变。

#### 1.3.4.7 活动图



### 1.3.5 修改密码用例规约

#### 1.3.5.1 简要说明

本用例允许学生和管理员对登录密码修改。用户不允许修改用户名（即学工号）。本系统设定每年新生入学时导入新生信息，用户名为学号，初始密码为 12345678。

#### 1.3.5.2 参与者

本校学生和系统管理员。

#### 1.3.5.3 事件流

##### 1.3.5.3.1 基本事件流

用例开始于用户已登录系统，并且点击“修改密码”选项。

A. 系统所需填写的内容列表显示给用户，内容包括原密码，新密码和新密码确认；

B. 用户输入原密码，新密码并点击确认；

B1: “原密码”未填写

B2: “原密码”不匹配

B3: “新密码”未填写

B4: “新密码”和原密码相同

B5: “新密码确认”未填写

B6: “新密码确认”与“新密码”内容不一致

C.系统提示“修改密码成功”，将密码更新到数据库，返回主页面。

##### 1.3.5.3.2 备用事件流

B1: “原密码”未填写

当用户未填写“原密码”而点击“确认”时，系统提示“原密码不能为空”；用户返回基本事件流第二步。

B2: “原密码” 与数据库中不匹配

当用户输入错误原密码并点击确认，系统提示“原密码错误”；  
用户返回基本事件流第二步。

B3: “新密码” 未填写

当用户未填写“新密码”而点击“确认”时，系统提示“新密码  
不能为空”；用户返回基本事件流第二步。

B4: “新密码” 和原密码相同

当用户填写的新密码和原密码相同时并点击“确认”后，系统提  
示“新密码不能和原密码相同”；用户返回基本事件流第二步。

B5: “新密码确认” 未填写

当用户未填写“新密码确认”而点击“确认”时，系统提示“请  
确认新密码”；用户返回基本事件流第二步。

B6: “新密码确认” 与 “新密码” 内容不一致

当用户填写的新密码和确认新密码两栏内容不一致而点击确认  
后，系统提示“密码确认错误”；用户返回基本事件流第二步。

1.3.5.4 特殊条件

无

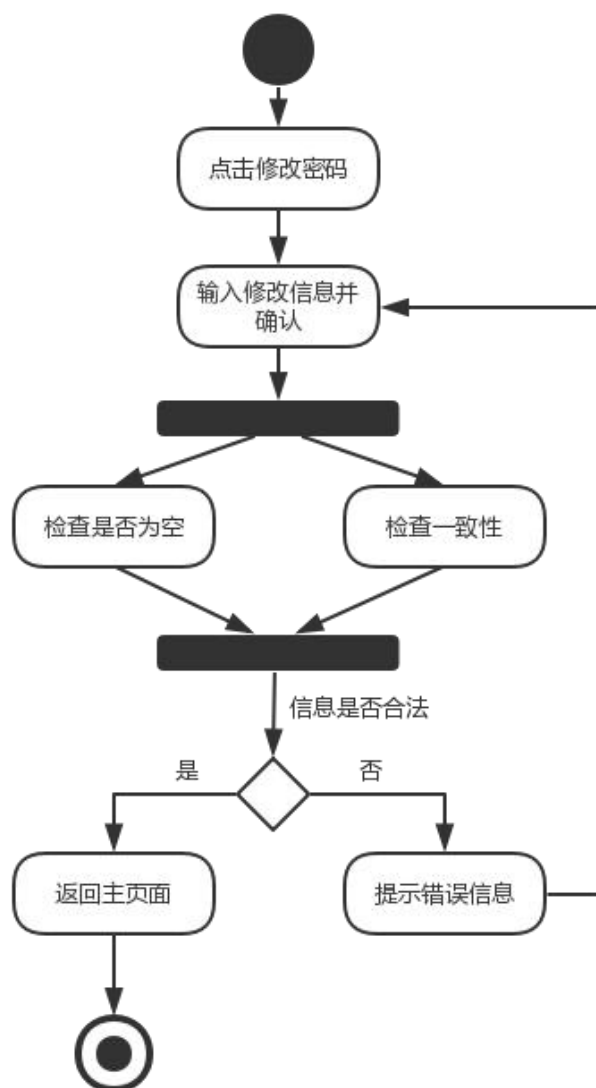
1.3.5.5 前置条件

本用例开始前用户已经登录系统并且点击“修改密码”。

1.3.5.6 后置条件

系统将数据库中用户的密码信息重置。

1.3.5.7 活动图



### 1.3.6 预约课室用例规约

#### 1.3.6.1 简要说明

这个用例实际上是本系统的主体部分。用户通过选定某个教室，查看该教室未来几天的安排情况，如果想要预约则点击预约按钮填写详细预约信息

#### 1.3.6.2 参与者

学生用户。

### 1.3.6.3 事件流

#### 1.3.6.3.1 基本事件流

用例开始于学生进入课室列表并查看某个课室具体课表，点击预约按钮进入预约页面。

- A. 学生填写学号和姓名
- B. 学生填写想要预约的日期和使用教室的开始和结束时间
- C. 学生填写申请用途和联系方式
- D. 点击确认预约
- D1：该课室已被占用
- D2：填写信息不完整，有漏填
- D3：申请日期格式不正确
- D4：手机号码格式不正确

#### 1.3.6.3.2 备用事件流

- D1：提示该课室已被占用
- D2：提示信息填写不完整，重新填写信息
- D3：提示日期格式不正确，重新填写日期
- D4：提示手机号码格式不正确，重新填写手机号码

#### 1.3.6.4 特殊条件

无

#### 1.3.6.5 前置条件

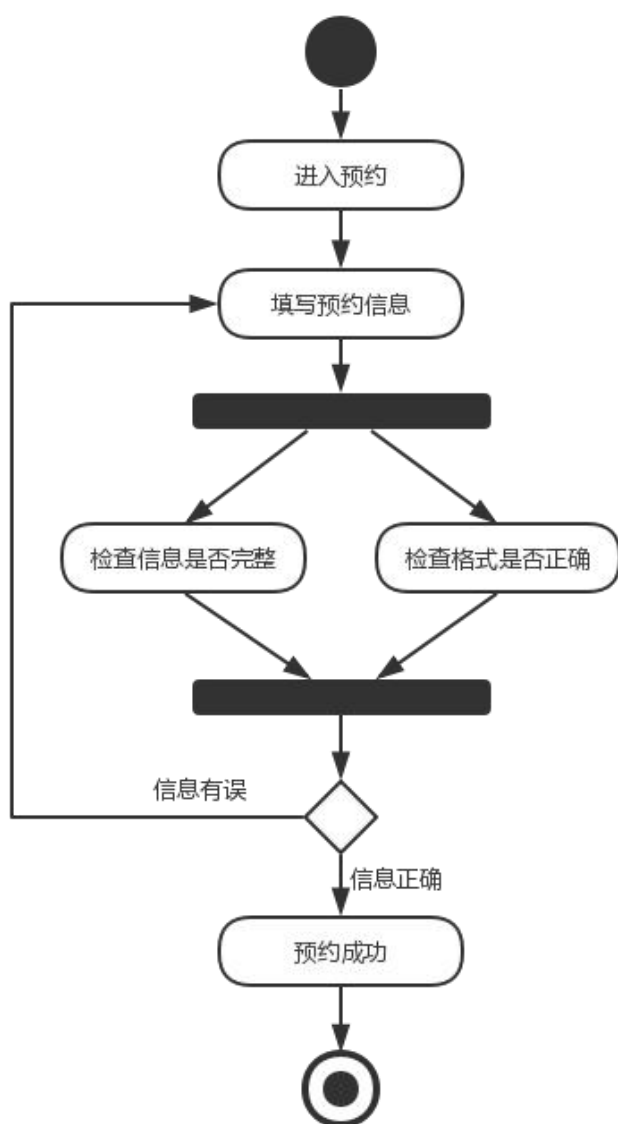
学生用户已经登录预约系统。

#### 1.3.6.6 后置条件



返回主页面

### 1.3.6.7 活动图



## 1.4 补充规约

### 1.4.1 可靠性

本系统可以保证 7\*24 小时连续运行，一周维护时间不会超过 2 小

时。

#### 1.4.2 安全性

本系统面向中山大学全体在校学生。用户名、姓名、学院、校区、入学年份由学校相关部门提供，在本系统中不得更改。

密码以密文存储在数据库中。

#### 1.4.3 可用性

本系统目标群体为中山大学在校学生。对于大学生而言，系统页面简单易理解、易操作。

#### 1.4.4 性能

本系统性能满足 1000 个人同时使用，页面反应时间不能超过 3 秒，允许大量的并发访问。

#### 1.4.5 可支持性

本系统支持主流浏览器，如 360 浏览器，IE 浏览器，谷歌浏览器等。

#### 1.4.6 可扩展性

本系统能接受对现有功能合理的改进，增加一些新的功能。

#### 1.4.7 实时性

本系统数据库信息会由管理员进行实时更新和维护。

#### 1.4.8 防止冲突

本系统同一项目的申请会依据时间进行选择，防止不同用户的选择冲突。

### 1.5 术语表

术语	含义
空闲课室查询及预约系统	提供中山大学全体在校学生在校园网上查询空闲课室、预约空闲课室以及审批相关课室预约申请的系统。文档中所涉及的“本系统”，均指空闲课室查询及预约系统。
学生用户	中山大学全体在校学生。
管理员	由系统设定，管理员可查看和审核学生用户申请预约的课室。
用户名	管理员和学生登录本系统时，所需要输入的唯一标识（用户名为学工号）。同时，需要给出相应匹配的密码才可以进入系统。
课室信息	描述一个课室的所有信息，包括：课室的位置、编号、课室的容量以及课室的空闲时间段。
课室状态	表示课室的空闲状态，共有三种：空闲中、已预约待审批、已申请。
用户预约状态	表示用户预约课室是否通过的状态，共有三种：待审批、已通过、审核不通过。
管理员审批状态	表示课室申请的审核状态，共有三种：待审核、审核通过、审核不通过。待审核表示管理员还未审核该课室申请，审核通过表示管理员已经审核并且通过该课室申请，审核未通过表示管理员已经审核但是该课室申请未成功。

## 2 架构设计

### 2.1 系统架构描述

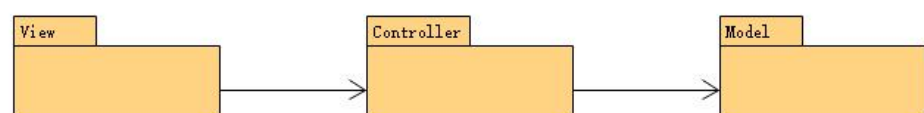
本系统基于 java 框架，采用 MVC ( Model View Controller ) 三层架构。该架构的使用实现了应用程序的分层管理，简化了后续对程序的修改和扩展，并且使程序某一部分的重复利用成为可能。

View 层又称视图层，是用户看到并与之交互的页面。在本系统中，视图层由众多 XML 文件组成，它们负责获取用户的输入及显示控制层处理的结果。

Controller 层又称控制层，它收到来自视图层的请求并对实体类进行增删改查。当单击 APP 中的 BUTTON 和发送 XML 表单时，控制层接收请求并处理请求，然后确定用哪个视图来显示处理返回的数据。

Model 层又称数据层，主要作用是存储和访问数据。

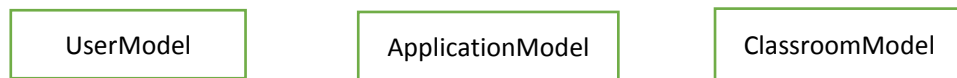
### 2.2 系统架构图



### 2.3 系统关键抽象

关键抽象即为找到系统实体类的过程。实体类为系统中存储和改

动的数据，可以从需求分析中的术语表中得到。经过分析，本系统有三个实体类，分别为用户表，申请表和课室表，用户表存储了与用户相关的一切信息，包含用户名（学工号）、密码、入学年份、联系方式等，申请表包含了用户申请课表的各种信息包括申请的课室号，具体的申请时段，申请课室的用处等，课室表包含教学楼各个课室的空闲时段和非空闲时段。



课室申请系统实体类图

## 3 用例分析

### 3.1 用例析取

#### 3.1.1 登录用例析取

根据用例规约，总结出的交互过程：

- 1、用户打开系统，进入登陆页面
- 2、用户填写用户名和密码，勾选用户身份，点击登录
- 3、用户名和密码合法，进入主页面，否则提示错误信息。

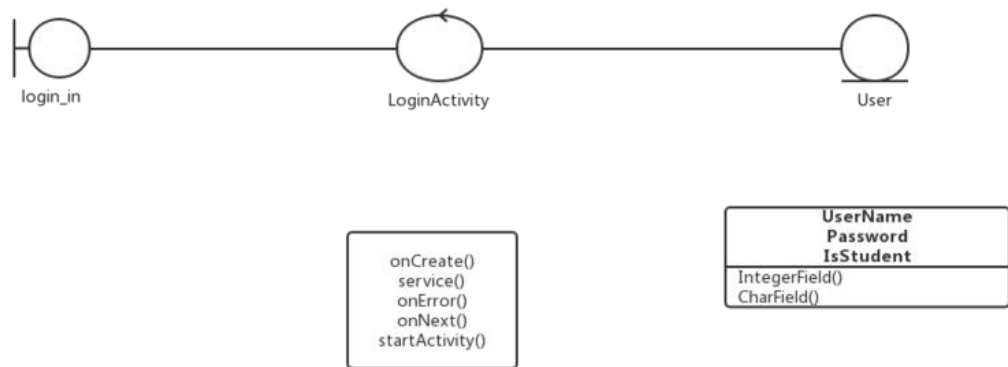
经过以上分析，得到三种类：

边界类：login\_in。login\_in 是登录的页面。

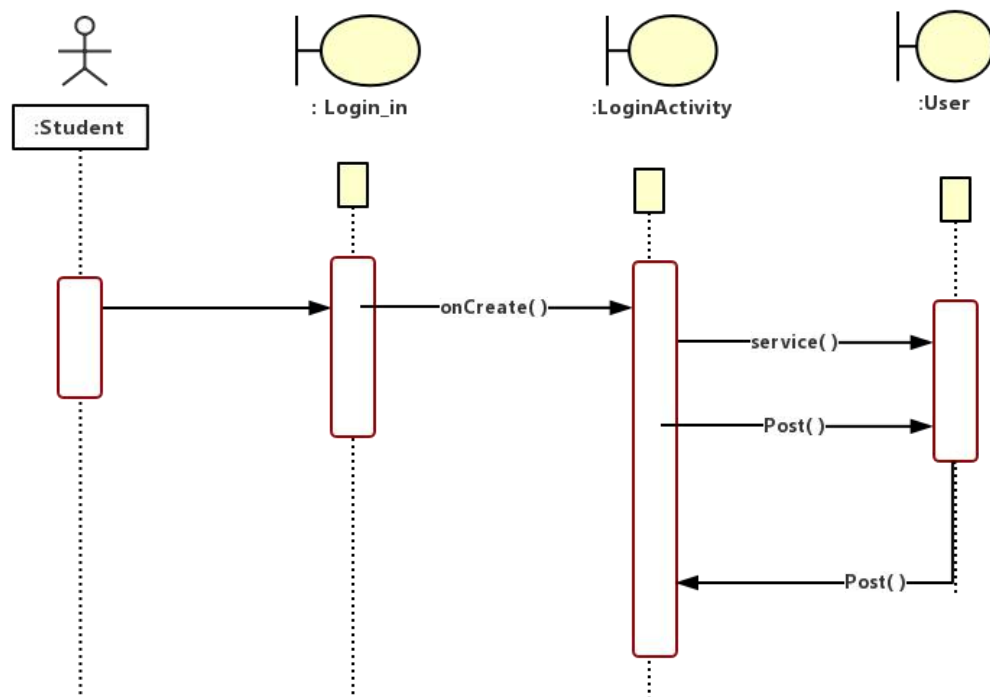
控制类：该用例中有一个控制类 LoginActivity，该类对用户填写的登录信息进行检查和提交

实体类：User，该类存储着用户所有的登录信息，包括用户名、密码、用户身份信息。

用例析取图如下：



时序图如下：



对时序图的简单描述：

用户打开该系统进入登录界面，在相应文本框填写相应信息，点击登录后系统调用 `getText()` 和 `isCheck()` 方法获取该用户的所有信息进行自动验证，调用

service()方法和后台进行数据交互，验证合法后调 startActivity()方法进入主界面。

### 3.1.2 审批申请用例析取

交互过程：

- 1、管理员已经登录系统后，点击查看申请列表
- 2、进入申请列表页面，该页面是表单形式，显示各个申请条目
- 3、管理员点击单个条目进入详细申请页面，查看申请详细信息，并进行审批，点击通过申请或拒绝申请

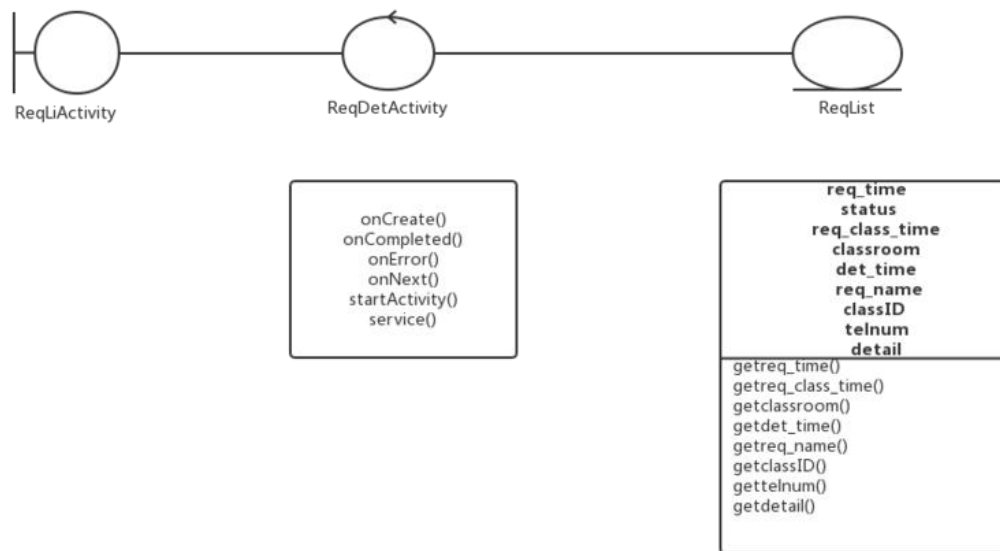
通过分析得到三种类：

边界类：ReqLiActivity，页面显示各个申请条目和简要信息

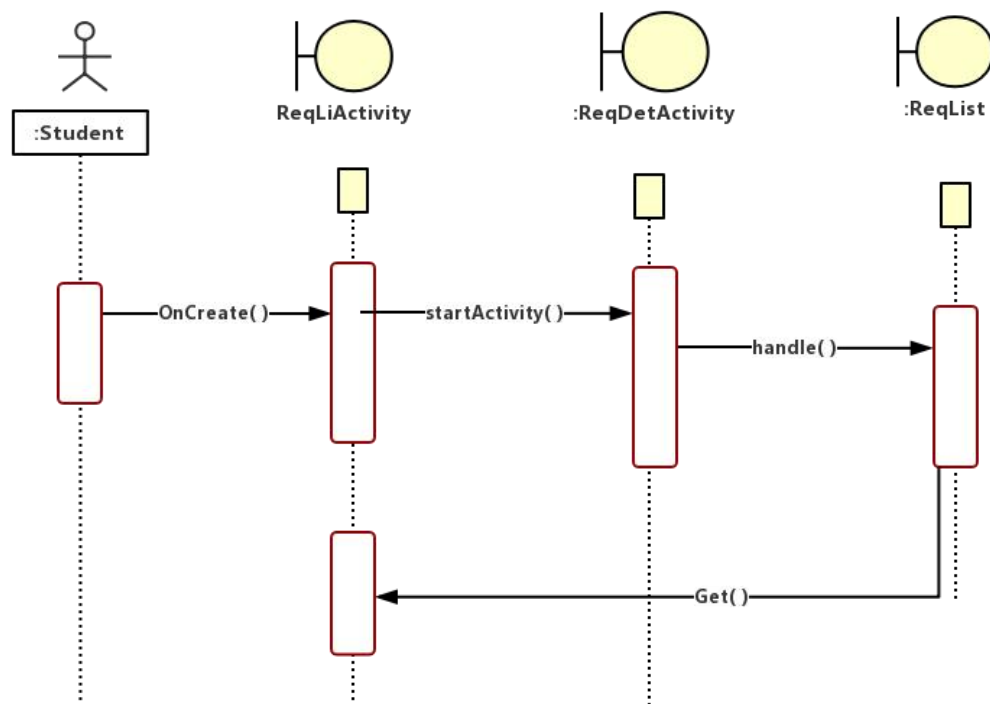
控制类：ReqDetActivity，查看和审批具体的申请条目

实体类：ReqList，保存着申请条目的具体信息，包括申请人，申请状态，申请时间，申请教室，预约的时间，申请理由等等。

用例析取图如下：



时序图如下：



时序图简单描述：



用户点击进入申请列表，调用 handler() 函数，从 ReqList 类获取申请的详细信息，并显示，点击具体条目，调用 OnClick() 函数，跳转进入详情页面，进行审批，审批成功，返回申请列表。

### 3.1.3 管理预约状态用例析取

交互过程：

- 1、用户登录系统，点击“查看预约”查看课室预约状态
- 2、进入预约状态页面后，用户已预约的教室条目会显示在页面内；若用户无预约，则页面为空

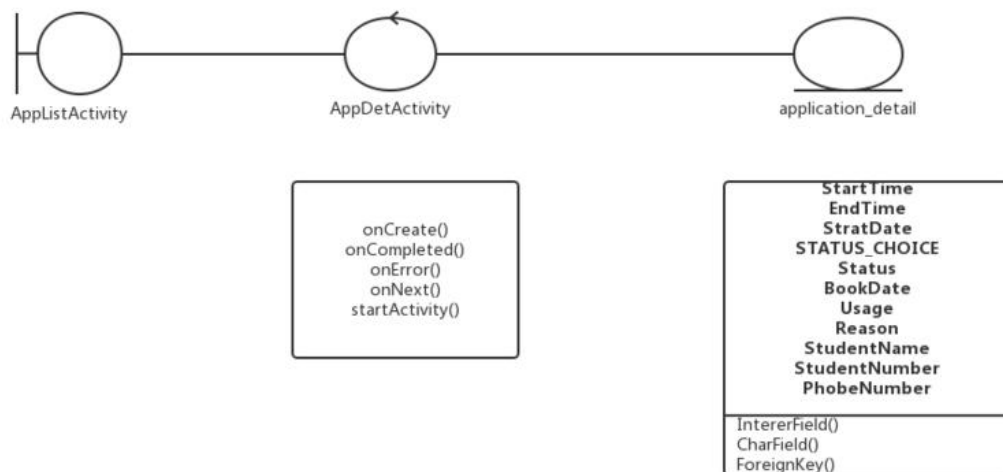
分析得到三种类：

边界类：AppListActivity 类，主要是展示学生预约课室的条目

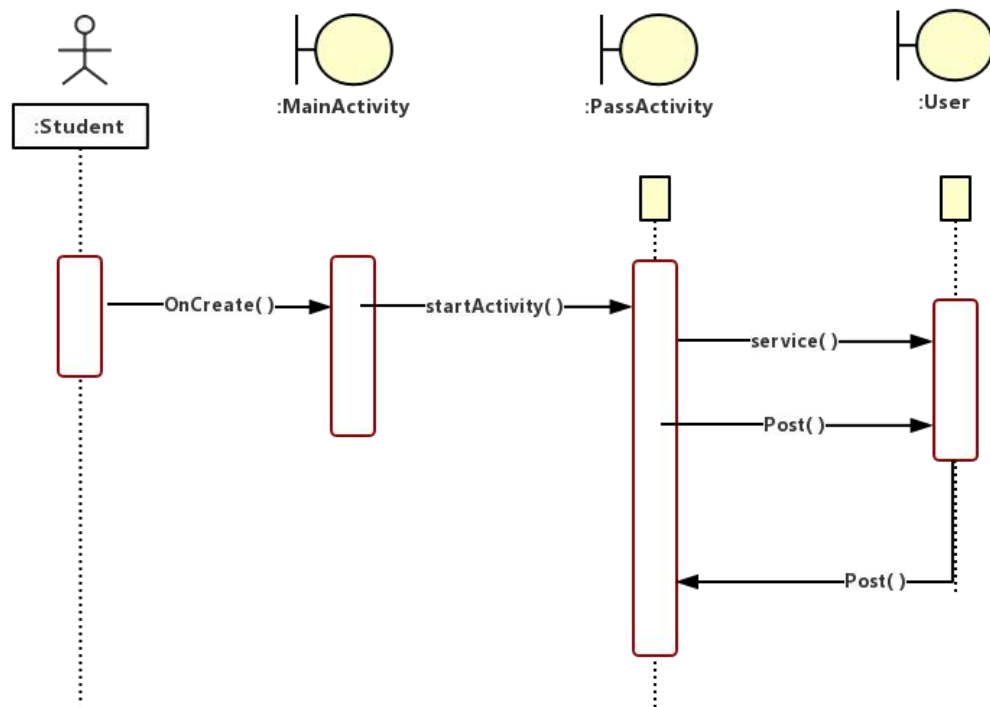
控制类：AppDetActivity 类，在 AppDetActivity 类中会展示预约课室的详情，并可以通过点击“取消”取消已有的课室预约并删除页面条目。

实体类：application\_detail 类，类中存储着学生预约信息

用例析取图如下：



时序图如下：



时序图简单描述：学生进入 AppListActivity 中后，点击课室条目，系统通过 AppDetActivity 类上的 putCharSequence()和 putInt()两种方法获取页面详细信息，学生点击“取消”按钮，系统调用

### 3.1.4 密码修改用例析取

交互过程：

- 1、用户登录系统，点击“修改密码”
- 2、输入原密码和新密码，点击确定按钮进行修改

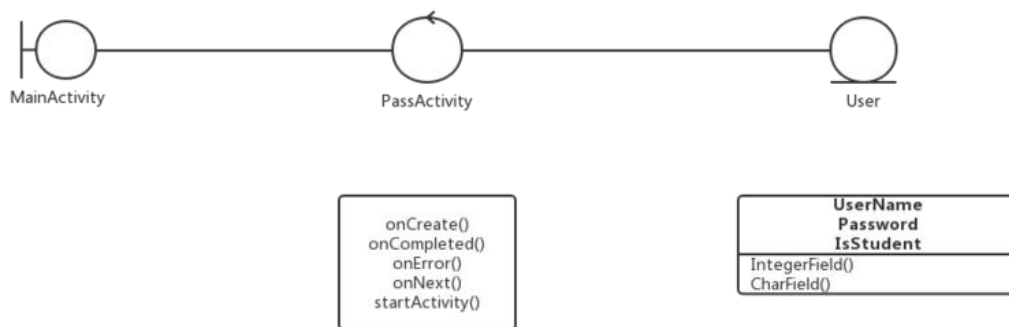
分析得到三种类：

边界类：MainAcitivity 类，学生登录后展示的主界面，用于各功能的交互

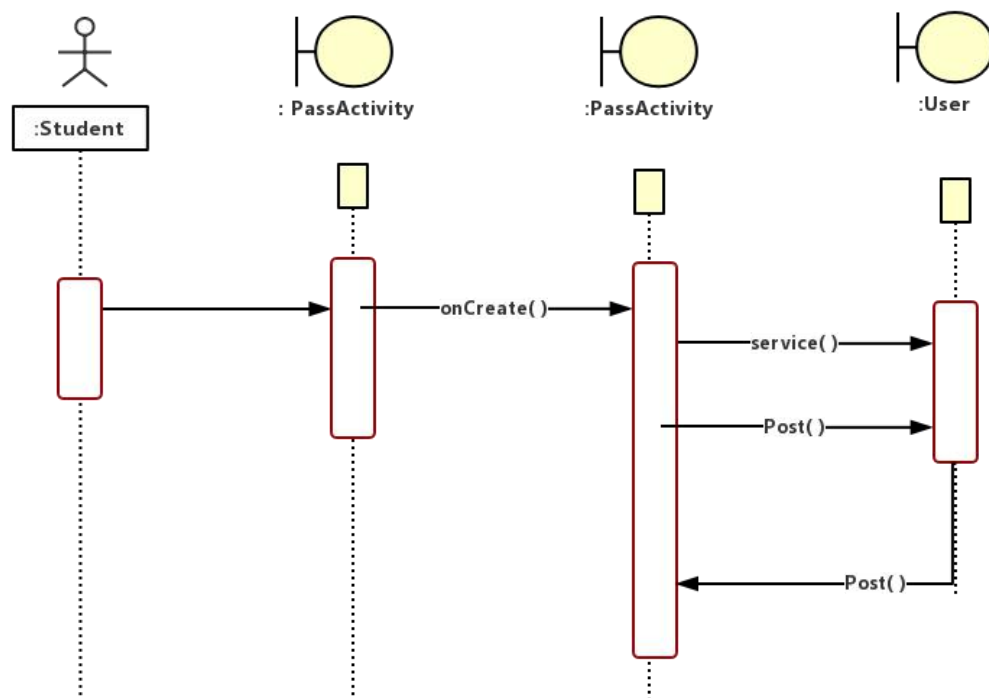
控制类：PassActivity 类，学生点击“修改密码”按钮后，对原有密码进行修改，并将新密码传至后台数据库

实体类：后台数据库中的 User 类，存储学生的学号和密码信息，修改后新密码将覆盖原密码

用例析取图如下：



时序图如下：



时序图简单描述：学生点击页面中的“修改密码”按钮，系统则会调用控制类 PassActivity，调用其中的 service 方法从服务器中获取学生用户的密码，服务器端采用 Post 方法返回学生的原密码

### 3.1.5 预约课室用例析取

交互过程：

- 1、用例开始于学生登录系统，在主页面点击“查看教室列表”按钮，进入课室列表，点击具体课室条目后进入预约界面。
- 2、学生在申请栏中填写申请人学号、姓名；时间栏里选择预约时间，分别选择开始时间和结束时间
- 3、学生填写预约理由
- 4、学生填写负责人联系方式
- 5、点击确认预约，若无冲突则可显示“预约成功”；若有冲突，则显示该申请与其他预约申请冲突。

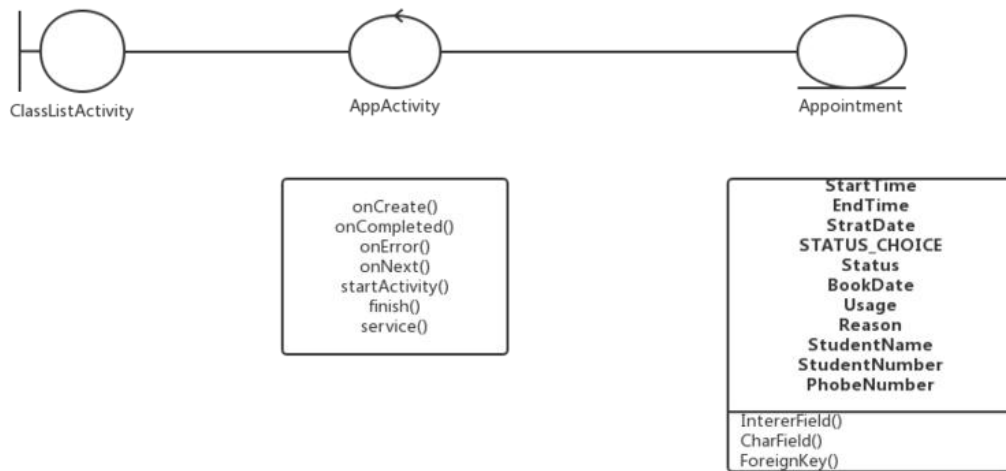
分析得到三种类：

边界类：ClassListActivity 类，界面展示了可预约的课室列表，用于进一步的交互

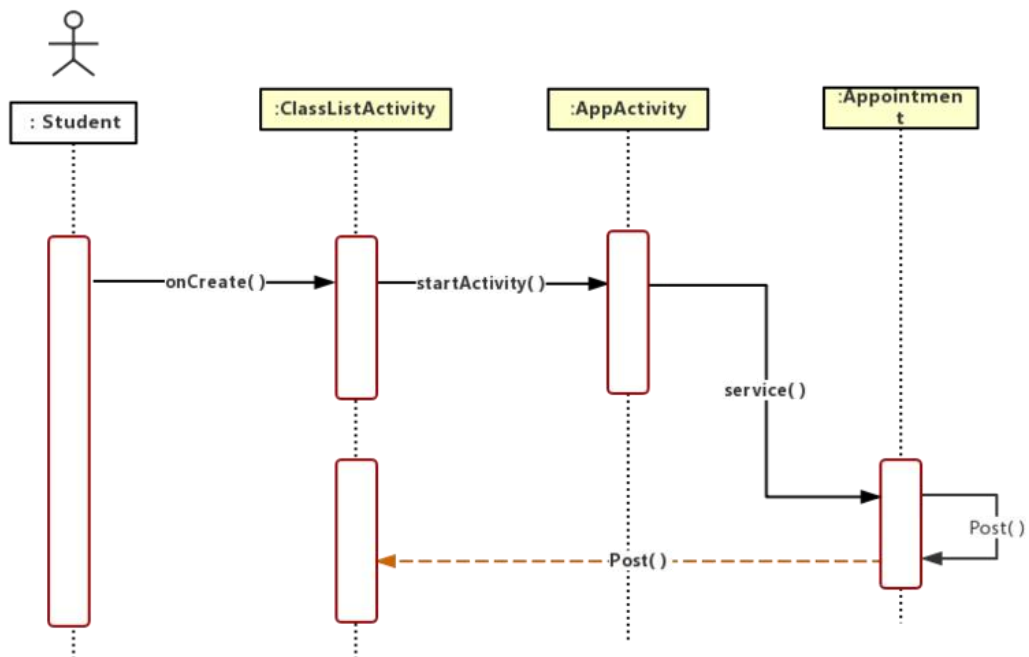
控制类：AppActivity 类，学生点击“确认”按钮对选中课室进行预约，点击“取消”解除当前预约

实体类：后台数据库 Appointment 类，存储详细的课室信息及预约情况

用例析取图如下：



时序图如下：



时序图简单描述：学生进入系统后通过 ClassListActivity 类进行交互，学生点击课室条目后，系统会调用其中的 service 方法从服务器中获取课室信息，服务端通过 Get 方法得到传输的课室信息，服务器调 HttpResponseMessage ( ) 函数返回状态码

### 3.2 分析机制

下表是中山大学课室申请系统的类分析机制表：

Table 课室申请系统的类分析机制

分析类	分析机制
LoginActivity	持久性，安全性
User	持久性，安全性
ReqLiActivity	持久性，安全性
ReqDetActivity	持久性，安全性
ReqList	持久性，安全性
AppListActivity	持久性，安全性
AppDetActivity	持久性，安全性
application_detail	持久性，安全性
MainAcitivity	持久性，安全性
PassActivity	持久性，安全性
ClassListActivity	持久性，安全性
AppActivity	持久性，安全性
Appointment	安全性

### 3.3 补充用例规约

无

## 4 部件设计

### 4.1 分析并发需求

- 1) 申请人数超过现有空闲课室数目

### 4.2 针对需求的设计方案

#### 4.2.1 针对申请人数超过现有空间课室数目

系统运行时，可能申请同一时间段课室的人数超过该时间段的所有空闲课室的数目，即同时间段同一课室被多人申请，此时可能会发生资源冲突问题。针对此问题，我们设计了管理员审批机制，对于所有的课室申请，都需要管理员在后台管理界面进行审批，因此在管理员进行审批之前，所有空闲课室均可被重复申请。

### 4.3 映射到现实系统

本系统主要由 Java 语言实现前端，由 Python 实现后台数据库，并在 Android 系统上进行运行，支持安卓 4.0 及以上版本。查看预约状态、管理个人和课室信息、预约课室、管理申请列表、审批申请等功能需要在联网状态下完成。