

系统分析与设计实践 课程设计

专 业 信息管理与信息系统

班 级 信管1811

指导老师 刘益玲、易燕

项目名称： Perfect Campus

组长：201821136022严旭 成 绩

组员：201821135009赵鑫 成 绩

2021年 1 月 22 日

**《系统分析与设计实践》课程设计自查表**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **小组成员** | **需求功能** | **分析** | | | **设计** | | | | **实现** | **自评等级（错开等级）** |
| **系统顺序图** | **顺序图** | **领域类图** | **架构图** | **顺序图、类图** | **数据库设计** | **其他** |  |  | |
| **严旭** | **申请，审批** | √ | √ | √ | √ | √ | √ |  | √ | **良** | |
| **赵鑫** | **登录，打卡，打卡信息** | √ | √ | √ | √ | √ | √ |  | √ | **良** | |

**目录**

[第一章 需求建模 7](#_Toc29069049)

[第二章 分析建模 7](#_Toc29069050)

[第三章 设计建模 7](#_Toc29069051)

[第四章 实现 8](#_Toc29069052)

[参考文献 8](#_Toc29069054)

[附录 8](#_Toc29069053)

# 需求建模

## 1.1项目功能介绍

系统名为PerfectCampus，满足疫情期间教师以及学生对于健康检测即每日打卡，学生对于外出申请的需求。

## 1.2用例图

整体用例图

## 1.3用例规约

**用例名称 ：外出申请**

用例编号：SA01

范围：Perfect Campus

级别：用户目标级别

参与者：学生

前置条件：学生登陆系统

后置条件：系统中增加一个申请信息

基本流：

1. 学生系统发起了外出申请请求
2. 系统提示用户输入出行方式，申请时间，审批人，申请理由。
3. 用户输入出行方式，申请时间，审批人，申请理由。
4. 系统验证用户输入的出行方式，申请时间，审批人，申请理由正确范围适当。
5. 系统保存用户的出校申请。
6. 系统提示申请成功等待审批。

备选流：

4a：系统验证学生信息出行方式，申请时间，审批人，申请理由是否填写正确，是否完成所有必填项，是否格式正确。

1. 提示学生信息出行方式，申请时间，审批人，申请理由，格式错误或未填写完整，提示学生重新填写
2. 回到基本流2步操作

**用例名称 ：导员审批**

用例编号：SA02

范围：学生疫情事务管理系统

级别：用户目标级别

参与者：Teacher

前置条件：导员登陆系统

后置条件：转变系统申请表中申请状态

基本流：

1. 导员发起了审批请求
2. 系统提示导员审批信息。
3. 导员选择通过审批，或不通过。
4. 系统保存申请状态。
5. 系统提示审批成功。

**用例名称: 每日打卡**用例编号:  SA04  
参与者：学生用户  
前置条件：系统验证用户身份成功  
后置条件：系统增加学生打卡信息  
  
基本流:  
1. 学生用户向系统发出打卡的请求  
2. 系统提示学生用户输入打卡信息  
3. 学生用户输入打卡信息  
4. 系统验证用户的打卡信息完全  
5. 系统提示用户打卡成功

备选流  
4a 系统验证用户打卡信息不完全  
  1 提示用户打卡信息不完全，提示用户重新输入  
  2 回到基本流第二步

**用例名称: 每日打卡**

用例编号: SA04

参与者：学生用户

前置条件：系统验证用户身份成功

后置条件：系统增加学生打卡信息

基本流:

1. 学生用户向系统发出打卡的请求
2. 系统提示学生用户输入打卡信息
3. 学生用户输入打卡信息
4. 系统验证用户的打卡信息完全，添加打卡记录
5. 系统提示用户打卡成功

备选流

4a 系统验证用户打卡信息不完全

1 提示用户打卡信息不完全，提示用户重新输入

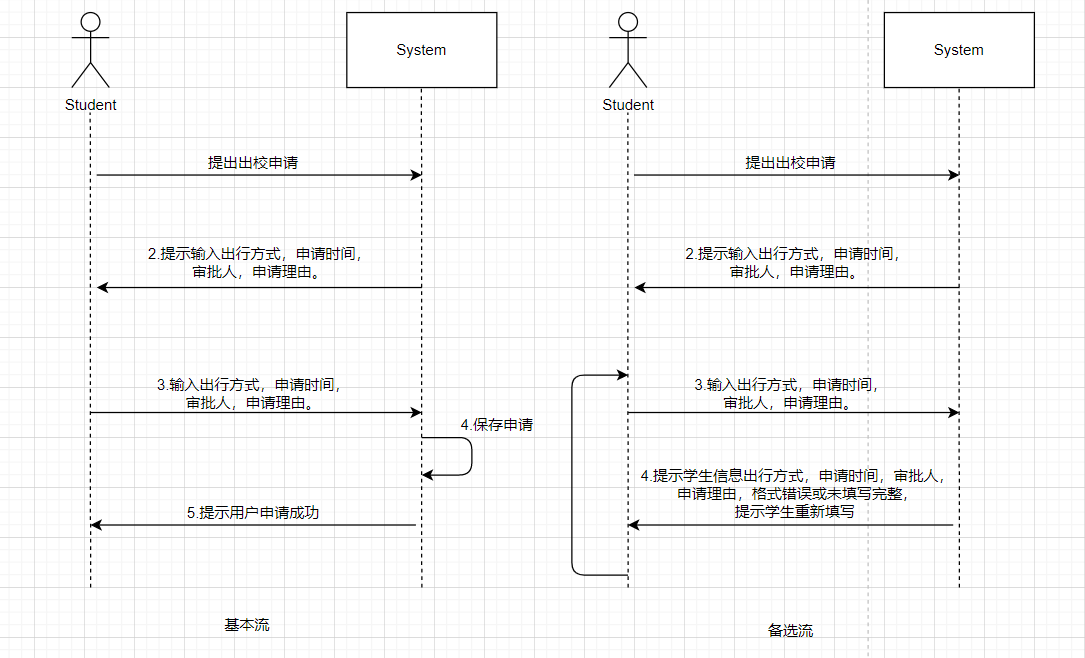
2 回到基本流第二步

# 分析建模

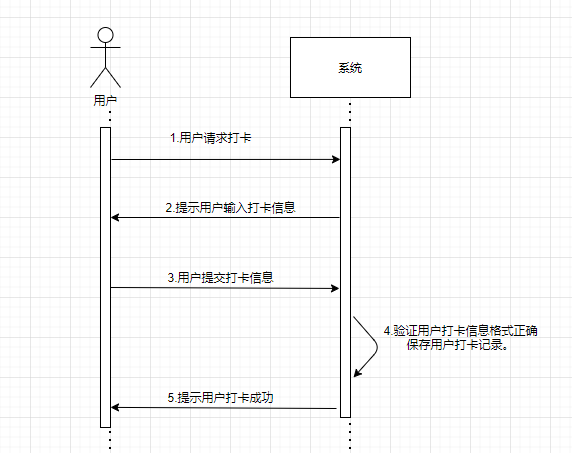
2.1 用例一的分析建模

2.1.1 系统顺序图

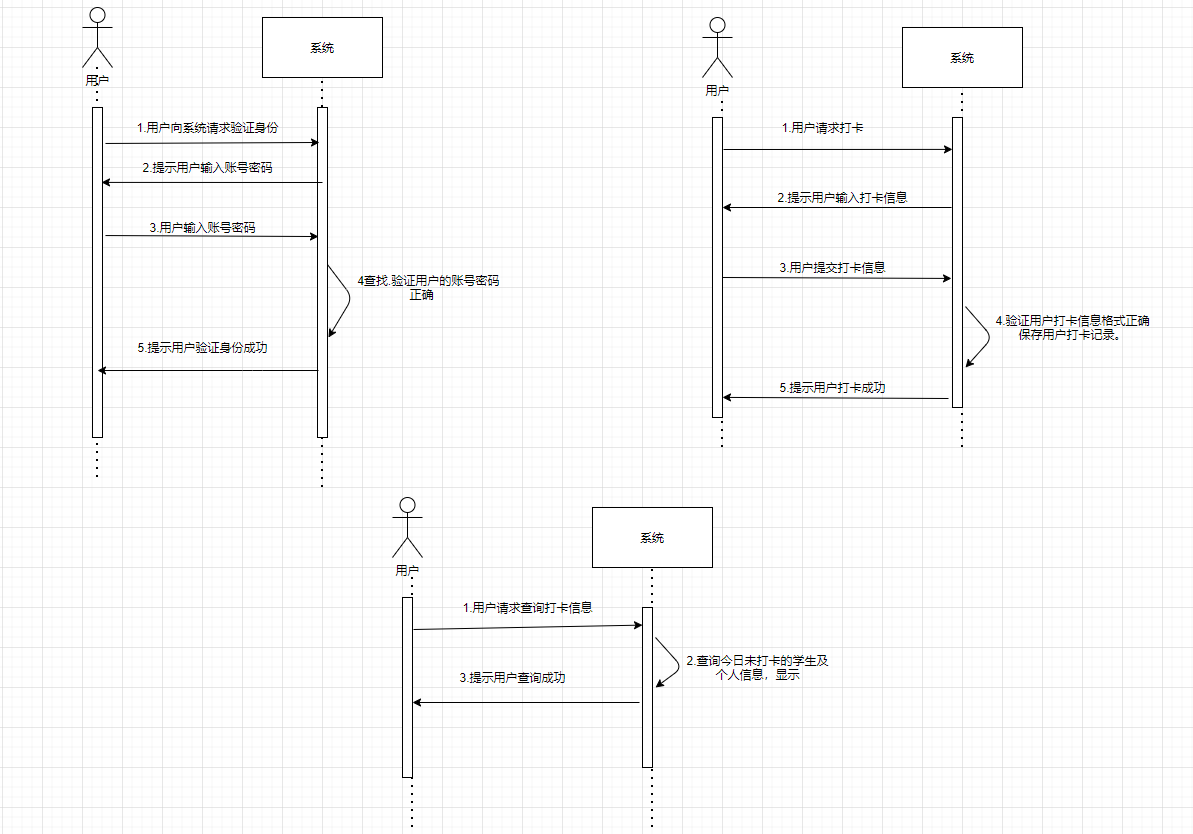
**外出申请**



**每日打卡**

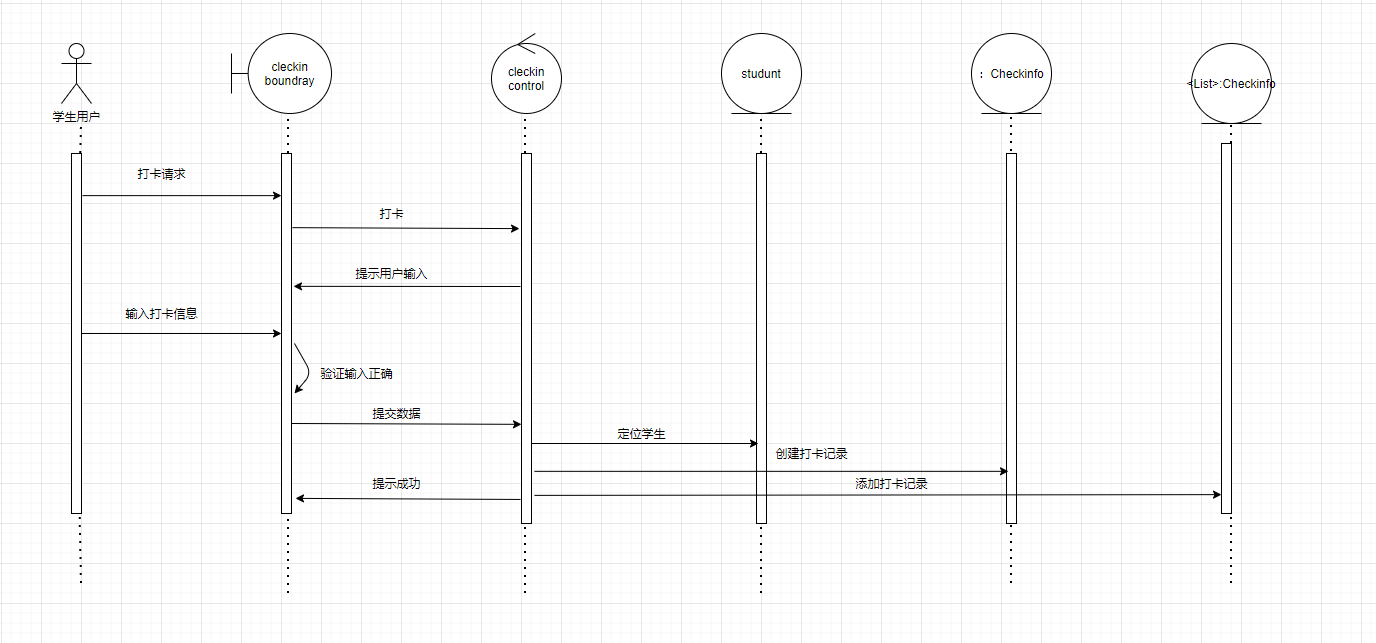


登录，打卡，查看打卡信息

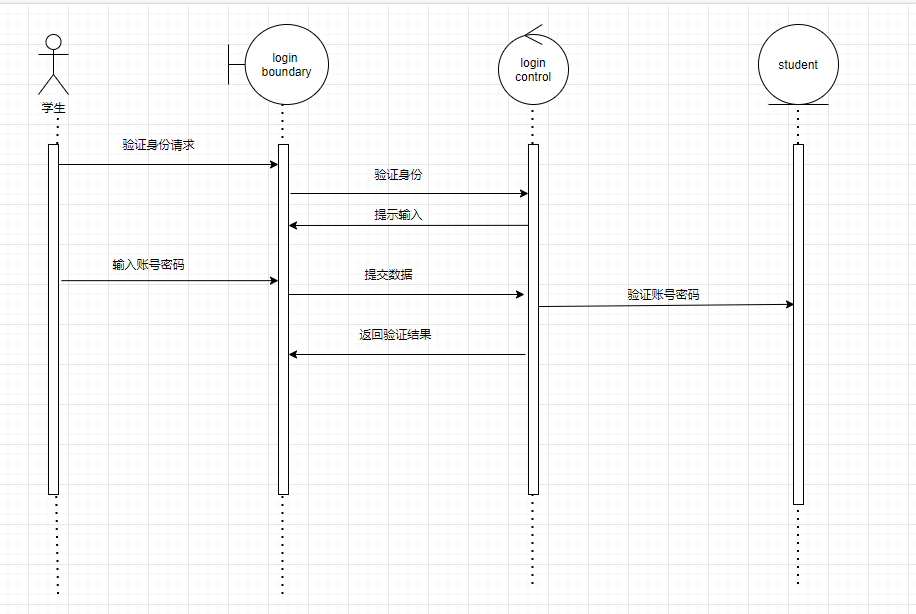


2.1.2 顺序图

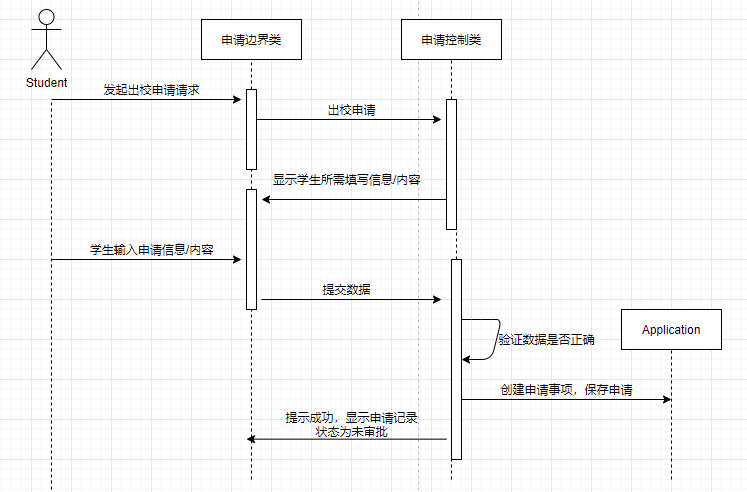
打卡：



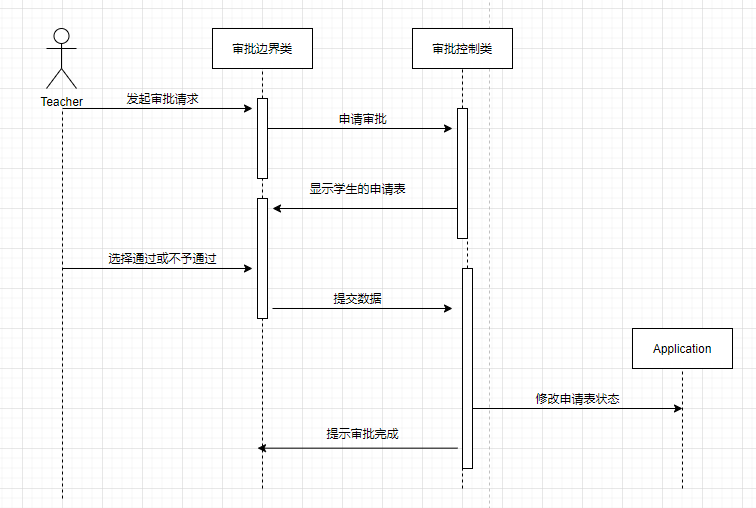
登录：



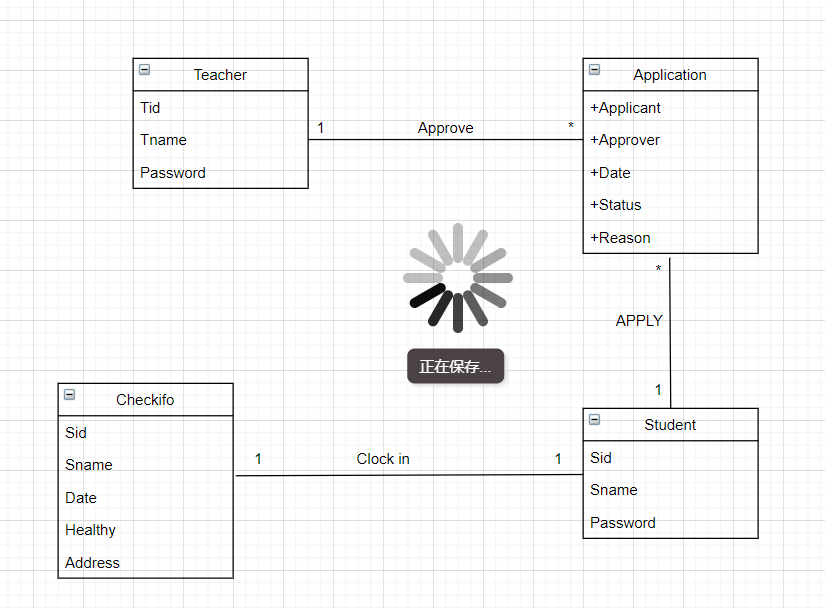
申请：



审批：



2.1.3 领域模型

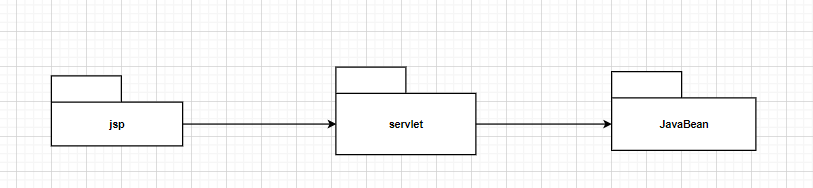


# 设计建模

## 3.1技术选择

### 使用idea ，DreamWaver平台， tomcat 技术， ajax 技术，和 jsp技术实现PerfectCampus。

## 3.2架构设计

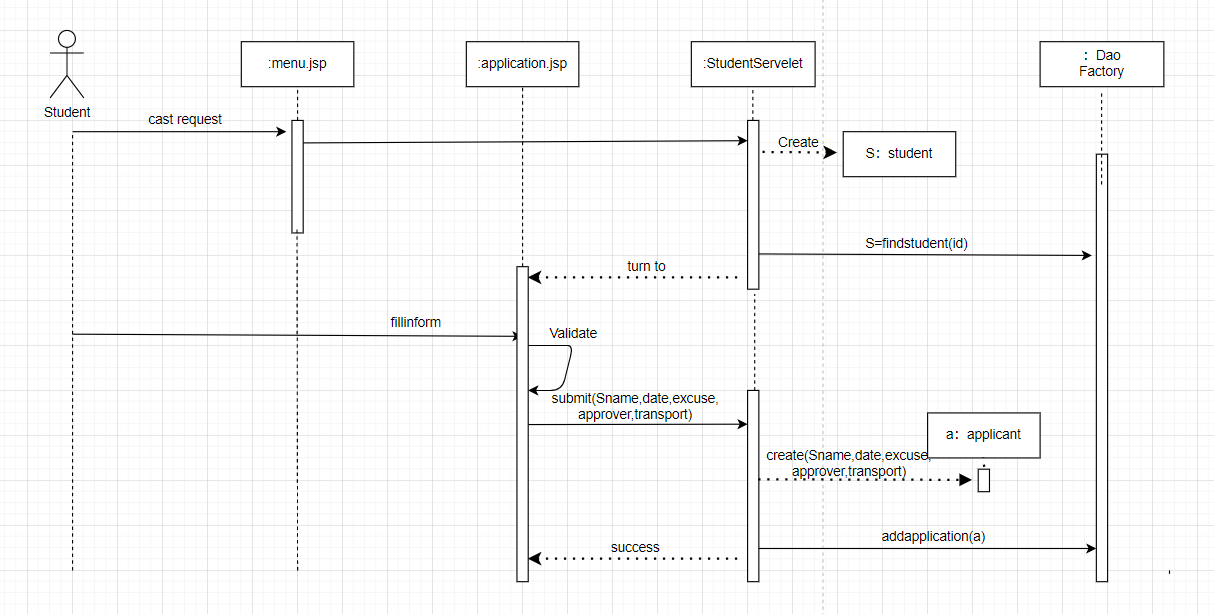


## 3.3类的设计

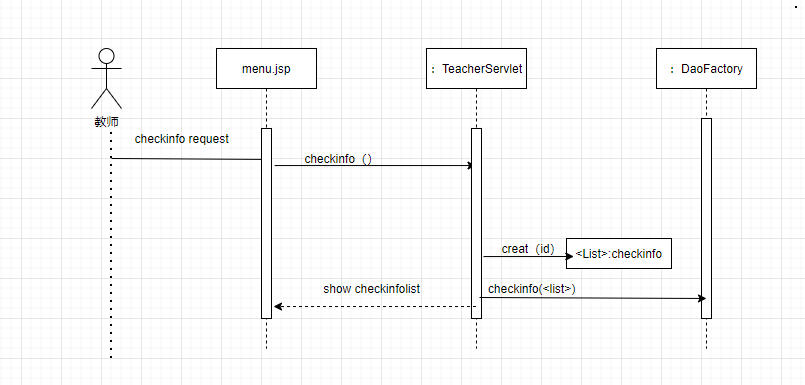
3.3.1 用例一的设计建模

3.3.1.1 顺序图

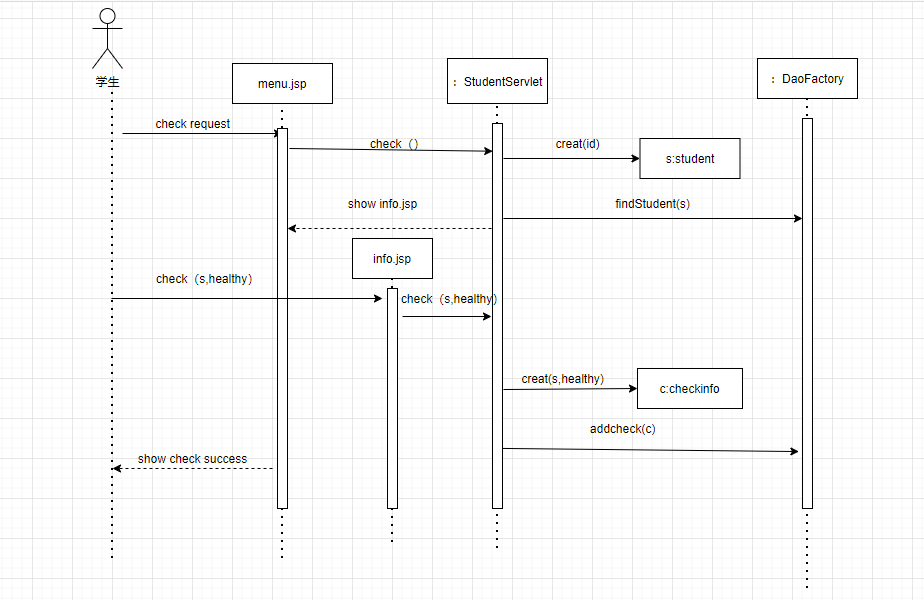
申请：



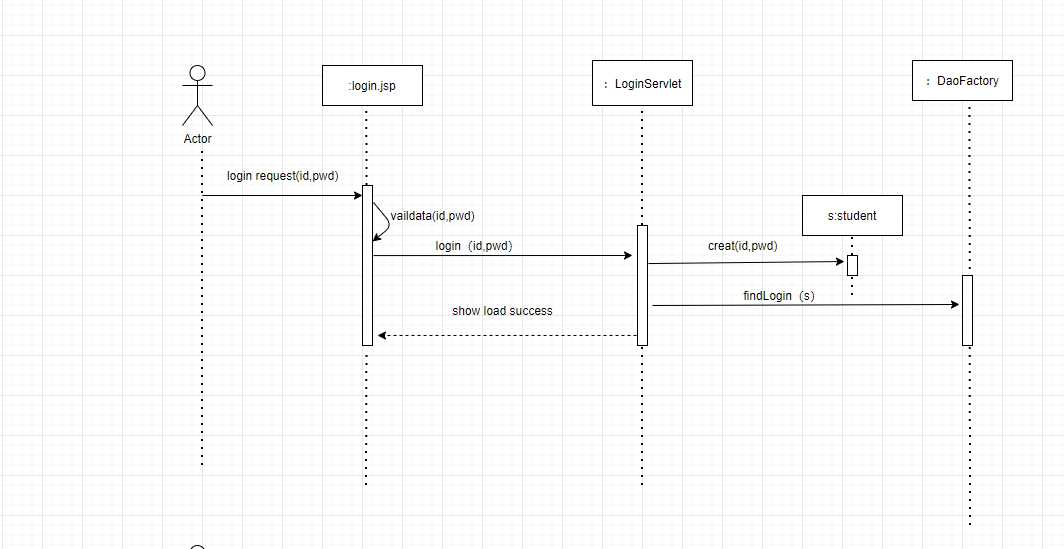
打卡信息查看：



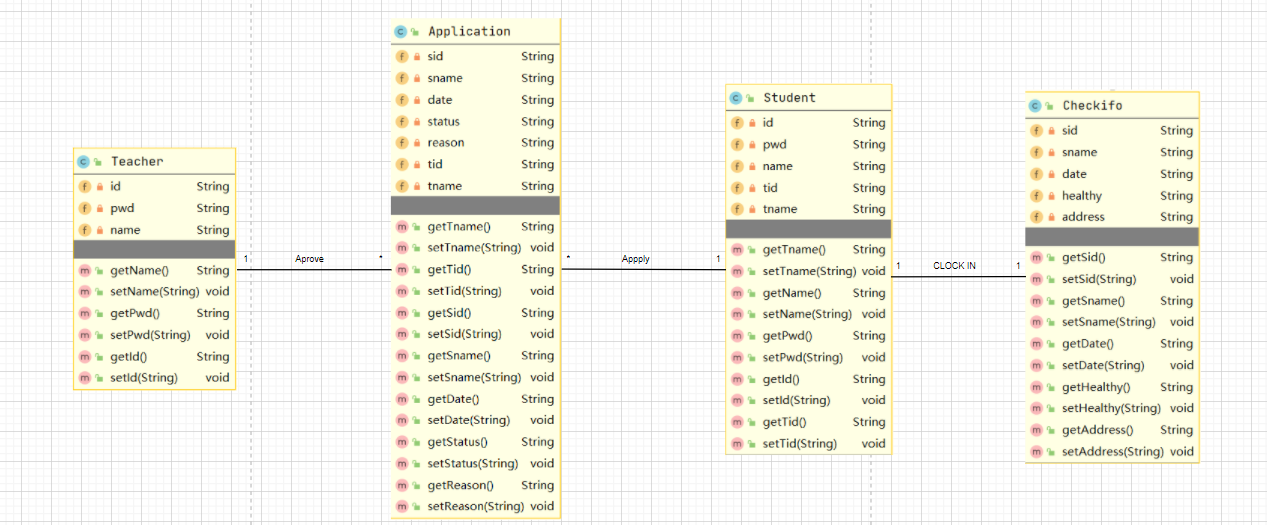
打卡：



登录**：**



3.3.1.2类图



## 3.4 数据库设计

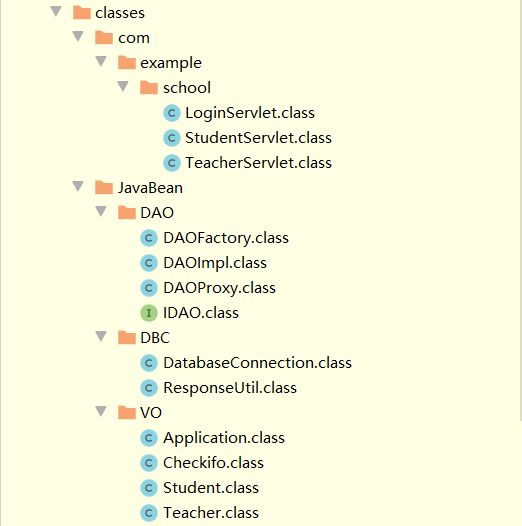
设计类图中需要持久化部分映射为关系模型

# 

# 第四章 实现

## 4.1 整体代码结构

对应架构的代码结构

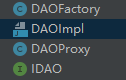


## 4.2 实现

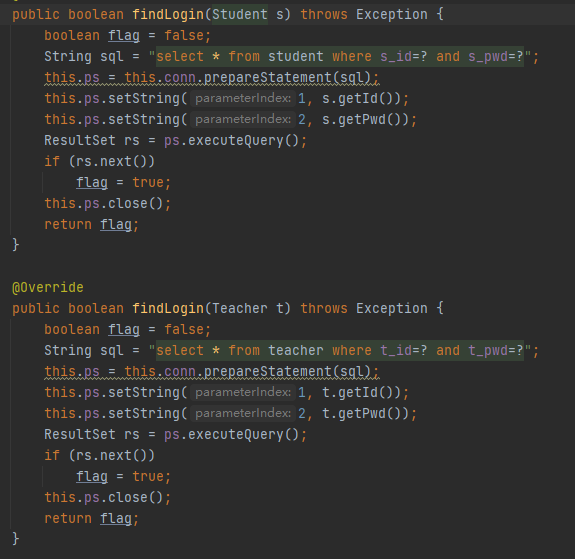
4.2.1 用例一的架构具体实现(以mvc为例)

4.2.1.1 m实现

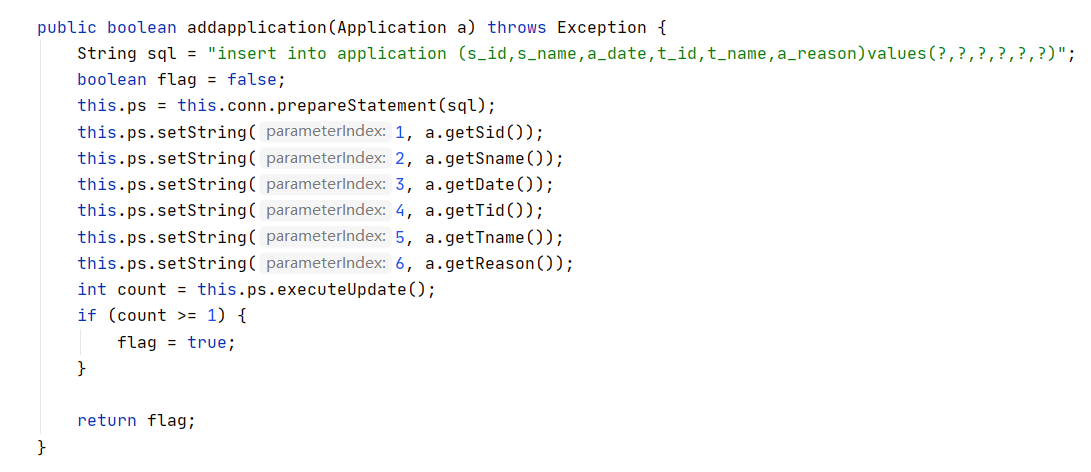
数据访问层采用工厂模式factory。



登陆：

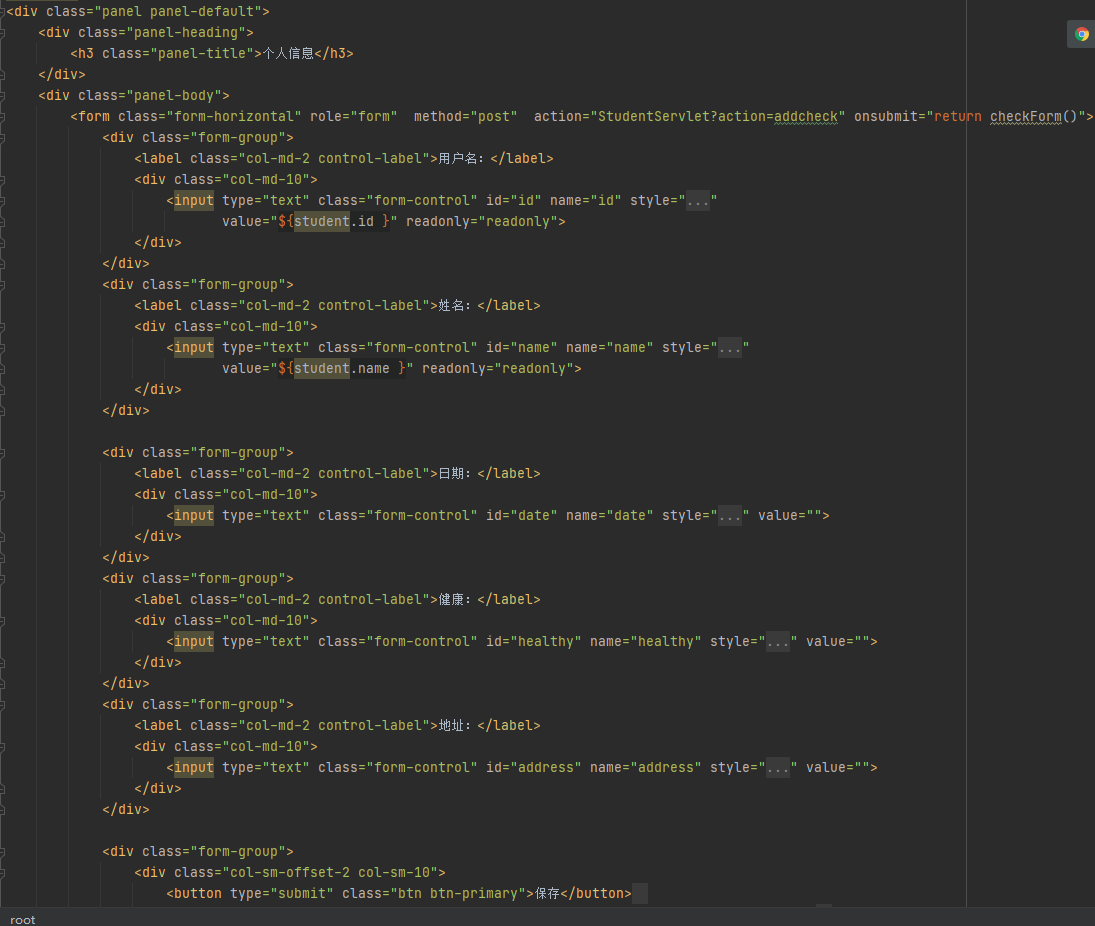


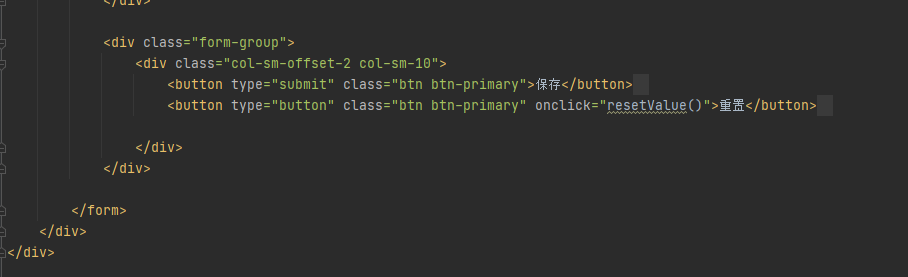
外出申请



4.2.1.2 v实现

打卡界面：





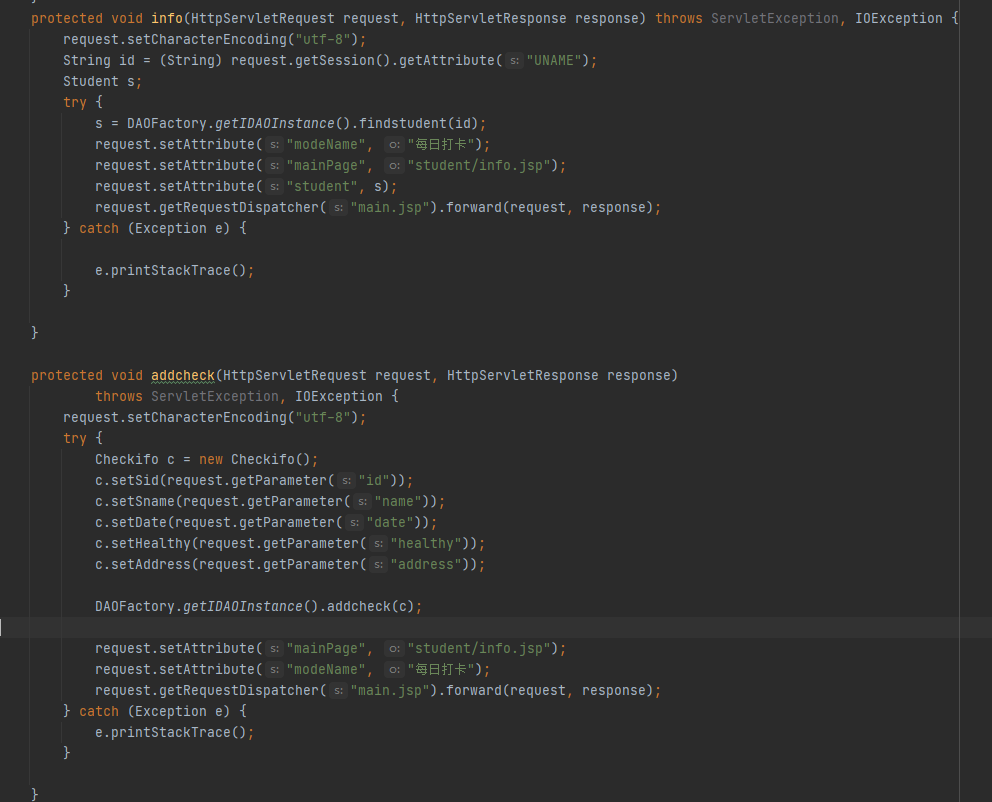
申请界面：





4.2.1.3 c实现

打卡：



外出申请：

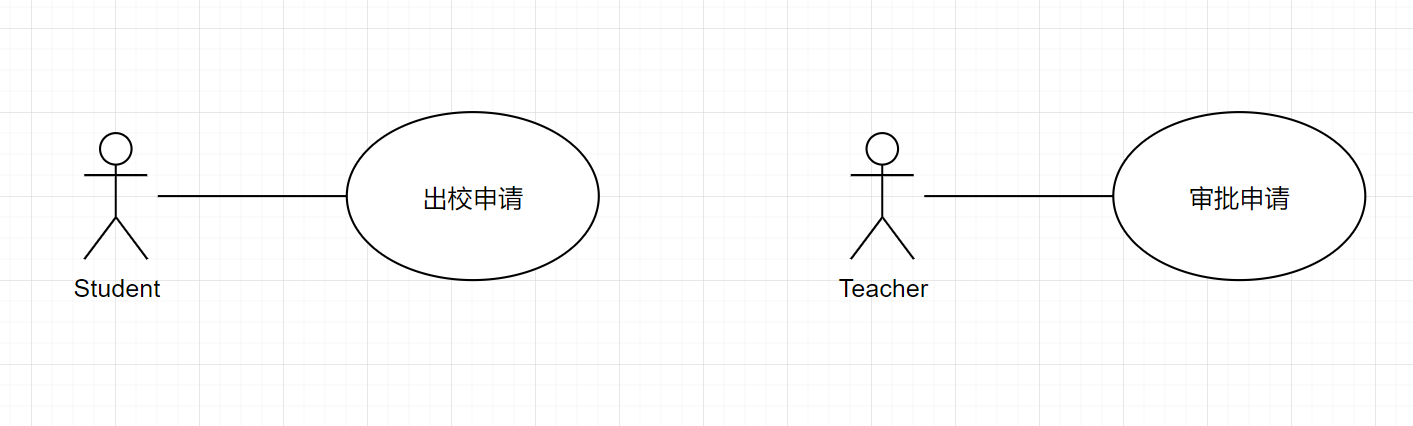


**附录**

个人课程设计报告文档

完成人： 严旭

# 需求建模



功能：出校申请

用例名称 ：外出申请

用例编号：SA01

范围：Perfect Campus

级别：用户目标级别

参与者：学生

前置条件：学生登陆系统

后置条件：系统中增加一个申请信息

基本流：

1. 学生系统发起了外出申请请求
2. 系统提示用户输入出行方式，申请时间，审批人，申请理由。
3. 用户输入出行方式，申请时间，审批人，申请理由。
4. 系统验证用户输入的出行方式，申请时间，审批人，申请理由正确范围适当。
5. 系统保存用户的出校申请。
6. 系统提示申请成功等待审批。

备选流：

4a：系统验证学生信息出行方式，申请时间，审批人，申请理由是否填写正确，是否完成所有必填项，是否格式正确。

1. 提示学生信息出行方式，申请时间，审批人，申请理由，格式错误或未填写完整，提示学生重新填写
2. 回到基本流2步操作

功能：审批

用例名称 ：导员审批

用例编号：SA02

范围：Perfect Campus

级别：用户目标级别

参与者：Teacher

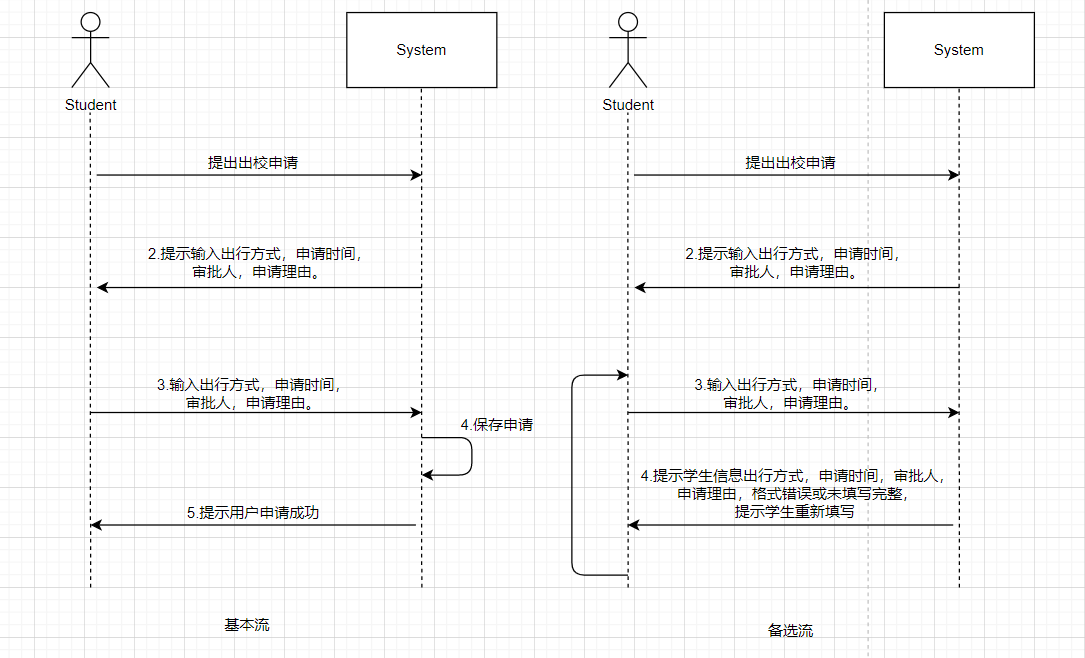
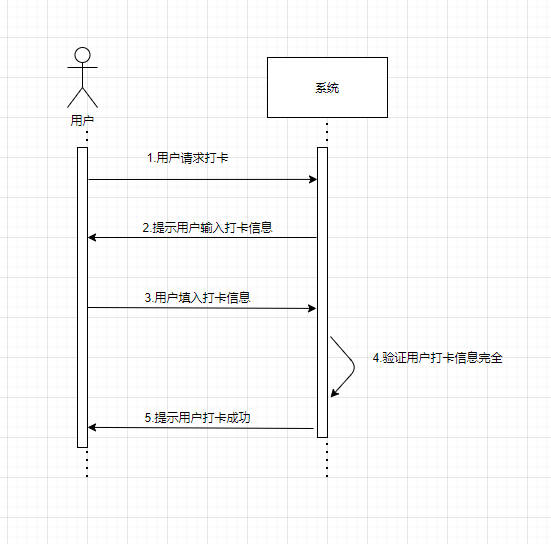
前置条件：导员登陆系统

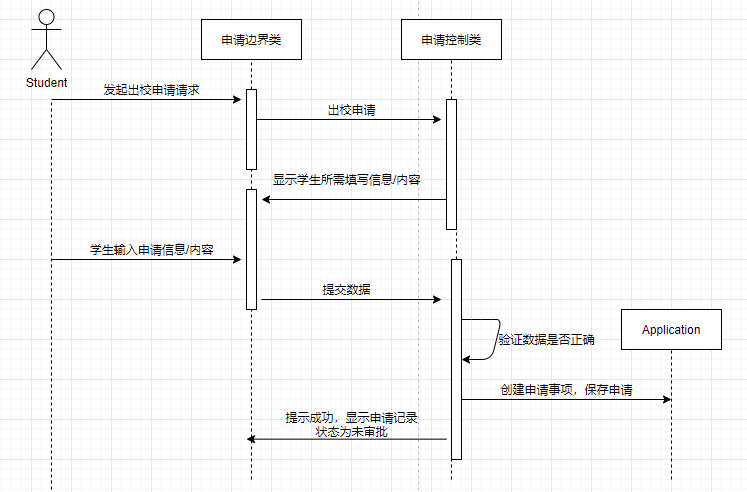
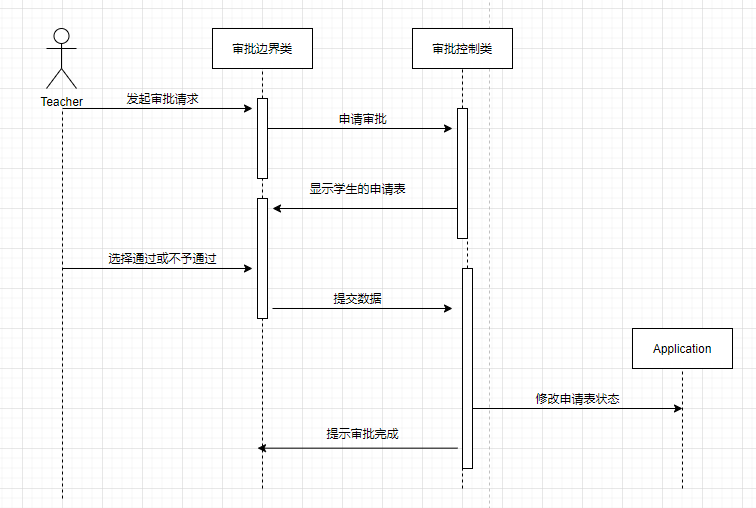
后置条件：转变系统申请表中申请状态

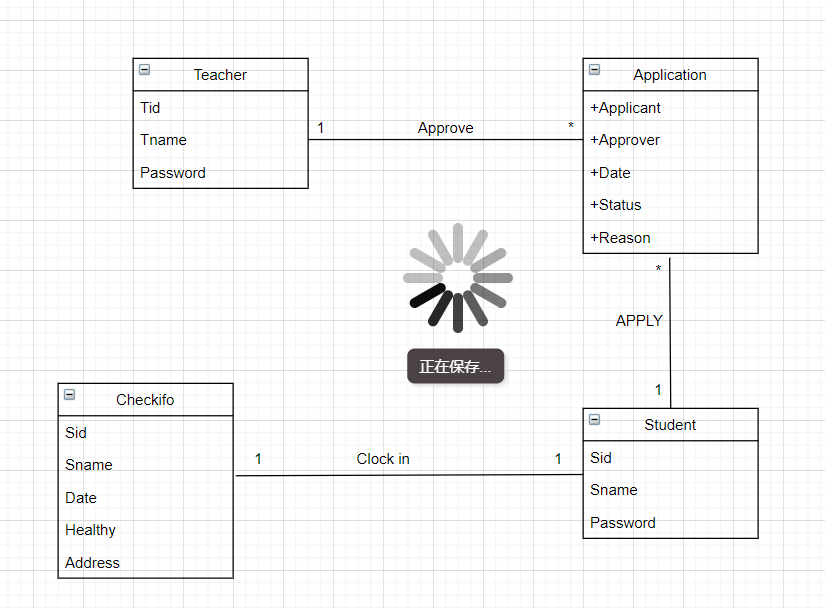
基本流：

1. 导员发起了审批请求
2. 系统提示导员审批信息。
3. 导员选择通过审批，或不通过。
4. 系统保存申请状态。
5. 系统提示审批成功。

# 第二章 分析建模

系统顺序图

顺序图

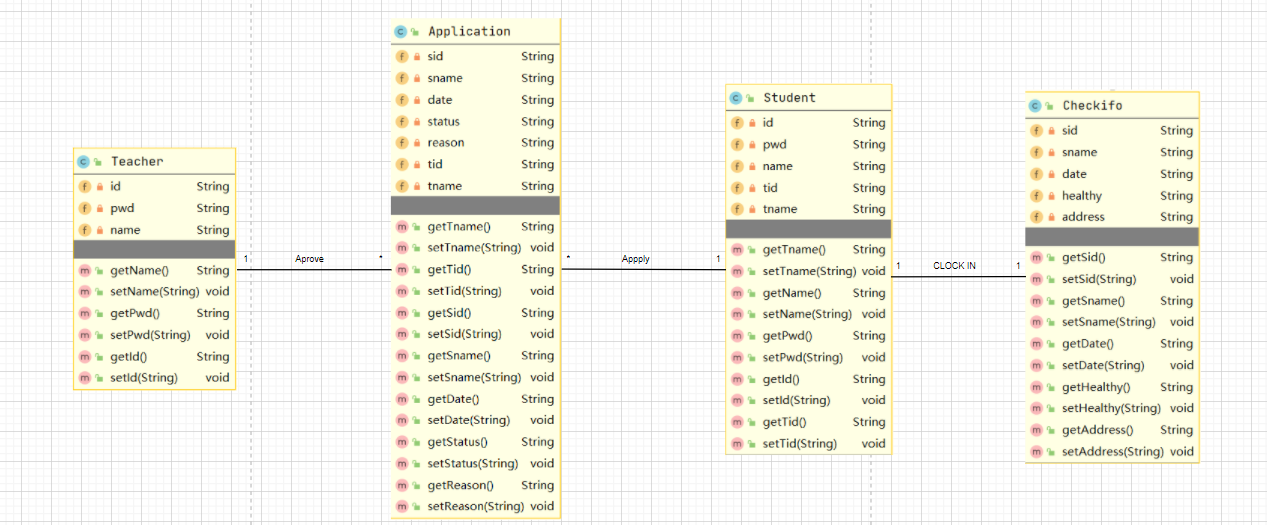
领域模型

# 第三章 设计建模

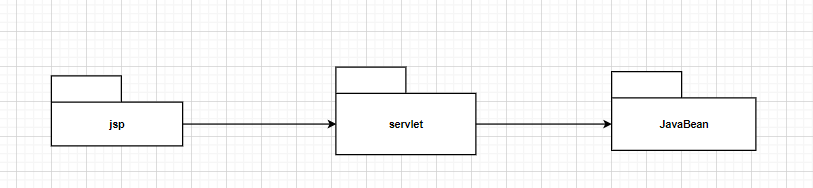
分配到的需求在小组架构设计的基础上，结合grasp原则设计的顺序图、类图，活动图等第

需在顺序图中标明运用到的grasp原则

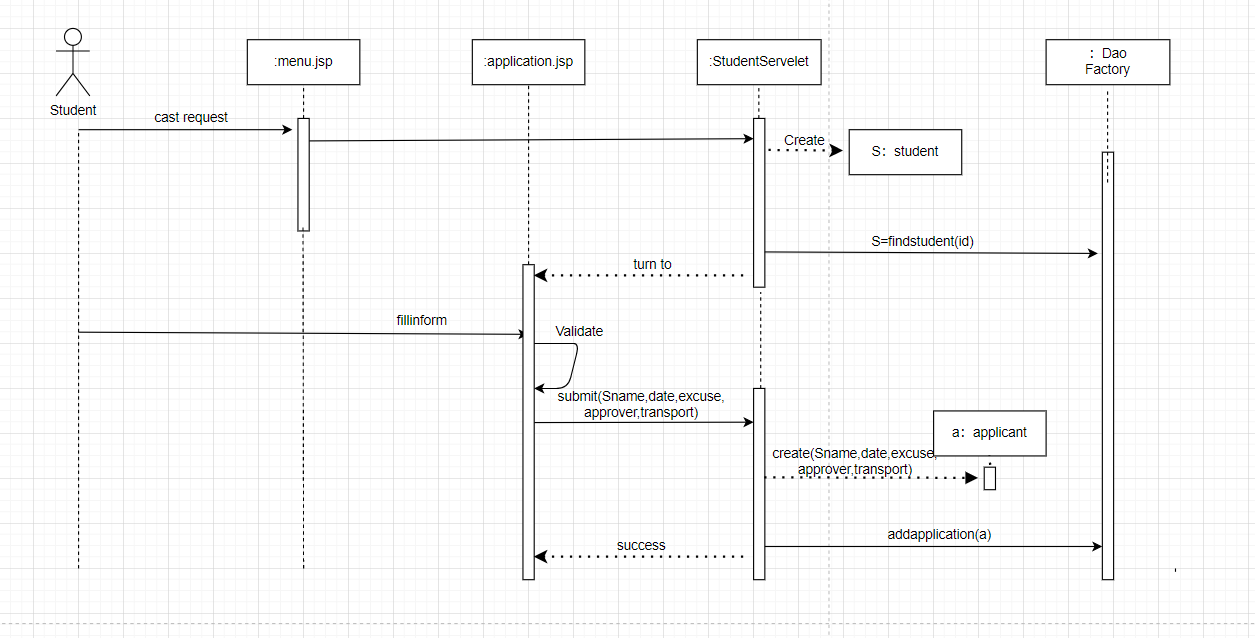
### 设计类图

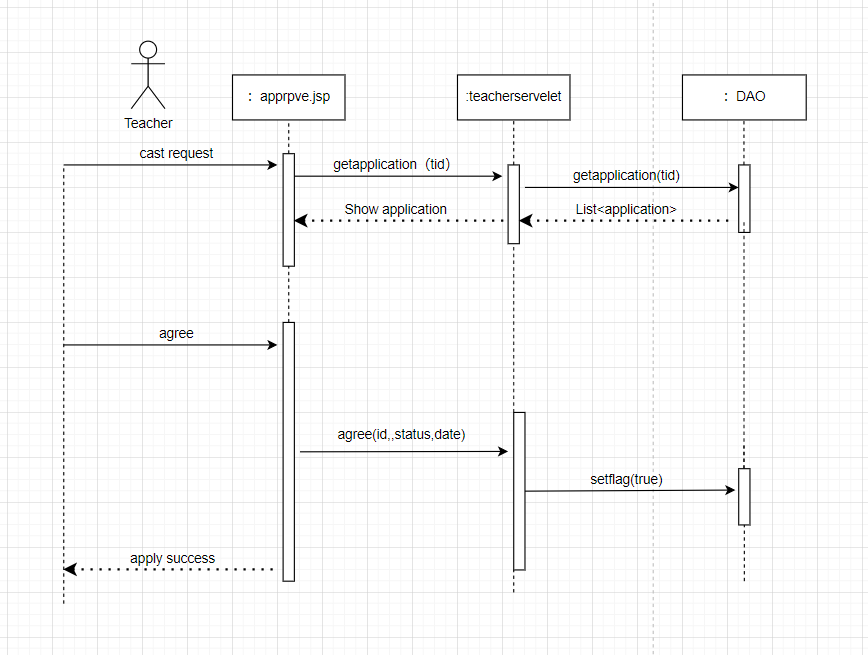


### 包图



**顺序图（出校申请，审批用例）**



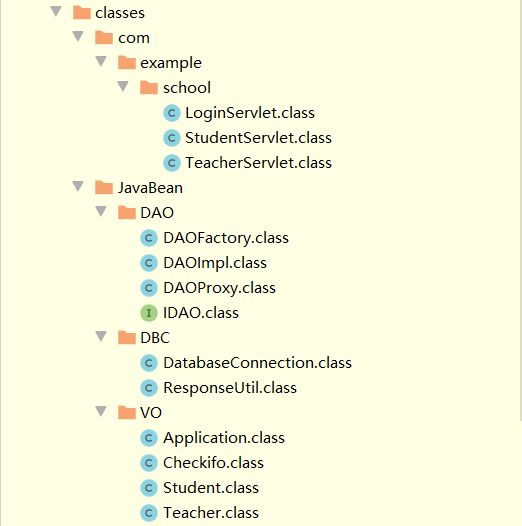


# 第四章 实现

对应设计模型的相应实现：分层分别介绍和代码对应

## 4.1整体代码结构

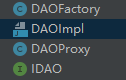
对应架构的代码结构



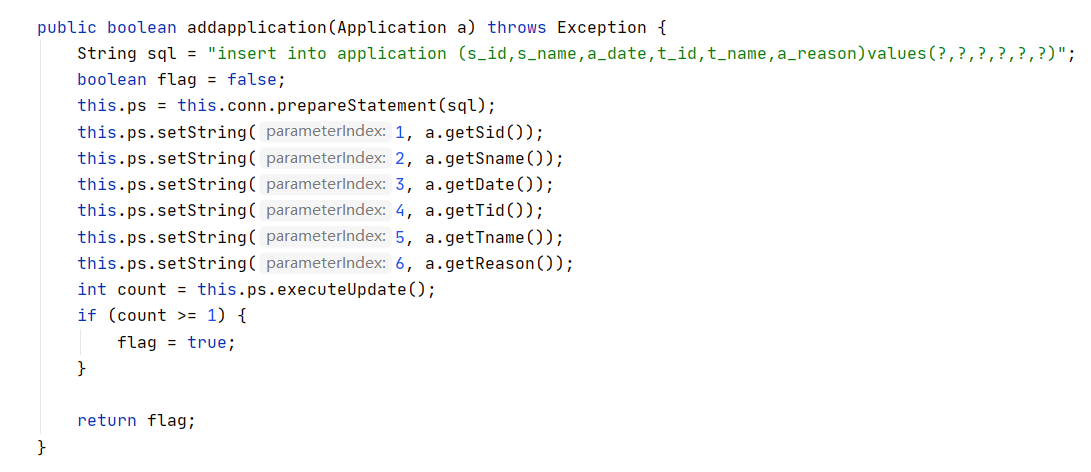
## 4.2 实现

m实现

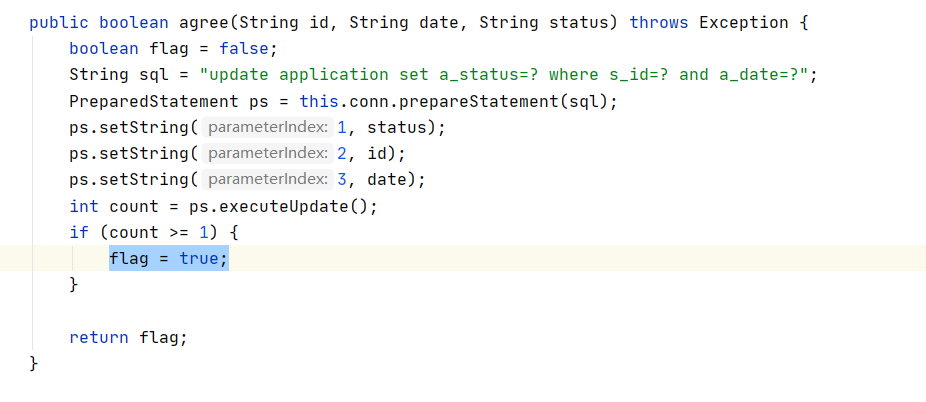
数据访问层采用工厂模式factory。



外出申请



申请审批：

v实现

申请界面：





审批界面：



c实现

申请：

审批：

# 个人总结

在本学期的信息系统分析与设计的课程学习后，我清楚地认识到设计系统的流程，明确了开发系统的基本步骤，学习了面向对象设计的迭代过程，从需求分析到设计再到最后的实现，过程清晰内容工作明确，不会再像从前那用无所适从。通过本学期该课程的学习，掌握了统一建模语言，使得原本复杂的软件设计，通过层层细化，明确了前期的需求，做好开发前的工作，只有这样才便于日后的维护与运行。

**附录**

个人课程设计报告文档

完成人： 赵鑫

# 需求建模

分配到三个用例，验证身份，每日打卡，打卡情况查看。

用例名称: 验证身份

用例编号: SA03

参与者：学生用户

前置条件：无

后置条件：系统赋予学生用户相关权限

基本流:

1. 学生用户向系统发出验证身份的请求
2. 系统提示学生用户输入其账号密码
3. 学生用户输入其账号密码
4. 系统验证用户的账号密码正确
5. 系统提示身份验证成功

备选流

4a 系统验证用户账号密码不正确

1 提示用户格式不正确，提示用户重新输入

2 回到基本流第二步

用例名称: 每日打卡

用例编号: SA04

参与者：学生用户

前置条件：系统验证用户身份成功

后置条件：系统增加学生打卡信息

基本流:

1. 学生用户向系统发出打卡的请求
2. 系统提示学生用户输入打卡信息
3. 学生用户输入打卡信息
4. 系统验证用户的打卡信息完全，添加打卡记录
5. 系统提示用户打卡成功

备选流

4a 系统验证用户打卡信息不完全

1 提示用户打卡信息不完全，提示用户重新输入

2 回到基本流第二步

用例名称: 打卡情况

用例编号: UC103

参与者：教师用户

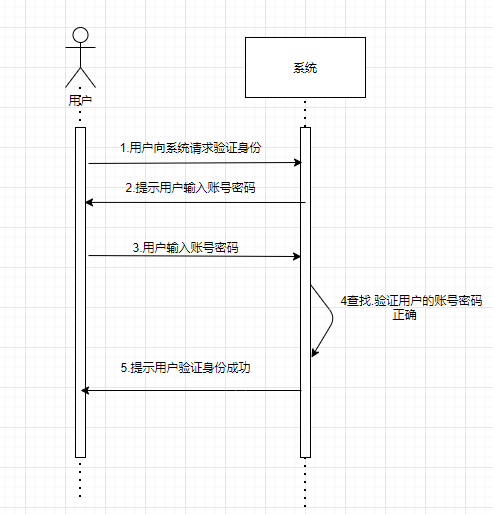
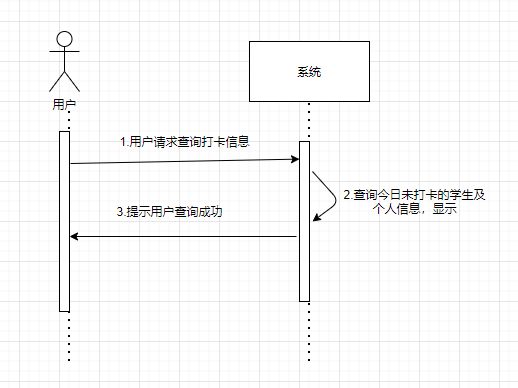
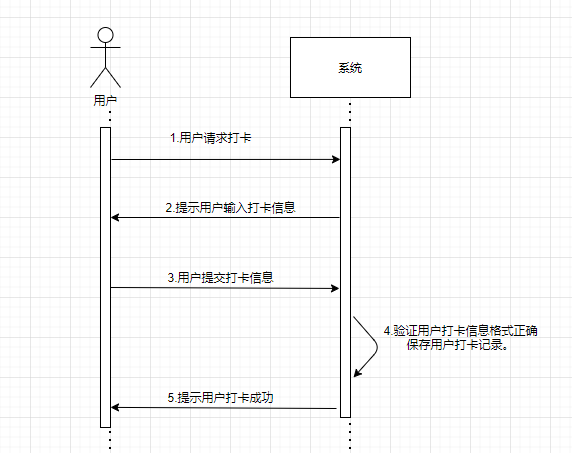
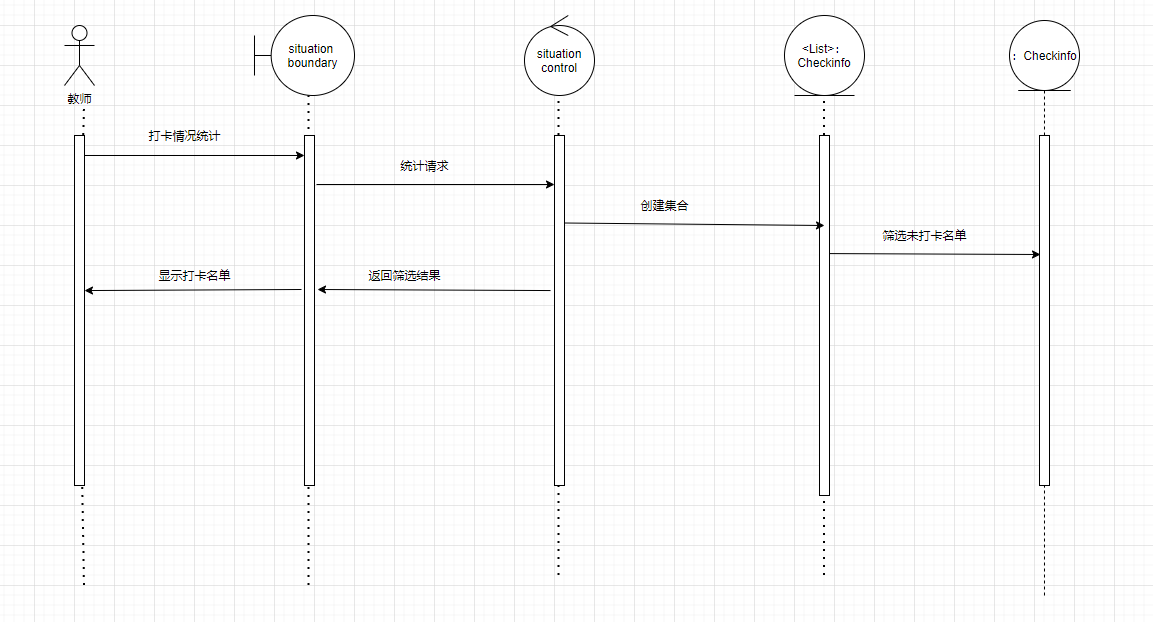
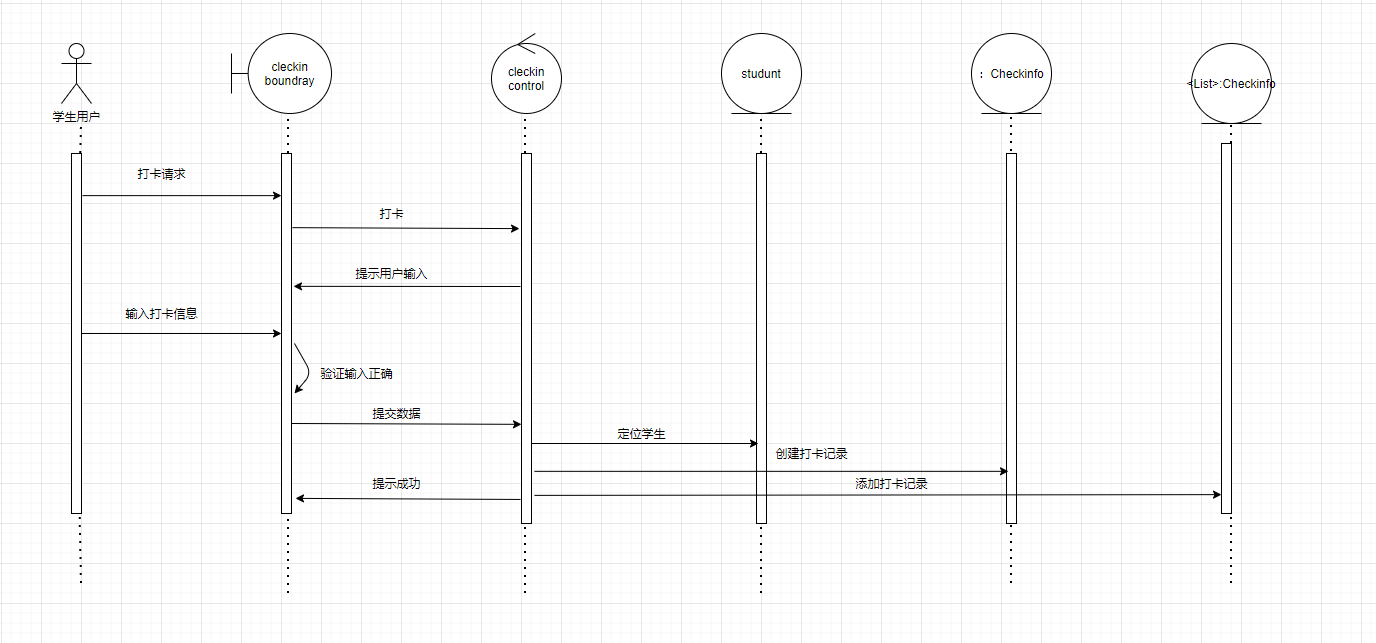
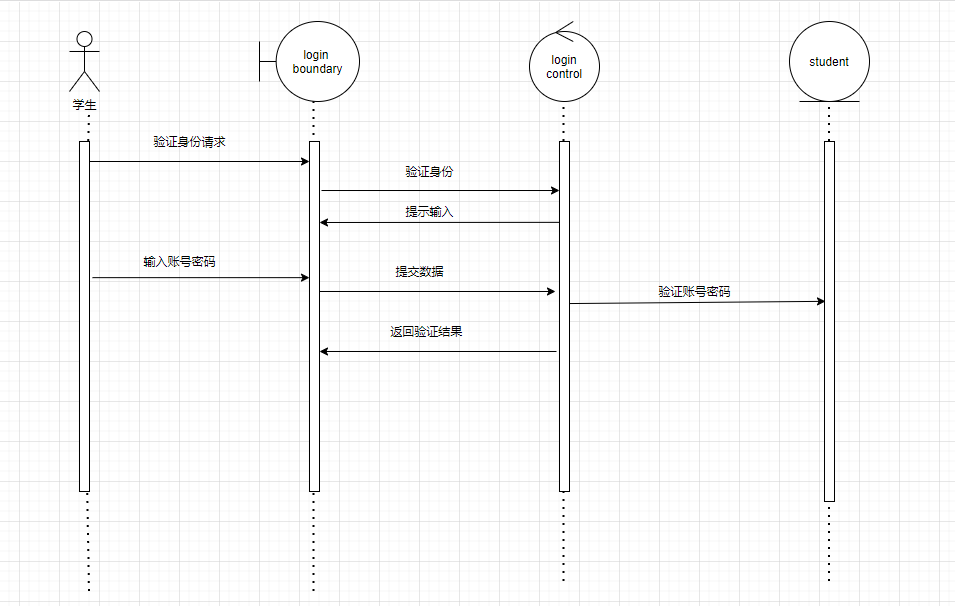
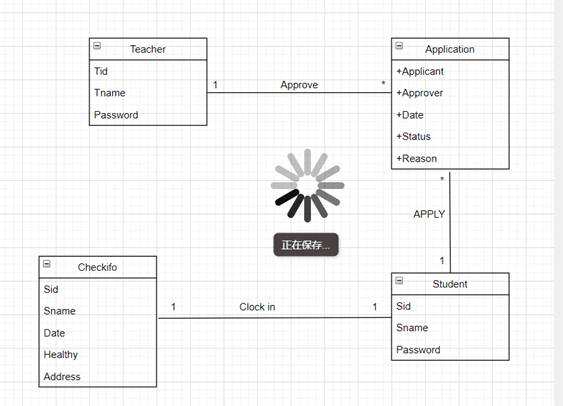
前置条件：系统验证用户身份成功

后置条件：系统增加学生打卡信息

基本流:

1. 学生用户向系统发出查询打卡情况的请求
2. 系统查询未打卡的名单，显示。
3. 系统提示用户查询成功

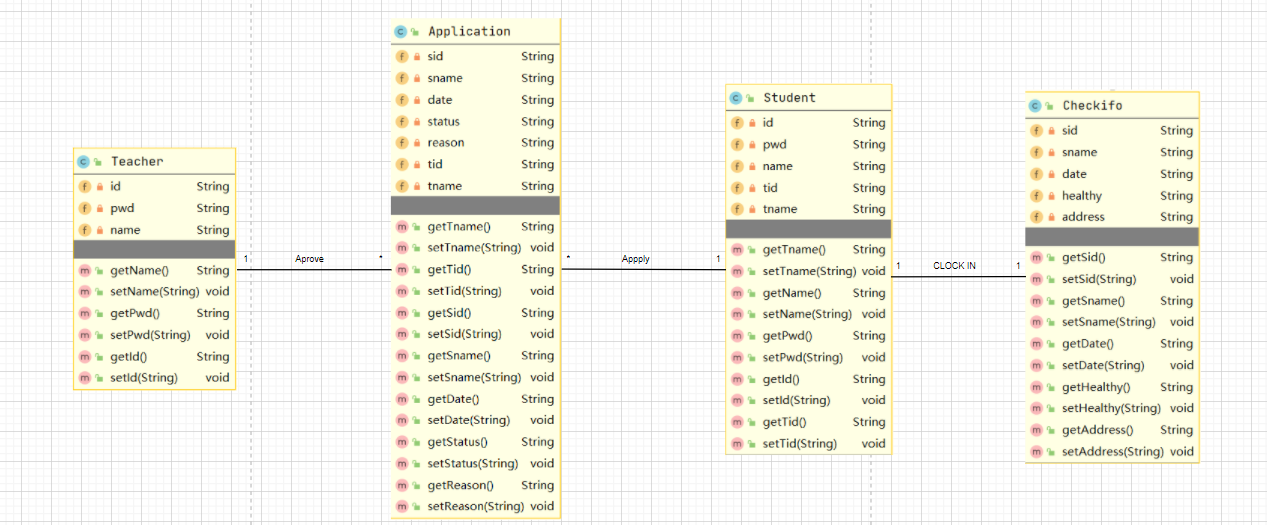
# 分析建模

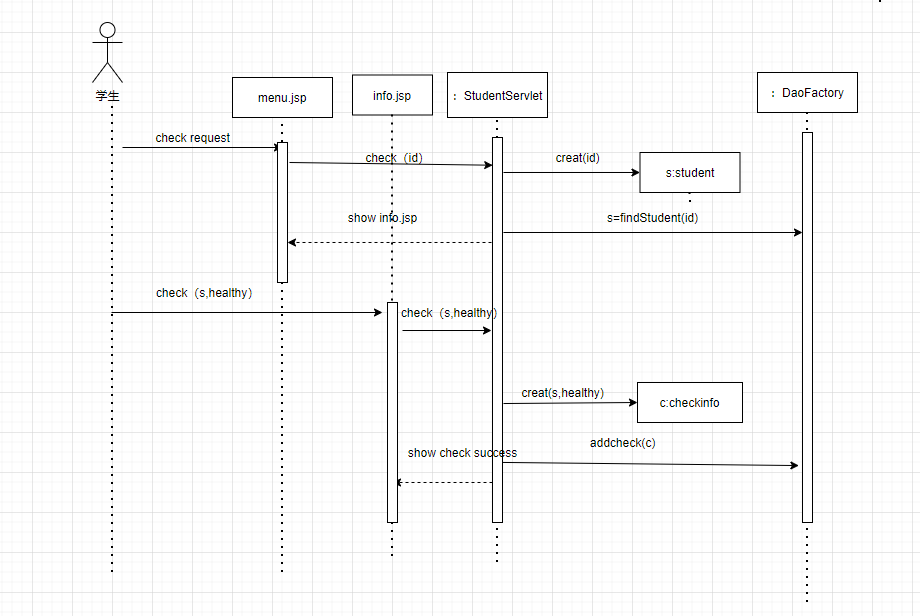
      

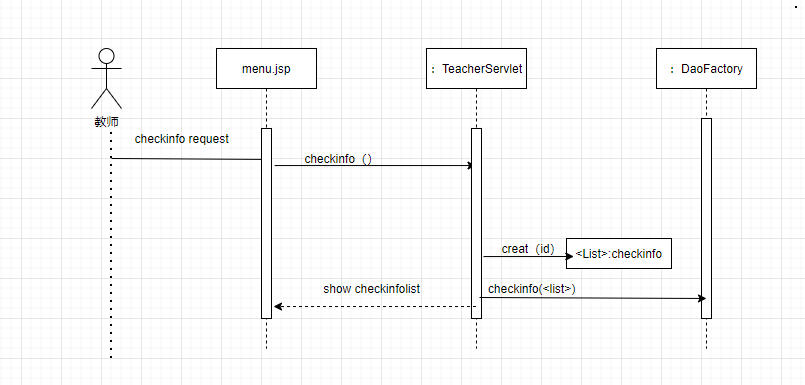
# 第三章 设计建模

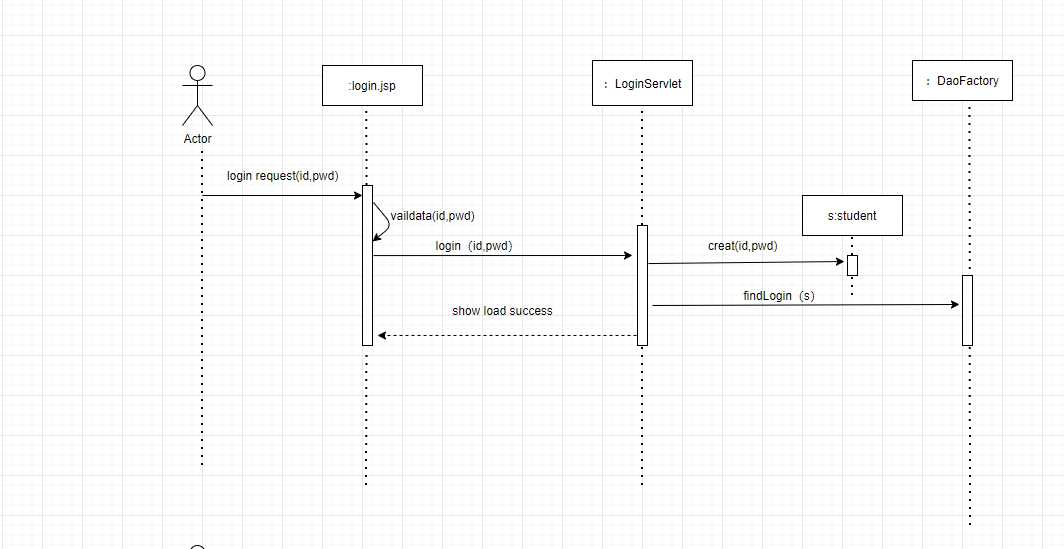
分配到的需求在小组架构设计的基础上，结合grasp原则设计的顺序图、类图，活动图等第

需在顺序图中标明运用到的grasp原则



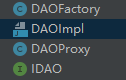




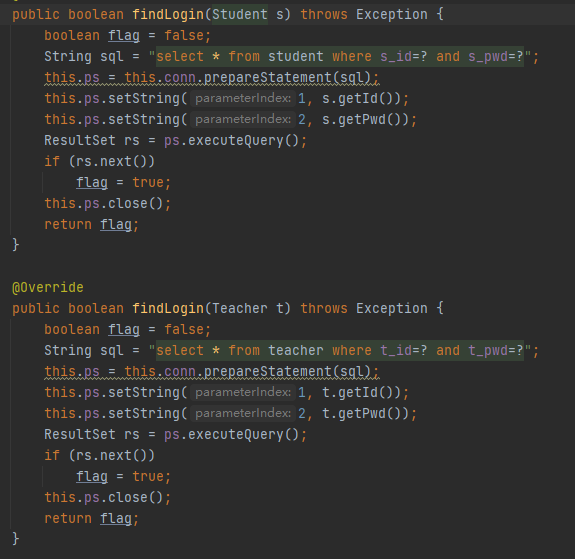


# 第四章 实现

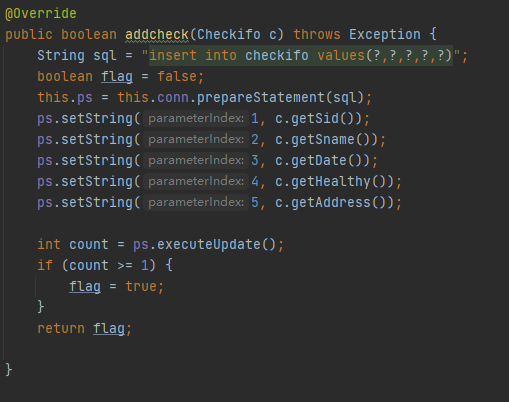
1. 4.1 m实现
2. 数据访问层采用工厂模式factory。



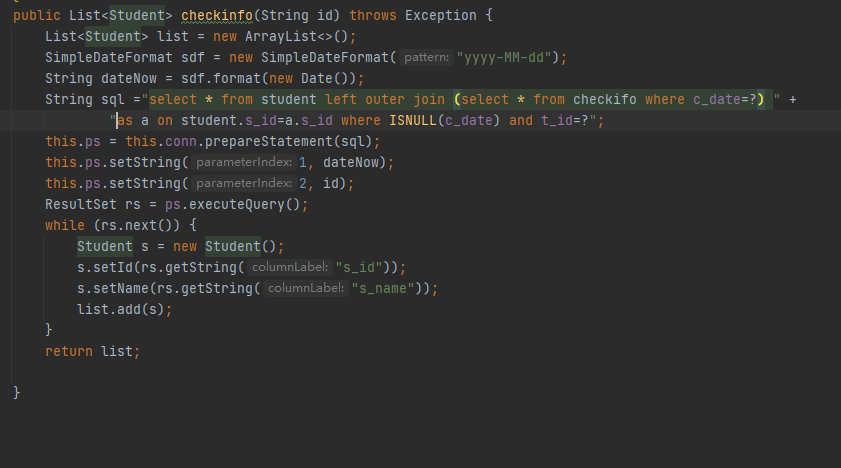
登陆：



打卡：

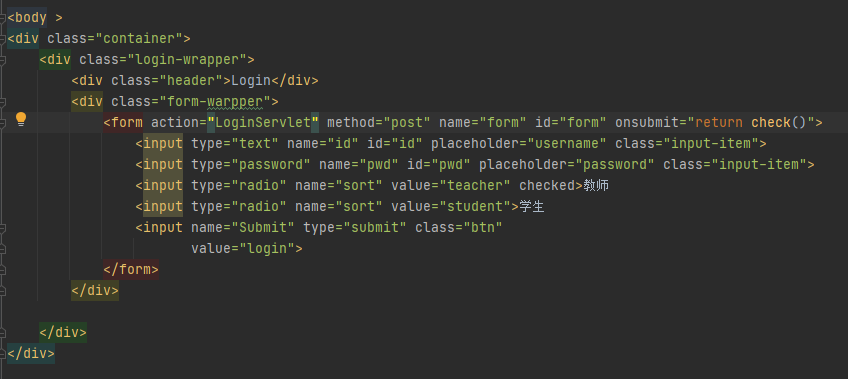


打卡情况

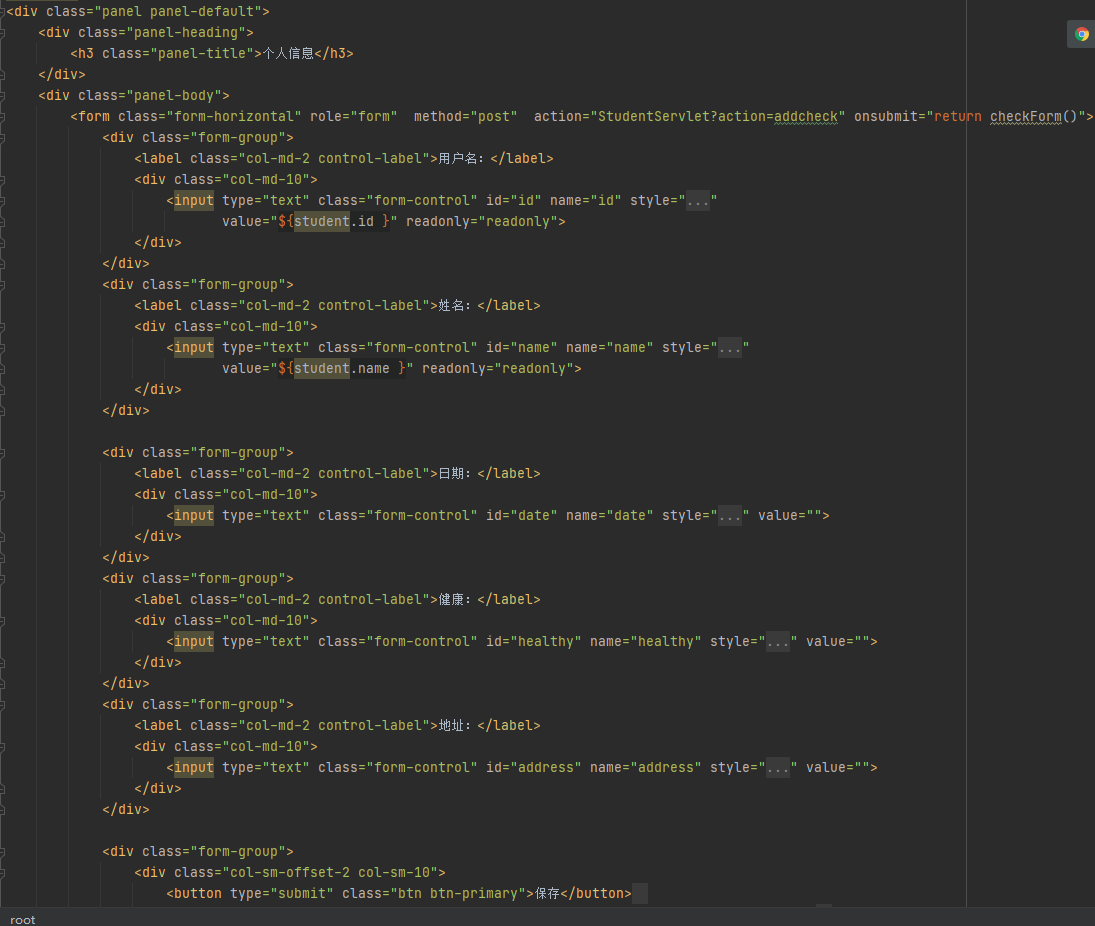


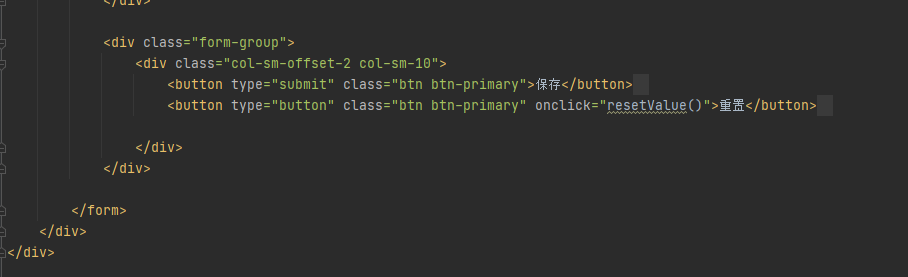
4.2 v实现

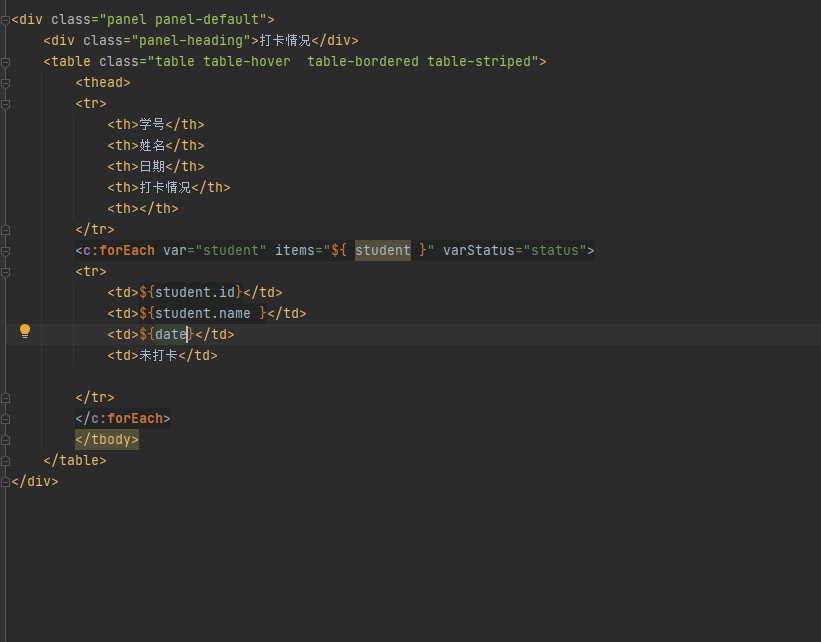
登陆界面：



打卡界面：

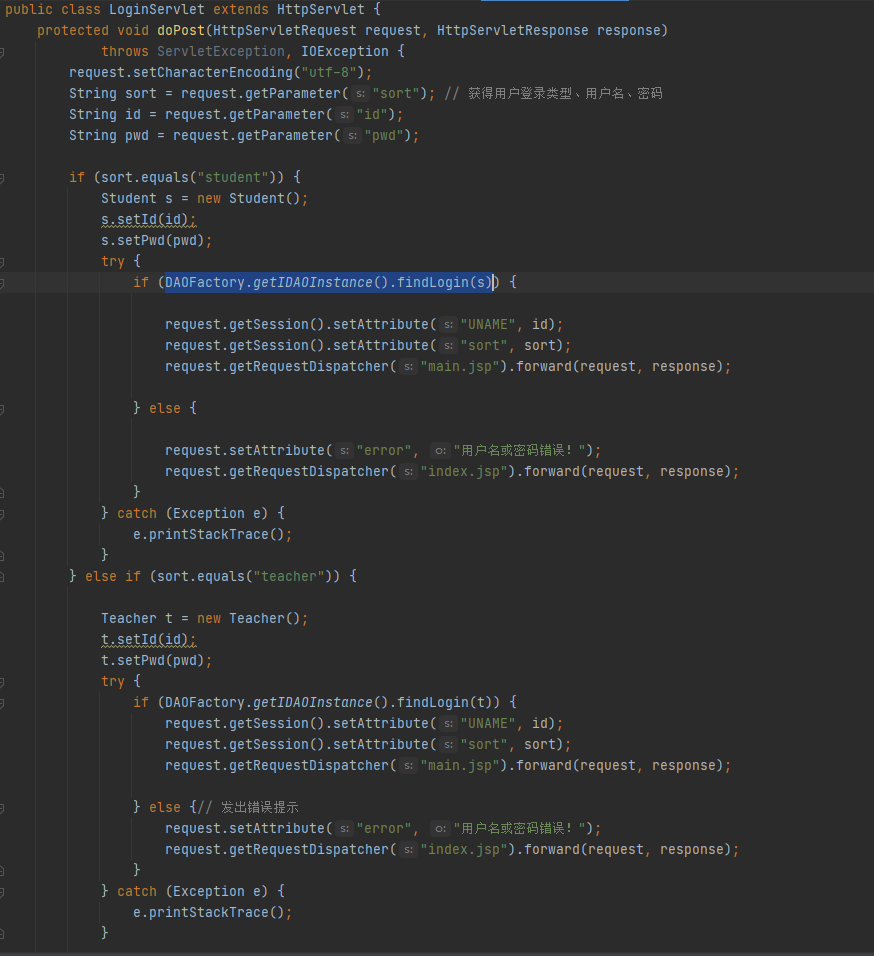


打卡信息界面

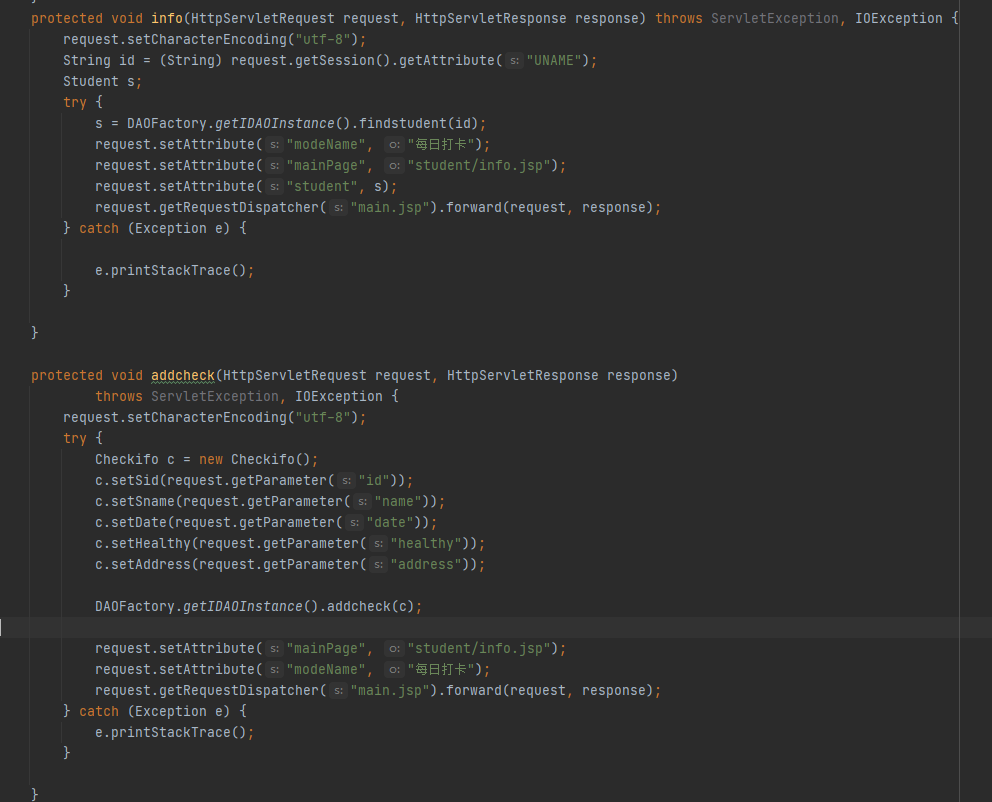


4.3 c实现

登陆：



每日打卡：



打卡信息：



# 个人总结

通过一学期的uml课程学习，让我知道了如何设计软件，之前写软件的时候总是不知所措无从下手，学习uml后，明确了开发软件的基本步骤，学习了软件设计一步步的迭代过程，从需求到分析到设计到最后的实现，一步一步慢慢学习软件设计，最后做出软件。他是统一建模语言，使复杂的软件设计更为简单。在实现一个庞大的系统时，如果在具体写代码时某个类的职责过于庞杂，那么必定会给系统带来很大的压力。或者说每个类之间的关系特别复杂，那么当后续需要更改某个类的时候，必定会影响到其他的类，带来十分高昂的维护成本。这时候如果通过建模分析，就能更好得实现。

在课程设计当中，应该把重心放在建模上面，有足够细心和完整的建模，软件实现起来非常轻松，架构清晰，不会像无头苍蝇一样无从入手