# Requirement Analysis of The Homework

# Project Report

Project Name： Where is the class-

-room **\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Name： **71118415-叶宏庭**

Date： **2019年12月4日**

Dept.: Software Engineering

# Contents

1. Introduction 3

1.1 Background of the project 3

1.2 Division of work 3

2. Requirements 3

2.1 User stories 3

2.2 Alternative flows 4

2.3 Use cases 4

2.4 Activity diagram 8

3.Design 10

3.1 Architectural design 10

3.2 the workflow of the system (if needed) 10

3.3 Classes 10

4. Implementation 11

4.1 Module 1 11

5. Testing 13

6. Conclusion 15

# 1. Introduction

## 1.1 Background of the project

**东南大学学校教务处需要开发一款手机应用程序“教室在哪儿？”给全校师生使用。**

(1) 学生可以登录App查看课表，以及上课提醒，教室位置导航。

(2) 教师可以登录App查看自己的课表，以及上课提醒，教室位置导航，空教室预约。

(3) 系统管理员可以为学生、教师注册账号，调整教室使用安排。

(4) 采用数据库存储个人信息及课程信息，在无网络环境下，调用本地数据库信息，在有网络环境下，从网络端获取信息。

# 2. Requirements

## 2.1 User stories

(1) 经过系统管理员的审核后，学生可以登录该App

(2) 使用学生账户登录后，学生方可使用该App

(3) 学生可以在App上查询自己得课表

(4) 学生可以使用该App的上课提醒功能

(5) 学生可以使用该App的教室导航功能

(5) 经过系统管理员的审核后，教师可以登录该App，与(1)相似

(6) 使用教师账户登录后，教师方可使用该App，与(2)相似

(7) 教师可以在该App上查询自己的上课安排

(8) 教师可以使用该App的上课提醒功能，与(4)相似

(9) 教师可以使用该App的教室导航功能，与(5)相似

(10) 教室可以使用该App的空教室预约功能

(11) 系统管理员使用账户登录后，管理员方可使用该App

(11) 系统管理员可以使用服务端为学生，教师注册账号，审核账号

(12) 系统管理员可以调整教室的使用安排

## 2.2 Alternative flows

(1) 若系统管理员调整教室使用安排，会对学生端，教师端发出提醒

(2) 若学生，教师账号存在异常，则无法登录

(3) 系统管理员可以处理学生，教师的账号异常问题

## 2.3 Use cases

(1) Use case diagram

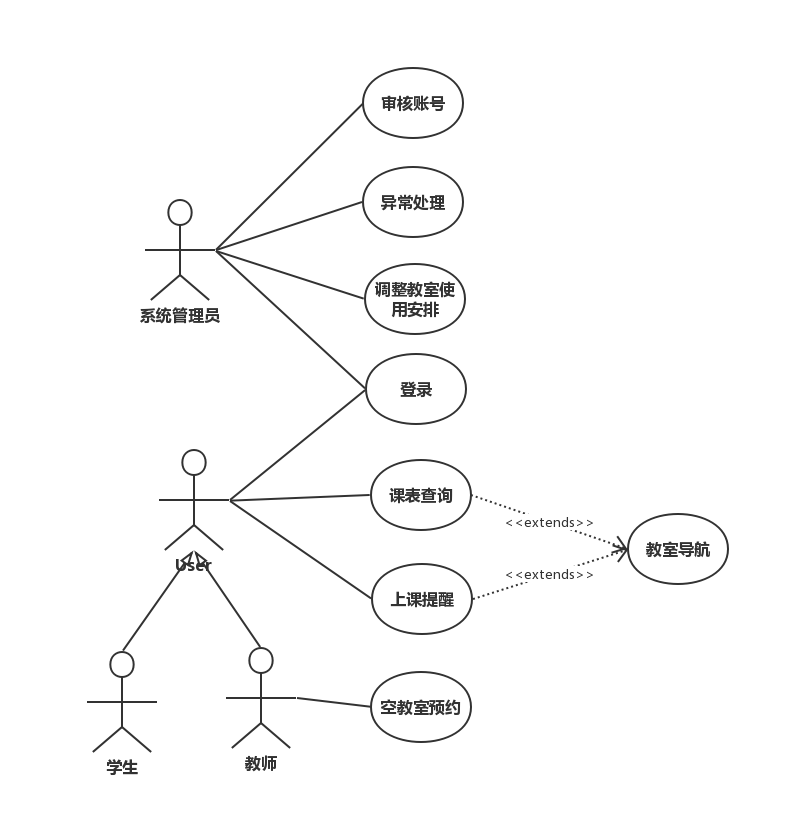


Figure 1. use case diagram

(2) use case specification

Table 1. 登录用例规约

|  |  |  |
| --- | --- | --- |
| **Items** | **Descriptions** | **Notes** |
| Name of the use case | **登录** |  |
| Brief description | 学生，教师，系统管理员登录该APP，之后才可以使用该App。 |  |
| Basic flow | 1. 使用者在界面上输入账号与密码。  2. 输入完毕后，点击登录按钮进行登录。 |  |
| Alternative flows | 1. 若登录失败，显示失败原因(密码错误，二次登录，账号异常等)。  2. 若密码不符合密码格式，提示用户。 |  |
| Precondition | 1. 该账户目前没有被登录。  2. 该账号经过注册且审核通过。 |  |
| Postcondition | 1. 为该用户加载个人信息。  2. 为该用户调取课表信息。 |  |

Table 2. 课表查询用例规约

|  |  |  |
| --- | --- | --- |
| **Items** | **Descriptions** | **Notes** |
| Name of the use case | **课表查询** |  |
| Brief description | 学生，教师通过该用例查询自己的课表 |  |
| Basic flow | 1. 成功登录之后，用户可以在界面上查看本周课表，或者点击切换(向前，向后)按钮(每次点击切换到前/后一周)，查看指定周次的课表。  2. 用户点击切换按钮后，界面显示切换后的周次课表。  3. 用户可以点击一个课程的图标，显示该课程的具体信息。(课程名称，教师名字，上课教室，上课周次等) |  |
| Alternative flows | 1. 用户可以在查看课程信息的界面上点击教室导航按钮，开始导航功能。 |  |
| Precondition | 1. 用户已登录。 |  |
| Postcondition | None |  |

Table 3. 上课提醒用例规约

|  |  |  |
| --- | --- | --- |
| **Items** | **Descriptions** | **Notes** |
| Name of the use case | **上课提醒** |  |
| Brief description | 提醒用户将要进行的课程 |  |
| Basic flow | 1. 在距离上课还有一小时的时间，对用户进行提醒。(提醒内容包括：课程名称，上课地点)  2. 若用户没有响应提醒，则过五分钟再次提醒直至用户响应提醒。(响应提醒指用户以收到该提醒并且点击关闭该提醒) |  |
| Alternative flows | 1. 在提醒界面中加入教室导航按钮，若用户点击该按钮，则开始导航功能。(导航终点为该提醒的课程地点) |  |
| Precondition | 1. 有即将进行的课程 |  |
| Postcondition | None |  |

Table 4. 空教室预约用例规约

|  |  |  |
| --- | --- | --- |
| **Items** | **Descriptions** | **Notes** |
| Name of the use case | **空教室预约** |  |
| Brief description | 教师使用该功能进行教室预约使用 |  |
| Basic flow | 1. 教室在主界面上点击空教室预约按钮，进入空教室预约界面。  2. 预约界面应包括：预约时间，使用人数。  3. 系统按照教师的时间，人数需求自动选择一个合适的教室。  4. 教师点击确认按钮，完成预约。 |  |
| Alternative flows | 1. 若不存在合适的教室，则提醒教师不存在合适教室。 |  |
| Precondition | 1. 教师进入空教室预约功能。 |  |
| Postcondition | 1. 若预约成功，则被预约的教室不能再被其他教师预约。  2. 在预约的时间即将到时，使用上课提醒功能提醒教师使用教室。 |  |

## 2.4 Activity diagram

(1)导航用例活动

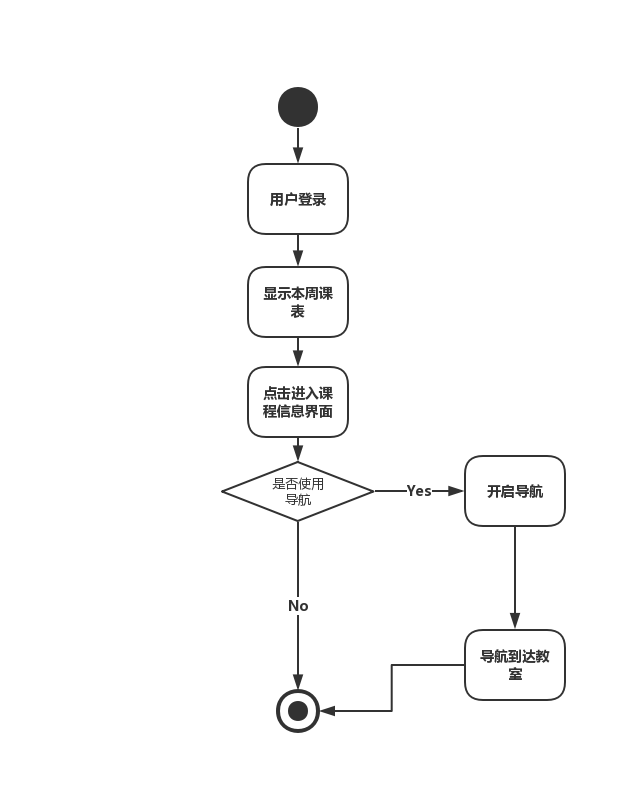


Figure 5. 导航用例活动图

(2) 上课提醒用例活动

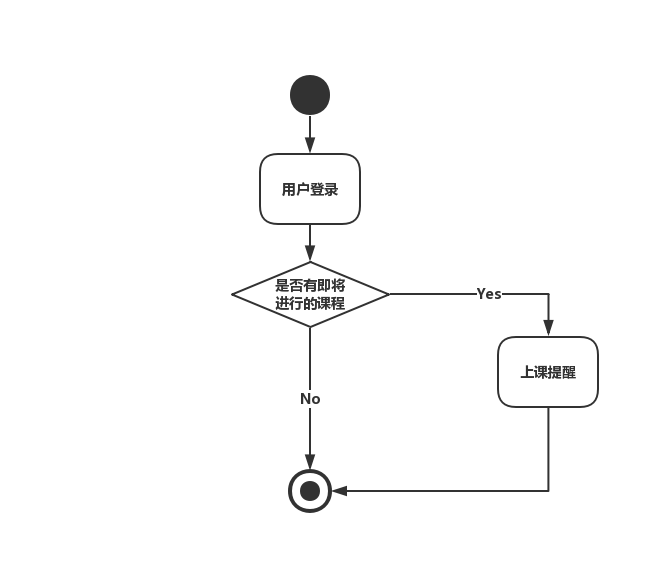


Figure 6. 上课提醒用例活动图

(3) 空教室预约用例活动

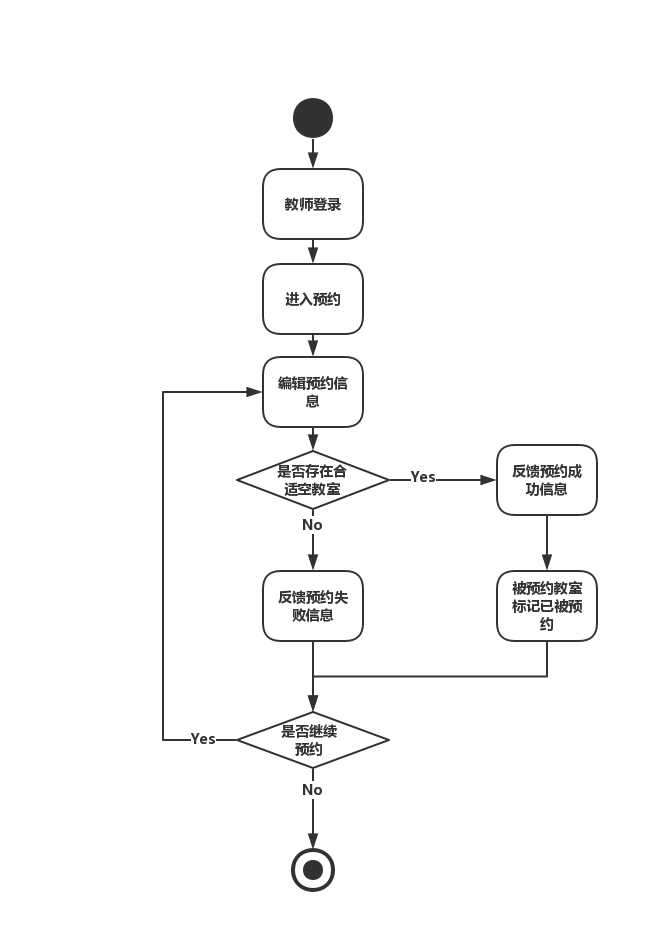


Figure 7. 空教室预约用例活动图

# 3.Design

## 3.1 Architectural design

*Give your architectural design and explanations to each module.*

*For example:*

XXXX System

Sub-System 1

Sub-System 2

Sub-System 3

Module 4

Module 3

Module 2

Module 1

Figure 7. The architecture of the system

*Explanations:*

(1) Module 1: the functions of Module 1.

(2) Module 2: the functions of Module 2.

.....

## 3.2 The workflow of the system (if needed)

*Use an activity diagram to illustrate the operating processes of your system & give some explanations about them.*

## 3.3 Classes

*Design the classes for your system by abstracting the objects and refining them.*

For example:

We designed 4 classes for our system.

(1) Student class: explanation of student class.

(2) Teacher class: explanation of teacher class.

....



Figure 8. The class diagram of the system

# 4. Implementation

## 4.1 Module 1

(1) Briefly introduce the main functions of Module 1.

(2) The classes involved in Module 1 should be specified and explained.

(3) If there are message transmissions between the classes, use a sequence diagram to illustrate them.

For example:

The sequence diagram (for the "Download File" case)



Figure 9. The sequence diagram of Module 1

(4) If there are some important algorithms in a method, please describe them specifically.

For example:

|  |
| --- |
| **Algorithm 1.** XXXXX  **Inputs:**  **Outputs:**  **Steps:**  ...... |

If the method is not very long (less than a page), you can just copy it here, with some comments.

For example:

|  |
| --- |
| Public void XXXXX\_function(parameters)  {  ..... //comments  .....  } |

(5) If you use databases to store your data, please describe the design of your tables.

For example:

Table2. Sys\_XXXXX\_Info

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Explanation** | **Data Type** | **Note** |
| Id | 编号 | Int | Primary Key |
| ProjectName | 工程名称 | Varchar |  |
| VersionID | 项目版本号 | Varchar |  |
| ProjPath | 项目路径 | Varchar |  |
| McCabe | 圈复杂度 | Int |  |

(6) The user interfaces of Module 1

The user interface of Module 1 should be represented and explained.

For example:

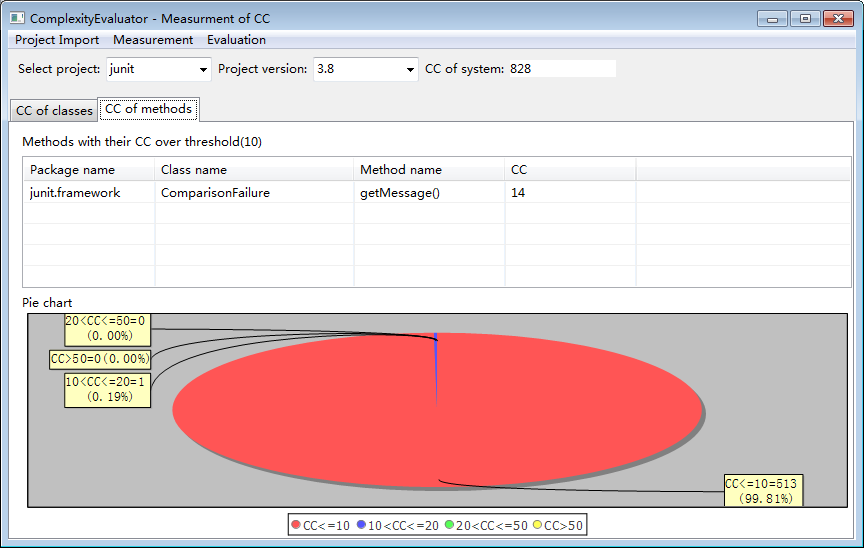


Figure 10. The user interface of Module 1

# 5. Testing

For the main scenarios, please show how your system runs with a sequence of user interfaces.

(1) Briefly introduce the scenario of the system usage.

(2) Show how your system runs according to the scenario.

For example:

For the use case xxxx, the users need to do xxxx, and the system will do xxxx.

The steps of this scenario are:

(1) The first step of xxxx is to do xxxx, and the system will show a user interface as Figure 11 shows:

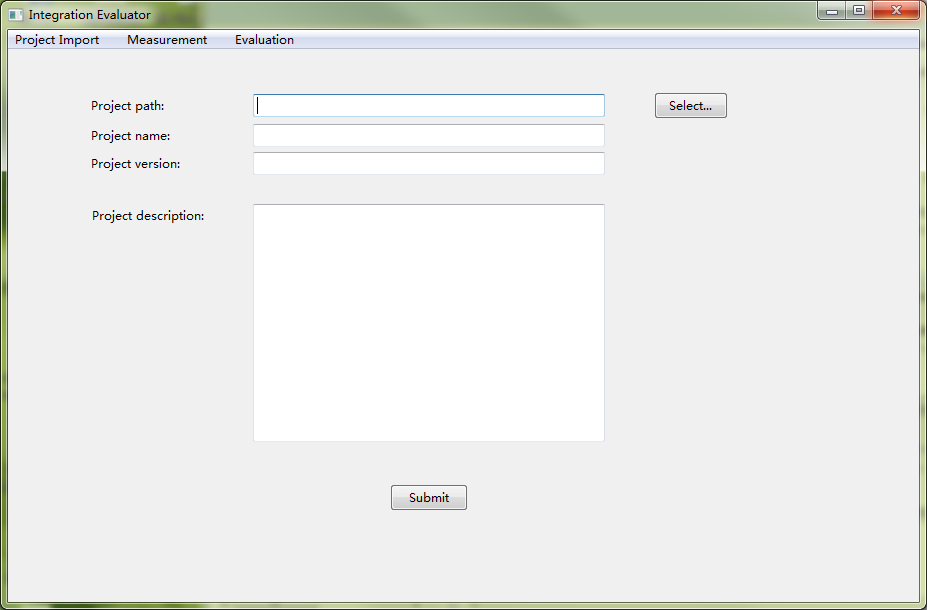


Figure 11. The user interface of step 1

(2) Then the user can do xxxx, and the system will show the results as Figure 12 shows:

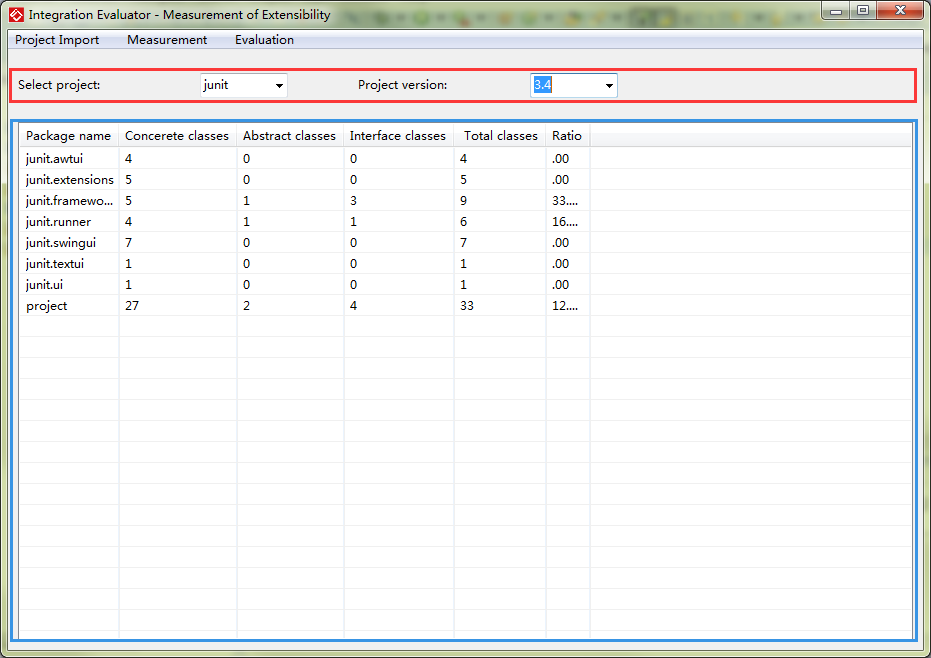


Figure 12. The result of step 2

# 6. Conclusion

Each team members should conclude their work, summarize the lessons they learned from the project. (About 3-5 lines)

Besides the technical gains, the team leader should say something more about the lessons learned from the organization and the management of the project.