<项目名称>

软件架构文档

版本 <1.0>

[注：用方括号括起来并以蓝色斜体（样式=InfoBlue）显示的文本，它们用于向作者提供指导，在发布此文档之前应该将其删除。按此样式输入的段落将被自动设置为普通样式（样式=Body Text）。]

[要定制 Microsoft Word 中的自动字段（选中时显示灰色背景），请选择 File>Properties，然后将 Title、Subject 和 Company 等字段替换为此文档的相应信息。关闭该对话框后，通过选择 Edit>Select All（或 Ctrl-A）并按 F9，或只是在字段上单击并按 F9，可以在整个文档中更新自动字段。对于页眉和页脚，这一操作必须单独进行。按 Alt-F9，将在显示字段名称和字段内容之间切换。有关字段处理的详细信息，请参见 Word 帮助。]

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <日/月/年> | <x.x> | <详细信息> | <姓名> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 2

1.1 目的 2

1.2 参考资料 2

2. 用例视图 2

3. 逻辑视图 2

3.1 概述 2

3.2 在构架方面具有重要意义的设计包 2

4. 进程视图 2

5. 部署视图 **错误!未定义书签。**

6. 实现视图 **错误!未定义书签。**

7. 数据视图（可选） 2

8. 核心算法设计（可选） 2

软件架构文档

# 简介

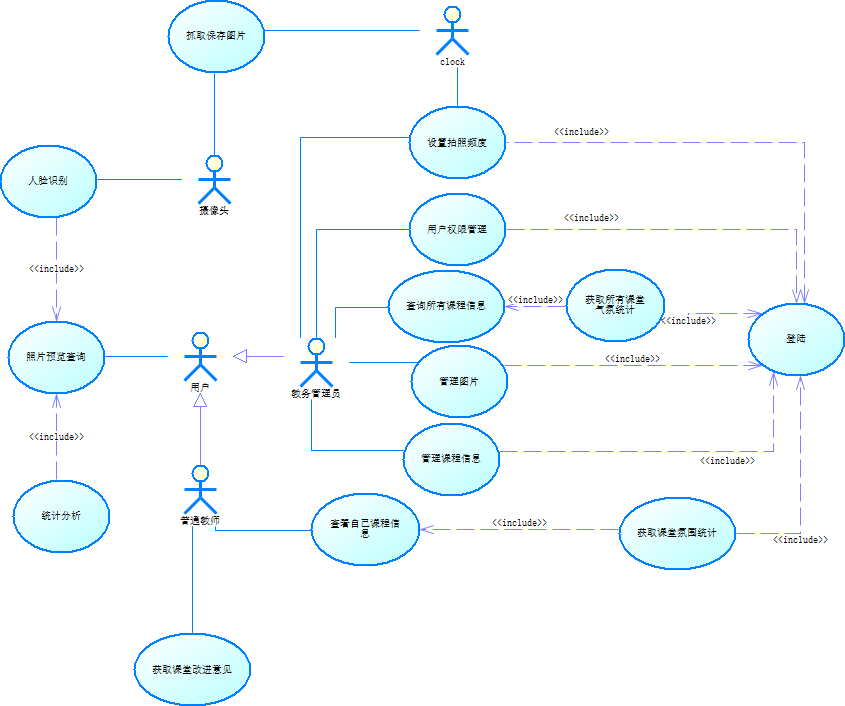
## 目的

本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

## 参考资料

无

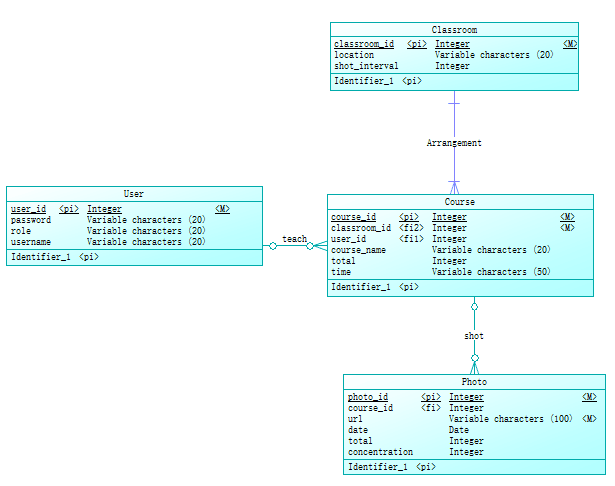
# 用例视图



# 逻辑视图

## 概述

以下为项目实体类的逻辑模型：



User类：负责用户在数据库里的登陆注册以及相应的权限认证。

Course类：负责记录每门课程的详细信息，例如教室编号、课程编号、课程名等

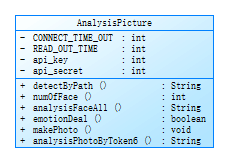
Photo类：负责记录每张照片的详细信息，例如拍摄的时间、地点、课程等

Classroom类：负责记录教师的详细信息，例如地理位置、拍摄间隔

此外，还有单独封装的AnalysisPicture.jar包，对分析图片这个过程进行了封装

## 在构架方面具有重要意义的设计包

AnalysisPicture 封装jar包0.0.1版使用



1. 0.0.1版使用要求:
   1. 安装python3.6
   2. 安装pillow，命令行输入 pip install pillow
   3. 在项目目录\src\main\resources下放置py文件markPicture.py，文件在github仓库中code\AnalysisPicture\src\main\resources目录下
2. 属性说明

2.1.CONNECT\_TIME\_OUT与READ\_OUT\_TIME限制了连接的时间

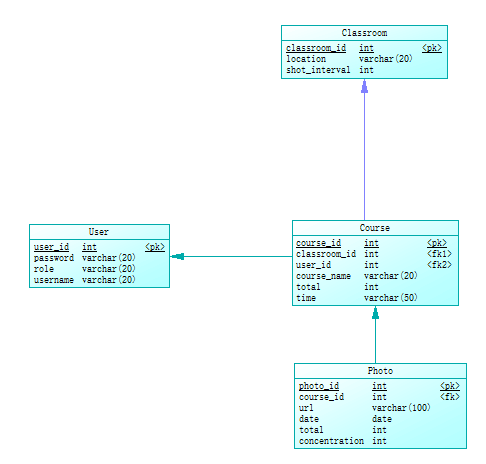
2.2.api\_key与api\_secret代表着你使用这个包的用户名和密码

1. 封装的函数
   1. DetectByPath(String filepath) 输入一个图片的完整路径，返回由face++返回的json字符串
   2. NumOfFace(String detectStr) 输入一个图片由detactByPath返回的json字符串 返回图片中被识别的人脸数(int)
   3. analysisFaceAll(String detectStr) 输入一个图片由detactByPath返回的json字符串 返回图片中被识别的人脸数和被认为是专注的人脸数，是个json字符串，形如 {"total":1,"concentrate":1}
   4. markPhoto(String filepath, String detectStr) 输入参数中前者为图片完整路径，后者为由detactByPath返回的json字符串， 返回值mark ok 表示标识成功，若成功标识，console会直接打出success，并且会直接显示图片，并在原图片路径下新建一个名为marked的文件夹，储存被标识的图片
   5. String analysisFaceByTokens(String facetoken) （尽量别用） 输入图片token，返回由face++analysis api的返回值

# 进程视图

本项目只有一个进程，即为server端，在该进程中有多个线程，用来应对多个用户同时使用这个项目。

# 数据视图



# 核心算法设计

表情属性加权

眼睛张合度

大多数学生专注意向数/所有学生数

表情属性加权：

surprise+ 2\*

Happiness+ 2\*

Neutural= 1

Sadness- 0.6

Fear- 0.6

Disgust- 0.4

Anger- 0.4

>0.7

眼睛张合度

>0.6

大多数学生专注意向数/所有学生数

Max（低头，抬头）/总人数

>0.7