
AITA

软件工程课程项目概要设计文档

1650262 梁峻浩

1652714 孙浩然

目录

AITA	1
目录	2
引言	3
概要设计依据	3
参考资料	3
假定和约束	4
概要设计	5
系统总体架构设计	5
系统软件结构设计	7
结构语境图	7
系统业务分析类图	8
课程管理子系统	8
学习结果管理子系统	9
用户管理子系统	9
系统软件架构图	10
接口设计	10
内部接口	10
外部接口	13
界面设计	14
数据库设计	18
系统出错处理设计	20
出错信息	20
补救措施	21
系统维护设计	21

引言

概要设计依据

- AITA——需求规约文档
- AITA——需求分析规约文档

参考资料

[1] 质量管理体系国家标准理解与实施(2008 版)

[2] Jeon, J., Park, J. C., Jo, Y. J., Nam, C. M., Bae, K. H., & Hwang, Y., et al. (2016). A Real-time Facial Expression Recognizer using Deep Neural Network. International Conference on Ubiquitous Information Management and Communication(pp.1-4). ACM.

[3] Kim, B. K., Roh, J., Dong, S. Y., & Lee, S. Y. (2016). Hierarchical committee of deep convolutional neural networks for robust facial expression recognition. Journal on Multimodal User Interfaces, 10(2), 1-17.

[4] Chang, T., Wen, G., Hu, Y., & Ma, J. J. (2018). Facial expression recognition based on complexity perception classification algorithm.

[5]Li, D., Wen, G., Hou, Z., Huan, E., Hu, Y., & Li, H. (2018). Rtc relief-f: an effective clustering and ordering-based ensemble pruning algorithm for facial expression recognition. Knowledge & Information Systems, 1-32.

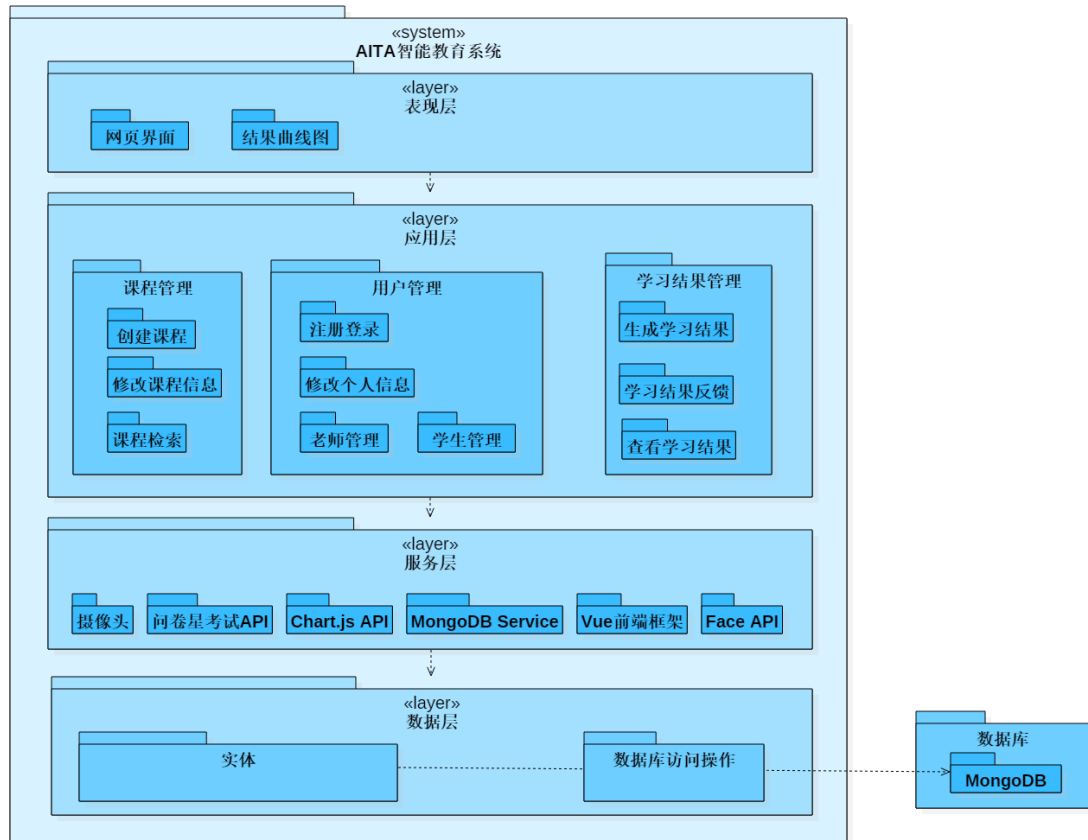
假定和约束

开发期限：2018 年 12 月 31 日

设备条件：具有摄像头的 PC 机

概要设计

系统总体架构设计



本架构以 **MVC 框架**为指导，将系统分为表现层、应用层、服务层、数据层四个层次。

表现层为系统的 webGUI 表面和图形化显示部分，接受用户操作等信息，通过 HTTP 协议与应用层进行通信。

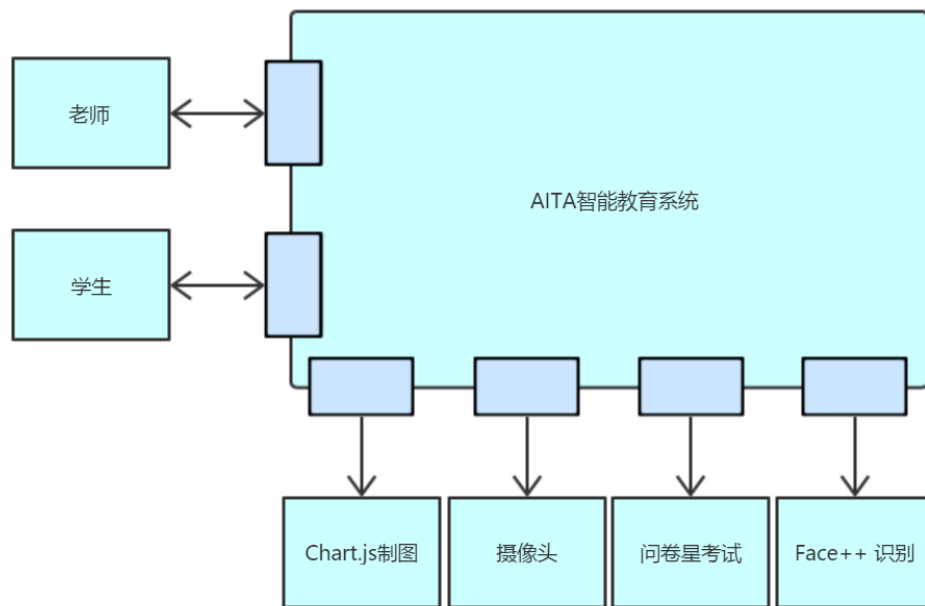
应用层为系统主要业务逻辑，处理表现层接收的前端请求，同时控制前端的显示内容，对前后端进行连接。系统主要业务分为三个子模块：课程管理、用户管理、学习情况管理。

服务层为应用层提供支持，包括前后端框架和系统使用到的第三方 API 的管理。

数据层作为 MVC 中的 Model 部分，提供实体类对象的持久化存储和读取方法。数据库访问操作中封装了对 MongoDB 数据库的连接、查询等基本操作，其它层在需要与数据库进行交互式只调用数据库访问。

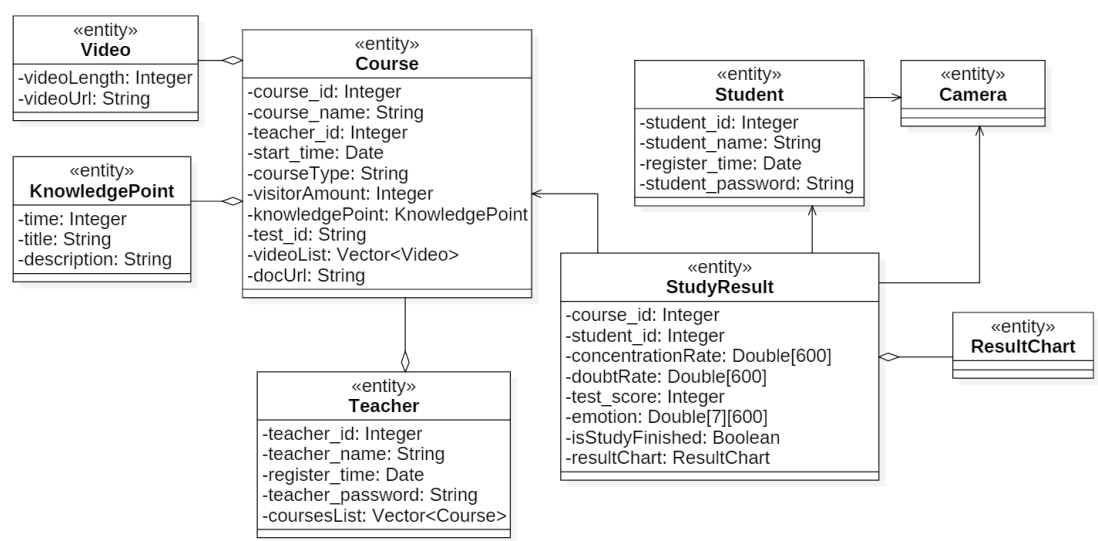
系统软件结构设计

结构语境图



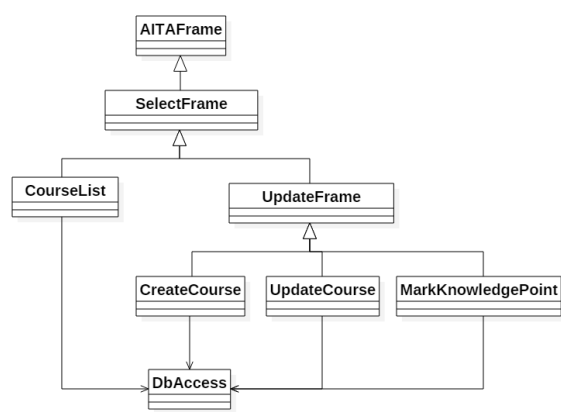
AITA 智能教育系统是一个相对独立的平台，与老师和学生用户进行交互，但几乎不与其他系统构成 peer 或 superordinate 关系；系统在使用过程中需要调用外部的系统来进行识别校准、绘图、摄像、生成试卷功能。

系统业务分析类图

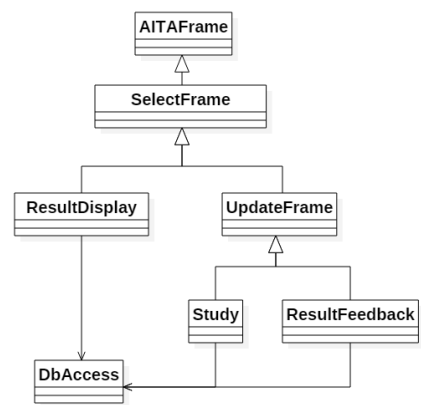


根据系统业务可以将整个系统分为课程管理、学习结果管理、用户管理三个子系统。

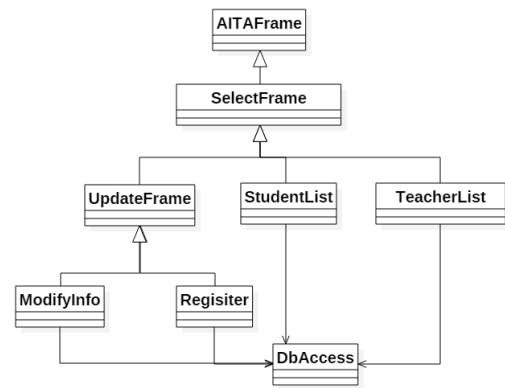
课程管理子系统



学习结果管理子系统

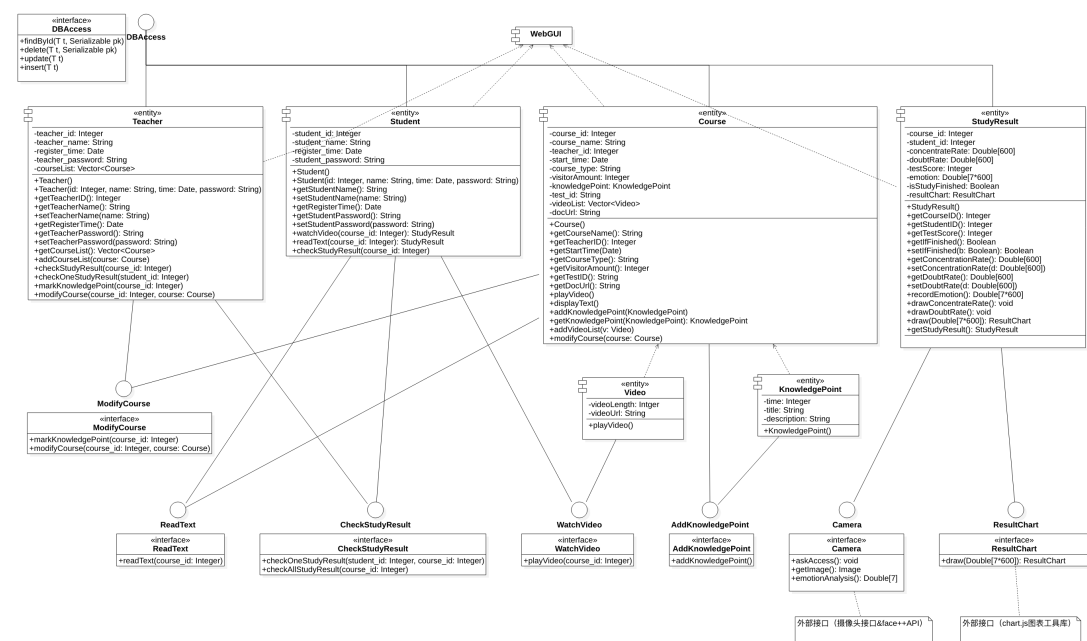


用户管理子系统



通过 problem frame 继承的形式针对每个子系统初步构建设计类图。

系统软件架构图



将系统业务设计类图初步加入接口类并构成组件，组件之间的层次调用是通过接口实现的。主要业务模块的操作对象抽象为组件，对于需要调用其他组件实现的功能定义接口，通过接口调用实现特定业务功能。

接口设计

内部接口

实体类从数据库表格映射得到，其属性值为私有类，可以通过对应的 `getter` 和 `setter` 进行访问和修改。同时还提供一部分接口完成业务逻辑。

- 《interface》ModifyCourse

ModifyCourse 接口用来实现 Teacher 类与 Course 类之间方法的调用：

Teacher 类的 markKnowledgePoint(course_id:Integer) 方法通过接口来调用 Course 类的 addKnowledgePoint(KnowledgePoint) 方法，完成标记知识点功能。

Teacher 类的 modifyCourse(course_id:Integer,course:Course) 方法通过接口 (modifyCourse(course_id:Integer,course:Course)) 来调用 Course 类的 modifyCourse(course:Course) 方法，完成改进课程功能。

- 《interface》 ReadText

ReadText 接口用来实现 Student 类与 Course 类之间方法的调用：Student 类的 readText(course_id:Integer) 方法通过接口来调用 Course 类的 displayText() 方法。

- 《interface》 CheckStudyResult

CheckStudyResult 接口用来实现 Student 类、Teacher 类与 StudyResult 类之间方法的调用：Student 类的 checkStudyResult(course_id:Integer) 方法通过接口 (checkOneStudyResult(student_id:Integer,course_id:Integer)) 来调用 StudyResult 类的 getStudyResult() 方法，得到特定学生的学习结果。

Teacher 类的 checkOneStudyResult(student_id:Integer)方法通过接口 (checkOneStudyResult(student_id:Integer,course_id:Integer)) 来调用 StudyResult 类的 getStudyResult() 方法，得到特定学生的学习结果。

Teacher 类的 checkStudyResult(course_id:Integer) 方法通过接口 (checkAllStudyResult(course_id:Integer)) 来调用 StudyResult 类的 getStudyResult() 方法 , 得到选择此门课程的所有学生的学习结果。

- 《interface》 WatchVideo

WatchVideo 接口用来实现 Student 类与 Course 类之间方法的调用 :

Student 类的 watchVideo(course_id:Integer) 方法通过接口来调用 Course 类的 playVideo() 方法。

- 《interface》 AddKnowledgePoint

AddKnowledgePoint 接口用来实现 Course 类与 KnowledgePoint 类之间方法的调用 : Course 类的 addKnowledgePoint(KnowledgePoint) 方法通过接口中的方法 addKnowledgePoint() 来调用 KnowledgePoint 类实现标记知识点。

对数据库的访问、操作 , 通过 DBAccess 提供的接口来实现 , DBAccess 为抽象接口 , 具体数据访问类要重写该接口以方便调用。

- DBAccess

insert(T t) 保存/新建某个对象到数据库 ;

delete(T t, Serializable pk) 删除数据库的某个对象 ;

findById(T t, Serializable pk) 由主码查找某些数据 ;

update(T t) 更新某条记录。

外部接口

- 《interface》Camera

Camera 是外部摄像头接口，askAccess() 用来获取用户的摄像头访问权限，getImage(): Image 用来捕捉学习时的图像（1s/次），emotionAnalysis() 用来调用 face++API 实现表情分析，返回七中表情对应的概率 Double[7]。

Course 类的 recordEmotion() 方法实现此接口来获得表情信息，以供根据表情数据进行学习情况分析。

- 《interface》ResultChart

ResultChart 是绘制图表的外部 Chart.js 接口，Course 类的 draw(Double[7*600]) 方法实现此接口来绘制学习情况图表。

界面设计

登录注册界面：

AITA

个人信息课程管理面板



EDUCATION

用户名称

输入密码

登录

注册

AITA

个人信息课程管理面板



EDUCATION

用户昵称

输入密码

确认密码

选择身份

学生/老师

注册

登录

用户管理：

用户信息

[退出登录](#)

用户名: ludan

用户身份: teacher

注册时间: 2018-12-25

您的收藏

课程名称	任课教师	开课时间 ↕	所用时间 ↕	浏览人数 ↕	分类 ↕	考试ID ↕	操作
将改革开放进行到底	ludan	2018-12-24	12h	15	Social	32931138	参加 移除

课程管理：

课程名称	任课教师	开课时间 ↕	所用时间 ↕	浏览人数 ↕	分类 ↕	考试ID	操作
将改革开放进行到底	ludan	2018-12-24	12h	14	Social	32931138	参加 收藏
测试	ludan	2018-12-25	1min	0	computer	32931138	参加 收藏

课程名称

请输入课程名称

课程分类

请选择课程分类

预计需要时间

请输入完成课程大致需要时间（如：24小时）

输入考试ID

关于考试ID生成的说明

请前往[问卷星](#)生成。将生成的考试ID填入上方输入框

点击上传文件

请上传一个md文件作为阅读材料和一个mp4文件作为课程视频

立即创建

取消

您的课程

课程名称	开课时间 ↕	所用时间 ↕	浏览人数 ↕	分类	考试ID	操作
将改革开放进行到底	2018-12-24	12h	15	Social	32931138	结束 删除

观看视频页面：

AITA

视频学习

阅读材料

开始考试

学习结果

个人信息

课程管理面板

0:15 / 9:56

请确认您的摄像头打开后再进行视频学习，否则您的学习结果将不会被记录

开始时间

结束时间

> 00:00 10:00

< 02:00 02:30

真的棒啊真的棒

考试页面：

AITA

视频学习

阅读材料

开始考试

学习结果

个人信息

课程管理面板

AITA

上课时间 [【多选题】](#)

☐ 周三~4节

☐ 周五3~4节

请输入学号和姓名

例如：张三 1658888

1. 1+1=?

☐ 1

☐ 2

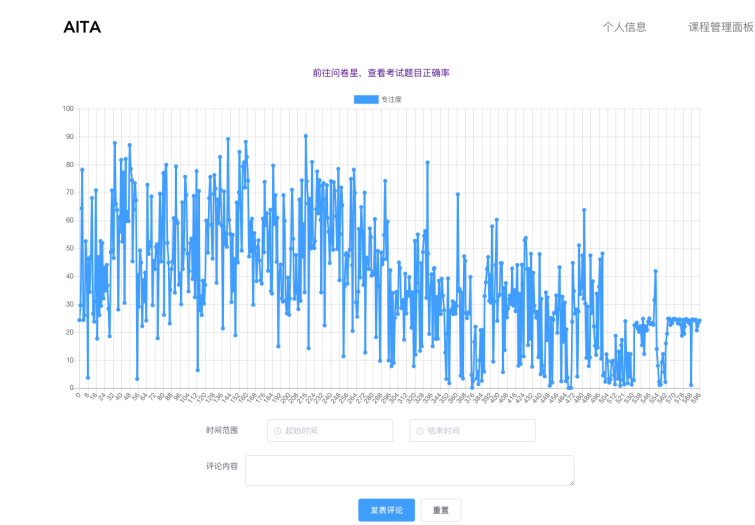
2. 地球绕着月球转

☐ 对

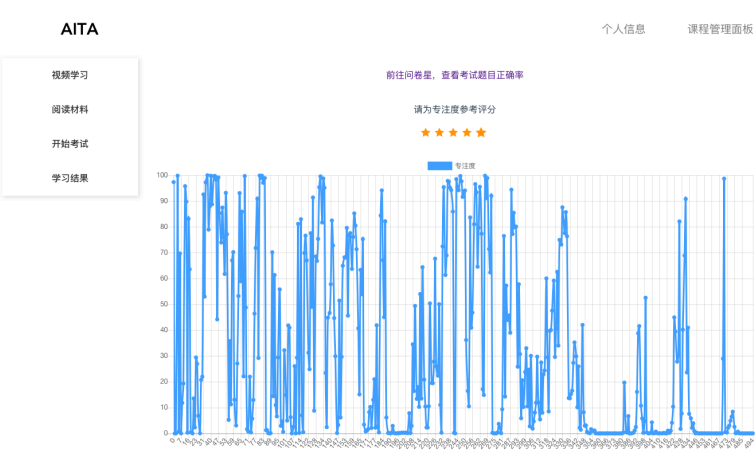
☐ 错

提交

老师查看课程学习情况：



学生查看学习结果页面：



数据库设计

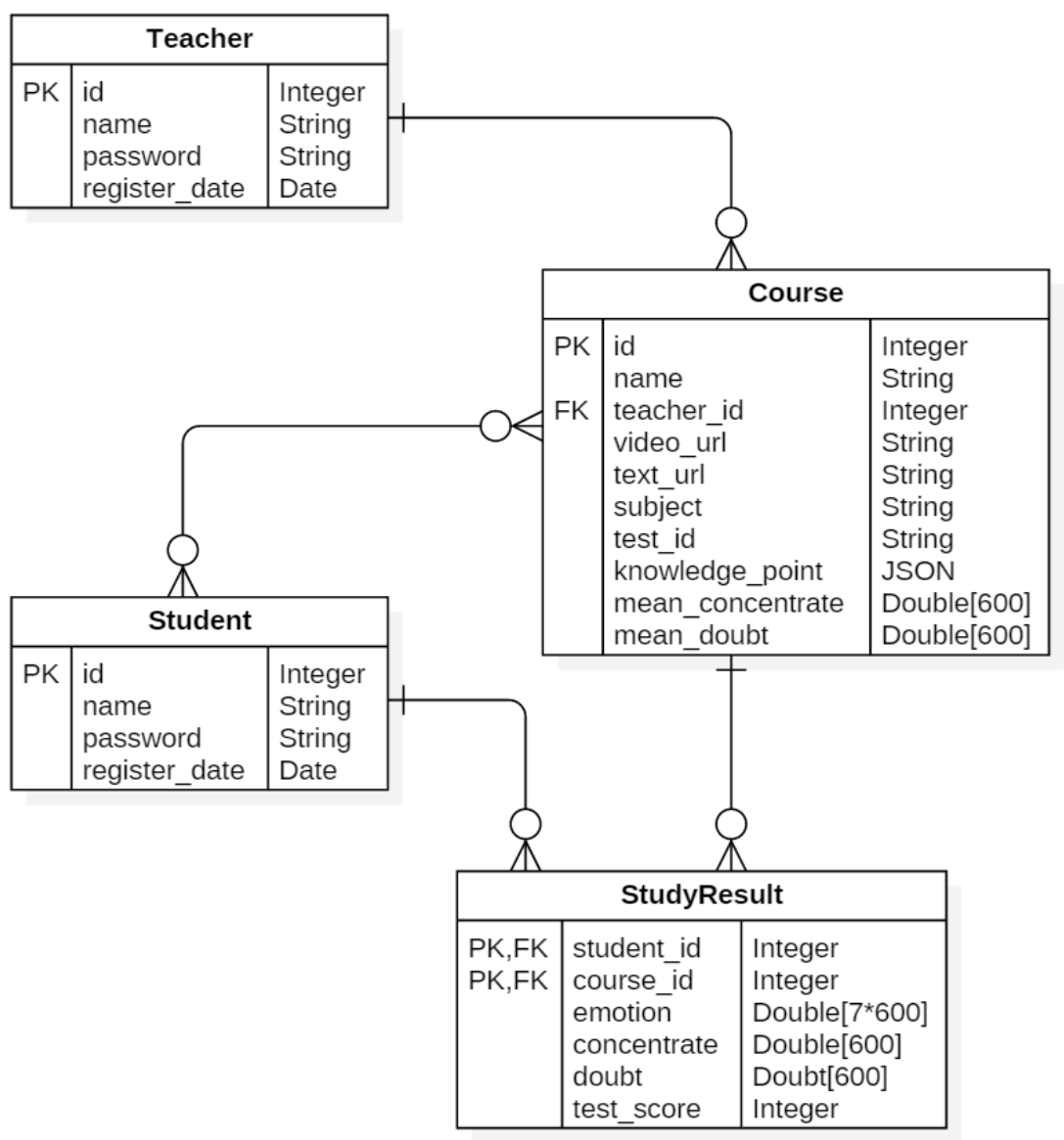


表 Teacher :

主键/外键	属性	数据类型	非空	默认值	备注
主键	id	Integer	Not Null		老师编号
	name	String	Not Null		老师账户名
	password	String	Not Null		
	register_date	Date	Not Null		注册日期

表 Student :

主键/外键	属性	数据类型	非空	默认值	备注
主键	id	Integer	Not Null		学生编号
	name	String	Not Null		学生账户名
	password	String	Not Null		
	register_date	Date	Not Null		注册日期

表 StudyResult:

主键/外键	属性	数据类型	非空	默认值	备注
主键/外键	student_id	Integer	Not Null		学生编号
主键/外键	course_id	Integer	Not Null		课程标号
	emotion	Double[7][600]		null	表情识别结果，在十分钟的视频中每秒解析并存储一个 JSON 数据
	concctrate_rate	Double[600]		null	每秒根据表情识别结果生成一个对应的专注度值
	doubt_rate	Double[600]		null	每秒根据表情识别结果生成一个对应的疑惑度值
	test_score	Integer		null	学生考试分数

表 Course :

主键/外键	属性	数据类型	非空	默认值	备注
主键	id	Integer	Not Null		课程编号
	name	String		未命名	课程名
外键	teacher_id	Integer	Not Null		创建课程的老师

	video_url	String		null	课程视频路径
	text_url	String		null	课程文档路径
	subject	String		null	课程所属科目
	test_id	Integer		null	课程测验编号，来自问卷星
	knowledge_point	JSON		null	老师在视频中标记的知识点 { time:Integer topic:String description:String }
	mean_concentrate	Double[600]		null	所有上课学生平均专注度
	mean_doubt	Double[600]		null	所有上课学生平均疑惑度

系统出错处理设计

出错信息

出现错误	系统输出信息	错误处理方式
注册时用户信息不符合规定	输出相应的错误信息	禁用注册按钮
登录时用户名和密码不匹配	提示“用户名或密码错误”	回滚之登录页面
注册用户名重名	后端返回 false	禁用注册按钮
未登录时的参加课程	返回 JSON 字符串 “haven't log in.”	
	返回 JSON 字符串 “No	

查询课程无此课程	available words. "	
查看学习情况失败	返回 JSON 字符串 “抱歉，我们无法对您的学习情况进行有效汇报.”	

补救措施

- 注册时用户信息不符合规定

系统将会在该输入右侧提示错误原因，诸如“用户名只能包括数字、英文、下划线”，“两次 输入密码不一致”等提示性语句。若不能准确填写所有注册信息，注册按钮处于被禁用状态

- 注册用户名重名

后端会返回 false，注册按钮处于被禁用状态

- 登录时用户名、密码不匹配

系统将会提示“用户名或密码填写错误”，并回滚至登录界面，要求再次输入。

系统维护设计

- 注册信息格式的判断，写于前端的 Javascript 中，进行检测。
- 注册时，判断用户名是否重名，写于 webAPP 中，通过发送 Post 请求，访问 DBAccess，从而对数据库中已存信息进行查询。
- 登录时判断用户名、密码是否匹配，写于 webAPP 中，通过发送 Post 请求，访问 DBAccess，从而对数据库中已存信息进行查询。
- 根据学生对于学习情况的反馈，优化推导学习情况的算法。