

中山大學



架构设计报告 报告

题 目	:	架构设计报告
上课时间	:	1-19周
授课教师	:	王青
姓 名	:	唐喆 贾思琪 杨锐佳 谢岳良 郑腾扬
学 号	:	20337111 20336022 20337143 20337133 20337161
组 别	:	第三组
日 期	:	2023.7.4

引言

1.1 编写目的

编写本文档主要用于说明关于人工智能(AI)教学在线评测与游戏网站的具体系统架构设计。说明对程序系统的设计考虑，包括程序系统的基本处理流程，程序系统的组织结构、模块划分、功能分配、接口设计、运行设计、数据结构设计和出错处理设计等，为程序的详细设计提供基础。本文将结合文字描述，概念图、组件图，以及关系图等来描述对网站系统的总体设计，接口设计，运行设计，系统数据结构设计，以及系统出错处理设计。本文档的预期读者是开发人员和维护人员，以及跟该项目相关的其他人员。

1.2 读者对象

系统管理者：在系统正式运行后，系统的管理者，主要用来维护系统的正常运行，保证系统数据库信息的安全，能够及时应对系统在运行时出现的一些小问题。

系统开发人员：整个系统的主要开发人员，也是系统架构和功能的主要设计人员，负责系统的web前端和后台的开发。

系统测试人员：在系统完成了初步实现后，对系统进行各方面的测试，其中包括系统的功能是否全面，系统的响应和事件的处理是否快捷，系统的安全性以及系统整体的业务流程等方面

1.3 定义

以下是说明书中会出现的各种术语的定义，以及各种简写和缩略语，以便读者阅读时进行参考。

术语、简写和缩略语	解释
Html	超文本标记语言，是一种用于创建网页的标准标记语言
CSS	层叠样式表(英文全称：Cascading Style Sheets)是一种用来表现HTML文件样式的计算机语言
JavaScript	JavaScript（简称“JS”） 是一种具有函数优先的轻量级，解释型或即时编译型的编程语言
JSON	JSON(JavaScript Object Notation, JS 对象简谱) 是一种轻量级的数据交换格式。
React	React是用于构建用户界面的JavaScript库
HTTP	HyperText Transfer Protocol，超文本传输协议
WebSocket	WebSocket是一种在单个TCP连接上进行全双工通信的协议
DDS	Data Distribution Service for Real-Time Systems，是一种物联网协议
Flask	Flask是具备的扩展性和兼容性的web开发框架
SQLite	SQLite是一个完全独立的、不要任何配置、支持SQL的、开源的文件数据库引擎
Vakuum	一个采用MVC（模型-视图-控制器）架构设计的判题系统

1.4 参考资料

[1]李伟/[吴庆海](#)，《软件架构的艺术》，电子工业出版社，2009-04

[2]邓春晖/秦映波，《Web前端开发简明教程》，人民邮电出版社，2017-12

[3]Till Adam，《架构之美》，机械工业出版社，2009-11

[4]许令波，《深入分析Java Web技术内幕》，电子工业出版社，2012-09

[5]陈康贤，《大型分布式网站架构设计与实践》，电子工业出版社，2014-09

[6][[施瓦茨 \(Baron Schwartz\)](#)](<https://book.douban.com/search/施瓦茨/>)/[扎伊采夫 \(Peter Zaitsev\)](#)/[特卡琴科 \(Vadim Tkachenko\)](#)，《高性能MySQL》，电子工业出版社，2013-05

2 系统概述

本系统以人工智能技术教学作为核心功能开展，用户可以通过网站上的指导教学来学习和实现各类统计学习、深度学习、强化学习等人工智能领域算法，并将所学的知识应用于多种有趣的游戏进行实践。项目提供给用户通过编写AI通关游戏和收集成就、多用户通过编写和优化AI进行对战和排名、基于博客社区的知识的扩展与分享等产品形态，从而帮助用户在更愉快、轻松的氛围中更深刻的理解和学习人工智能知识。

2.1 需求规定

2.1.1 性能需求

作为一体化的人工智能(AI)教学在线评测与游戏网站，为了保证学习者、教学者和开发者的用户体验，性能需求是必须得到保障的，主要的性能规定包括以下三个方面：

- 每个用户最多支持三个终端登录，超过数量应按登录时间依次弹出之前的登录。
- 在带宽和用户负荷方面，我们初期估计用户数为 10000 人，每天登录用户数为3000人左右，网络的带宽为100M。服务器应能够同时负荷预期熟练的人数同时运行。
- 在用户响应时间方面，在非高峰时间运行用户代码，应该能在 3 秒内得到运行结果，高峰时段不超过 5 秒。其他如查询学习资料等功能，在 2 秒内得到查询结果，高峰时段不超过 3 秒。

2.1.2 输入输出需求

我们定义用户的数据输入或是用户对系统的操作均视为对项目系统的输入，定义项目系统针对用户的输入进行的反馈动作为系统的输出。当用户对项目系统进行输入时，前端应根据用户所采取操作的类型，反馈给后端或其他独立组件如数据库、判题器等等，在后端根据用户的操作类型和输入信息，后端采取相应的动作，并反馈给前端，进而反馈给用户。以部分输入输出场景为例，如下所示：

- 当用户注册账号时，用户输入手机号、用户名等个人信息后，前端应将输入反馈给数据库，由数据库进行表的添加和权限的赋予，并反馈唯一、未使用过的用户id给用户，提示注册成功等信息
- 当用户请求查看课程资料时，前端应将消息发送给后端，由文件服务器通过HTML协议传输，以图片、文件、视频或者压缩包等格式反馈给用户进行查看。
- 当教学者对考试进行批改时，前端应将消息发送给后端，后端应及时的将数据写入文件，并实时的反馈给教学者显示已批改的情况。

2.1.3 灵活性需求

当本项目发生了更新或维护，该项目应该在灵活性方面加以保证，使得用户、管理人员和开发人员能够快速适用项目的变化，主要包括以下方面的规定：

- 操作方式：用户前端的主体操作方式、管理人员前端的管理操作页面、开发人员内部的操作接口不应发生较大的变迁，应保持与过去相同风格、相同模式，尽可能保证只增不删
- 运行环境：不管发生怎样的更新和维护，都应保证用户、管理人员和开发人员的运行环境不发生改变
- 其他软件接口：当项目中所使用的其他软件接口发生变化，该项目能够快速定位到变迁接口的位置，并替换为最新兼容版本的接口。

2.1.4 数据管理能力需求

不同身份的角色拥有着不同的权限，这要求特定身份的角色只能根据自己所具备的权限对自己可访问的数据进行访问、更新或删除，每个角色都不可以越权对数据进行管理，也不可以对其他角色所管理的数据进行访问和更新。以部分数据管理需求场景为例，如下所示：

- 学习者不可以对自己未加入的课程的数据进行越权访问，包括对未加入课程的资料查看、作业提交等。但可以对自己已加入的课程的资料进行查看、作业提交、加入考试等等。
- 教学者不可以对未加入课程的学习者进行管理，包括对未加入课程的学习者进行发起作业、发起考试等。但可以对已加入课程的学习者进行成绩批改，考试批改等等
- 用户不可以对他人所发表的用户博客进行修改，但可以对他人所发表的用户博客进行评论，以及对自己所发表的用户博客进行修改。

2.1.5 故障处理需求

当用户使用本项目发生系统故障时，该项目应该能够迅速定位到错误发生的问题和类型，明确提示错误信息，指导用户如何从系统故障中恢复，如若无法恢复，该项目应该能够快速自我定位和修复，并提示用户刷新或重新登入。以部分常见故障类型和故障处理为例，如下所示：

- 当物理服务器发生硬件故障，应当紧急关闭项目网站，提示所有在线用户下线通知，保存用户数据。同时系统监控程序应能发现故障发生，传递消息给开发人员，开发人员进行硬件维修后，做好数据恢复工作，并提示用户维护完成。
- 当软件接口发生故障时，以评测系统评测软件进程崩溃为例。监控程序应迅速定位发生故障所在，提示所有用户发生故障，等待重启，并自动重启服务端进程，在重启进程恢复服务后，应提示用户进行浏览器刷新
- 当用户操作发生异常时，以学习者搜索不存在的课程为例。当对数据库访问发现不存在该搜索课程时，应该由前端提示给用户该课程不存在，提示用户重新搜索。

3 架构设计目标和约束

3.1 架构设计目标

架构设计的主要目的是为了解决软件系统复杂度带来的问题。

架构设计是宏观性考量，在整体上理解问题的复杂性，给出方案，并论证方案的可行性，提供一系列准则指导执行。

因此，架构设计的目标主要有以下几点：

- (1) 提前识别问题的复杂性和关注点，提供可行的经过论证的解决方案；
- (2) 建立服务质量指标，确定设计方案可以满足指定的质量指标；
- (3) 规划整体设计，提供长远的可扩展性；

(4) 为实际开发确定基本框架。

3.2 约束需求

3.2.1 软件、硬件环境（包括运行环境和开发环境）的约束

软件开发和运行是基于64位Windows操作系统，所以软件可在64位Windows操作系统上的网页浏览器中正常打开和运行。采用WebStorm-2020完成Web前端的开发，Eclipse IDE for Java Developers-2020完成后端开发，MySQL 8.0进行数据库的设计和编辑，最终在Windows 10操作系统上的Google Chrome浏览器完成系统的测试和最终运行。

3.2.2 接口/协议的约束

(1) 内部接口/协议的约束

在内部接口方面，各模块之间采用函数调用、参数传递、返回值等方式进行信息传递。接口传递的信息将是数据结构封装的数据，以参数传递或返回值的形式在各模块间传输。

(2) 外部接口/协议的约束

在输入方面，对于键盘、鼠标等外部设备的输入，可用VISUAL C++的标准输入/输出，对输入进行处理。

在输出方面，如打印机的连接及使用，也可用VISUAL C++的标准输入/输出对其进行处理。在网络传输部分和网络硬件部分，为了实现高速传输，将使用高速ATM。

3.2.3 用户界面的约束

UI的设计开发基于html5、css3和JavaScript等前端开发语言。在UI的设计中要具备适用性，易于用户理解；当错误操作时会有提示信息和防错处理，整体的布局和色彩的适用要合理，在事件的响应方面要最高效率，整体界面可复用，控件设置满足单一应对原则。网络通信协议采用 HTTP和TCP/IP协议，调用接口采用 Web Service，系统通过内嵌浏览器技术与扩展对象插件接口集成。

3.2.4 软件质量的约束

(1) 正确性

UI上的控件能够正确且及时响应，实现该控件对应功能，错误操作会进行信息提示，同时添加防错处理，及时应对出现的错误操作。

(2) 可靠性

设置备用服务器保证系统正常工作。当遇到断电重启的情况，能够实现系统断电后的自动重启，且较少人工维护量和实现系统稳定运行。

(3) 效率

数据信息的采集通讯支持10秒以上的实时数据传输（受网络环境影响），并且要求系统的响应时间不超过5秒。

(4) 安全性

用户注册/登录系统，关键数据操作，需要验证码进行验证，关于数据采用MD5加密技术。

(5) 可扩展性

系统在整体设计上要充分考虑可扩展性，当用户的需求变化时，系统可进行相应的处理；当系统在运行中，客户如果对原有功能作调整或变更，系统应该能够灵活地修改。同时，预留接口，能扩展到其他监测系统。

3.3 运行环境

3.3.1 服务端运行环境

我们采用了如下配置的服务器作为我们的项目搭建服务器，通过系统所支持的指令集作为基础进行优化，例如采取并行向量化AVX-512编程作为软件加速，同时V100-SXM2型号GPU能够提供高可靠的强大算力，满足了我们高并发、低延迟的用户代码评测的需求，而Intel的最新Xeon Gold 6132型CPU满足我们对搭建大量数据库、文件服务器和Web页面的支撑；最后我们整体的硬件框架都基于市场主流硬件，在软件层次上几乎支持所有的常见主流协议。

- CPU：Intel(R) Xeon(R) Gold 6133 CPU @ 2.50GHz
- GPU：NVIDIA A100 40GB
- RAM：2GB
- Hard disk: 100G SSD SATA
- HCA card：InfiniBand Mellanox ConnectX-6 HDR card
- 接口类型：Serial ATA
- 接口速率：Serial ATA 300
- 电源类型：后备式 UPS
- 操作系统：Centos 7
- 指令集：MMX/SSE/SSE2/SSE3/SSE4.1/SSE4.2/SSSE3/Sup-SSE3/EM64T/AVX/AVX2/AVX-512/IMCI

3.3.2 客户端运行环境

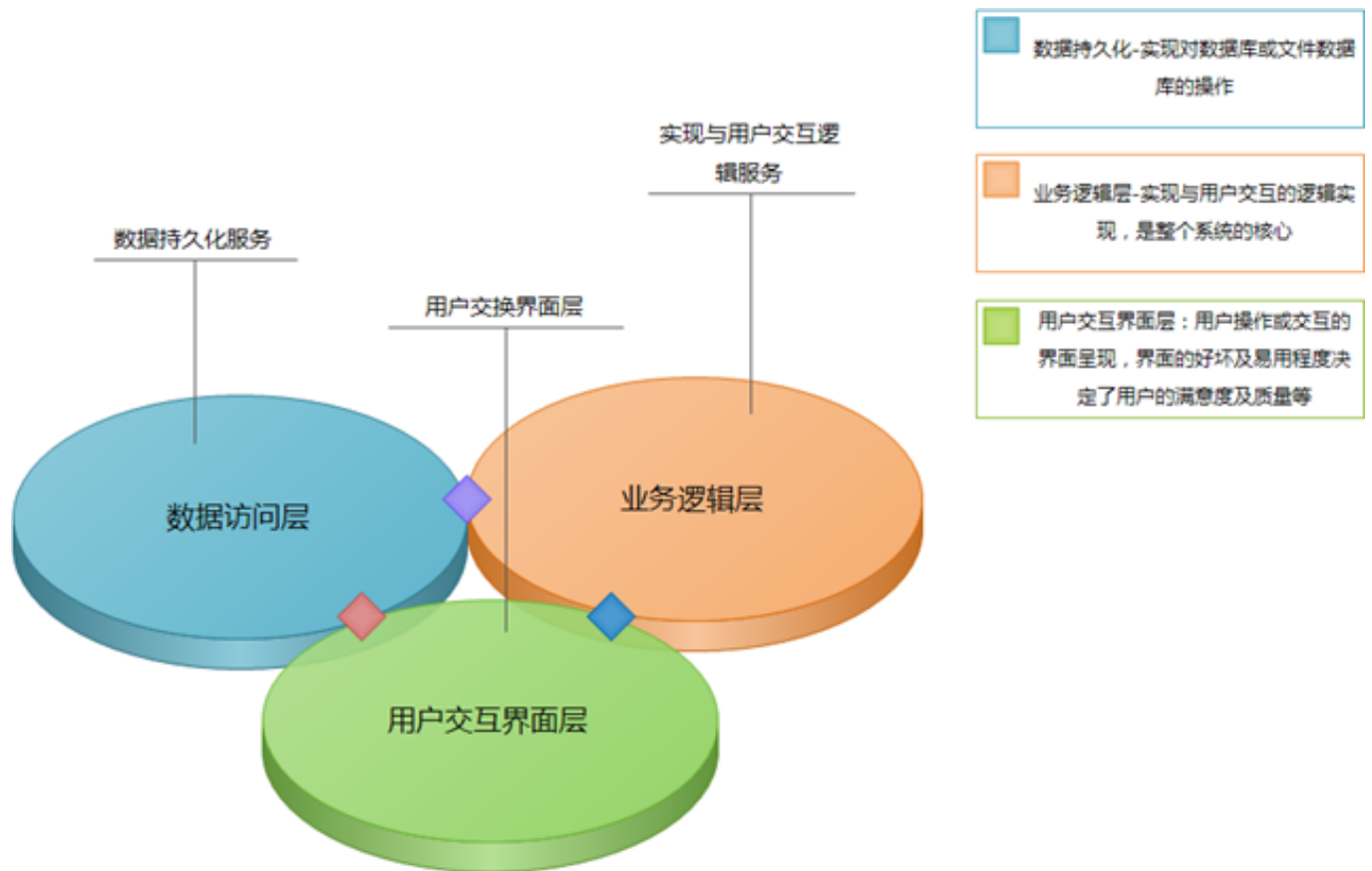
而对于用户来说，无需特别限定硬件配置，主流笔记本电脑配置以及主流浏览器即可完成对本项目的访问、课程的学习和代码的编写。下面我们给出最低电脑配置作为示例：

- CPU：Intel Core(TM) 2 Duo CPU E4600 @ 2.40GHZ
- GPU：512MB显存及以上
- RAM：2GB
- Hard disk: 5GB及以上(空闲)
- 操作系统：Window XP、Windows 7/8/9/10、Windows 2003 Server、Vista等
- 浏览器：Microsoft Edge、Chrome、QQ浏览器、火狐浏览器等

4 系统总体设计

4.1 设计思想

三层架构(3-tier architecture) 通常意义上的三层架构就是将整个业务应用划分为：界面层（User Interface layer）、业务逻辑层（Business Logic Layer）、数据访问层（Data access layer）。区分层次的目的即为了“[高内聚低耦合](#)”的思想。3个层次中，系统主要功能和业务逻辑都在业务逻辑层进行处理。



所谓三层体系结构，是在客户端与数据库之间加入了一个“中间层”，也叫组件层。这里所说的三层体系，不是指物理上的三层，不是简单地放置三台机器就是三层体系结构，也不仅仅有B/S应用才是三层体系结构，三层是指逻辑上的三层，即使这三个层放置到一台机器上。三层体系的应用程序将业务规则、数据访问、合法性校验等工作放到了中间层进行处理。通常情况下，客户端不直接与数据库进行交互，而是通过COM/DCOM通讯与中间层建立连接，再经由中间层与数据库进行交互。

4.2 软件逻辑架构设计

在基本设计上，本系统由采取了3-tier的设计模式，让不同的功能模块相互分离，彼此相互协作工作，使得结构清晰明了，功能独立性强，分工明确，层次清晰，进而使得整个系统的可扩展性强，这提高了整个业务系统的开发质量和开发效率，为日后的更新和维护都提供了帮助。

按照3-tier设计，一个AI教学在线评测与游戏网站的系统架构可以分为以下三个层次：

1. 表现层（Presentation Tier）：负责处理用户界面和交互，包括用户的注册、登录、个人信息管理、教学评测页面、游戏页面等。该层通过Web浏览器或客户端应用程序与用户进行交互。
2. 业务逻辑层（Business Logic Tier）：负责处理应用程序的业务逻辑，包括人工智能教学评测模块、人工智能游戏模块、通知模块、支付模块等。该层处理应用程序的核心业务逻辑，与表现层和数据访问层进行通信。
3. 数据访问层（Data Access Tier）：负责处理数据的读、写、更新等操作，包括数据库模块和API模块。该层与业务逻辑层进行通信，提供数据的持久化和查询功能。

每个层次的模块可以根据实际需求和技术实现能力进行调整和扩展。例如，表现层可以使用Web技术（如HTML、CSS、JavaScript）或客户端应用程序（如Android、iOS等）来实现，数据访问层可以使用关系数据库（如MySQL、PostgreSQL等）或NoSQL数据库（如MongoDB、Cassandra等）来实现，业务逻辑层可以使用Java、Python、C#等编程语言来实现。

本系统的所有运行模块包括：

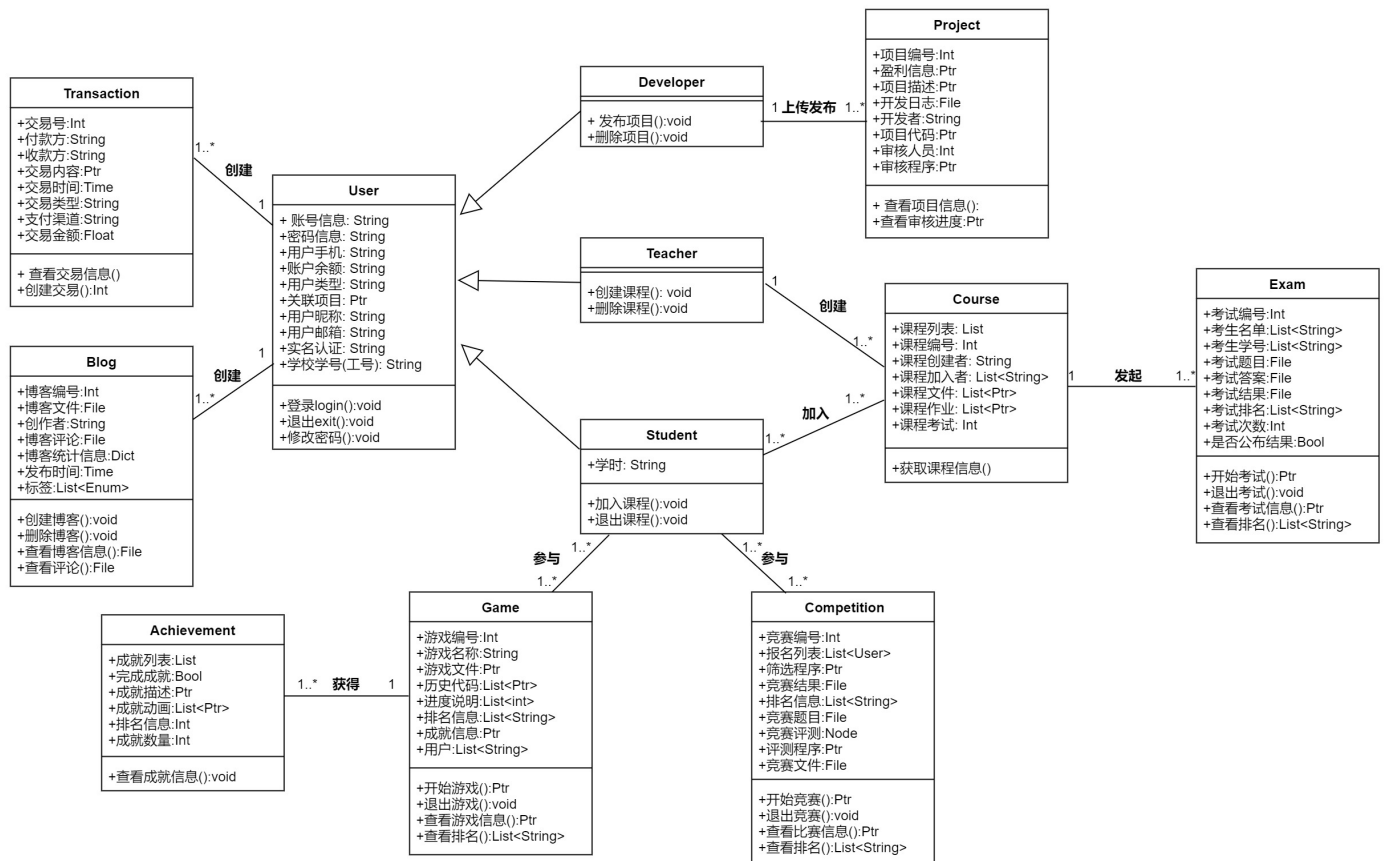
- 展示层：游戏音像渲染引擎、Web渲染引擎
- 业务逻辑层：
 - 服务组件：练习模块、教学模块、博客模块、开发者模块、竞赛模块、管理模块。

- 功能组件：评测系统模块、游戏执行引擎、博客系统、数据展示模块、匹配系统、通知模块、用户管理模块
- 基础组件：安全验证模块、错误处理模块、HTTP服务器
- 持久化层：
 - 持久化模块：提供对数据库的存取接口。

5 系统数据结构设计

5.1 逻辑结构设计要点

下面给出本软件系统所使用的各个数据库的记录类型以及字段，它们之间的相互关系如下：



5.1.1 用户信息数据库

用户信息数据库用来存储每个用户的最基本身份信息，包括其个人资料、账号密码信息等等，用来帮助系统对其进行身份识别、账号登入、权限修改等等。在用户数据库中，每条记录维护了一个用户的个人信息，其中每种个人信息设定为字符串类型，每条记录设定为由多个字符串组合的复合结构体类型。

具体的，数据库的每条记录组成字段如下：

字段说明	类型	长度	字段说明	类型	长度
账号信息	String	50	用户昵称	String	50
密码信息	String	50	用户邮箱	String	50
用户手机	String	50	实名认证	String	50
账户余额	String	50	学校	String	50
用户类型	String	20	学号(工号)	String	20
关联项目	Ptr	8			

5.1.2 课程数据库

课程数据库维护了每个用户所加入或创建的课程信息，并对关键的数据进行了保存和维护。每条记录代表一个用户所关联的课程信息。每条记录由复合结构体类型组成。

具体的，数据库的每条记录组成字段如下：

字段说明	类型	长度	字段说明	类型	长度
课程列表	List	50	课程考试	Int	4
课程编号	Int	4			
课程创建者	String	50			
课程加入者	List	200			
课程文件	List	500			
课程作业	List	500			

5.1.3 考试数据库

考试数据库维护了考试功能所关联的全部数据信息。每条记录代表一场考试的信息，由复合结构体类型组成。

具体的，数据库的每条记录组成字段如下：

字段说明	类型	长度	字段说明	类型	长度
考试编号	Int	4	考试排名	List	200
考生名单	List	200	考试次数	Int	4
考生学号	List	200	是否公布结果	bool	1
考试题目	File	文件长度			
考试答案	File	文件长度			
考试结果	File	文件长度			

5.1.4 游戏数据库

游戏数据库记录了所有游戏的相关数据，以及包括各个参与用户的历史代码、游戏排名等等。每条记录代表一个游戏的信息。每条记录由复合结构体类型组成。

具体的，数据库的每条记录组成字段如下：

字段说明	类型	长度	字段说明	类型	长度
游戏编号	Int	4	排名信息	List	1000
游戏名称	String	50	成就信息	Ptr	8
游戏文件	Ptr	8	用户	List	1000
历史代码	List	500			
进度说明	List	500			

5.1.5 成就数据库

成就数据库记录了每个用户在项目学习过程中所获得的成就信息。每条记录代表一个用户所完成、未完成的全部成就系统所涉及的数据。每条记录由复合结构体类型组成。

具体的，数据库的每条记录组成字段如下：

字段说明	类型	长度	字段说明	类型	长度
成就列表	List	500	成就数量	Int	4
完成成就	Bool	1			
成就描述	Ptr	8			
成就动画	List	500			
排名信息	Int	4			

5.1.6 竞赛数据库

竞赛数据库记录了所有竞赛所涉及的用户信息、排名信息等等。每条记录代表一场竞赛从竞赛报名、竞赛途中到竞赛结束、结果公开所产生的全部数据信息。每条记录由复合结构体类型组成。

具体的，数据库的每条记录组成字段如下：

字段说明	类型	长度	字段说明	类型	长度
竞赛编号	Int	4	竞赛题目	File	文件长度
报名列表	List	200	评测程序	Ptr	8
筛选程序	Ptr	8	竞赛文件	File	文件长度
竞赛结果	File	文件长度			
排名信息	List	200			

5.1.7 博客数据库

博客数据库记录了用户发布的每条博客的相关信息。每条记录代表一篇博客的所有相关数据。每条记录由复合结构体类型组成。

具体的，数据库的每条记录组成字段如下：

字段说明	类型	长度	字段说明	类型	长度
博客编号	int	4	发布时间	Time	8
博客文件	File	文件长度	标签	List	50
创作者	String	50			
博客评论	File	文件长度			
博客统计信息	Dict	100			

5.1.8 项目数据库

项目数据库记录了每个开发者所提交、开发的项目信息。每条记录代表着一个项目所涉及的全部信息。每条记录由复合结构体类型组成。

具体的，数据库的每条记录组成字段如下：

字段说明	类型	长度	字段说明	类型	长度
项目编号	int	4	项目代码	Ptr	8
盈利信息	Ptr	8	审核人员	Int	4
项目描述	Ptr	8	审核程序	Ptr	8
开发日志	File	文件长度			
开发者	String	50			

5.1.9 交易数据库

交易数据库记录了用户和平台每次的交易，每条交易均可为用户与平台的交易、用户与开发者之间的交易、平台与开发者之间的交易。每条记录维护了单条交易信息，其类型为复合结构体类型。

具体的，数据库的每条记录组成字段如下：

字段说明	类型	长度	字段说明	类型	长度
交易号	Int	4	交易类型	String	20
付款方	String	50	支付渠道	String	20
收款方	String	50	交易金额	Float	4
交易内容	Ptr	8			
交易时间	Time	8			

5.2 物理结构设计要点

下面给出每个数据库的物理结构设计要点。

	存储要求	访问方法	存取单位	索引	设备	保密级别
用户信息数据库	关系型数据库	sqlite3	记录	聚集索引	云服务器	强
课程数据库	关系型数据库	sqlite3	记录	聚集索引	云服务器	强
考试数据库	文档型数据库	sqlite3	记录	聚集索引	云服务器	强
游戏数据库	关系型数据库	sqlite3	记录	聚集索引	云服务器	强
成就数据库	关系型数据库	sqlite3	记录	无	云服务器	弱
竞赛数据库	文档型数据库	sqlite3	记录	非聚集索引	云服务器	弱
博客数据库	文档型数据库	sqlite3	记录	非聚集索引	云服务器	强
项目数据库	文档型数据库	sqlite3	记录	非聚集索引	云服务器	强
交易数据库	关系型数据库	sqlite3	记录	非聚集索引	云服务器	强

5.3 数据结构与程序的关系

下面给出各个数据结构与访问这些数据结构的各个程序之间的对应关系。

	课程模块	博客模块	游戏模块	教学模块	盈利模块	开发模块
用户信息数据库	√	√	√	√	√	√
课程数据库	√			√		
考试数据库	√					
游戏数据库			√			
成就数据库			√			
竞赛数据库			√			
博客数据库		√				
项目数据库						√
交易数据库					√	

6 系统出错处理设计

6.1 出错信息

出错/故障情况	系统输出信息形式	处理方式
用户注册/登录信息错误	userError	提示注册/登录失败，请重新输入
用户权限错误	permissionError	提示用户无权访问/修改相关内容
数据库异常	dataError	提示用户当前数据库异常，需要检查维修，请稍后重试
网络异常	networkEtrror	提示用户现在网络环境不稳定，请稍后重试
服务器异常	serverError	提示用户当前服务器出错，需要检查维修，请稍后重试

6.2 补救措施

1. 后备技术：利用云服务商提供的创建镜像功能，周期性地保存系统状态，一旦系统出错则读取最近的保存，降低损失和影响。
2. 降效技术：人工审查用户代码异常运行日志以及项目平台的异常运行记录。
3. 恢复及再启动技术：让多件事务做绑定，一旦其中一个撤销，其余所有相关的事务也要撤销。若事务已经写入数据库，其余相关的事务也要重做。采用投票机制进行恢复：多个组件或子系统采用相同或不同地算法计算同一件事，如果各自结果不一致，则少数服从多数。

6.3 系统维护设计

1. 定时检查模块

一方面，通过自动的方式，定时扫描服务器的内存使用情况、工作负载和硬盘占用情况，以及数据库的一致性，从而可以预防异常情况的发生；另一方面，通过人工的方式，定期修复用户报告的bug，从而不断提高系统的鲁棒性。
2. 日志模块

系统运行的过程中，会有大量的用户操作，大量的数据更新以及产生大量的输出，通过记录这些内容，在出错时可以快速恢复到出错前的状态，同时也可以通过分析这些数据，预防可能发生的错误，及时地解决问题。

3. 运行数据分析模块

如前所述，系统运行的过程中，会产生大量的数据，同时也有大量的用户行为记录。一方面，可以通过一些PL技术自动分析出系统运行是否正常；另一方面，也可以通过ml技术分析出用户的行为模式以及偏好，从而为下一个版本的迭代提供基础。