项目名称：tzlOJ

文档名称：用户手册

*HUSTZL*

*侯皓斐 软件2003班 U202010851*

*刘铭宸 软件2003班 U202010783*

*甘凤轩 软件2003班 U202010822*

版本：V2.1

文 档 信 息

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√ ] 正式发布  [ ] 正在修改 | 文件标识： | HUSTZL\_010 |
| 当前版本： | V2.1 |
| 作者： | 甘凤轩 |
| 发布日期： | < 2023.4.23 > |

文 档 更 改 记 录

| 版本 | 更改日期 | 更改人 | 更改原因 | 说明 |
| --- | --- | --- | --- | --- |
| V1.0 | 2023.4.15 | 甘凤轩 |  | 初版 |
| V2.0 | 2023.4.20 | 甘凤轩 | 校对 | 勘误并调整行文语句 |
| V2.1 | 2023.4.23 | 甘凤轩 | 调整格式 | 完善文档大纲，调整段落格式，修改序列编号 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1.用途 1](#_Toc133031644)

[1.1功能 1](#_Toc133031645)

[1.2质量 2](#_Toc133031646)

[1.2.1精度 2](#_Toc133031647)

[1.2.2时间特性 2](#_Toc133031648)

[1.2.3灵活性 3](#_Toc133031649)

[1.3安全保密 4](#_Toc133031650)

[1.3.1用户身份验证和权限控制 4](#_Toc133031651)

[1.3.2数据安全和加密 4](#_Toc133031652)

[1.3.3安全审计和日志记录 4](#_Toc133031653)

[1.3.4备份和恢复 4](#_Toc133031654)

[2.运行环境 4](#_Toc133031655)

[2.1硬件和网络 4](#_Toc133031656)

[2.2支持软件 4](#_Toc133031657)

[3.软件系统的使用 5](#_Toc133031658)

[3.1安装与初始化 5](#_Toc133031659)

[3.1.1Linux环境： 5](#_Toc133031660)

[3.1.2Windows环境 5](#_Toc133031661)

[3.2输入 5](#_Toc133031662)

[3.2.1输入数据的现实背景 5](#_Toc133031663)

[3.2.2输入格式 6](#_Toc133031664)

[3.3输出 8](#_Toc133031665)

[3.3.1输出数据的现实背景 8](#_Toc133031666)

[3.3.2输出格式 8](#_Toc133031667)

[3.4出错处理和恢复 9](#_Toc133031668)

[3.4.1如何备份 9](#_Toc133031669)

[3.4.2恢复备份 9](#_Toc133031670)

# 1.用途

## 1.1功能

tzlOJ系统旨在为华中科技大学软件学院的教师和学生提供一个在线编程评测平台，以帮助提升编程技能和竞赛能力。学生可以注册账号并登录该系统，同时老师拥有高权限的管理员账号来进行一系列管理操作，并且该平台将提供一系列可以搜索、筛查的编程题目和自动评测系统，用户可以在平台上提交自己的代码，系统将自动评测代码的正确性和效率，并给出相应的评分和反馈。本项目支持应该支持多种竞赛模式，支持创建比赛并创建比赛的题单供用户参与并答题，在赛后用户还可以查看自己的排名。

**用户功能表：**

|  |  |
| --- | --- |
| 功能 | 说明 |
| 登录与注册 | 用户可以通过输入用户名和密码来登录，如果没有账号，可以通过注册页面内创建新账号。 |
| 题目列表 | 系统展示所有可用编程题目列表，用户可输入搜索条件来对题目进行筛选和排序。 |
| 题目详情页 | 用户可以查看题目的详情页，包括题目描述、输入输出样例、评测结果和题目提交历史等信息。 |
| 代码编辑器 | 系统提供一个代码编辑器，用户可在其中编辑和提交自己的代码。编辑其支持多种语言 ，并提供自动缩进、语法高亮、自动补全等功能。 |
| 评测结果查看 | 用户可以查看自己提交的代码的评测结果，包括编译错误、运行错误、通过测试点的数量等信息。 |
| 个人信息修改 | 用户可以查看和编辑自己的个人信息，包括用户名、密码、邮箱等信息。 |
| 参加比赛 | 用户可以通过比赛页面查看和参加系统中的比赛。比赛页面应该展示比赛的基本信息、排名、比赛规则和题目列表等信息。 |
| 排行榜 | 用户可以通过积累积分和参加比赛等活动来提高自己的排名。排行榜页面应该展示用户的排名和积分等信息。 |

**管理员功能表：**

|  |  |
| --- | --- |
| 管理用户 | 管理员可以导入、删除、增添、修改用户信息，并拥有对用户进行封禁的权限。 |
| 管理题目 | 管理员可以修改、增添、删除题目的信息，如类型、描述、输入输出样例、提交历史等，同时还可以增添新题目或删除旧题目。 |
| 管理比赛 | 管理员可以创建比赛并设置或修改比赛的相关信息，如时间、题目列表、相关规则等 |

## 1.2质量

### 1.2.1精度

本系统的密码可以为数字与字母的组合，最低不能少于10位，最高不能超过32位。

### 1.2.2时间特性

该系统的时间特性对于用户体验至关重要。下面是几个需要考虑的关键时间特性：

1. 响应时间：响应时间指用户提交请求后系统返回响应的时间。这对于用户来说是最直接的感受，因为它关系到用户等待时间的长短，从而影响用户对系统的印象和满意度。为了确保良好的用户体验，平均响应时间应在200毫秒以内，且有百分之九十的请求响应时间在该数值以下，同时最大响应时间应该在1秒以内。
2. 更新处理时间：更新处理时间是指当用户进行某项操作时，系统需要将这些更改同步到后端数据库并处理，这通常需要一定的时间。如果系统需要更新的数据量很大，更新处理时间可能会很长，这会影响用户操作体验。为了确保及时响应用户操作，平均处理时间应在200毫秒以内，且有百分之九十的请求响应时间在该数值以下，同时最大响应时间应该在1秒以内。
3. 数据传输与转换时间：数据传输与转换时间是指在系统内部或与外部系统之间传输数据的时间。如果数据量很大或者需要进行复杂的转换，数据传输与转换时间可能会很长。为了确保系统的数据交换效率，平均传输时间应在50毫秒以内，且有百分之九十的请求响应时间在该数值以下，同时最大响应时间应该在0.8秒以内。
4. 计算时间：计算时间是指系统执行计算所需的时间，例如评测程序的运行时间。评测程序是 OJ 系统的核心功能之一，如果评测时间过长，会直接影响用户的体验。为了保证系统的快速评测能力，平均计算时间应在200毫秒以内，且有百分之九十的请求响应时间在该数值以下，同时最大响应时间应该在2秒以内。

### 1.2.3灵活性

为了确保OJ系统的灵活性，应该采取以下措施：

1. 运行环境：OJ系统应该支持多种操作系统和网络环境。无论用户是在Windows、macOS、Linux或移动设备上访问系统，都应该能够无缝使用。
2. 结果精度：OJ系统应该提供不同的输出选项，以满足用户对结果精度的不同要求。例如，对于数值计算问题，可以提供不同的精度选项；对于字符串匹配问题，可以提供不同的匹配算法和精度级别。
3. 时间特性：OJ系统应该优化性能，确保系统具有快速的响应时间和处理能力。同时，系统应该能够根据用户的需求进行灵活的调整，例如提高服务器的硬件配置，采用分布式计算等方式来提高系统的性能。
4. 其他要求：OJ系统应该具有良好的扩展性和可定制性，以满足用户的不同需求。例如，可以提供API接口，以便用户将系统集成到自己的应用程序中。同时，系统也应该支持第三方插件和扩展，以增强系统的功能和灵活性。

通过以上措施，OJ系统可以在不同的用户需求变化时保持灵活性，并能够快速适应不同的运行环境、结果精度和时间特性等需求。

## 1.3安全保密

### 1.3.1用户身份验证和权限控制

本系统提供强大的用户身份验证和权限控制功能，以确保只有经过身份验证的用户才能访问和操作系统的各项功能，同时可以根据用户的权限来控制其访问系统的不同部分和功能。例如，管理员可以访问系统的全部功能，而普通用户只能访问特定的功能。

### 1.3.2数据安全和加密

为了保护系统的数据安全，本系统采用多种安全措施，例如对用户密码进行加密存储、使用SSL加密传输数据等。在对用户输入进行处理和计算时，采用安全的算法和方法，以避免黑客攻击和数据泄漏。

### 1.3.3安全审计和日志记录

为了检测潜在的安全问题并保证数据安全，本系统提供安全审计和日志记录功能，可以追踪所有的系统访问记录、用户操作记录和异常事件记录，并存储在安全的地方，以便进行日后的审计和分析。

### 1.3.4备份和恢复

为保证数据安全，需要定期备份，使用数据库导出sql文件备份脚本（OJD目录中的data文件夹是系统的所有的数据，包括日志、数据库、测试用例、上传的文件等）。恢复数据时，将备份文件进行覆盖即可。

# 2.运行环境

## 2.1硬件和网络

无特殊硬件设备配置要求，普通计算机即可。

## 2.2支持软件

支持的操作系统；Linux或Win10x64

多语言支持：C, C++, Java, Python2, Python3，题目可以选择使用的语言

其他支持软件：Docker

# 3.软件系统的使用

## 3.1安装与初始化

对于不同的操作系统（Linux或Windows），本系统提供了不同的安装方式，具体如下。

### 3.1.1Linux环境：

#### （1）安装必要的依赖

sudo apt-get update && sudo apt-get install -y vim python3-pip curl git

pip3 install --upgrade pip

pip install docker-compose

#### （2）安装Docker

国内用户使用脚本一键安装: sudo curl -sSL https://get.daocloud.io/docker | sh  
国外用户使用脚本一键安装: sudo curl -sSL get.docker.com | sh

详细步骤参照： <https://docs.docker.com/install/>

### 3.1.2Windows环境

仅适用于Win10 x64下的PowerShell

* + 1. 安装 Windows 的Docker 工具
    2. 右击右下角Docker 图标，选择Settings 进行设置
    3. 选择Shared Drives 菜单，之后勾选你想安装 OJ 的盘符位置（例如勾选D盘），点击Apply
    4. 输入Windows 的账号密码进行文件共享
    5. 安装Python、pip、git、docker-compose

## 3.2输入

### 3.2.1输入数据的现实背景

根据系统的不同功能，输入数据的显示背景会有所不同：

1. 用户管理：输入数据包括用户账号、密码、个人信息等，场景为人员变动，来源为管理员和用户本人，出现频率较低。
2. 在线评测：输入数据为程序代码和测试数据，场景为学生提交代码、教师出题，出现频率较高且不规律。
3. 题库管理：输入数据为题目信息、答案、解析等，场景为教师出题、修改题目信息，出现频率较低。
4. 试题分类：输入数据为题目信息、题目分类、难度等级等，场景为教师进行试题分类，出现频率较低。
5. 排行榜：输入数据为用户提交记录、通过记录等，需要对访问这些数据进行限制，仅管理员可以查看所有数据，用户只能查看自己的数据。
6. 举行比赛：输入数据包括比赛名称、比赛时间、比赛规则、比赛题目等，场景为教师或管理员创建比赛，出现频率较低。

对于所有输入数据，需要对其合理性进行检验，如果发现错误，应该建立相应的记录。输入数据的处理要求可以根据具体情况来定，对于不符合要求的数据，可以根据系统设计来保留或废弃。例如，在线评测中，如果输入的代码无法编译或者执行结果不符合要求，可以废弃该代码，并记录相关信息。

### 3.2.2输入格式

针对该系统的功能，以下是对初始输入数据和参量的格式要求：

1. 用户管理：
   1. 长度：用户名和密码长度必须在规定范围内，范围为6-20个字符。
   2. 格式基准：用户名和密码的基准应该是左边缘。
   3. 标号：每个用户必须有唯一的标识符——用户ID。
   4. 顺序：用户ID和用户名应该按照一定的顺序排列。
   5. 标点：用户名和密码中不允许使用空格、斜线等标点符号。
   6. 词汇表：对于用户名和密码，只允许使用字母和数字，不允许使用其他字符。
   7. 省略和重复：用户名和密码不能为空，不能重复。
2. 在线评测：
   1. 长度：代码长度必须在规定范围内，不超过10000个字符。
   2. 格式基准：代码应该是以左边缘为基准。
   3. 顺序：代码应该按照提交的顺序排列。
   4. 词汇表：代码只允许使用指定的编程语言关键字和函数名等。
   5. 省略和重复：代码不能为空，且不能重复提交同一份代码。
   6. 控制：需要输入命令来提交代码和运行测试用例。
3. 题库管理和试题分类：
   1. 长度：每道题目的标题和描述长度应在规定范围内，不超过500个字符。
   2. 格式基准：题目标题和描述应以左边缘为基准。
   3. 标号：每个试题需要一个唯一的标识符——题目ID。
   4. 顺序：试题应该按照一定的规则排列，可按照题目难度或类型进行分类。
   5. 标点：题目描述中不允许使用非法字符。
   6. 词汇表：题目描述中只允许使用指定的单词和术语。
   7. 省略和重复：题目标题和描述不能为空，且不能重复。
   8. 控制：需要输入命令来创建、修改或删除试题。题目的标号和标识符需要按照一定的规则来命名，例如采用阿拉伯数字和英文字母的组合，避免重复或混淆。
4. 排行榜：
   1. 长度：输入数据行数或项数需符合预设要求，限定每个用户的信息只能占用一行，每行信息的字符数不能超过20个字符。
   2. 格式基准：每个用户信息的排版格式应按照预设规则进行。
   3. 标号：对于每个用户有唯一的标识符——用户ID。
   4. 顺序：每个用户信息的数据项次序按照预设规则进行，按照姓名、学号、得分等依次排列。
   5. 标点：用户信息之间使用用逗号或制表符隔开不同的数据项，用换行符分割不同的用户信息。
   6. 词汇表：输入数据中允许出现的字符组合应该符合预设的规则，禁止出现特殊字符，同时限制输入数据中只能使用数字和字母等等。
   7. 省略：输入数据中某些数据项可以省略或重复，用户信息中可以省略一些可选的数据项。
   8. 控制：输入数据中的控制信息应该按照预设规则进行，每个输入文件中只包含一组用户信息，同时控制输入数据的开始和结束位置。
5. 举行比赛：
   1. 长度：应根据比赛规则和需要输入的信息而定，选手信息需要输入姓名、学号、所属学院、比赛时间等信息，不同信息的长度也不同。
   2. 格式基准：以左对齐为基准。
   3. 标号：对于每个用户和每场比赛有唯一的标识符——用户ID和比赛编号。
   4. 顺序：选手信息按照姓名、学号、所属学院的顺序输入，比赛信息按照比赛名称、时间、地点的顺序输入。
   5. 标点：使用空格、斜线、星号、冒号等符号进行分隔。
   6. 省略和重复：使用特定的符号来表示某些信息的省略或重复，例如使用“-”来表示选手信息中某一项为空。

## 3.3输出

### 3.3.1输出数据的现实背景

1. 使用：输出数据是为用户和管理员服务的，用于不同的目的。用户可以查看他们的考试成绩、排行榜信息和课程表等，管理员可以使用输出数据来管理用户信息、题库信息和比赛信息等。
2. 输出频率：输出频率会因不同的功能而异。排行榜信息和比赛信息可能会每天更新，而用户信息可能只会每周更新一次。
3. 输出方式：输出方式也因输出数据的类型而异。排行榜和考试成绩会在网站上直接显示，而用户个人信息会被写入数据库中。
4. 质量管理：输出数据的质量对于用户体验至关重要。因此，系统有一些合理性检验和出错纠正的规定，以确保输出数据的准确性和可靠性。
5. 处理：处理输出数据的方式也很重要。一些输出数据需要保留，用户个人信息和课程表信息，而一些临时数据可以被废弃。

### 3.3.2输出格式

1. 首部：输出数据的标识符通常用来标识输出的数据类型，以便用户能够快速识别数据。输出日期可以显示数据生成或更新的日期和时间。输出编号可以用来跟踪数据的历史和变化。
2. 主体：输出信息的主体是包含实际数据或信息的部分。在一个OJ系统中，主体输出信息可以包括题目描述，测试用例，代码提交结果等。
3. 尾部：尾部是输出信息的最后部分，通常包括累计总数和结束标记。累计总数可以用于告诉用户输出信息的数量或统计数据。结束标记可以用于标识输出信息的结束，以便用户知道何时数据到达了末尾。

## 3.4出错处理和恢复

### 3.4.1如何备份

为保证数据安全，需要定期备份，OnlineJudgeDeploy 目录中的 data 文件夹是系统的所有的数据，包括日志，数据库，测试用例，上传的文件等，其中需要备份的数据为 backend/public，backend/test\_case 两个目录。

对于数据库，请不要使用复制数据库数据文件的方法。在OnlineJudgeDeploy 中，backup 目录提供了数据库导出 sql 文件备份脚本，每次备份后检查生成的 sql 文件的大小和内容，确保备份成功。

请不要把备份数据和 OnlineJudge 系统放在同一台机器上，这样数据丢失的风险仍然较高。

### 3.4.2恢复备份

如果只是想不同机器之间迁移部署，docker stop $(docker ps -aq) 然后复制 OnlineJudgeDeploy 文件夹到新机器后重新 docker-compose up -d 即可。

如果要恢复数据，首先要保证已经新部署了一套 OnlineJudge，然后需要恢复数据和测试用例文件。

测试用例存储在data/backend/test\_case 文件夹中，覆盖即可。

* 在新的机器上执行下面的操作可以恢复数据库
* docker cp db\_backup\_xxxxxxx.sql oj-postgres:/root
* docker exec -it oj-postgres bash
* psql -U postgres 然后运行 drop database onlinejudge; (请一定注意！！！看清楚自己再哪台机器上）
* \q 退出，然后 psql -f /root/db\_backup\_xxxxxxx.sql -U postgres