

概要设计文档

编写人员：贺思超、陈杰、陈增耀、何毅、江桀

指导老师：张健

编写日期：2023.7.14

概述

目的

本文档是针对云体测（在线体测）系统的设计，包括系统动态设计的概要、关键数据结构设计、流程设计、人机交互设计、非功能性设计、系统设计、系统部署与实施设计、版本维护设计等内容。

系统设计与需求分析旨在将文字与形式化设计相结合，提供半形式化的设计与实现的低层次设计。通过与需求分析内容相对应，保证系统设计的语言文档、形式建模、主要类图等可在 UML 中进行表达，并实现系统设计。

本文档的读者包括云体测系统的产品经理、设计人员、开发人员、测试人员以及适用的维修人员。他们将通过阅读本文档了解系统的设计细节、功能实现和系统架构，以便更好地理解系统的工作原理和进行相应的开发、测试、维护工作。

通过本文档，读者将获得以下内容：

- 系统动态设计概要：提供系统的整体架构和模块划分，描述各个模块之间的交互关系和功能。
- 关键数据结构设计：介绍系统中关键的数据结构，包括数据库表设计、数据流设计等，确保数据的有效组织和存储。
- 流程设计：描述系统的核心流程和各个模块之间的交互流程，确保系统能够按照预期的方式运行。
- 人机交互设计：考虑用户体验，设计系统的界面和交互方式，使用户能够方便、直观地使用系统。
- 非功能性设计：关注系统的性能需求和安全性需求，包括处理能力、响应时间和数据安全等方面的设计。

6. 系统设计：基于需求分析和上述设计内容，详细描述系统的架构、模块设计和关键技术选型等。

7. 系统部署与实施设计：描述系统的部署和实施计划，包括硬件环境、软件环境和系统配置等。

8. 版本维护设计：考虑系统的可维护性和可扩展性，提供版本维护和升级的设计方案，确保系统能够适应未来的需求变化。

通过本文档的阅读，读者将全面了解云体测系统的设计和实现，为后续的开发、测试、部署和维护工作提供指导和参考。

设计原则

1. 搜索速度与资源占用：对于文件搜索功能，主要考虑的问题是提高搜索速度和降低资源占用。通过优化搜索算法、建立索引、使用合适的数据结构等方法，可以提高搜索效率。同时，需要合理利用系统资源，确保搜索功能不会过度占用系统资源，影响其他系统操作。

2. 桌面协议的主要考虑问题：在设计桌面协议时，需要考虑多个方面的问题。首先，设计需要考虑不受协议软件的约束，即使在不同操作系统或平台上也能够良好运行。其次，需要关注协议的性能，确保协议在数据传输和渲染方面具有高效性。此外，还需要考虑协议的安全性、可扩展性和可维护性等方面。

3. 架构设计的问题：在设计系统架构时，需要考虑多个因素。首先，需要根据系统的需求和规模选择合适的架构模式，例如分层架构、微服务架构等。其次，需要考虑系统的可伸缩性，以便能够应对未来的扩展需求。此外，还需要关注系统的性能和可靠性，确保系统能够在高负载和故障情况下保持稳定运行。

约束和假定

软件环境约束

- (1) 开发软件：IDEA, PyChorm, VSCode, Andriod Studio
- (2) 配置环境：Gradle, JAVA17, Python 3.10
- (3) 应用服务器：TOMCAT
- (4) 数据库：MYSQL

硬件环境约束

- (1) windows 系统

(2) 硬盘空间：16G 以上

(3) 处理器：64 位处理

接口/协议的约束

采用 HTTP 和 UDP 协议

软件质量的约束

1. 性能：系统需要能够支持 3000 人以上同时登录云体测系统，并且能够处理多人同时在线操作。系统的响应延迟不能超过 50 毫秒，以保证用户体验和操作的实时性。
2. 可用性：系统的各个按钮、输入框等操作元素需要正常运行，并且当用户发生错误操作时（如输入非法数字）系统应能及时提示用户。这样可以提供友好的用户界面，减少用户错误并提升用户体验。
3. 可靠性：即使在系统发生错误时，系统仍需保持基本的登录功能和数据交互与存储的能力。这样可以确保系统不会完全崩溃，并能够维持基本的功能和数据安全。
4. 健壮性：系统需要具备一定的承载压力的能力，能够在高负载情况下保持稳定运行，不易受到外部干扰而崩溃或失效。
5. 安全性：系统需要能够阻止非授权用户的企图，并保证只向合法用户提供服务。这包括用户认证、数据加密、防止恶意攻击和拒绝服务等安全措施。
6. 可修改性：系统应具备快速进行变更的能力，以便在需求变化或修复漏洞时能够迅速进行系统改进，并保持较高的性能。
7. 易用性：系统的用户界面设计应简明易懂，用户能够轻松理解和操作，以降低用户学习成本和提高用户满意度。
8. 可测试性：当系统发生故障时，系统应具备隔离和定位故障的能力，同时能够进行有效的测试设计和测试执行，以便快速发现和修复问题。
9. 功能性：系统需要能够实现所期望的功能，并满足用户的需求，确保系统能够正常运行并提供所需的功能。
10. 互操作性：系统需要与外部系统或其他系统之间能够进行相互作用，以实现数据交换、系统集成和协同工作等功能。

额外约束

1. 可扩展性：系统应具备可扩展性，能够根据需求的增长进行水平或垂直扩展，以支持更多用户和更大的数据量。这可以通过良好的系统架构和模块化设计来实现。
2. 可靠性：系统应具备高度可靠性，能够在面临故障、错误或异常情况时继续正常

运行。这可以通过实施错误处理机制、备份和恢复策略等措施来提高系统的可靠性。

3. 数据保护和隐私：系统应采取适当的措施保护用户数据的安全性和隐私性，包括加密敏感数据、访问控制、数据备份和灾难恢复等措施。
4. 符合法规和合规性：系统应符合适用的法规、标准和合规性要求，如数据保护法规、隐私法规、安全标准等。这可以确保系统在法律和行业规范的框架内运行。
5. 跨平台兼容性：如果系统需要在不同平台或设备上运行，系统应具备跨平台兼容性，能够适应不同的操作系统、浏览器或移动设备，并提供一致的用户体验。
6. 文档和培训支持：系统应提供清晰、详细的文档和培训材料，以帮助用户和技术人员理解系统的功能和操作方法，并提供必要的支持和培训。
7. 可追溯性：系统应具备追溯性，能够记录和跟踪用户操作、系统事件和错误日志，以便在需要进行故障排查、审计和问题解决。
8. 资源效率：系统应合理利用计算资源、存储资源和网络带宽，以确保系统的高效运行并减少不必要的资源消耗。

系统设计

系统概述

该系统由网页前端、安卓 APP、数据后端和算法后端四个部分组成。每个部分具有不同的功能和职责。

算法后端使用 PyCharm 进行开发。它接收数据后端传回的数据值，并利用计算机视觉技术实时检测用户正在进行的体育动作。通过获取相关推流服务器上摄像头实时拍摄到的画面，算法后端会计算出运动的详细数据，并将结果返回。

数据后端使用 Spring Boot 框架进行开发。它接收网页前端和安卓 APP 传来的摄像机、用户等相关信息，并调用算法后端的接口进行动作识别和计数。然后，数据后端将相关数据返回到网页前端的界面和安卓界面。

网页前端采用 Vue 框架进行开发。它实现了基本的客户功能，如登录、注销，以及成绩查询和历史运动数据查看等查询功能。网页前端的核心功能是动作展示和识别结果的确认。

安卓 APP 前端采用安卓原生框架进行开发。它具有与网页前端相同的一系列功能。同时，安卓 APP 将算法部分移植到本地进行动作识别，并调用数据后端的接口进行数据上传。

通过这个系统，用户可以通过网页前端或安卓 APP 进行体育动作的检测和数据分析。

系统利用计算机视觉技术实时检测用户的动作，并提供详细的运动数据。用户可以通过查询功能查看自己的成绩和历史运动数据，以便进行进一步的分析和改进。

整个系统的设计目标是提供用户友好的界面和高效准确的动作识别和数据分析功能。通过结合网页前端、安卓 APP、数据后端和算法后端的各自优势，实现了一个全面的体育动作监测系统。

系统架构图

云体测系统采用了多层架构，将系统分为展示层、业务层和数据层。每个层次具有不同的功能和职责，以实现系统的各项功能。

1. 展示层：

在展示层中，包括网页前端和安卓 APP 两个部分。网页前端采用 Vue 框架进行开发，安卓 APP 采用安卓原生框架进行开发。这两部分负责展示用户界面、接收用户输入，并通过 Ajax 技术与业务层进行交互。用户可以通过网页前端或安卓 APP 访问系统的各种功能。

2. 业务层：

业务层包括登录注册、个人信息修改、人员管理、统计报表，人脸照片上传，人脸比对，动作识别，成绩上传，运动记录上传等具体功能

3. 数据层：

数据层负责数据库的访问和操作。它包含自定义函数，用于提供对数据库的访问接口，例如执行 SELECT 语句进行数据查询或执行 UPDATE 语句进行数据更新。业务层通过调用数据层的接口来获取或修改数据库中的数据。

整个系统的架构是基于多层架构的分层设计，通过将不同的功能和职责分离，实现了系统的模块化和可扩展性。表示层负责用户界面的展示和用户交互，业务层处理各种业务逻辑和功能，数据层负责数据的访问和管理。这种架构使系统更易于维护、扩展和重用，并提高了系统的可靠性和性能。

[该类型的内容暂不支持下载]

WEB 前端

1. 用户管理和信息维护：提供用户管理功能，包括注册、登录、个人信息维护等。
2. 人脸数据收集和存储：体测前，用户需要上传身份证照片以及 WEB 端摄像头实时拍摄的照片，核验为同一人后即可开始体测
3. 在线测评和数据分析：提供在线测评功能，根据用户提供的数据进行评估和分析。
4. 结果展示和报告生成：根据测评结果生成个性化的体测报告或结果展示。

逻辑后端

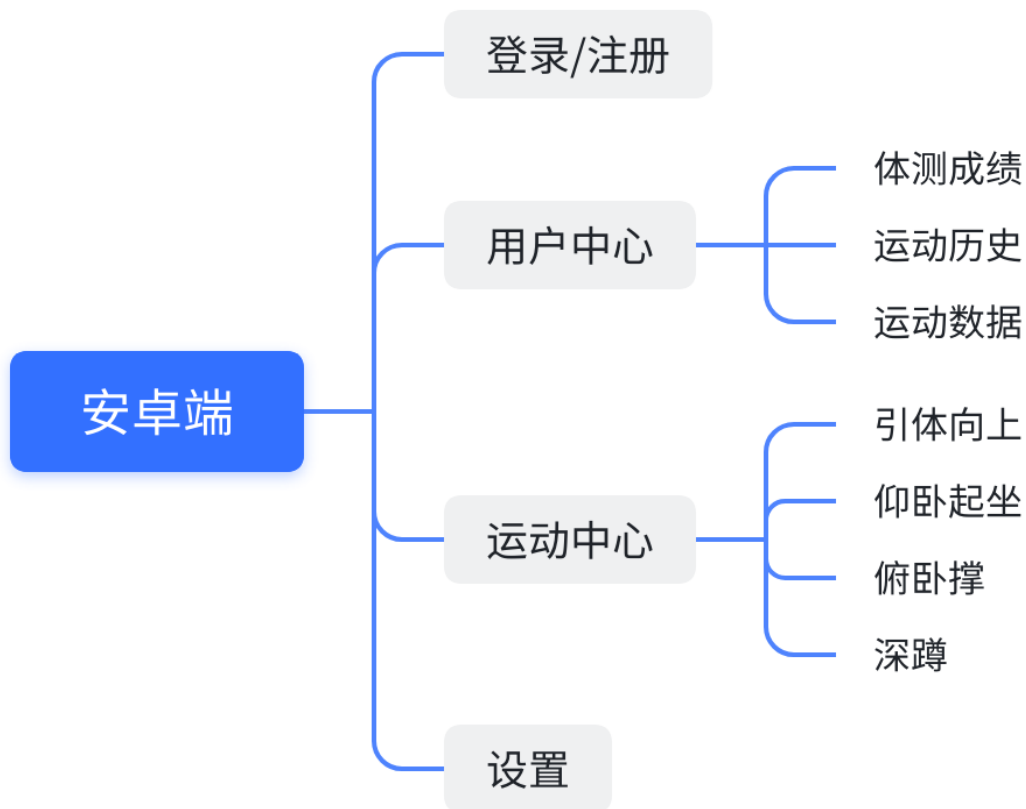
1. 数据库设计
2. 管理员模块
3. 腾讯云上传模块
4. 人脸比对模块
5. 动作识别模块
6. 成绩记录模块

算法后端

1. 环境配置
2. 动作定义
3. 动作识别
4. 接入 Flask

安卓端

安卓端类似 WEB 前端，不过要实现实时的运动计数算法，并上传成绩，以及可以让用户进行查询体测成绩和运动记录



推拉流服务端

1. 高性能异步网络架构 Boost.Asio
2. 流媒体协议支持: rtmp, rtsp, HTTP-FLV
3. 编码支持: H265/H264/AAC
4. 存储与分发 (尚在开发)
5. 客户端管理
6. 日志监控调度

非功能性设计

性能需求

处理能力

在线平台作为一个高并发的系统, 需要考虑系统能够承载的最大并发用户数。这个数字应该根据实际情况进行规划, 但至少应该能够支持全校学生总人数乘以一个常数 φ 的并发用户数。这个常数 φ 通常取决于服务器容量和性能的预期。通过合理的服务器

配置和优化，确保系统能够在高负载情况下保持稳定，并能够处理大量的并发请求。

响应时间

为了能够提供快速的在线测评服务，系统需要具备快速响应用户请求的能力。除了系统的响应速度，网络状况也会对用户最终获取结果的响应时间产生影响。因此，对于 Web 服务器端，需要设置较高的要求，以确保能够在最短的时间内响应用户的请求。这可以通过优化系统架构、减少响应时间和使用缓存等技术手段来实现。

安全性需求

1. 数据加密：为了保证数据的安全性，在传输过程中，所有的数据应采用高强度的加密算法进行加密，例如使用 DES 加密算法。通过加密，即使数据发生泄漏或被截获，也无法识别相关的数据内容，从而确保数据的机密性和完整性。
2. 安全套接层（SSL）：在客户端和服务端之间的数据交互中，需要使用安全套接层（SSL）进行信息交换。SSL 是一种基于加密技术的安全传输协议，用于保护数据在传输过程中的安全性。通过在客户端移动终端和服务端之间使用 SSL 加密传输，可以确保重要信息在传输过程中的安全性。
3. 客户端和服务端安全：除了数据传输的安全性，还需要确保客户端和服务端之间的通信安全。这可以通过实施身份验证、访问控制和安全审计等措施来实现。通过限制未经授权的访问和监控系统的使用情况，可以保护系统免受恶意攻击和非法访问。