ICS 点击此处添加 ICS 号
CCS 点击此处添加 CCS 号

团　　体　　标　　准

T/QGCML XXXX—XXXX

通讯设备检修服务管理平台技术规范

Technical specifications for communication equipment maintenance service management platform

XXXX－XX－XX 发布　　XXXX－XX－XX 实施

全国城市工业品贸易中心联合会　　发 布

目　　次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由××××提出。

本文件由全国城市工业品贸易中心联合会归口。

本文件起草单位：

本文件主要起草人：

本文件为首次发布。

通讯设备检修服务管理平台技术规范

1 范围

本文件规定了通讯设备检修服务管理平台技术规范的术语和定义、技术要求、功能要求、性能要求、安全要求。

本文件适用于通讯设备检修服务管理平台的设计和检验。

2 规范性引用文件

本文件没有规范性引用文件。

3 术语和定义

下列术语和定义适用于本文件。

3.1

通讯设备检修服务管理平台 Communication equipment maintenance service management platform

指用于提供通讯设备检修服务的信息系统平台，包括设备信息管理、检修任务管理、检修过程监控、数据分析等功能。

4 技术要求

4.1 技术架构

4.1.1 平台应采用稳定、可靠且广泛认可的技术架构，如微服务架构、分布式架构等，以适应复杂多变的业务需求。

4.1.2 平台应采用模块化设计思想，将功能划分为独立的模块，便于模块的独立开发、测试和维护。

4.1.3 平台应提供标准化的接口，包括 API 接口、数据接口等，以便于与其他系统或平台进行集成和交互。

4.2 技术选型

4.2.1 选择成熟、稳定且广泛使用的开发语言，如 Java、Python 等，以确保代码的可读性、可维护性和可扩展性。

4.2.2 根据业务需求和数据量大小，选择适合的数据库系统，如关系型数据库（MySQL、Oracle 等）或非关系型数据库（MongoDB、Redis 等）。

4.2.3 采用成熟的中间件技术，如负载均衡器、消息队列、缓存系统等，以提高系统的性能和稳定性。

4.3 技术实现

4.3.1 实现设备信息的统一管理和维护，包括设备信息的录入、查询、修改、删除等功能。支持设备信息的分类管理和统计分析，方便用户快速定位和管理设备。

4.3.2 实现检修任务的创建、分配、跟踪、完成等功能。通过任务进度监控和任务超时提醒，确保检修任务能够按时完成。

4.3.3 实时监控检修过程中的关键指标和数据，如设备状态、检修进度等。提供检修过程回放和数据分析功能，帮助用户了解检修情况并优化检修流程。

4.3.4 对检修数据进行统计和分析，提供可视化报表和图表。支持数据导出功能，方便用户进行进一步的数据处理和分析。

4.4 技术安全与可靠性

4.4.1 采用强密码策略和多因素认证机制，确保用户身份的真实性和合法性。实施严格的权限管理策略，确保用户只能访问其授权范围内的数据和功能。

4.4.2 对敏感数据进行加密存储和传输，确保数据的机密性和完整性。采用安全的通信协议（如 HTTPS），确保数据传输过程中的安全性。

4.4.3 定期进行安全漏洞扫描和风险评估，及时发现并修复潜在的安全隐患。关注最新的安全漏洞信息，并采取相应的防护措施。

4.4.4 设计合理的容错机制和灾备方案，确保在发生故障时能够迅速恢复服务。采用数据备份和恢复技术，确保数据的可靠性和完整性。

4.5 技术文档与支持

4.5.1 提供详细的技术文档，包括系统架构图、数据流程图、接口文档等，方便开发人员和用户了解系统的技术实现和使用方法。

4.5.2 提供全面的技术支持服务，包括技术咨询、故障排查、系统升级等，确保用户在使用过程中能够得到及时有效的帮助和支持。

5 功能要求

5.1 设备管理

5.1.1 平台应提供友好的用户界面，允许用户录入设备的基本信息，如设备名称、型号、序列号、生产厂家、安装位置等。同时，应支持批量导入设备信息，以提高录入效率。

5.1.2 平台应支持多种查询方式，如按设备名称、型号、序列号等进行精确查询，或按生产厂家、安装位置等进行模糊查询。查询结果应展示详细的设备信息，并支持导出功能。

5.1.3 当设备信息发生变化时，平台应允许用户进行修改，并确保修改后的信息能够及时同步到系统中。

5.1.4 平台应能够展示设备的当前状态，如正常、故障、维修中等。同时，应支持用户对设备状态进行手动调整，并记录状态变化的历史记录。

5.2 检修任务管理

5.2.1 平台应允许用户创建检修任务，并填写任务的基本信息，如任务名称、任务描述、优先级、开始时间、结束时间等。同时，应支持关联设备信息和维修人员信息。

5.2.2 平台应支持任务的自动分配或手动分配。自动分配可根据维修人员的空闲时间、技能水平等因素进行智能匹配；手动分配则允许用户自行选择维修人员。

5.2.3 平台应实时跟踪任务的执行情况，包括任务的进度、维修人员的工作状态等。同时，应提供任务超时提醒功能，以确保任务能够按时完成。

5.2.4 当任务完成后，平台应允许用户进行任务完成确认，并记录任务完成的时间、维修人员等信息。同时，应支持生成任务报告，以供后续分析和评估。

5.3 检修过程监控

5.3.1 平台应能够实时监控检修过程中的关键指标和数据，如设备状态、维修人员位置、维修进度等。通过可视化界面展示实时数据，帮助用户了解检修情况。

5.3.2 平台应支持检修过程回放功能，允许用户查看历史检修记录，包括维修人员的工作轨迹、维修

步骤等。这有助于用户了解检修过程的全貌，并进行问题追溯。

5.3.3 平台应对检修数据进行统计和分析，提供可视化报表和图表。通过数据分析，用户可以发现潜在的问题和规律，为优化检修流程提供依据。

5.4 用户管理

5.4.1 平台应提供用户注册和登录功能，确保用户能够安全地访问和使用系统。同时，应支持多种认证方式，如用户名密码认证、手机验证码认证等。

5.4.2 平台应实施严格的权限管理策略，确保用户只能访问其授权范围内的数据和功能。通过角色管理、权限分配等方式，实现用户权限的灵活配置。

5.4.3 平台应记录用户的操作日志，包括登录时间、操作内容、操作结果等。通过日志分析，可以了解用户的使用情况，及时发现潜在的安全问题并进行处理。

6 性能要求

6.1 响应时间

6.1.1 平台应保证在用户发出请求后，能够迅速作出响应。一般情况下，系统的平均响应时间应控制在 X 秒以内，且对于高优先级或紧急的请求，应优先处理并尽快返回结果。

6.1.2 平台的前端页面应优化加载速度,确保用户在访问页面时能够迅速获取所需信息。一般情况下，页面的首次加载时间应控制在 Y 秒以内，且页面间的跳转和刷新时间也应控制在较短的时间范围内。

6.2 吞吐量与并发处理能力

6.2.1 平台应能够处理大量的业务请求和数据交换。在正常情况下，平台的吞吐量应满足业务需求，并能够根据业务量的增长进行弹性扩展。

6.2.2 平台应具备良好的并发处理能力，能够同时处理多个用户的请求而不影响系统性能。在高并发场景下，平台应能够保持稳定的运行状态，并确保用户请求的及时处理。

6.3 稳定性与可靠性

6.3.1 当平台发生故障或异常时，应能够迅速恢复服务。故障恢复时间应控制在较短的时间范围内，以减少对用户的影响。

6.3.2 平台应确保数据的准确性和一致性。在数据传输、存储和处理过程中，应采取适当的数据校验和冗余备份措施，确保数据的完整性和可靠性。

6.4 可扩展性与可维护性

6.4.1 平台应具备良好的可扩展性，能够支持业务需求的不断增长和变化。通过模块化设计、微服务架构等技术手段，平台应能够方便地进行功能扩展和升级。

6.4.2 平台应具有良好的可维护性，便于开发人员进行系统的维护和优化。通过清晰的代码结构、合理的注释和文档、友好的开发工具支持等方式，降低维护成本并提高维护效率。

7 安全要求

7.1 网络安全

7.1.1 设立网络边界防火墙，限制维修服务人员的网络访问权限，防止未授权访问。

7.1.2 定期更新防火墙规则，以适应新的安全威胁和攻击模式。

7.1.3 实施访问控制列表（ACL），明确允许和拒绝的网络流量。

7.1.4 对维修服务过程中的敏感数据进行加密传输和存储，确保数据的机密性。

7.1.5 采用强密码策略，包括定期更换密码、强制密码复杂度要求等，防止密码泄露和破解。

7.2 设备安全

7.2.1 设置强密码，并定期更换，确保设备只能被授权人员访问。

7.2.2 采用多层次的访问控制策略，如基于角色的访问控制（RBAC），确保只有特定权限的用户能够访问和管理设备。

7.2.3 根据具体的安全需求，合理配置设备防火墙，包括设置访问规则、限制传输协议和端口等。

7.2.4 定期审查和更新设备防火墙配置，以适应新的安全威胁和攻击模式。

7.3 数据安全

7.3.1 定期备份设备数据，确保在发生数据丢失或损坏时能够迅速恢复。

7.3.2 对备份数据进行加密存储，以防止未经授权的访问和泄露。

7.3.3 使用数据完整性校验机制，如哈希函数或数字签名，确保数据的完整性和真实性。

7.3.4 在数据传输和存储过程中,采取适当的数据校验和冗余备份措施,确保数据的完整性和可靠性。

7.4 漏洞管理

7.4.1 使用专业的漏洞扫描工具对系统进行定期扫描，及时发现并修复潜在的安全漏洞。

7.4.2 定期进行安全评估，对系统的安全状况进行全面分析和评估，发现潜在的安全风险并提出相应的改进措施。

7.4.3 及时更新操作系统、应用程序和安全补丁，以修复已知的安全漏洞和缺陷。

7.4.4 在更新之前进行充分的测试和验证，确保更新不会对系统的稳定性和性能产生负面影响。

7.5 安全审计与监控

7.5.1 定期对系统进行安全审计，检查系统的安全配置、访问权限、日志记录等方面是否符合安全要求。

7.5.2 对安全审计结果进行分析和评估，发现潜在的安全风险并提出改进措施。

7.5.3 实时监控系统的运行状态和安全事件，及时发现并处理异常情况和安全威胁。

7.5.4 建立安全事件应急响应机制，对安全事件进行快速响应和处理，减少损失和影响。