摘要：

2023年5月份，我所在公司承担了某银行“智能终端管理系统”的开发工作，我作为该项目的系统架构设计师，负责该系统的需求分析与架构设计工作。该系统主要实现对智能收银终端软硬件的管理及维护，包括应用APP管理、设备管理、远程协助以及屏媒广告等多个模块。本文以该平台为例，论述系统三层架构在该项目中的具体应用。表现层，包括Web管理后台和Android客户端，前者采用Vue、iView等前端技术，构建友好的用户操作界面；业务逻辑层，负责处理系统的核心业务，采用SprintBoot框架，将不同业务功能进行模块化，按功能模块进行开发和维护；数据层，采用Mybatis作为持久层框架，数据库选择MySql，并使用Redis缓存技术提升访问效率。整个项目历时11个月开发完成并上线，到目前运行稳定。实践证明，三层架构的使用极大提升了开发效率，减少了开发成本，也为后续的持续优化奠定了基础。

正文：

随着互联网技术的飞速发展，传统的收银设备功能单一，无法远程升级或升级过程繁琐，设备的使用状态无从得知，所以智能设备替换传统收银设备是一个趋势。为了增加商户的粘性，提升服务质量，各大银行开始加大智能收银终端的投放。不同于传统的收银设备，智能设备联网能力强，操作系统强大，能够实现更多更复杂的业务。为了充分发挥智能设备的优势，银行提出建设一个智能终端管理系统的需求，实时监控设备状态，并对设备的软硬件生命周期进行有效的管理。

2023年5月份，我司承担了某银行的智能终端管理系统的开发工作，我作为该项目的系统架构设计师，参与整个开发过程。该系统的主要功能模块包括应用APP管理、设备管理、归属管理、远程协助以及屏媒广告等多个模块。建设该系统旨在简化银行对设备的管理，提升对商户设备的监管能力，以及对设备的维护效率。首次使用时，银行可在平台针对不同商户配置个性化装机任务，实现一键装机，极大提升装机效率。商户使用过程中，银行通过平台的可视化界面，能够清晰、实时的关注到设备当前的软件使用情况和硬件状态。针对软件常见的BUG问题，在商家授权后，通过该平台的远程协助功能，可远程操控设备，获取设备日志，便于软件开发人员排查问题。此外，硬件方面，银行也可以通过平台查看每个模块的使用寿命情况，在硬件寿命到期前，提前联系售后团队进行更换，提升商户的体验。

三层架构是一种经典的系统架构设计模式，应用广泛，它将系统分为三个独立的层次：表现层、业务逻辑层以及数据层。各层次间职责分离，相对独立，便于系统的维护和扩展。下面我将详细阐述三层架构在该系统的应用过程。

首先，表现层负责为用户提供直观的操作界面，包括Web管理后台以及Android客户端。Web管理平台采用Vue、iView等前端技术进行开发设计。Vue是一个轻量级的框架，易于上手，项目组中的新人能够快速掌握。另外，它采用虚拟DOM和组件化的方式来提高渲染效率，性能较高，表现层作为和用户之间交互，流畅的操作体验非常重要，所以我最终选择Vue作为前端技术。iView是一个开源的高质量UI组件库，它提供了覆盖导航、布局、数据输入等多个方面的组件，组件库丰富，几乎覆盖了所有常见web应用场景。同时，它的API设计非常友好，开发者能够快熟上手，所以我选择了它作为前端UI组件库。Ajax可以异步加载数据，能够快速响应用户操作，避免不必要的页面刷新和重新加载，具备较好的用户体验。同时，Ajax对不同浏览器的兼容性较好，该项目的用户包括银行和商户，使用环境是参差不齐的，它的跨浏览器兼容性也是我选择它的一个考虑因素。

其次，业务逻辑层是系统的核心部分，采用主流的SprintBoot框架进行开发。为了保证系统的可扩展性，业务逻辑层设计为模块化，包括应用APP管理模块、设备管理模块、远程协助模块、屏媒广告模块等等。从设备通信的实时性角度，为了实现智能管理系统和Android端应用的长连接及消息推送场景，我们引入了MQTT消息中间件。通过MQTT，系统能够将银行或者商户的配置，实时推送到设备，这非常重要，比如屏媒广告任务。当银行或商家配置了新的广告时，是希望能够即时生效的；从设备数据的安全性角度，为保证数据的安全性，所有的数据传输都使用HTTPS进行加密传输，在通信过程中采用令牌机制，确保数据的完整性不被篡改；从安全角度，在业务逻辑层实现请求限制，当客户端短时间内频繁请求超过设定的阈值时，服务器会拒绝该请求并返回错误。

最后，数据层选择Mybatis作为持久层框架，MySQL作为数据库。Mybatis简单易学，灵活并且解除了sql语句与程序代码的耦合，通过DAO层，将业务逻辑和数据访问逻辑分离，使得系统的设计更加合理、清晰，并具备更好的维护性。为了应对高并发数据访问可能导致的数据库压力过大，我们使用数据库读写分离技术，主数据库负责写操作，从数据库负责读操作，减轻主库的压力。另外，为了提升设备数据、应用APP数据的查询速度，一方面我们引入了Redis缓存机制，针对热点数据进行缓存，另一方面，我们对SQL语句和索引进行优化，使用批量操作的SQL语句减少数据库的访问次数。例如在更新设备状态时，将多条语句合并成一条批量更新语句。

整个项目历时11个月开发完成，并于2024年4月顺利上线，到目前运行稳定，市场反馈良好。尽管该架构带来了许多优势，但实际应用中仍然存在一些不足之处。在高并发场景中，如何保证数据的实施一致性是一个难题。例如，设备管理中的状态更新和应用管理中的安装操作，可能因为网络延迟，数据冲突等原因导致数据一致性问题。针对这个问题，可以通过引入消息队列，实现事件驱动架构。对于某些操作（如设备状态变更），可以通过事件驱动的异步处理来保证数据的一致性，同时也能减少对主数据库的直接压力。

实践证明，通过基于三层架构设计的智能终端管理平台，不仅降低了开发成本，提高了团队开发效率，同时，系统在大规模设备管理中展现出了良好的稳定性。通过表示层、业务逻辑层和数据访问层的分工和协作，实现了系统的高效、灵活和安全，有效提升了银行智能终端管理效率，减少了设备维护成本，从项目完工后的使用效果来看，达到了预期目的，也获得了行方的一致好评。

最后，为了提高自己的架构设计水平，我将在以后的工作时间中持续不断的学习，AI是趋势，之后将重点放在系统的AI智能分析模块开发上，提升系统的智能化，在为公司创造价值的同时，也能够为客户开发出更稳定可靠的系统。