

# 安卓平板家长控制系统的设计与实现

郝晓川<sup>1</sup>, 代磊<sup>1</sup>, 温泉<sup>1</sup>, 程树欣<sup>2</sup>, 文家义<sup>2</sup>, 李迎新<sup>2</sup>

<sup>1</sup>联想(北京)有限公司, 北京

<sup>2</sup>南京畅索软件科技有限公司, 江苏 南京

收稿日期: 2023年1月21日; 录用日期: 2023年2月17日; 发布日期: 2023年2月24日

## 摘要

随着智能手机、智能平板电脑在家庭中的普及,越来越多的学生都用上了自己专属的设备。K12教育从传统的线下教育培训逐步转向网络教育,推动了学生电子设备的普及。网络的世界是精彩的且不可预测的,成年人尚且无法实现自控,更不用说正处于成长中的学生,防止学生过度沉迷电子产品,刻不容缓。安卓平板家长控制系统,是基于安卓平板电脑的一整套完善的应用系统,可以有效帮助家长了解学生的平板电脑使用情况,监督学生学习。考虑到逐步增长的市场需求,需要提升应用服务器的高并发处理能力,借助Netty、Redis等技术来构建家长端与学生端之间的高性能通信,基于HTTPS与TCP协议完成加密数据发布与接收;同时利用现有安卓平板设备的定制化SDK接口,满足更多场景需求。本文将重点介绍家长控制系统的整体设计与实现,包括安卓家长端应用、学生端应用、后台管理系统,重点解决的问题如下:家长端与学生端应用之间采用Netty直接与间接通信的方式实现;Netty服务器的性能调优、心跳检测与断线重连;学生端应用的信息亭模式,通过调用平板设备定制化SDK接口来实现。

## 关键词

安卓, 家长控制, 信息亭模式, 远程控制

# Design and Implementation of Android Tablet Parent Control System

Xiaochuan Xi<sup>1</sup>, Lei Dai<sup>1</sup>, Quan Wen<sup>1</sup>, Shuxin Cheng<sup>2</sup>, Jiayi Wen<sup>2</sup>, Yingxin Li<sup>2</sup>

<sup>1</sup>Lenovo (Beijing) Co., Ltd., Beijing

<sup>2</sup>Nanjing Changsuo Software Technology Co., Ltd., Nanjing Jiangsu

Received: Jan. 21<sup>st</sup>, 2023; accepted: Feb. 17<sup>th</sup>, 2023; published: Feb. 24<sup>th</sup>, 2023

## Abstract

With the popularity of smart phones and smart tablets in families, more and more students have used their own devices. K12 education has gradually shifted from traditional offline education and training to online education, promoting the popularization of student electronic equipment. The

world of the Internet is wonderful and unpredictable. Adults can't control themselves, let alone growing students. It is urgent to prevent students from indulging in electronic products. The Android tablet parent control system is a complete set of application systems based on Android tablets, which can effectively help parents understand the use of students' tablets and supervise students' learning. Considering the growing market demand, it is necessary to improve the high concurrent processing capability of the application server, build high-performance communication between parents and students with Netty, Redis and other technologies, and complete the release and reception of encrypted data based on HTTPS and TCP protocol; at the same time, the customized SDK interfaces of existing Android tablet devices can be used to meet the needs of more scenarios. This article will focus on the overall design and implementation of the parent control system, including Android parent application, student application, and background management system. The key problems to be solved are as follows: Netty direct and indirect communication between parent and student applications; Netty server performance tuning, heartbeat detection and disconnection reconnection; the Kiosk mode of the student application is realized by calling the customized SDK interface of the tablet device.

## Keywords

Android, Parental Control, Kiosk Mode, Remote Control

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

随着互联网的快速发展,各种智能电子设备在青少年学生群体中得到广泛的普及,这给青少年带来了便利性和娱乐性的同时,引发了不少社会问题。由于低龄人群的智能设备持有率不断上升,并缺乏辨识能力,极易受到互联网内容信息的影响,这样不仅会影响他们正常的学习和生活,也会损害他们的身心健康,所以家长有必要对学生的智能设备进行管控,规范青少年对智能设备的使用行为,防止青少年沉迷虚拟网络世界,促进青少年的健康成长。

目前市面上有很多厂商已经对家长控制这一功能进行了实现,例如 Apple 的家人共享功能,Google 的数字健康与家长控制功能,华为的家长助手功能,微软的 Family Safety 功能。各大厂商对于家长控制的功能实现还是十分看重的,但是在具体使用上是有一定限制的,如大部分 Android 系统在海外市场都是使用 Google 的原生功能,这对于其他不能使用 GMS 服务的国家和地区并不具有通用性,并且不能满足个性化的功能需求。联想同样作为 OEM 厂商,拥有许多的智能设备,如 PC 电脑、平板电脑、手机等,需要对此功能进行实现和拓展。目前平板设备使用的仍是基于 Google 的原生功能,面临着类似的问题。为了向用户提供更好的个性化服务,有必要实现并完善家长控制功能。

本文提出的家长控制系统可以多地区部署提供服务,满足国内外用户的使用需求,另外,为了向用户提供更好的个性化服务,本文基于 Android 系统深度设计并调用定制化 SDK,以满足不同用户的需求,这对于用户具有重要意义。

## 2. 系统设计

### 2.1. 系统目标与功能

本系统设计的目标是为了在安卓平板电脑上的实现家长控制功能,保证学生在学习中健康使用平板

电脑，系统的主要功能如下：

1) 家长端功能列表

包括账户管理、设备管控功能，如图 1 所示：

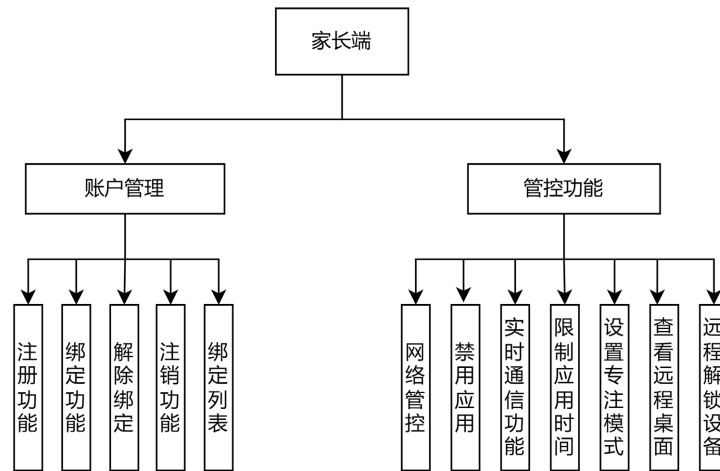


Figure 1. Diagram of the parent side functional structure  
图 1. 家长端功能结构图

2) 学生端功能列表

包括信息亭模式、应用限额控制，以及远程控制，如图 2 所示：

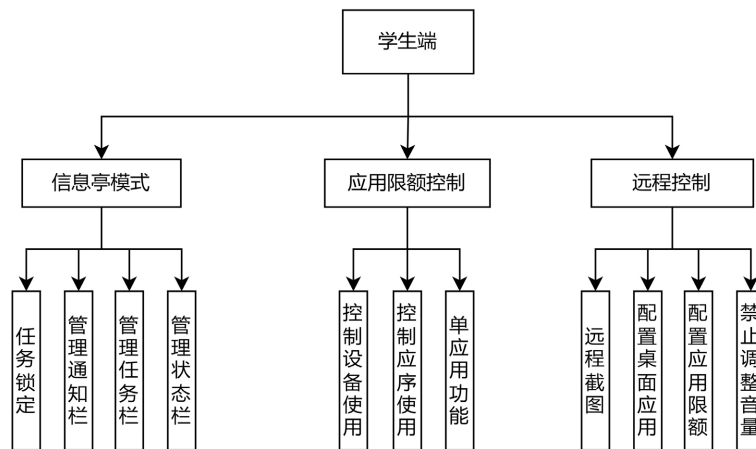


Figure 2. Diagram of the student side functional structure  
图 2. 学生端功能结构图

## 2.2. 系统结构设计

本文介绍的家长控制系统，主要包括家长端安卓应用、学生端安卓应用和云平台三部分组成。总体系统架构如图 3 所示，家长端和学生端都采用 Android Studio 来完成环境搭建与编译[1]，数据库采用主从服务和灾备存储，后台业务服务架构采用 Spring Boot 和 Vue.js，学生端 Web 服务采用 Netty 框架[2]，利用消息队列服务完成学生端的数据传递与延迟入库[3]，利用 Redis 来缓存家长端的管控命令并实现管控命令的发出与接收[4]。

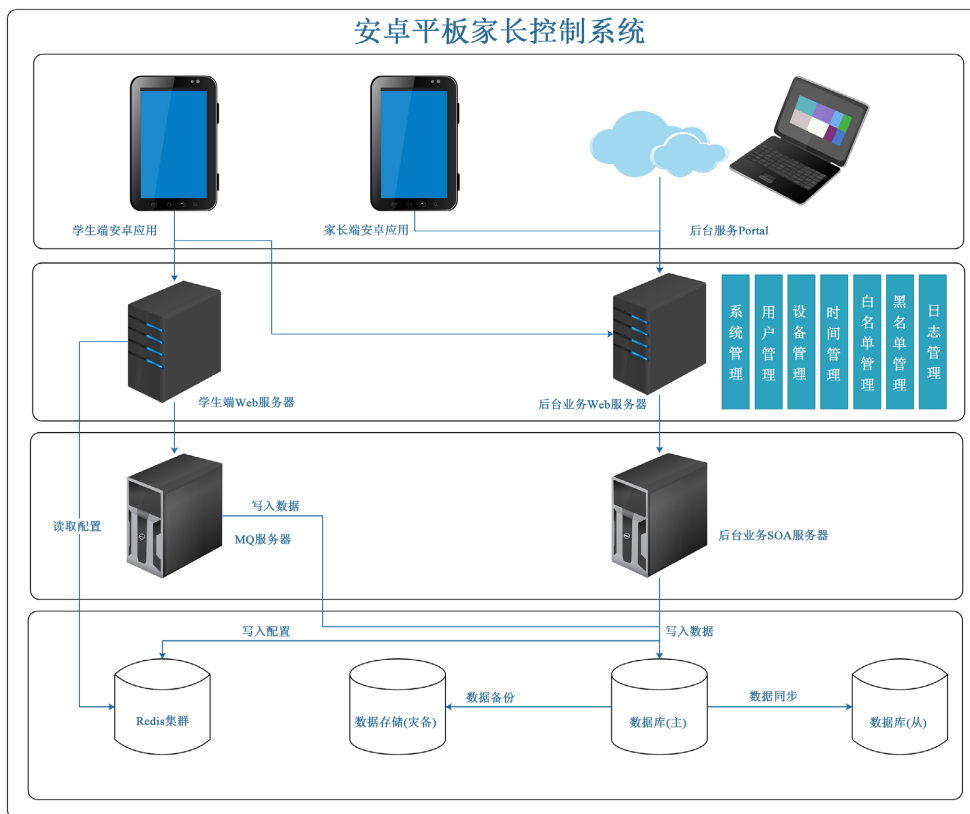


Figure 3. Diagram of the parental control system architecture  
图 3. 家长控制系统架构图

1) 家长端安卓应用

实现家长账号与学生设备的绑定，读取学生设备相关信息，远程控制学生设备，向学生设备下达管控命令，并实时获取学生设备的统计分析情况[5]。

2) 学生端安卓应用

接收来自家长端安卓应用的管控命令，并将学生端设备应用、设置状态与使用情况统计实时保存到云平台，用于家长端的实时监控。

3) 云平台

作为家长端与应用端的枢纽，首先保证了两者的安全通信与数据传输，保证学生端可以实时获取到家长端的管控命令[6]，并对学生端的数据进行采集，用于家长端的界面展示，供家长参考。另一方面，细化对设备的权限管理，保证家长端与学生端的一一对应绑定关系，接收量级化的数据，进行筛选与汇总入库。

2.3. 数据库设计

数据库采用主从同步、读写分离架构，为了保证数据量大的表的存储和读写能力，采用分库分表和定期迁移数据的方式，保证数据库读写的效率和容灾，数据库版本需要 MySQL5.7 以上，表之间关联键适当增加索引，将大表做视图，优化查询速度(图 4)。

1) **user** 表：存储用户基本信息，包括手机号、用户名、邮箱、公司等信息；

2) **deviceinfo** 表：存储设备的基本信息，包括绑定用户 ID、设备号、型号、安卓版本等；

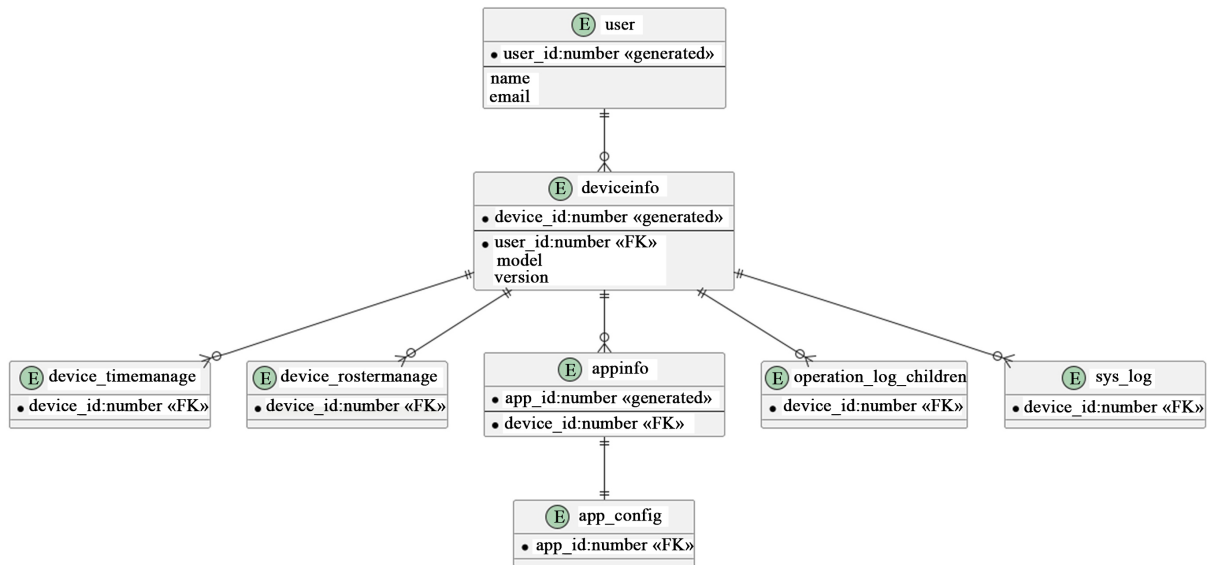


Figure 4. Diagram of the database object relationship

图 4. 数据库对象关系图

3) **device\_timemanage** 表: 设备使用时间方案的信息设置, 包括时间段、读书日、节假日、锁屏策略设置、所属设备 ID;

4) **device\_rostermanage** 表: 设备黑(白)名单管理, 包括应用列表、网站列表、所属设备 ID、名单类型(黑/白);

5) **appinfo** 表: 存储设备 APP 的基础信息, 包括 APP 名称、包名、所属设备 ID、安装时间、APP 图标地址、APP 状态等;

6) **app\_config** 表: APP 相关设置, 包括累计使用时间(例如: 分钟/天)、使用时间段设置、所属 APP (`app_id`)、APP 状态等;

7) **operation\_log\_children** 表: 学生端设备操作日志, 包括设备 ID、APP 使用记录、使用时间、网站访问记录等;

8) **sys\_log** 表: 系统 HTTPS 请求日志表, 包括请求 URL、请求参数、响应结果等信息, 用于系统日志监控。

### 3. 系统实现

#### 3.1. 服务端实现

后台管理服务采用的 SpringBoot 框架, 主要用于接收家长端的 HTTPS 请求, 例如家长端账号注册, 家长端绑定与移除学生端设备, 以及后台针对学生端的日志的深度分析, 并推送给家长端。

另外, 本系统采用 Netty 框架来建立服务端与客户端之间的通讯链接, Netty 是一个基于事件驱动的异步非阻塞的网络通讯框架, 底层采用 Reactor 线程模型和内存零拷贝的读写方式来设计实现, 可以快速开发高性能、高扩展性的网络通信应用[7]。服务端与客户端之间通过 Netty 实现长连接通讯, 主要靠心跳来维持服务器端及客户端连接, 搭建高性能的 Netty 服务器, 需要重点完成对服务性能的优化。

##### 3.1.1. Netty 服务性能调优

###### 1) 心跳检测

为应对 NAT 老化以及及时感知客户端在线状态,使用心跳机制必不可少。心跳检测可以及时发现当前链路的通达性,判断双方是否可以正常收发数据。及时断开假死和空闲连接,减轻服务器压力;设置合理的心跳周期,防止心跳定时任务积压,造成频繁的老年代 GC 而导致应用暂停;同时在数据交互的空闲期进行心跳检测可以在数据链路上产生数据,防止 NAT 老化而导致链路中断;使用 Netty 提供的链路空闲检测机制,可以有效降低系统负担。

#### 2) 断线重连

增加断线重连机制,保证每个连接能稳定连接服务器,及时收到服务器消息。服务器在网络空闲操作一定时间后,服务端失败心跳计数器加 1;如果收到客户端的 Ping 心跳包,则清零失败心跳计数器,如果连续 n 次未收到客户端的 Ping 心跳包,则关闭链路,释放资源,等待客户端重连。释放资源时,要及时删除 Channel 属性等内存中的数据,避免资源浪费;客户端要增加空闲检测与连接超时机制,当检测到连接断开时启动重连服务[8]。

#### 3) 离线消息缓存

家长端发出的指令与对应 Hash 值,需要存储到 Redis 和数据库,如果学生端设备因为断网、关机等情况无法及时接收到指令,待学生端启动应用并联网时,通过 Netty 服务器直接优先读取 Redis 的指令数据,如果未查询到 Redis 指令数据,说明 Redis 存储的指令数据已过期失效,需要从数据库获取最新的指令数据,用于应用启动时的数据初始化,需要保证学生端的指令 Hash 与数据库一致,确保能够准确完整的将家长端指令发送到学生端。

#### 4) 设置合理的线程数

对于线程池优化,主要集中在用于接收大量设备 TCP 连接、TLS 握手的 Acceptor 线程池 BossGroup 上,以及用于处理网络数据读写、心跳发送的工作线程池 WorkGroup 上。Netty 的默认值是 CPU 核数\*2,对于非阻塞 I/O,线程数并不建议设置太大,建议 I/O 线程数是[CPU 核数 + 1, CPU 核数\*2]之间。

#### 5) 合理设置接收和发送缓冲区

对于长链接,Netty 提供的 ByteBuf 支持容量动态调整,无论是接收缓冲区还是发送缓冲区,缓冲区的大小建议设置为消息的平均大小,Netty 提供了两种:

**FixedRecvByteBufAllocator:** 固定长度的接收缓冲区分配器,由它分配的 ByteBuf 长度都是固定大小的,并不会根据实际数据报的大小动态收缩。

**AdaptiveRecvByteBufAllocator:** 容量动态调整的接收缓冲区分配器,它会根据之前 Channel 接收到的数据报大小进行计算,如果连续填满接收缓冲区的可写空间,则动态扩展容量。

#### 6) 合理使用内存池

为了尽量重用缓冲区,Netty 提供了基于内存池的缓冲区重用机制。Netty 内存池从实现上可以分为两类:堆外直接内存和堆内存。由于 ByteBuf 主要用于网络 IO 读写,因此采用堆外直接内存会减少一次从用户堆内存到内核态的字节数组拷贝,所以性能更高[9]。由于 DirectByteBuf 的创建成本比较高,因此如果使用 DirectByteBuf,则需要配合内存池使用,否则性价比可能还不如 HeapByte。

#### 7) 网关限流;

为避免短时间内大量设备接入,造成网络拥堵,给服务器带来严重的性能压力;可在后台引入网关限流策略,常见的限流算法比如:计数器算法、漏桶算法和令牌桶算法等。

#### 8) 提高数据传输效率

根据家长端与学生端应用之间传输数据的类型与数据量的大小,以及功能业务的分类,可以直接和间接混合调用 Netty 服务完成数据的传输,有效提高数据的传输效率。

### 3.1.2. Netty 直接通信

对家长端与学生端的传输数据进行定义时，有一部分数据的数据量较小，不需要进行后台管理服务的校验与记录，可以采用图 5 的通信方式来进行，通过 Netty 服务器可以快速完成数据的发送与接收。

家长端、学生端应用在与 Netty 服务器建立 TCP 连接时，初始化各自的 SocketChannel，并记录应用客户端 ID 与 SocketChannel 的映射关系到 ChannelGroup，用于请求或转发时传递消息；当家长端向 Netty 服务器发出指令，Netty 服务通过 SocketChannel 的 ChannelPipeline 接受家长端的指令，并通过 ChannelGroup 查找学生端的 SocketChannel 并完成指令的转发，学生端应用就可以接收到家长端的指令，并沿原路径反馈处理结果。

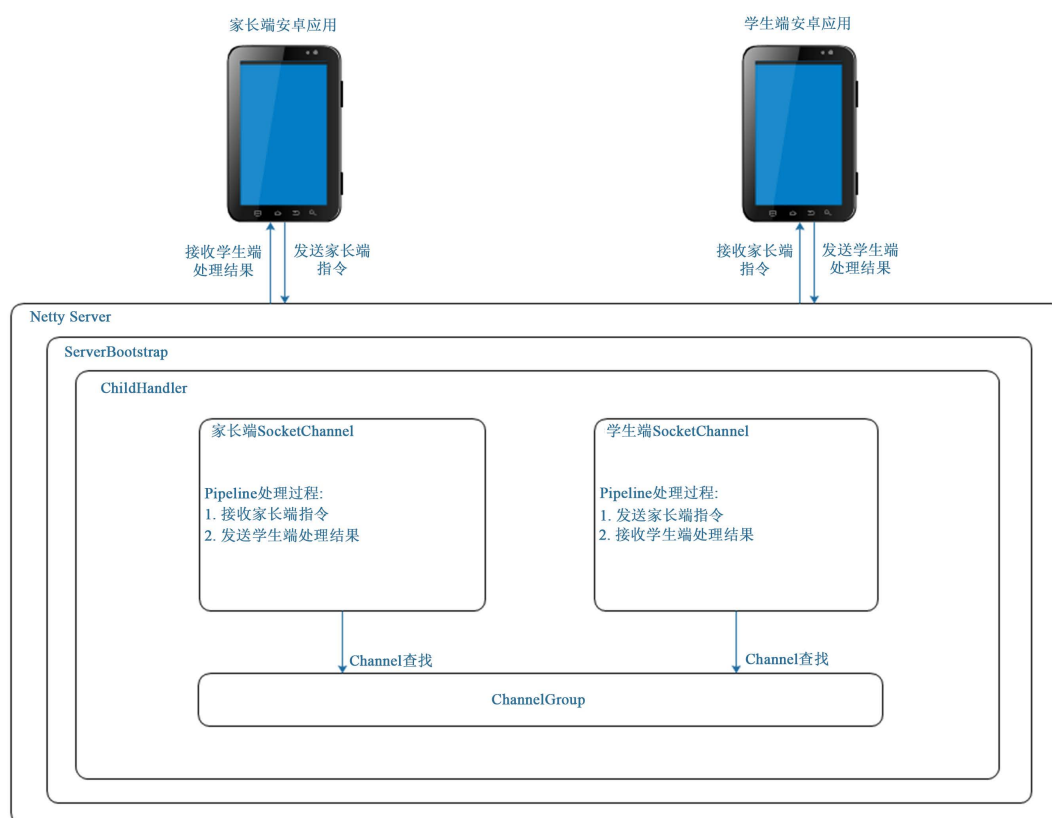


Figure 5. Diagram of Netty direct communication architecture  
图 5. Netty 直接通信架构图

### 3.1.3. Netty 间接通信

1) 学生端通过心跳线程，定时将设备信息通过 Netty 服务器，推送到 MQ 消息队列服务器，并保存到数据库，家长端可以直接通过 HTTPS 请求后台管理服务器，获取到设备的最新情况；

2) 学生端回传大文件数据，例如截屏等信息，可以通过 Netty 服务器，推送到云存储，家长端可以通过 HTTPS 请求后台管理服务器，获取到云存储的文件信息。

### 3.1.4. 服务器端——面向管控应用

设备端管控应用启动后，通过长连接与服务端建立通讯，主要接收服务端发送过来的指令，收集设备使用信息，执行管控指令。它需要实现的功能主要是服务端管控指令具体的执行过程，功能如下：

**服务端：**发送指令；

**客户端：**接收/执行指令；

#### 1) 信息收集

用户登录学生账号后，进入学生模式，收集设备信息(设备名称、设备型号、系统版本、SN 等)和账号信息(账号登录退出时间、账号在线状态、在线时长等)，回传给服务器。

#### 2) 时间管理

管控应用根据服务端的时间管理方案，在指定的时间范围内对相关应用进行暂停，启用以及对设备进行锁屏或关机操作[10]。

#### 3) 白名单管理(与黑名单管理互斥)

管控应用根据服务端设置的白名单列表，授予白名单中 APP 和网站(主要是设备中自带的工具类和学习类应用)允许访问的权限[11]，非白名单列表中的应用都将被限制访问。当用户访问非白名单列表中应用时，提醒用户“访问受限，您没有某应用访问权限”。

#### 4) 黑名单管理(与白名单管理互斥)

管控应用根据服务端设置的黑名单列表，将限制黑名单中 APP 和网站(主要是社交类、游戏类和娱乐类应用)的访问权限[12]，黑名单列表中的应用都将被限制访问。当用户访问黑名单列表中应用时，首次提醒用户“某应用是受限应用，您没有访问权限”，并开启系统锁屏 30 秒。若超过一定频次，向用户发出警告信息“您已多次访问受限应用，超过 X 次，系统将关机”，开启系统锁屏 5 分钟。

#### 5) 远程管理

**加锁：**接收到“加锁”指令后，管控应用对设备进行锁屏操作，设备不可操作；

**解锁：**接收到“解锁”指令后，管控应用对设备进行解锁操作；

**远程关机：**客户端接收到“关机”指令后，管控应用对设备进行关机操作；

**回复消息：**客户端收到服务端的聊天信息后，可以及时进行回复。

#### 6) 功能定制

客户端接收到服务端的定制策略后，管控应用根据策略中的指令逐一进行执行[13]，对设备进行管控；可定制功能包括：开机启动、默认模式、休息时间开关、远程管理开关、锁屏开关和关机开关。

### 3.1.5. 服务器端——面向家长应用

家长控制应用主要通过功能面板去定制一些针对设备端的管控指令，然后发送给客户端，让客户端管控应用执行这些指令，达到远程控制设备的目的。主要实现的功能如下：

#### 1) 用户管理

包括注册、登录和退出系统、修改账号信息(密码、职务、工作单位、手机号、邮箱等)、添加、修改并删除子账号信息。

#### 2) 设备管理

用户可以添加手机、平板等管控设备信息(设备名称、设备型号、系统版本、SN 号、设备上网起止时间、设备在线状态、子账户在线时长等)，并对设备信息进行修改和删除。

#### 3) 时间管理

用户可以添加一个时间管理方案(如：读书日，节假日)，设置设备可使用的起止时间段，超出这个时间段，可以设置关机或锁屏；也可以设置间歇性锁屏，每间隔 45 分锁屏 15 分钟；用户可以对方案和时间段进行添加、修改和删除操作。

#### 4) 白名单管理(与黑名单管理互斥)

用户可以根据需要维护 APP 和网站管控白名单，可以对白名单中的 APP 和网站进行添加、修改和移



除操作。白名单应用主要是系统自带的一些学习类和工具类的 APP 和网站。

### 5) 黑名单管理(与白名单管理互斥)

用户可以根据需要维护 APP 和网站管控黑名单,可以对黑名单中的 APP 和网站进行添加、修改和移除操作。黑名单应用主要是一些社交类、游戏类、娱乐类、资讯类和购物类 APP 和非法网站。

### 6) 远程管理

**加锁解锁:** 用户可以对学生所持有设备设置一个密码进行加锁,此时设备端处于锁屏状态;用户对 学生所持设备进行解锁时,属于加锁密码,对设备端进行解锁;

**远程关机:** 用户打开“远程关机”按钮,发送指令到客户端,可以对 学生所持设备进行关机;

**发送消息:** 用户可以根据学生上网游戏时间,发送消息提醒学生上网时间不能太长;用户可以根据 学生学习时间,发送消息告诉学生可以适当休息啦。

### 7) 功能定制

用户可以按照模板一次性定制一套完整的管控策略,然后把定制结果发送给设备端“管控应用”, 让“管控应用”去执行这些指令。

**开机启动:** 设置设备端是否开机自启动“管控应用”,“启用”或“禁用”;

**默认模式:** 设置设备端默认打开模式为“学生模式”;

**休息时间开关:** 可灵活设置设备每使用 X 分钟,休息 Y 分钟;

**远程管理开关:** 设置服务器是否可以远程管理设备端,“启用”或“禁用”;当启用时,可以进一 步设置是否可以远程关机,是否可以设置加锁解锁,是否开启发送消息;当禁用时,远程关机和加锁解 锁功能都不可用[14];

**锁屏开关:** 设置设备端是否锁屏同时关闭桌面;

**关机开关:** 设置 X 分钟后自动关机。

### 8) 家长帮助

**用户操作指南:** 指导用户设置账号、对设备进行远程管控的描述文档,比如:时间管理方案设置步 骤,黑名单/白名单设置步骤,功能定制须知等;

**常见问题收录:** 对一些常见问题的解答,比如:无法登录或注册,远程管理失效,如何找回密码等。

### 9) 用户反馈

收集用户在使用系统时遇到的任何问题,管理员可对问题进行回复解答,也可以将热评问题置顶。

## 3.2. 客户端实现

### 3.2.1. 家长端

这里展示下家长端应用的功能主界面与个人中心,功能界面可以快速进入对应的功能设置,在功能 项下侧可以实时监控学生端安卓设备的使用情况,更直观的帮助用户来管控学生的学习情况。其主要提 供对家长的注册、绑定学生端、设置管控策略及其他功能,分为两个模块:账户管理和设置管控功能。 通过这些模块可以实现对学生端的管控功能,如设置应用程序的限时、远程辅导等[15]。

个人中心即账户管理模块是对家长账号进行管理,也是家长端的个人信息页,展示家长昵称、账 户 ID 和其他信息,可以实现对学生端的绑定、解除绑定设备、显示绑定列表以及查看管控历史的功能, 最重要的功能就是绑定学生设备,实现家长端与学生端的安全网络通信与映射关系。其主要页面如图 6 所示。

功能主界面即设置管控模块是家长端的主要功能,实现发送管控策略到服务器,再由服务器查找对 应的学生端设备并下发管控指令,最后由学生端解析配置文件并执行对应管控指令;其主要分为如下功

能：查看设备位置功能，家长若需要了解学生此时的位置信息，点击此图标即可得到学生端设备此时所在位置；应用管理功能主要是对学生端设备进行管理，如图 7 所示，设备限时和允许显示的应用都是在这里进行设置，如对学生端某些指定应用设置限时，当学生使用达到限时后，无法再继续使用，只能向家长端进行申请更长时限或输入密码进行解除限时；制定计划功能，如家长需要给孩子指定一个学习计划，完成语文作业，当孩子完成后，会向家长端进行反馈；查看桌面功能，如果家长想要了解学生端目前的桌面使用情况，家长使用此功能，学生端会生成一张本地快照截图并传回家长端。

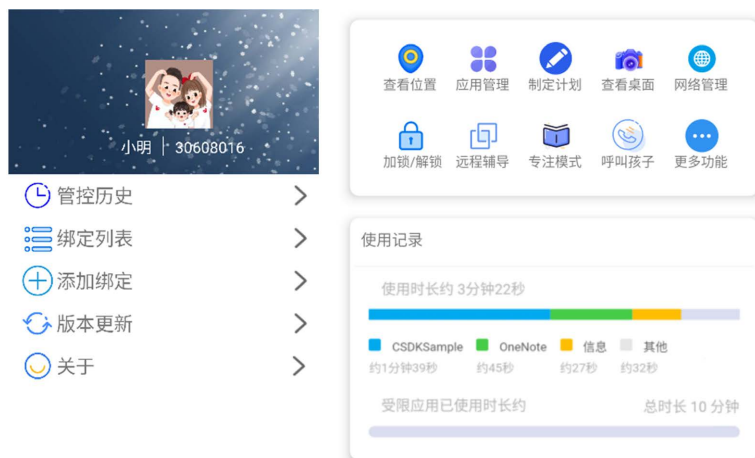


Figure 6. Diagram of the parent side application function display  
图 6. 家长端应用功能示意图



Figure 7. Diagram of application management page on parent side  
图 7. 家长端应用管理页面

### 3.2.2. 学生端

学生设备端主要有信息亭模式、应用程序限额控制及远程控制三个模块，信息亭模式(Kiosk Mode)是该设备的第二套桌面系统，可以指定某些应用程序可以显示在桌面上供使用者使用。应用程序限额控制可以实现对指定应用程序进行时间限额配置，相关应该程序使用限额到达后锁定应用程序。远程控制

模块则配合家长端对信息亭模式和应用程序限额控制进行远程配置(图 8)。

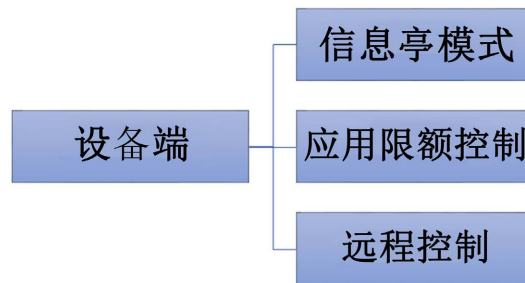


Figure 8. Diagram of the main functions of student terminal  
图 8. 学生端主要功能示意图

学生端采用经典 MVC 方式进行程序架构, 分开三个层次进行部署, 同时采用了多进程的方式分割了页面操作和后台通信服务, 如图 9 所示:

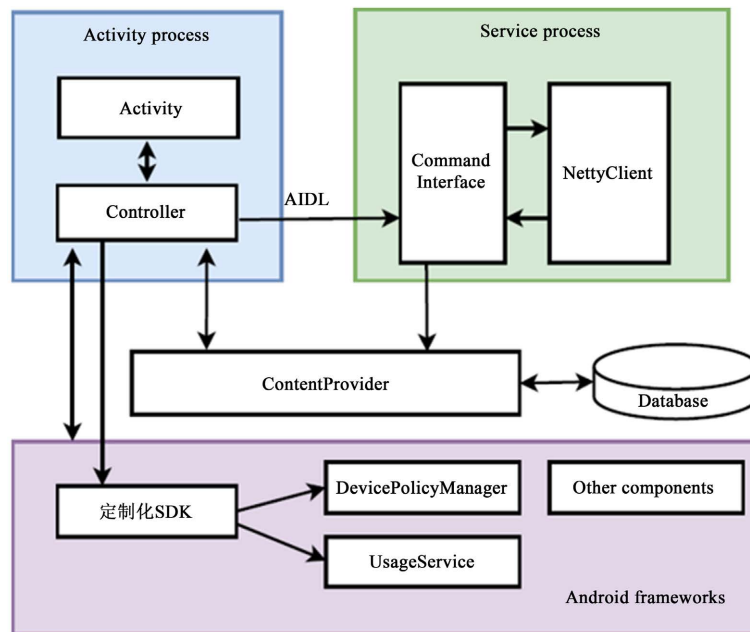


Figure 9. Diagram of Student application architecture  
图 9. 学生端应用架构图

在学生端应用架构中, 使用 ContentProvider 封装数据库底层, 充分使用 ContentProvider 的 Notify 机制解决了 Service 进程和 Activity 进程之间的通信, 同时也解决了不同页面与组件之间的通信问题。Service 包含 Netty 组件, 并设置成为前台服务, 实现与 Activity 操作解耦, 提高 Service 存活率。

在 Netty 组件之上封装了 CommandInterface。CommandInterface 提供了对外统一的操作接口, Activity 进程可以通过 AIDL 与其进行通信, 将学生端指令发送到家长端。CommandInterface 对来自家长端的指令进行了解析, 并通过 ContentProvider 将数据写入了数据库, 同时通过 ContentProvider 通知到 Activity 进程。

### 1) 信息亭模式(Kiosk Mode)

信息亭模式，也称作 Kiosk 模式。可以将系统的状态栏，通知栏，任务栏进行锁定或隐藏显示，可以将使用者的操作区域规范在应用程序内部，减少了外部干扰同时降低了误操作率，非常受商业客户的青睐，常被用在自助服务器(如自助点餐机，超市自助收银机，银行 ATM 取款机)上。在个人设备上也有相当多的应用(如 Windows 10 上就提供了 KioskMode)。

学生端使用了 Kiosk 模式。在 Android 操作系统中，Kiosk 模式是通过 DevicePolicyManager 的 setLockTaskPackages 方法来提供。使用 setLockTaskPackages 方法需要激活 DeviceOwner，在 Android 平台上，一台机器只能安装一个激活 DeviceOwner 的应用程序，清除 DeviceOwner 还需要恢复出厂设置，这些条件决定了学生端不能使用标准的 DevicePolicyManager 接口。本文使用了内置定制化 SDK 解决了这个问题，具体实现如下：

- 在 SystemServer 中增加一个自定义的系统级别的 SystemService: MySdkService;
- 在 MySdkService 中实现了 setLockTaskPackages 接口;
- 在系统中实现一个自定义的 Manager: MySdkManager, MySdkManager 通过系统调用的方式将 MySdkService 的接口暴露出来。

在 Android 上借助定制化 SDK 提供的 API，可以将正在运行的应用程序任务进行锁定，同时还可以锁定状态栏、通知栏，以达到信息亭模式的效果(图 10)。

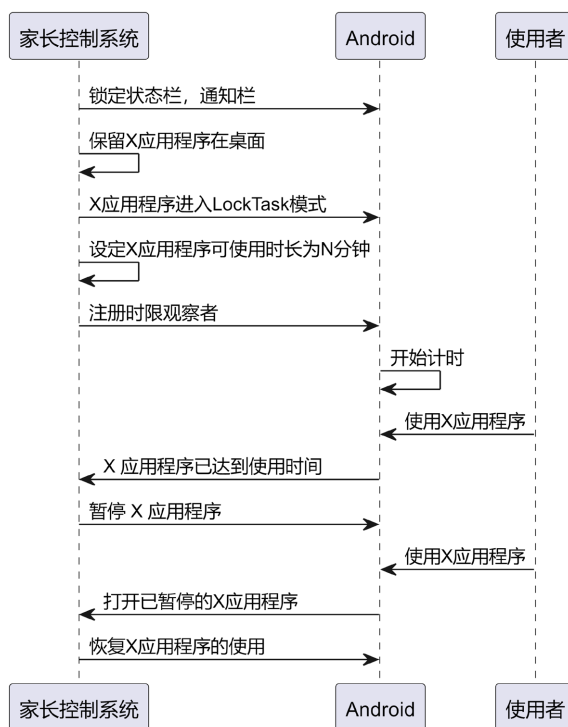


Figure 10. Sequence diagram of Kiosk mode  
图 10. 信息亭模式时序图

## 2) 应用限额控制

应用限额控制，即控制电子设备使用时长，大致分为两类，包括控制设备使用时长和控制应用程序使用时长。前者对整个设备进行控制，只要时长到达所设定的额度就锁定设备或是关闭设备，限制学生继续使用。后者则是分开控制逻辑的，仅对指定的，可能对学生造成影响的应用程序进行时间额度的设

定，当指定应用程序的使用时间到达额度后即刻停止该应用程序的运行。

“应用程序使用时长统计”与“限制应用使用时长”两项功能，调用了 Android UsageService 提供的 registerAppUsageLimitObserver 接口，该接口可以向系统注册一个监听器来监听相关应用程序的使用时长，当注册应用的使用时长达到所设定的时长时，系统将会回调监听器，来通知应用程序。

学生端的应用程序限制功能，使用了 Android 提供的 PackageManager 提供的 setPackageSuspended 方法，使用此方法可以将应用程序设置为 Suspended 状态，Suspended 状态的应用程序是无法被打开和调起的；通过组合 registerAppUsageLimitObserver 和 setPackageSuspended 学生端实现了应用使用时长限制。

- 学生端离线设置配额(图 11)

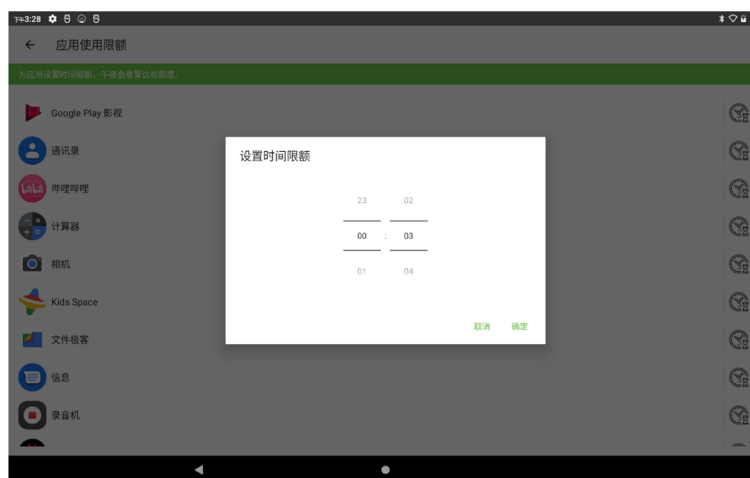


Figure 11. Diagram of offline quota setting on student side  
图 11. 学生端离线设置配额示意图

- 当学生端使用时长耗尽，如果学生还想增加配额，家长可以远程增加时长，或者通过学生平板输入管理密码，临时增加固定使用时长(图 12，图 13)。

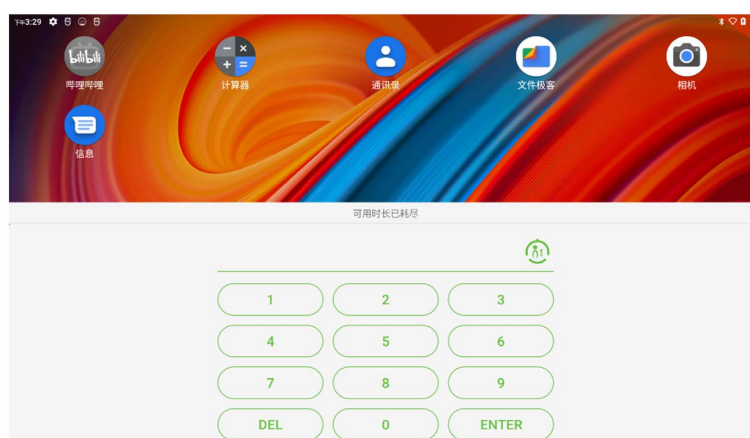


Figure 12. Diagram of temporarily increasing quota entrance on student side  
图 12. 学生端临时增加配额入口示意图

### 3) 远程控制

远程控制主要辅助家长端对设备端信息亭模式和应用程序时间限额进行控制。远程控制模块使用 Netty 长连接到服务器，家长端任何的远程控制操作，如配置桌面上可显示的应用程序，配置应用程序使用时间限额，禁止调整设备音量等，都可以在毫秒时间内部署到设备端，完成家长对设备的远程控制 (图 14)。

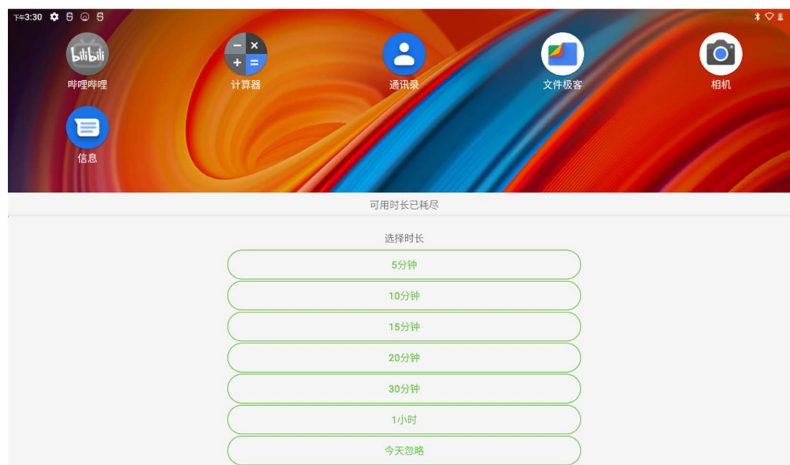


Figure 13. Diagram of temporary quota increase on student side  
图 13. 学生端临时增加配额示意图

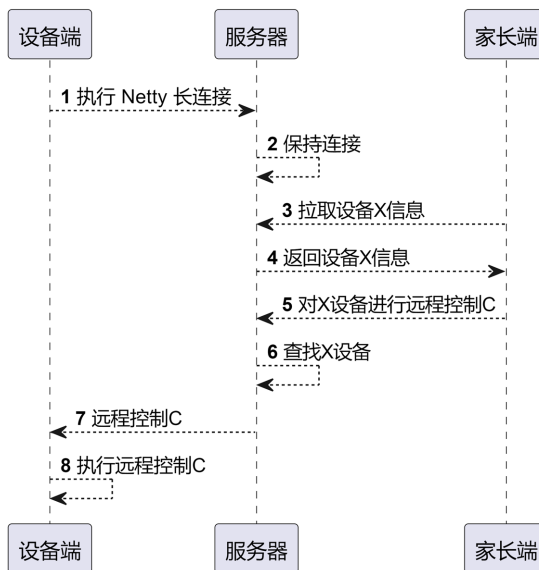


Figure 14. Sequence diagram of the remote control  
图 14. 远程控制时序图

#### 4. 结语

本文设计了一种基于安卓平板的家长控制系统，通过家长端与学生端的配合使用，可以覆盖家庭、学校多种场景，在学生使用安卓平板增加学习便利性的同时，保证学生良好的学习习惯，依托于信息亭模式、使用配额、远程控制等功能，为家长解决后顾之忧，通过云平台的有力技术支撑，为更多的家庭提供优质的服务，保证各项功能的稳定准确执行，在疫情下居家办公与学习成为常态，家长工作之余也

可以更加有效地支持学生的学习, 实时了解学习动态, 让家教平板发挥更大的作用。

## 参考文献

- [1] 明日科技. *Android 开发从入门到精通*[M]. 北京: 清华大学出版社, 2017.
- [2] 李林锋. *Netty 权威指南*[M]. 第 2 版. 北京: 电子工业出版社, 2015.
- [3] 倪炜. *分布式消息中间件实践*[M]. 北京: 电子工业出版社, 2018.
- [4] 黄健宏. *Redis 设计与实现*[M]. 北京: 机械工业出版社, 2014.
- [5] Widyatri, I., *et al.* (2018) Parental Control Application on Android Platform. *Library Hi Tech News*, **35**, 18-24. <https://doi.org/10.1108/LHTN-08-2017-0057>
- [6] Lim, J.D., Choi, B.C., Han, S.W., *et al.* (2013) Adult Contents Analysis and Remote Management Framework for Parental Control Based on Android Platform. In: Jeong, H.-Y., *et al.*, Eds., *Advances in Computer Science and Its Applications*, Springer, Berlin, 161-166. [https://doi.org/10.1007/978-3-642-41674-3\\_24](https://doi.org/10.1007/978-3-642-41674-3_24)
- [7] 张状状, 杜星熠, 王啸, 胡静, 宋铁成. 基于 Netty 的 MQTT 协议消息系统的设计与实现[J]. *信息化研究*, 2022(3): 48.
- [8] 胡喜明, 胡淼. 基于响应式的服务推送框架设计[J]. *计算机工程与设计*, 2021, 42(8): 2137-2143.
- [9] 万建民. 基于 Netty 和 Redis 应对高并发场景的研究和实现[D]: [硕士学位论文]. 南京: 南京邮电大学, 2022.
- [10] Wisniewski, P., Ghosh, A.K., Xu, H., *et al.* (2017) Parental Control vs. Teen Self-Regulation: Is There a Middle Ground for Mobile Online Safety? *ACM Conference on Computer Supported Cooperative Work & Social Computing*, Portland, 25 February-1 March 2017, 51-69. <https://doi.org/10.1145/2998181.2998352>
- [11] 宁强. *Android 手机的家长控制*[D]: [硕士学位论文]. 南京: 南京大学, 2013.
- [12] 张晓. 基于 Android 平台家长控制软件的设计与实现[D]: [硕士学位论文]. 济南: 山东大学, 2014.
- [13] 蒙晓红. 学生安全监护系统的设计与实现[D]: [硕士学位论文]. 成都: 电子科技大学, 2018.
- [14] 贾亚龙. 基于家长模式的智能家居控制系统设计[D]: [硕士学位论文]. 合肥: 中国科学技术大学, 2017.
- [15] 张晓明. 基于 uni-app 和 Android 的学生手机管控系统的设计与实现[D]: [硕士学位论文]. 兰州: 兰州大学, 2020.