

文章编号: 2095-2163(2020)01-0173-06

中图分类号: TP311-5

文献标志码: A

# 基于 Android 的防走失系统的设计与实现

张 斌, 邬晓静, 邵 想, 张 帅

(郑州大学 软件学院, 郑州 450002)

**摘 要:** 针对当前老人儿童走失现状以及市面上依赖于第三方设备的定位产品价格成本高、实时性低等局限性, 为提高找回效率与实用性, 提出一种基于 Android 的防走失系统。系统 Android 客户端基于 kotlin 语言, 采用 Okhttp、Gson 等框架与 MVP 设计模式完成实现, 同时数据采用 MD5 算法进行加密处理缓存到本地, 保证用户数据的安全性及友好的人机交互性。系统 Web 服务器端基于 Java 语言, 由 Spring Boot 框架搭建而成; 系统数据库通过 MongoDB 来保存数据; 系统提供可利用的百度地图 API 接口。客户端与服务器端使用 Socket 通信为主、Http 为辅的通信机制, Socket 通信具有传输时间短、性能高、数据安全性强等优点。系统可实现实时定位、电子围栏、强制录音等功能, 完成对老人儿童的安全监护, 做到自动报警, 保障老人儿童的出行安全。

**关键词:** Android; Spring Boot; 网络通信; 实时定位; 轨迹回放; 位置校验算法

## Design and implementation of anti-loss system based on Android

ZHANG Bin, WU Xiaojing, SHAO Xiang, ZHANG Shuai

(Software School, Zhengzhou University, Zhengzhou 450002, China)

**[Abstract]** In order to improve the recovery efficiency and practicability, an Android-based anti-loss system is proposed, aiming at the current situation of missing elderly and children and the limitations of positioning products relying on third-party devices in the market, such as high price cost and low real-time performance. The Android client of the system is based on kotlin language, and uses Okhttp, Gson and other frameworks and MVP design mode to achieve the design. At the same time, MD5 algorithm is used to encrypt and cache the data locally, ensuring the security of user data and friendly human-machine interaction. The Web server of the system is based on Java language and built by Spring Boot framework. The system database saves data through MongoDB. The system provides available Baidu map API interface. Socket communication between client and server is the main mechanism, while Http communication is the auxiliary, which has the advantages of short transmission time, high performance and good data security. The system can realize real-time positioning, electronic fence, mandatory recording and other functions, complete the safety monitoring of the elderly and children, and send out automatic alarm, so as to ensure the safety of the elderly and children.

**[Key words]** Android; Spring Boot; network communication; real-time location; track playback; location verification algorithm

## 0 引言

基于 Android 的防走失系统是指通过移动设备对人的位置坐标进行 GPS 定位, 通过位置校验算法提取精确位置信息, 并对实时上传的数据进行分析处理, 达到实时监控的目的。该技术在车载导航、测绘、军事等领域已有广泛应用。老人儿童安全问题一直是一个社会热点问题, 艾瑞发布的《2019 年中国亲子陪伴质量研究报告》中指出, 2008 年来中国宏观经济增速放缓, 已婚人群养老与抚养孩子的压力更加明显<sup>[1]</sup>, 陪伴老人与孩子的时间也越来越少。虽然市面上已有的防走失定位产品很多, 但依赖于第三方设备的定位产品在续航时间、价格成本、快速升级等方面有明显的局限性。在 2017~2018 年中, 智能手机在所有年龄段的普及率、覆盖率不断

增长<sup>[2]</sup>, 超过九成的人不会在进行家庭活动时关闭自己的移动设备<sup>[3]</sup>。智能手机具有系统升级快, 价格低廉等优点, 而且 Android6.0 可使设备续航时间提升 30%。基于 Android 的防走失系统能够精确定位、自动报警、帮助用户快速导航到老人儿童身边, 保障老人儿童出行安全, 满足人们需求, 同时其在各个层面相互独立便于后续开发与维护。

## 1 Android 客户端架构设计

系统 Android 端使用 IntelliJ IDEA 2019 为开发工具, 基于 kotlin 语言, 结合 MVP (Model-View-Presenter) 设计模式, 采用 Nv-websocket-client+Okhttp+Gson 框架, 在使用 Android SDK 27 的基础上, 兼容 Android6.0 及以上版本, 实现了实时定位、轨迹回放、强制录音、电子围栏等功能。

**作者简介:** 张 斌(1998-), 男, 本科生, 主要研究方向: 安卓开发; 邬晓静(1996-), 女, 本科生, 主要研究方向: 软件开发与应用; 邵 想(1994-), 男, 本科生, 主要研究方向: 移动互联网开发; 张 帅(1997-), 男, 本科生, 主要研究方向: Web 服务器开发。

**通讯作者:** 邬晓静 Email: 2413352439@qq.com

**收稿日期:** 2019-10-14

## 1.1 Application 设计

Android 系统会为每个程序运行时创建一个单例 (singleton) 模式的 Application 类,且其生命周期等于此程序生命周期,所以可以通过 Application 来进行权限申请、数据共享、数据缓存等操作。程序首次运行时申请如下权限。

(1) 授予用于进行网络定位的权限。此时需用到设计代码为:

```
android.permission.
```

```
ACCESS_COARSE_LOCATION
```

(2) 授予该程序向外部存储器写入数据的权限。此时需用到设计代码为:

```
android.permission.
```

```
WRITE_EXTERNAL_STORAGE
```

(3) 授予该程序录制声音的权限。此时需用到设计代码为:

```
android.permission.RECORD_AUDIO
```

(4) 程序在手机屏幕关闭后,后台进程仍然运行。此时需用到设计代码为:

```
android.permission.ACCESS_FINE_LOCATION
```

等。

在 Application 中建立 2 个 HashMap 用于 Activity 之间进行数据传递和缓存用户数据,同时将缓存的数据经过 MD5 算法进行加密处理缓存到本地,既保证用户数据安全性,也预防出现程序意外中断导致数据丢失情况,避免用户打开应用重复登录,提高人机交互友好性,实现后台实时定位的功能。

在 Application 的 onCreate() 的方法中通过 LogUtil.logoff = false,设置屏蔽打印;LogUtil.level = log.ERROR,指定把高于或等于 Error 的信息保存到 sdcard 日志文件中;LogUtil.saveDirName = "/GuardianshipModify/log/" 指定错误信息日志目录。LogUtil.trace (int type, String tag, String msg, boolean isContinueWrite) 中 type: log 的类型 (ERROR, WARN 等), tag: 输出标志, msg: 输出日志, isContinueWrite: true (默认), 表示继续在原有文件写入,若为 false 则表示创建一个新的覆盖原来的文件。如果原来并没有创建过,则无论是 false, 还是 true 都会创建一个新文件。globalExceptionHandler.setUncatchExceptionHandler() 可以回调处理程序崩溃后自定义的用户操作,如上传错误信息,保存数据信息等操作。

EventBus 提供了 PostThread (当前线程)、MainThread (主线程)、BackgroundThread (后台线程)

和 Async (异步线程) 四种线程模式,是 Android 下高效的发布/订阅事件的消息总线。因此,本系统采用此对象代替传统的 Intent, Handler, Broadcast 或接口函数在 Fragment、Activity、Service 和线程之间传递数据并执行方法。EventBus 有 3 个主要元素,分别是: Event (事件)、Subscriber (事件订阅者,接受特定的事件) 和 Publisher (事件发布者,用于通知 Subscriber 有事件发生)。

## 1.2 Socket 和 Http 网络通信

### 1.2.1 客户端发送请求设计

NetworkManager 通信管理类是一个重量级类,采用单例模式,用于管理频繁出现的网络请求。该管理类把请求分发给目标业务逻辑组件去处理,目标业务逻辑组件根据相应的请求调用相应的方法去执行,通过 JSON 接口和继承的方式利用 Gson 框架将请求对象和返回对象设计为 JSON 对象。由于 Socket 通信<sup>[4]</sup>具有传输数据量小 (费用低)、传输时间短、性能高、适合于客户端和服务器之间信息实时交互、可以加密、数据安全性强等优点,因此研究采用以 Socket 通信为主、Http 通信<sup>[5]</sup>为辅的通信机制。请求过程如下:

**Step 1 准备请求。**在请求对象中加入 seqId (实现当服务器响应长连接请求时能够找到对应的回调) 和 reqCount (记录请求超时次数) 两个参数,同时添加超时任务并且将该请求的回调添加到回调集合。

**Step 2 开始请求。**请求成功或者失败都通过 seqId 找到对应回调执行并从回调集合中移除该回调,再取消超时任务。如果超时则判断当前请求次数,当前请求次数小于或等于 3 次时则再次通过 WebSocket 发送请求;当前请求次数大于 3 次时则走 Http 补偿通道,并根据请求成功或失败情况执行对应回调。客户端发送请求过程图如图 1 所示。

### 1.2.2 授权、心跳和重连循环设计

为实现服务器端向客户端主动发送通知,客户端需要在用户进行登录后尝试建立连接,建立连接成功后进行授权,授权就是发送一个携带用户信息的请求,而在服务器端通过这个请求后验证用户信息,验证成功后服务器端就知道当前长连接属于哪个用户。授权成功后开始心跳。心跳是指每隔一段时间,服务器端发送请求,如果服务器端有响应,就认为这条连接是稳定的。心跳连接成功后进行数据同步,如果心跳连续失败三次或尝试建立连接失败则开始继续尝试重连。当重连成功后会再次进行授

权, 然后再次开启心跳, 至此形成了一个循环。如果多次重连失败后系统会向用户发送通知, 提醒用户检查当前网络状态。授权、心跳和重连循环过程如图 2 所示。

图会实时显示当前位置和正前方的方位。用户产生的运动信息会被存储到对象中, 通过 SecureData 类进行数据格式化, 而后被系统存储到本地的 track.xml 文件里。系统根据上传机制将 track.xml 文件上传到服务器。

监护端可以查看被监护端当前位置和运动轨迹, 位置信息由 Download 类从服务器上下载 track.xml 文件, 经过 SecureData 类解析为数据信息。系统通过 mapView 对象将解析后的数据绘制为被监护端的位置点和运动轨迹。

### 1.3.2 运动轨迹监测

系统调用百度地图 API 接口时存在定位精度<sup>[9]</sup>的问题, 可能会产生错误的位置信息。为了防止错误的位置信息被记录到运动轨迹中, 研究特别设计了位置校验算法。

当设备信号不良时, 百度地图定位时会出现跳跃现象, 地图定位会产生一定程度的偏差, 这个错误的定位点被称为跳跃点。跳跃点的位置和实际位置相差约 5~10 m (1 m 相当于 0.000 01° 经纬度<sup>[10]</sup>), 可根据经纬度的瞬时变化检测当前定位点是否属于跳跃点。当被监护端处于高速移动时, 间隔定位点的距离可能会和跳跃点变化相似。但跳跃点是瞬时变化, 高速移动是持续性变化, 据此可区分跳跃点和高速点。研发判断是否为有效轨迹点代码详见如下。

```

var highSpeedSign = 10, error = 0
val LatOffsetLimit = 0.000 1, LonOffsetLimit = 0.001
var offsetList; MutableList < LatLng >,
motionList; MutableList < LatLng >
fun savePos(currentPos; LatLng, lastPos; LatLng) {
//儿童未进行移动
if (currentPos == lastPos) return
var offsetlati = Math. abs(currentPos.
latitude - lastPos.latitude)
var offsetlong = Math. abs(currentPos.
longitude - lastPos.longitude)
val savePos = offsetlati < LatOffsetLimit
&& offsetlong < LonOffsetLimit
if (savePos) { // 有效轨迹点
if (error != 0) {
if (highSpeedSign < error) // 判定此
前运动为高速运动

```

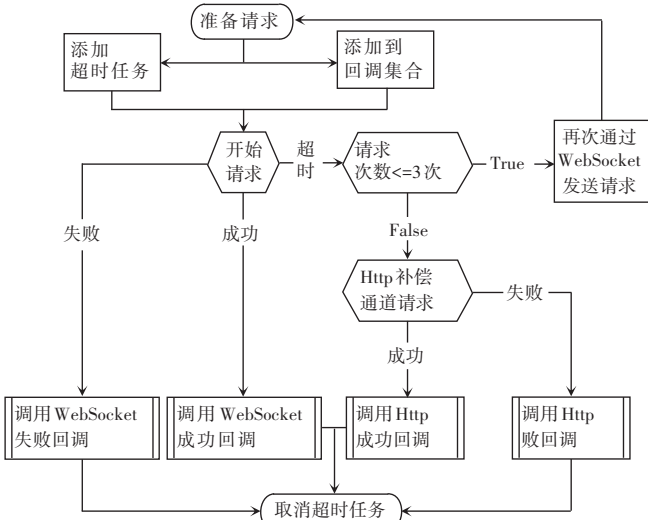


图 1 客户端发送请求过程图

Fig. 1 Client sending request process diagram

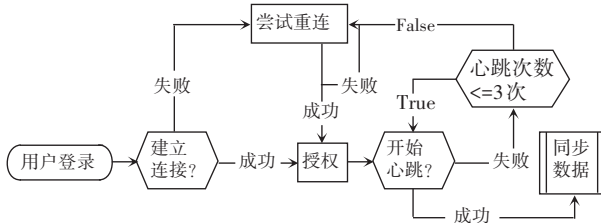


图 2 授权、心跳和重连循环过程图

Fig. 2 Diagram of authorization, heartbeat, and reconnection cycles

### 1.3 地图搭建与运动轨迹监测

Android 端依托于百度地图 Android SDK, 采用 BD09 坐标系实现绘制定位。BD09 坐标系是在 GCJ02 坐标系 (国家测绘局)<sup>[6]</sup> 基础上再次加密形成的, 具有高度的可靠性和安全性, 可以更好地保护用户个人隐私, 防止位置信息泄露。百度地图服务具有高精度、覆盖广、功耗低、支持电子围栏和室内定位等优点, 并为开发者提供了简单易用且功能丰富的操作接口。

#### 1.3.1 地图绘制和数据处理

项目导入百度地图 jar 包<sup>[7]</sup> 并在 Application 类中初始化 Android SDK<sup>[8]</sup>。SDK 会产生 mapView 和 baiduMap 两个对象, 分别用于显示和控制地图元素。

被监护用户登录 App 之后, 系统会为其创建一个对象, 并初始化运动轨迹。用户使用系统期间, 地

```

for (i in offsetList)
    motionList.add(i)
    error = 0 // 清空错误轨迹点信息
    offsetList.clear()
emotionList.add(currentPos)
} else { // 无效轨迹点
    error = error + 1
    offsetList.add(lastPos) } }

```

## 2 Web 服务器端架构设计

Web 服务器端基于 Java 语言,采用 Spring Boot 框架<sup>[11]</sup>,使用百度地图 API 接口,前端使用 Bootstrap 框架,数据库使用 MongoDB<sup>[12]</sup>,来实现数据分析、信息管理、定位与录音管理、地图展示等功能。

Spring Boot 框架简化新 Spring 应用的初始搭建及开发过程,不再需要定义样板化配置,也无需部署 WAR 文件,简化 Maven 配置提供生产就绪功能,为客户提供数据访问接口和处理机制<sup>[13]</sup>。客户端通过 URL 调用 Spring Boot 方法的过程如下:

(1) 指定 Controller 方法名,并添加 @RequestMapping(兼容接受 POST 和 GET 请求)注解。

(2) 通过 @RequestParam 和 @PathVariable 设置参数。

(3) 客户端通过 HttpGet 或者 HttpPost 请求访问服务端资源。

(4) 服务端接受请求并通过 @ControllerAdvice 进行全局错误验证,并返回结果。

系统数据服务器采用 Spring Boot 内置 Tomcat,连接 DataSource 数据源<sup>[14]</sup>。使用 MVC 分层,各个层面单独开发并提供访问接口(TrackService、PositionDao 等),便于后期开发和维护。具体分层如下:

(1) View 层。根据接收到的数据展示页面给用户。

(2) Controller 层。响应用户请求(@RequestMapping(value = "/updatesetting")、@RequestMapping(value = "/modifyGuardian")等)。

(3) Service 层。即为业务逻辑层,通过调用 DAO 层的底层数据操作完成用户访问所要求的业务逻辑。

(4) DAO 层。把数据放到持久化的介质中,提供增删改查<sup>[15]</sup>操作。

(5) Module 层。存放实体类,并与数据库中的

属性值基本保持一致。

由于客户端需要频繁访问服务端,采用 Spring Boot 线程池能够显著提升服务端的稳定性。Spring Boot 使用 ThreadPoolExecutor 线程池 + Queue 队列。设定线程最大数量和队列大小,当周期上传的线程池满,就进入缓冲队列,线程结束或超时之后,就从线程池中删除线程,从缓冲队列中按序调用新线程。Spring Boot 线程池流程如图 3 所示。

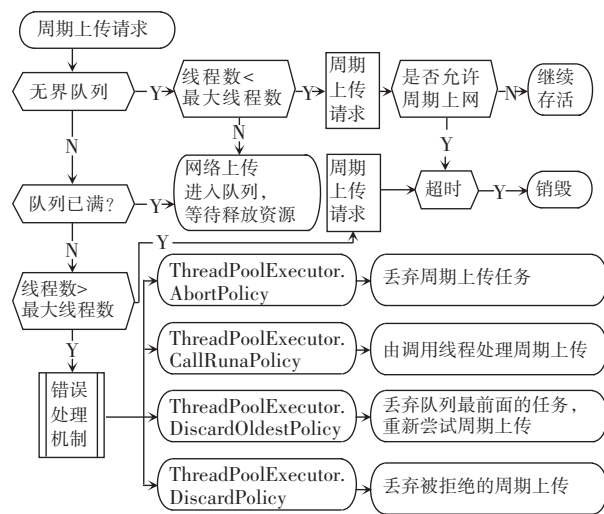


图 3 Spring Boot 线程池流程图

Fig. 3 Spring Boot thread pool flowchart

### 2.1 数据访问与服务设计

为实现对用户多条定位信息、轨迹信息以及录音信息的高效管理,系统设计采用非关系型数据库 MongoDB。MongoDB 具有查询速度快、高并发(可达 2 万并发)、高容量(支持 10 TB 以上数据量)的优点。MongoDB 能有效提高系统抗压性。MongoDB 在项目中使用过程如下:

(1) 在 pom 文件引入 spring-boot-starter-data-mongodb 相关依赖。

(2) 在 application.properties 中添加配置:  
spring.data.mongodb.uri = mongodb://name:pass@localhost:27017/test。

(3) 创建 Track、Position 等实体。

(4) 实现 TrackDao、PositionDao 等的增删改查操作(加入 @Component 注解)。

数据控制层提供的数据服务接口返回格式是 application/json,数据接口通过方法注解提供访问方式、查看轨迹方法 @RequestMapping(value = "/trackplayback") 和上传定位录音方法 @RequestMapping("/upload") 等。服务器数据管理如图 4 所示。

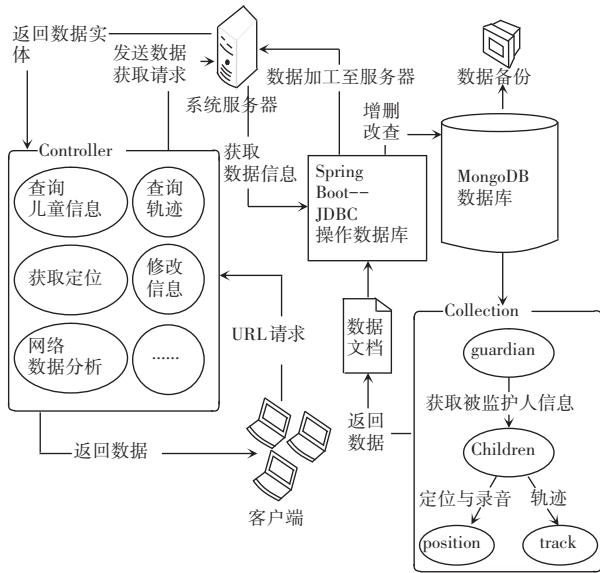


图 4 服务器数据管理图

Fig. 4 Server data management diagram

### 2.2 定位录音与轨迹回放设计

通过 HTML5+JavaScript+百度地图 API 实现定位录音信息管理和轨迹回放功能。

通过请求对象中的 account 和 datetime 参数查询被监护人的信息并加工为 position 对象返回给前端,View 层将 position 对象转化为 JavaScript 数组作为标注,使用百度地图 API 中 BMap 的 Marker 添加标注,并通过 BMap 中 InfoWindow 为标注添加信息,信息以标注形式展示,点击可查询其对应的地点名称(精确到街道、门牌号)和同步上传的录音,录音文件通过 MongoDB 数据库访问服务器录音文件下载播放。

轨迹回放则通过被监护人 id 查询数据并格式化为 List<Track> track 对象,应用层将 track 对象转化为 String 对象,通过分隔符“,”将经纬度分开并通过 BMap 的 Polyline 方法逐一绘制轨迹,通过 Icon 为轨迹添加起点和终点图标。

### 3 防走失系统的实现

防走失定位录音系统旨在为用户提供一个通过手机实现定位录音的功能,利用手机方便携带,易于访问网络的特点,为老人儿童的出行提供安全保障。客户端分为监护端与被监护端,其中被监护端账号密码由监护端创建并自动绑定,保障被监护端的安全性。

#### 3.1 Android 客户端的实现

Android 客户端设置了主动上传、强制上传和周期上传 3 种通信上传方式实现对被监护端的实时监测。客户端的功能划分为 6 个基本模块:

(1)实时定位:监护人可查看被监护人实时位置,也可快速导航到被监护人身边。

(2)电子围栏:监护人可通过创建家、学校、社区等电子围栏来将被监护人设定在安全的围栏内,防止被监护人走失或发生其他意外等。

(3)轨迹回放:可以满足监护人及时了解被监护人之前的活动情况。

(4)强制录音:突发意外时,监护人可以强制录音以便听取被监护人现场声音信息,进行及时处理。

(5)多人监控:该模块主要满足多个监护人对同一个被监护人或一个监护人对多个被监护人同时监控的需求。

(6)模式设置:设置被监护端上传定位录音的模式、周期与录音时长。

### 3.2 Web 服务端的实现

服务端的功能划分为 4 个模块:

(1)网站分析:采用 Chart.js 插件统计用户分布和定位数量。

(2)用户信息管理:通过 Bootstrap 框架展示用户信息,并向管理员提供数据管理和备份功能。

(3)定位与录音管理:将系统数据库中的定位与录音信息提取出来并展示,管理员可以对定位与录音进行管理。

(4)地图展示:用于查询某个用户指定时间所处位置以及录音内容,同时提供用户的轨迹回放。地图展示如图 5 所示。

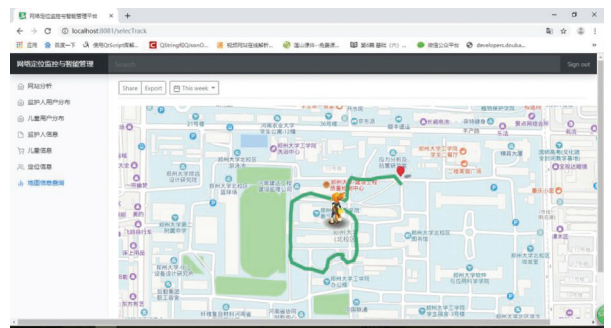


图 5 地图展示界面

Fig. 5 Map display interface

### 4 结束语

通过对 Android 系统架构的研究,应用 MVP 设计模式,设计并实现一款基于 Android 的防走失系统。该系统能够应用在移动设备上,并且充分利用硬件系统的性能,实用性高。基于 Android 的开发平台,能够短时间内升级新功能,并且该系统界面简洁,功能清晰,易于操作,为用户提供一种便捷实用的监控方式。(下转第 182 页)