**大连理工大学本科毕业设计（论文）**

**基于Android的门户网站监控系统设计与实现**

**Design and Implementation of Web Portals Monitoring System Based on Android**

学 院（系）： 软件学院

专 业： 软件工程

学 生 姓 名： 苏王荣

学 号： 201492059

指 导 教 师： 丁 锋

评 阅 教 师：

完 成 日 期： 2018.5.30

大连理工大学

Dalian University of Technology

# 摘 要

随着互联网的普及和迅速成长，越来越多的个人网站，企业网站，社区网站出现在市场上。同时，网站之间的竞争也越来越强。网站的健康状况对网站的质量有着至关重要的影响。一个及时，可靠，方便的监控系统可以很好的解决这一个需求。目前世面上的绝大多数监控系统的是基于PC端的网站应用，在便携性上有着很大的缺陷。在这样的背景下，本文提出了一个基于Android智能手机的解决方案，并且给与了实现。

本系统采用C/S模式，服务端由Nginx和PHP实现，部署在Ubuntu操作系统上。客户端使用Android Studio进行开发，通过使用MPAndroidChart图标库，实现了一个高效，稳定，简洁的Android应用，通过图表的形式，将门户网站的健康状况实时，简明的反应给使用者。

本文提出的系统相对于传统的监控系统，最大的优势是有着很高的便携性。本文提出的系统可以实时的反应门户网站所被部属的服务器的健康状况，不需要依赖于PC，使用者在有Android智能手机和网络的情况下，可以在任何时间任何地点都可以通过系统获得门户网站的健康状况。

关键词：门户网站；实时监控；Android开发；便捷性

**Design and Implementation of Web Portals Monitoring System Based on Android**

# Abstract

With the popularity and rapid growth of the Internet, more and more personal websites, corporate websites, and community websites appear on the market. At the same time, the competition between websites is also getting stronger. The health of the site has a crucial influence on the quality of the site. A timely, reliable and convenient monitoring system can well solve this demand. Most majority of monitoring systems on the world today are based on PC-side web applications, which have a lot of limitations in portability. In this context, this article proposes a solution based on Android smart phones and gives them a realization.

The system uses C/S mode. The server is implemented by Nginx and PHP and deployed on Ubuntu operating system. The client uses Android Studio for development. By using the MPAndroidChart icon library, an efficient, stable, and concise Android application is implemented. Through the form of a chart, the portal's health status is immediately and concisely reflected to the user.

The biggest advantage of the system proposed in this paper over traditional monitoring systems is its high portability. The system proposed in this paper can reflect the health status of the servers of the portals in real time, without relying on the PC. Users can obtain the portals through the system at any time and any place with Android smartphones and networks. The health of the site.

**Key Words:** Web portals; Real time monitoring; Android Development; Convenience

目 录

[摘 要 I](#_Toc514675992)

[Abstract II](#_Toc514675993)

[1 绪论 1](#_Toc514675994)

[1.1 项目背景 1](#_Toc514675995)

[1.2 项目的目的和意义 1](#_Toc514675996)

[1.3 本文主要研究内容 1](#_Toc514675997)

[1.4 章节安排 2](#_Toc514675998)

[2 相关技术介绍 3](#_Toc514675999)

[2.1 Android系统发展历史 3](#_Toc514676000)

[2.2 Android架构介绍 3](#_Toc514676001)

[2.3 Android应用开发技术介绍 5](#_Toc514676002)

[2.3.1 Activity 5](#_Toc514676003)

[2.3.2 Intent 6](#_Toc514676004)

[2.3.3 Nginx 7](#_Toc514676005)

[2.3.4 MySQL 8](#_Toc514676006)

[2.3.5 Redis 8](#_Toc514676007)

[2.3.6 MPAndroidChart图表库 9](#_Toc514676008)

[2.3.7 OkHttp3框架 9](#_Toc514676009)

[3 系统分析 11](#_Toc514676010)

[3.1 系统可行性分析 11](#_Toc514676011)

[3.1.1 技术可行性分析 11](#_Toc514676012)

[3.1.2 社会可行性 11](#_Toc514676013)

[3.1.3 经济可行性 11](#_Toc514676014)

[3.1.4 操作可行性 12](#_Toc514676015)

[3.2 系统需求分析 12](#_Toc514676016)

[3.2.1 系统实现目标概述 12](#_Toc514676017)

[3.2.2 功能需求分析 12](#_Toc514676018)

[3.2.3 性能需求分析 13](#_Toc514676019)

[4 系统设计与实现 13](#_Toc514676020)

# 1 绪论

## 1.1 项目背景

随着互联网越来越普及，门户网站越来越多，门户网站的监控系统的需求也越来越大。想要建立可靠，实用的网站就必然需要健康，稳定的服务器来支持。如何对服务器进行良好的管理，关系到网站的稳定运行，这对树立单位，公司的形象，保证业务的稳定，展示单位，公司的能力有着至关重要的作用。为了确保网站的稳定运行,就必须要做好网站服务器的监控。做好对服务器运行的各类异常、服务的运行状态进行及时的监控预警工作,可以帮助网站管理员及时修复一些服务器潜在异常状态,及时处理和修复服务器问题。目前市场上已经有了不少的监控系统，但是这些监控系统存在着这样那样的不足。市面上大多数的监控系统的是基于PC端的，这些监控系统要求使用者在使用的时候拥有一个PC，这样的设计大大的降低了系统的便携性和实时性，使用者无法随时随地的了解到自己网站服务器的状况。

在如今移动通信技术和智能手机飞速发展，各个操作系统的智能手机层出不穷。基于PC的监控系统已经无法满足使用者随时随地了解服务器状况的需求。现在的移动手机操作系统有IOS，Android两大主流操作系统。其中，Android是一个基于Linux内核的手机智能操作系统，是一款完全免费的智能手机平台，是一个开放源代码的系统，具有简单性，开放性。本文提出的基于Android智能手机的监控系统可以很好的解决这一个需求，帮助使用者随时随地获得自己的网站服务器的信息。

## 1.2 项目的目的和意义

伴随着智能手机手机的普及，传统的监控系统也需要做出改变。本文的目的在于实现一个基于Android设备的实时监控系统。该系统可以让网站管理者轻松的获得网站服务器的各项数据，检查服务器是否有异常，是否正常工作。本文的意义是开发出一个高效，简洁，实用的网站监控系统，帮助网站管理者更好的管理自己的网站。

## 1.3 本文主要研究内容

本文所介绍的是一个基于Android的门户网站监控系统。本系统采用C/S模式，其中手机客户端使用Android Studio开发，使用OkHttp3通信库实现客户端数据的实时更新，使用MPAndroidChart图表库实现简洁直观的数据界面，通过饼状图，折线图等图表，将数据直观的展示给使用者，并且实现了系统主动推送的功能，在发生异常情况时主动向使用者发出推送消息，告诉使用者异常情况。服务器端是通过Nginx和PHP搭建，使用MySQL关系型数据库和Redis 缓存型数据库。MySQL用来存储历史数据，当用户想要查看历史数据时，从该数据库获得对应数据，Redis用来存储近期的数据，该数据库为缓存型数据库，用于提高系统的响应速度。

本系统实现之后，经过了详细的测试，证实本系统已经能够满足所提出的需求。

## 1.4 章节安排

本系统的开发过程符合标准的软件开发流程，从技术介绍、需求分析、设计与实现、功能测试等方面进行。本文分为五章，各章节的主要安排如下：

第一章：绪论

包括项目背景、目的以及意义。

第二章：相关技术概述

对本文提出的监控系统所采用的主要技术进行了分析和介绍，其中包括Android基本技术介绍、MPAndroidChart图表库介绍，Redis缓存型数据库介绍等。

第三章：系统分析

包括系统可行性分析和系统需求分析。其中系统可行性分析包括技术可行性分析、经济可行性分析以及社会可行性分析，系统需求分析包括系统功能性需求分析以及系统非功能性需求分析。

第四章：系统设计与实现

包括系统总体设计、详细设计以及具体实现。

第五章：系统测试

首先，介绍了系统测试的指导原则，之后对本系统测试所运行的环境进行了介绍。最后，根据详细的测试用例，给出了实际的测试结果，并进行了分析。

# 2 相关技术介绍

## 2.1 Android系统发展历史

Android这个名字第一次出现在小说《未来夏娃》中，这本小说是法国作家利尔亚当在1886年写的科幻小说。小说中，作者将外形像人的机器起名为Android。这也是Android名字的起源。

Android系统一开始并不是由Google研发出来的。起初，Andy Rubin创办了两个手机os公司，分别是Danger和Android。其中Danger被微软收购。而Android只是Google收购的众多项目中的一个。当时的Android还是名不见经传的一个操作系统，在资金用完之后，陷入不温不火的尴尬地步。2007年，苹果推出了第一款iPhone。2007年11月05日，在苹果的刺激下，Google正式发行了名为Android基于Linux平台的开源操作系统，并且提倡创立一个全球性的组织，由34家手机制造商、软件开发商、电信运营商以及芯片制造商共同组成。但是这个版本还有获得广泛的关注。

2009年5月，发布了1.5版本，该版本提供了一个比较丰富的用户界面，同时还实现了蓝牙技术。这个版本的Android赢得了大量开发者的支持。

接下来，Android版本更新速度变快，目前最新的是Android6.0，每一个新的版本都修复了上一个版本的bug，并且增加了上一个版本没有的功能，Android的功能越来越强大。生产使用Android操作系统的手机的厂商也在逐渐变多，Android手机平台在真正意义上，变成一个完全的开放平台，而且出于费用的考虑，免费的Android显然受到了更多的青睐。

目前，已经发布使用Android系统的手机有：三星、HTC、LG等；国内厂商有华为，小米，联想，中兴等。

## 2.2 Android架构介绍

Android系统的底层系统是Linux，由操作系统、中间件、用户界面和应用软件四层组成。Android的软件架构的结构是分层的，采用一种Software Stack（软件叠层）的方式进行构建，从上到下分别为Application（应用层）、Application Framework（应用框架层）、Android Runtime & Libraries（运行时库和本地库层）、Linux Kernel（内核层），如图2.1所示。

（1）Android应用层

Android系统将会包含用户安装的应用程序及系统自带的应用程序，这些程序包括有电子邮件客户端、便签、计算器、日历、相机、相册等，这些应用程序都是用Java写的。开发人员在该层进行应用层开发，除了对第三方的开发，还有对系统原生的应用程序的开发。



图2.1 Android软件架构图

（2）Android应用层

Android系统将会包含用户安装的应用程序及系统自带的应用程序，这些程序包括有电子邮件客户端、便签、计算器、日历、相机、相册等，这些应用程序都是用Java写的。开发人员在该层进行应用层开发，除了对第三方的开发，还有对系统原生的应用程序的开发。

（3）Android应用框架层

这是一个系统框架层，是Google发布时编写核心应用时所用的API框架，Android开发人员也可以用来开发出自己的应用，其中封装了大量应用程序所使用的类，达到组件重用的目的，大大精简了项目开发的流程，同时开发者应该遵守框架开发的基本原则。该层是Android应用开发的基础，负责封装应用层的接口函数，向上层应用层提供API，比如有Activity用来管理全部的Activity的画面导航、回退等与生命周期有关的操作，PackageManager主要用来管理程序安装包的安装、卸载、更新等操作。

（4）Android Runtime & Libraries

本层分为两个部分，其中Runtime是Android的运行环境，负责解释和执行核心库字节码，而Libraries是本地库层，也表示开放源码的函数库。这一层主要包含了手机系统平台必须的C/C++核心库、Dalvik虚拟机运行环境和HAL子层。常用的有SSL层，SQLite层等。SSL层处于各个应用层协议跟TCP/IP之间，主要是提供数据通信。SQLite层是关系型数据库，用来操作数据库。

（4）Linux Kernel

Android核心系统服务依赖于Linux内核。Linux为Android提供主要的系统服务如：Security、Memory Management、Process Management、Network Stack等。同时也作为软件栈和硬件之间的抽象层。

## 2.3 Android应用开发技术介绍

### 2.3.1 Activity

Activity是一个应用组件，它用来提供一个窗口，用户可以通过与该窗口交互来完成一些任务，比如拨电话，照相，发送邮件，或者寻找地图。为每个activity分配一个窗口，用户在这个窗口中绘制UI。这个窗口通常都是铺满屏幕，但是也可能比屏幕小或者漂浮在其他的窗口上面。

一个应用通常由多个activity组成，这些activity之间保持松散的绑定关系。通常来说，一个应用中的某个activity被定为主activity后，当第一次启动应用时，这个activity会呈给用户。然后每个activity可以启动别的activity来执行不同的动作。当每次一个新的activity启动，前一个activity处于stop状态，但是系统会将前一个activity保存在一个栈（后退栈）里。当一个新的activity开启时，它会被系统放到后退栈的栈顶并且会获得用户的焦点。后退栈遵守后进先出的栈的机理，所以当用户在当前页面完成任务之后，并且按下后退键，它就被从栈顶弹出（并且被销毁），然后前一个activity就会被恢复（变成resume状态）。

当一个activity因为一个新的activity开启而处于stop状态时，它可以通过调用activity的生命周期的回调方法在状态中标明这个变化。由于activity自身的状态变化，无论系统是在创建它，停止它，恢复它，或者销毁它，该activity都会接受到多个回调方法，并且每个回调方法可以提供给你一个针对某个状态变化执行特定操作的机会。例如，当处于stop状态时，你的activity应该释放任何一个大的对象，比如网络或者数据库连接。当activity恢复时，你可以重新获取必要的资源并且恢复之前被中断的操作。这些状态转变是一个activity生命周期的全部。

当activity进行状态切换时，系统会调用以下七种方法来标识这些变化：

onCreate()在activity第一次被创建时调用。在这个函数里，你应该完成所有的标准的静态方法，比如创建视图，为视图设置监听器，将数据绑定到list上等等。这个方法有一个Bundle对象的参数，Bundle对象中包括了这个activity的前一个状态。通常来说，随后将会执行onStart()方法。

onStart()仅当activity对用户变成可见之前，才会被调用。随后，如果activity进入前台，则会调用onResume()方法，否则，如果被隐藏，则调用onStop()方法。

onResume()仅当activity开始与用户交互才会被调用。Activity处于activity栈的顶部，此时用户可以与该activity交互。之后一般会调用onPause()方法。

onPause()当系统将要开始恢复另一个activity时，才会被调用。这个方法通常用来把未保存的数据变为永久的数据，停止动画以及其他消耗CPU的任务等等。因为只有在onPause()方法完成之后，下一个activity才会被恢复，所以本方法必须快速地完成。

onStop()在activity不再显示给用户时，才会被调用。发生这种情况有两种可能：该activity被销毁或者有另一个activity（已经存在的或者新的activity）被恢复并且正在覆盖它。onStop()方法之后，如果本activity要返回前台与用户交互，就会调用onRestart()方法，否则，如果要消失，则会调用onDestroy()方法。

onRestart()在activity已经处于stop状态之后，当需要再次启动时调用。

onDestroy()在activity被销毁前被调用。这是activity会接受到的最后的一个调用方法。当那个activity将要完成或者系统为了节省内存而暂时地销毁本activity的实例时，onDestroy()就会被调用。

### 2.3.2 Intent

一个程序中的三个核心的组件activities，services和broadcast receivers都是通过intent来传递消息。使用intent传递消息能很便利地在运行时绑定组件，无论这些组件是来自同一个或者不同的程序。一个intent对象是一个被动的数据结构，保存一个将要被执行操作的一个抽象描绘，或者通常对于广播来说，是一个对于已经发生的动作的描述。Intent对象是一个信息集合。它包含了收到它的组件所感兴趣的信息，以及android系统感兴趣的信息。它主要包含：

Action：用来指定要进行的操作是什么。

Data：要被操作数据的URI（统一资源定位符）和该数据的MIME类型。不同的操作对应不同种类的数据规格。

Category：一个字符串，包含一些额外的信息，这些信息是关于用来处理本intent的组件。

Extras：一个键值对，表示了发给正在处理本intent的组件的额外信息。

Intent分为显示intent和隐式intent。

显式的intent通过名字来指定目标组件。由于组件的名字通常不会被其他应用程序的开发者所知道，显式的intent通常用于一个应用内部的消息传递，比如一个activity启动一个附属服务或者启动某个应用程序的activity。

通常来说，隐式intent用来激活其他应用程序的组件。隐式的intent没有指定组件名，Android通过intent中足够多的信息使用intent filter在所有的组件中过滤信息来匹配目标组件。

### 2.3.3 Nginx

Nginx是一个高性能的HTTP和反向代理服务器，也是一个IMAP/POP3/SMTP代理服务器。

Nginx是一款轻量级的Web服务器/反向代理服务器以及电子邮件代理服务器，并在一个BSD-like协议下发行。由俄罗斯的程序设计师Lgor Sysoev所开发，供俄国大型的入口网站及搜索引擎Rambler使用。其特点是占有内存少，并发能力强，事实上Nginx的并发能力确实在同类型的网页服务器中表现较好。

Nginx相较于Apache\Lighttpd具有占有内存少，稳定性高等优势，并且依靠并发能力强，丰富的模块库以及友好灵活的配置而闻名。在Linux操作系统下，Nginx使用epoll事件模型,得益于此，Nginx在Linux操作系统下效率相当高。同时Nginx在OpenBSD或FreeBSD操作系统上采用类似于epoll的高效事件模型kqueue.

Nginx 是一个高性能的 Web 和反向代理服务器, 它具有有很多非常优越的特性:

作为 Web 服务器：相比 Apache，Nginx 使用更少的资源，支持更多的并发连接，体现更高的效率，这点使 Nginx 尤其受到虚拟主机提供商的欢迎。能够支持高达 50,000 个并发连接数的响应，感谢 Nginx 为我们选择了 epoll and kqueue 作为开发模型.

作为负载均衡服务器：Nginx 既可以在内部直接支持 Rails 和 PHP，也可以支持作为 HTTP代理服务器 对外进行服务。Nginx 用 C 编写, 不论是系统资源开销还是 CPU 使用效率都比 Perlbal 要好的多。

作为邮件代理服务器: Nginx 同时也是一个非常优秀的邮件代理服务器（最早开发这个产品的目的之一也是作为邮件代理服务器），Last.fm 描述了成功并且美妙的使用经验。

Nginx 安装非常的简单，配置文件 非常简洁（还能够支持Perl语法），Bugs非常少的服务器: Nginx 启动特别容易，并且几乎可以做到7\*24不间断运行，即使运行数个月也不需要重新启动。你还能够在 不间断服务的情况下进行软件版本的升级。

### 2.3.4 MySQL

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件。MySQL是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

由于其社区版的性能卓越，搭配 PHP 和 Apache 可组成良好的开发环境。

与其他大型数据库例如Oracle、DB2、SQL Server等相比，MySQL 自有它的不足之处，但是这丝毫也没有减少它受欢迎的程度。对于一般的个人使用者和中小型企业来说，MySQL提供的功能已经绰绰有余，而且由于 MySQL是开放源码软件，因此可以大大降低总体拥有成本。

### 2.3.5 Redis

Redis是一个key-value存储系统。和Memcached类似，它支持存储的value类型相对更多，包括string(字符串)、list(链表)、set(集合)、zset(sorted set --有序集合)和hash（哈希类型）。这些数据类型都支持push/pop、add/remove及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。在此基础上，Redis支持各种不同方式的排序。与memcached一样，为了保证效率，数据都是缓存在内存中。区别的是Redis会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了master-slave(主从)同步。

Redis 是一个高性能的key-value数据库。 Redis的出现，很大程度补偿了memcached这类key/value存储的不足，在部 分场合可以对关系数据库起到很好的补充作用。它提供了Java，C/C++，C#，PHP，JavaScript，Perl，Object-C，Python，Ruby，Erlang等客户端，使用很方便。

Redis支持主从同步。数据可以从主服务器向任意数量的从服务器上同步，从服务器可以是关联其他从服务器的主服务器。这使得Redis可执行单层树复制。存盘可以有意无意的对数据进行写操作。由于完全实现了发布/订阅机制，使得从数据库在任何地方同步树时，可订阅一个频道并接收主服务器完整的消息发布记录。同步对读取操作的可扩展性和数据冗余很有帮助。

### 2.3.6 MPAndroidChart图表库

MPAndroidChart图表库是一个开源的Android图表库，是当下最热门的Android图表库。该图表库提供多种美观，可用性高，个性化强的图表。该图表库用于帮助开发人员绘制出符合需求且美观，简洁的各种图表。该图表库具有如下特点：

8种不同的图表类型

两轴缩放（支持触摸手势,两轴单独或同时的放缩）

拖 / 平移 / 抛（触摸手势）

Combined-Charts 组合图表（线状、柱状、散点图等）

双轴（比如说有两个独立的Y轴数据）

画值 （draw values into the chart with touch-gesture）

高亮显示值（我们可以自定义Popup-views来高亮显示我们选中的值）

多个 / 单独的轴

预定义的颜色模板

自定义轴（x轴和y轴）

动画（建立x和y轴动画）

限制线（比如提供附加信息、最大值 …）

完全自定义（paints、字体、legends、颜色、背景、手势、虚线 …）

平滑缩放和滚动 30.000 数据点（线状，柱状图表）

### 2.3.7 OkHttp3框架

OkHttp是一个处理网络请求的开源项目，Android 当前最火热网络框架，由移动支付Square公司贡献，用于替代HttpUrlConnection和Apache HttpClient(android API23 6.0里已移除HttpClient)。

OkHttp框架具有如下优点:

支持HTTP2/SPDY（SPDY是Google开发的基于TCP的传输层协议，用以最小化网络延迟，提升网络速度，优化用户的网络使用体验。）

socket自动选择最好路线，并支持自动重连，拥有自动维护的socket连接池，减少握手次数，减少了请求延迟，共享Socket,减少对服务器的请求次数。

基于Headers的缓存策略减少重复的网络请求。

拥有Interceptors轻松处理请求与响应（自动处理GZip压缩）。

# 3 系统分析

本章节的主要内容是对所开发的门户网站监控系统进行系统分析，包括系统可行性分析和需求分析。

## 3.1 系统可行性分析

可行性分析的目的，不是为了解决问题，而是尽量用较小的代价，花费较短的时间，来确定问题是否值得去解决。可行性分析是项目正式开发之前的必要流程，可以预测到工程的可行性，从各个方面为项目开发提供理论依据。有效合理的分析可以有效的降低风险，增加项目的可用性。因此在项目开发前，对项目进行可行性分析是必不可少的过程。

### 3.1.1 技术可行性分析

技术是工程设计和开发的支撑，如果没有成熟的技术作为驱动，工程开发就很难达到预期的要求。之前很多失败的项目开发经历，就是由于开发之前，对于所需要的技术分析不够，导致在开发过程中出现了技术瓶颈，出现了临时改动需求或者项目开发半途而废，因此在项目开发之前，进行技术可行性分析是一件很有必要的事。本文设计的基于Android的门户网站监控系统主要有以下核心技术：

（1）Android客户端开发技术，包括Bottom Navigation Bar，ViewPager，图表绘制等等。

（2）服务器端开发技术，包括PHP，MySQL数据库技术，Redis数据库技术等。

以上所介绍的技术都是目前Android平台开发应用所采用的主流技术，并且市场流行的众多Android应用中都广泛使用了这些技术，因此本文系统的开发在技术上是具有可行性的。

### 3.1.2 社会可行性

所有的门户网站都需要一个可靠的监控系统来支持，本文提出的监控系统不仅拥有普通监控系统的各项功能的同时，还具备有普通监控系统没有的便携性性和实时报警的功能。因此相对于上面上普通的监控系统，本文提出的监控系统拥有很大的优势。

### 3.1.3 经济可行性

随着时代的进步，高校的基础设施不断改善，规模也在扩大，成本也在提高，但是资金来源却很有限，因此高校都很谨慎地控制管理方式改革上的成本，因此本系统能否在一定程度上降低开发成本，并获得一定的收益，是一个很重要的问题。

本系统使用的开发平台是Android（Android Studio + Android SDK），服务器是Nginx，数据库是MySQL，Redis。

相比其他的大型软件开发项目来说，该系统开发的硬件成本很小，只需要租赁一台服务器，它的客户端的要求很低，因为智能手机已经在各年龄群体中都已普及，所以客户端的投入可以忽略不计。

在系统开发完成并投入使用之后，可以通过手机随时随地了解到网站服务器的状况。

### 3.1.4 操作可行性

本系统在界面设计上，以图表作为数据的载体，使得数据简明清晰，并且可以一眼就看出数据的变化趋势，手动更新操作使用了目前广泛使用的下拉式，是的软件可以轻松的上手。

## 3.2 系统需求分析

本系统是一个基于Android的监控系统，使用的是Java语言。本系统的基本需求是准确的反应门户网站对应的服务器的健康状况。当使用者打开该系统中的Android应用后，可以很快的了解到服务器的健康状况，简单快速的获得自己需要的信息。该系统中的Android应用会周期性的刷新数据，并且支持使用者主动刷新数据，并且在服务器出现的突发错误时可以发出警告，主动提醒使用者。使用者还可以通过该Android应用来查看历史的服务器数据，从而来判断服务器中的数据是否正常。

### 3.2.1 系统实现目标概述

本系统的目标是通过在Android智能手机上实现门户网站的监控数据显示并且在服务器出现异常时发出报警。使用图表的形式可以很好的展现服务器的监控数据，使用推送功能可以实现异常报警的功能。

### 3.2.2 功能需求分析

对于客户端来说，本系统通过Android手机作为客户端。由于服务器的相关数据有着一定的机密性，本系统中的Android应用需要一个登陆界面，来验证用户的身份。对于服务器传来的数据，Android应用需要一个概要数据展示界面，以图表的形式将数据展示给用户。由于每一个图表展示的数据有限，所以Android应用需要提供一个详细数据界面来让用户获得详细数据，例如过去几天的历史数据数据等。

对于服务端来说，本系统为了减轻MySQL数据库的压力，使用了MySQL与Redis协同服务的策略，所以需要把数据分别存储到两个不同的数据库当中，MySQL中存储服务器的历史数据，Redis存储服务器当天的数据，并且要把数据的获取和返回分散开来，以此来保证服务器的反应速度。对于客户端发出的普通周期性更新请求，服务器从Redis中获取数据，并且把数据返回给客户端，对于客户端发出的主动更新请求，服务器获取实时的数据，把数据存入Redis并且返回给客户端。

3.2.3 性能需求分析

本系统是基于Android设备的系统，因此风格和系统效率要符合Android程序开发的准则，交互的友好性、界面的简洁性、运行效率以及程序的可扩展性。

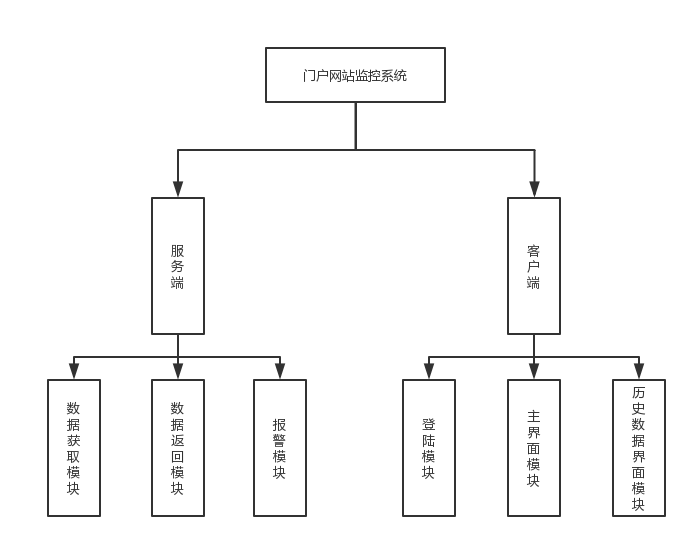
1. 交互的友好性。本系统通过图表的形式展示数据，可以通过简单常见的手势对图表进行放大和缩小。刷新操作采用了目前最广泛使用的下拉刷新的方式，使得操作简单且容易上手。并且在界面的变化过程中有过渡动画进行平滑处理，是的界面的转化变得合理且流畅。
2. 界面的简洁性。本系统专注于高效和简洁，在界面中只留有必要的图表数据，并且使用的是简洁大方的单色背景，不使用容易使人眼花的背景图票，对于每一个图表，都只留下了必要的数据。可以让使用者在非常短的时间内获得自己想要的所有数据。
3. 运行效率。本系统使用了数据获取和发送分离的方式，并且使用了Redis缓存型数据库来降低系统的响应时间。最大限度的来缩短服务器的反应时间和数据返回的时间。
4. 可扩展性。本系统的设计和开发严格符合正规的软件开发流程，使用了合理的构架和合理的函数，变量命名方式。可以简单快速的进行后续的迭代操作。

# 4 系统设计与实现

需求分析之后需要对系统进行实现，而在系统实现之前需要对系统进行合理的设计。本章将会从系统的设计开始，介绍系统的设计和实现。

## 4.1系统总体设计

本系统包括客户端和服务端两个部分。服务端的数据需要使用客户端来展示，客户端需要服务端的数据支持，两端之间通过网络请求进行数据传递。服务端可以分为获取服务器数据的数据获取模块和响应客户端请求的数据返回模块，还有服务器出现异常时对客户端进行主动提醒的报警模块。客户端可以分为最初验证使用者身份的登陆模块和展示服务器数据的主界面模块，还有查询历史记录的历史数据界面模块。如图4.1 门户网站监控系统总体设计图所示。



## 4.2 系统详细设计

本系统的详细设计将被分为功能详细设计和数据库设计。

### 4.2.1 功能详细设计

本系统按照需求被分成六个模块：数据获取模块，数据返回模块，报警模块，登陆模块，主界面模块，历史数据界面模块。

#### 4.2.1.1 数据获取模块

该模块负责门户网站对应的服务器的各个有价值的参数，如CPU使用率，带宽使用率等。该模块通过Linux系统提供的Shell命令和PHP的系统命令接口获得服务器的数据，并且把数据存储到MySQL数据库和Redis数据库中。MySQL数据库负责存储历史数据，Redis数据库具有很快的响应速度，因此用来存储访问频率较高的数据，也就是当天的数据。通过分析门户网站对于服务器的日志可以获得门户网站的访问量。通过在门户网站对应的接口中设置计数器，并且把结果存储的Redis数据库中，可以获得对应接口的访问记录。

#### 4.2.1.2 数据返回模块

该模块负责向客户端返回客户端请求的数据。对于一般的客户端周期性更新请求，该模块通过从Redis获取数据返回给客户端，以此来保证服务端的响应速度。对于客户端发起的查看历史数据的请求，该模块通过查询MySQL数据库来获取数据，并且把数据返回给客户端。对于客户端发起的主动更新请求，该模块先通过PHP的系统命令接口和Linux系统提供的Shell命令获得服务器实时的数据，并且将数据返回给客户端，然后再将数据写入到MySQL数据库和Redis数据库中。

#### 4.2.1.3 报警模块

该模块负责在服务器出现异常时进行记录，并且向客户端发起报警推送。当数据获取模块获取到了异常的数据或者门户网站对应的服务器错误日志文件中出现了报错日志时该模块会被调用。该模块将会将错误信息存入到ElasticSearch中，并且将错误信息与意见存在于ElasticSearch中的错误进行对比。如果该类错误在近期内没有出现过，则向客户端发起一个报警推送；如果该类错误在近期出现过说明该错误已经发送过报警推送，为了避免过渡的报警推送，这种情况下将不会发送报警推送。

#### 4.2.1.4 登陆模块

登陆模块是客户端的第一个模块，当用户打开应用时，第一个看到的就是登陆模块。用户需要在该模块填写用户名和密码，因为一台服务器可以部署多个门户网站，所以用户还需要填写门户网站的名称。服务端通过将数据与数据库中的数据进行对比，返回登陆成功或者登陆失败。如果登陆成功，客户端将会保存用户的用户名和密码，在下一次打开程序时，将不用再输入密码和账号。如果登陆失败，客户端将会提示用户对应的错误信息。无论登陆是否成功，服务端多会将访问结果记录到MySQL

数据库中，以便将来进行查阅。



#### 4.2.1.5 主界面模块

该模块负责以图表的形式向用户展示服务端传递过来的数据。该模块中将会有两个饼状图，一个用来展示门户网站的访问量，其中包括总访问量和主要接口的访问量；另一个用来展示服务器的磁盘存储状态。因为内存的使用率，CPU的使用率，带宽的占用率，这3个数据需要近期的一系列数据来一起查看才能分辨服务器是否在正常运行，所以该模块将会使用折线图来展示这3项数据。并且该模块需要对数据进行等级划分。将内存，CPU，带宽的使用率按照30%，70%两个界限划分为3个等级，分别是良好级，正常级，危险级。当数据处于良好级时，折线图的背景为绿色；当数据处于正常级时，折线图的背景为白色；当数据处于危险级时，折线图的背景为红色。

#### 4.2.1.6 历史数据界面模块