

**本科生毕业设计[论文]**

**基于OAuth2.0的第三方用户信息系统设计与实现**

专业班级 软件工程1301

姓 名 邢铭哲

学 号 U201317429

指导教师 卢力

2017年 4月 18日

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名： 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于 1、保密囗，在 年解密后适用本授权书

2、不保密囗 。

（请在以上相应方框内打“√”）

作者签名： 年 月 日

导师签名： 年 月 日

摘 要

随着移动互联网规模的不断扩大、移动设备的不断普及，网络技术的迅速更新，各式各样的网络应用逐渐出现到人们的日常生活中。由用户为主导而生成的内容互联网模式——Web 2.0的发展更是成为互联网的重大变革，推动了人与人之间对于网络互动资源的共享。但随着互联网世界的迅猛发展，随之产生了一系列问题，比如人们的账号数量不断增加，用户的账号密码很难维护，也会随之出现用户隐私信息泄露问题，并且各个网络应用之间的账号系统以及资源不能进行有效的沟通和共享，从而形成数据资源孤岛，不利于数据价值挖掘和利用。OAuth协议便是针对这种问题应运而生的，现已成为Web服务的标准授权协议，广泛的运用在各种社交应用，电商服务等。OAuth 是一个开放的授权协议，允许用户让经过用户授权的第三方应用访问该用户在某一网站上存储的私密的资源（如照片，视频，联系人列表），而无需将用户名和密码提供给第三方应用。

本文主要介绍OAuth 2.0的工作原理和在工程中的实现、用户管理中心的设计以及面向Gitlab等第三方应用的OAuth2.0接入设计。

**关键词：**OAuth 2.0；开放式授权；单点登录；用户中心

Abstract(need update)

With the mobile Internet, the scale of things has been expanding, the rapid updating of network technology, a variety of network applications gradually appear in people's daily lives. User-driven content generated by the Internet model - Web 2.0’s development is a major change in the Internet, and promote the sharing of human resources and human resources. But there appear a series of problems, people gradually increase the number of users, the user's account and password is difficult to maintain, there will be a user privacy disclosure problems, the various network applications between the account system resources can not be effective communication and sharing, thus forming a data resource island, is not conducive to the value of data mining and use. In this case, OAuth protocol came into being, has now become a Web services standard licensing agreement, widely used in a variety of social applications, electricity business servioces.

OAuth is an open licensing agreement that allows users to have third-party applications access private resources (such as photos, videos, contacts lists) that users store on a site without having to provide a username and password for third-party applications. This article mainly introduces the working principle of OAuth 2.0 and the realization of project, the design of user management center and OAuth2.0 access design for third party applications such as Gitlab.

**Key Words：**OAuth 2.0；Open Authorization；Single Sign On；User Center

# 绪论

## 网络账号系统的研究背景及意义

在传统互联网时代，各个网络应用或者服务之间的资源是封闭的，形成了数据资源孤岛，对于这些海量的数据资源更是得不到有效的共享利用。因此网络应用与服务之间海量数据的整合与共享已经成为必然趋势，网络信息逐渐从分散的状态进行聚合。而数据共享的需求导致系统服务之间的相互协作日益增加，对用户的认证和授权也提出了更高的要求。2007年，Facebook最先提出开放平台的概念，开放平台提供开放的API（Open API），开发者和第三方应用可以利用简洁易懂的Open API获取系统的数据。开放平台的产生代表了互联网数据共享的发展，国内各个IT公司逐渐开发了自己的开放平台，如新浪微博开放平台，人人网开放平台，腾讯开放平台，百度开放平台等。但随之而来的问题是如何对这些开放API进行认证和授权，如今，互联网上主要有两种认证授权协议：OpenID和OAuth。本文主要介绍基于OAuth 2.0的用户信息管理系统的设计实现，以下简称网络账号系统。

本课题来源于中国科学院软件研究所操作系统基础版研发支撑平台（ISDE），随着支撑平台的开发日益完善，其子系统的数量和规模也逐渐扩大，支撑平台集成了如Gitlab，JIRA，mantis，QDS，TestLink，LogCat等经过二次开发的开源应用以及针对项目开发周期中开发的特定应用，我们在日常开发工作中建立了项目组、用户、任务的基本概念，我们需要将这三个对象进行对应管理，当数据资源发生变化时需要与各个子系统进行消息推送，使每个系统做到数据一致性，并且每个子系统都有自己的账号，导致账号数量繁多，不易进行管理、维护、统一，我们也需要将各个子系统的用户信息统一收敛到一个第三方的用户信息管理系统，进行统一的管理维护，因此如何将这几个子系统整合到一起，并进行数据资源共享就成为了一个问题。针对以上问题，我们设计开发了基于OAuth 2.0的第三方用户信息管理系统，此系统即为本文进行讨论的网络账号系统。此网络账号系统需要维持用户在全局子系统中的在线状态并保持数据一致性以及对项目组、用户、任务等信息进行有效管理，此外，网络账号系统也要提供开放平台，Open API以供第三方应用、社区等的注册接入。意义是通过实现、完善这些功能，能够提升网络账号的管理能力、提高安全稳定性，并进一步更好地支撑操作系统基础版研发支撑平台（ISDE）各子系统。

## 国内外研究现状：

如今，网络应用主要是由B/S架构和C/S架构组成。B/S（Browser/Server）结构也就是浏览器和服务器结构，它是随着互联网的兴起，对C/S结构的变化和改进，B/S结构少部分逻辑处理在浏览器端进行处理，大部分逻辑任务在服务器端处理，服务器如IIS、Apache、Nginx等处理网络请求，维护管理整个Web应用。C/S（Client/Server）架构也就是客户端和服务器架构，其主要是应用在桌面应用上，服务器是网络的核心，而客户端是网络的基础，客户端依靠服务器获得所需要的网络资源，而服务器为客户端提供网络必须的资源，将任务合理的分配到客户端和服务端，可以降低系统的通讯开销，但是由于移动客户端计算性能的局限性，复杂的逻辑计算逐渐转移到服务器端进行处理，这时客户端就像B/S架构中的浏览器端一样只负责数据的展示以及少部分逻辑事务处理，C/S架构和B/S架构应用之间的界限逐渐缩小，因此我们可以将这种C/S架构应用和B/S架构应用统称为网络应用。随着Web、网络技术的迅速发展以及以社交分享为代表的Web 2.0时代的到来，人们越来越注重人与人之间在互联网上的交互。从较早的博客、论坛、BBS、等，到如今社会性网络服务（Social Network Site）应用、电子商务等日益普及，与此同时，随着移动互联网的速度、稳定性和普及度不断提高，致使一批批的移动互联网应用逐渐兴起，如现在较为流行的照片实时分享应用Instagram，根据统计结果，Instagram在2010年10月13日时的用户数量仅仅不到10万，而到了2010年12月21日用户数就突破了100万。2011年10月31日时，其用户数达到1200多万，到了2012年5月1日就达到了5000多万用户。2012年7月27日，Instagram的注册用户数已经达到8000多万，距离该网站用户数突破5000万仅仅2个多月。这意味着，每月Instagram的用户数增长数超过1000万，Instagram同时公布统计结果，其应用的用户已分享了近40亿张照片。那么在拥有规模如此巨大的网络应用以及网络应用用户的今天，用户信息的保护与管理就成了一个很严峻的问题，很显然，用户账号数量越少，其数据隐私，账号信息泄露的几率就越少，针对这个问题，先后出现了OpenID和OAuth两种协议来进行跨应用授权和资源共享，用户不必记忆繁杂的账号密码，并且可以跨应用分享自己的数据。

2007年，OpenID正式发布，OpenID是一个去中心化的网上身份认证系统。对于以往的序言用户进行身份凭证校验的网站，用户需要记住像用户名和密码这样的传统验证标记来进行登录操作。而对于支持OpenID的网站，他们只需要预先在一个作为OpenID身份提供者（identity provider, IdP）的网站上注册。任何网站都可以使用OpenID来作为用户登录的一种方式，任何网站也都可以作为OpenID身份提供者，因此OpenID是去中心化的。OpenID既解决了问题而又不需要依赖于中心性的网站来确认数字身份。OpenID正在被越来越多的大网站采用，比如作为身份提供者的雅虎和微信、MySpace等。

OAuth是一个关于开放式授权的开放网络协议，在全世界的许多互联网公司和服务提供商上都得到了广泛的应用，它当前的版本是2.0.它允许用户让第三方应用访问该用户在某一网站上存储的私密的资源（如照片，视频，联系人列表），而无需将用户名和密码提供给第三方应用。2009年4月23日，OAuth宣告了1.0协议上的一个的安全漏洞。该漏洞影响了以OAuth 1.0为核心规范的OAuth的认证流程。于是，发布了OAuth Core协议1.0a版本来解决这一漏洞。后来，一位名叫王晶的新加坡南洋理工大学数学系博士生，发现了OAuth和OpenID开源登录工具的“隐蔽重定向漏洞（Covert Redirect）”。该漏洞首先由CNET报道，搜狐网、网易、新浪微博、阿里巴巴、腾讯QQ、人人网、亚马逊、微软Live、开心网、Facebook、Google、、GitHub Yahoo、WordPress、eBay、PayPal、LinkedIn等大量知名网站受到漏洞影响并造成损失。黑客可利用该漏洞来伪造钓鱼网站，用知名大型网站链接误导用户访问并登录钓鱼网站，一旦当用户访问钓鱼网站并成功登陆授权，黑客即可根据认证服务器发放的令牌获取读取其在网站上存储的私密信息。该漏洞被网易，人民网，CSDN，搜狐，凤凰网，太平洋电脑网等大量中文网站报道。之所以存在问题不是因为OAuth这个协议本身，这个协议本身是没有问题的，而是因为各个厂商没有严格参照OAuth授权协议的官方文档进行设计开发，只是实现了OAuth授权协议的简版。这个漏洞产生的原因在于OAuth的提供方提供OAuth授权过程中没有对客户端注册的回调URL进行校验，从而导致被黑客赋值为非原定的回调URL，就可以导致跳转、XSS攻击等问题。目前大部分涉及该漏洞的网站都已经修复该问题。OAuth2.0是OAuth协议的下一版本，但不向下兼容OAuth 1.0，即完全废止了OAuth1.0。OAuth 2.0关注客户端开发者的简易性。2012年10月，OAuth 2.0协议正式发布为RFC 6749。

在如今的Web领域中，OAuth授权协议在国内外的实施已经成熟并且得到广泛运用。国内外知名Web服务商，IT公司都提供了开放平台，提供了Open API，用户可以十分便捷的接入第三方授权认证，获取第三方数据资源并进行共享，OAuth协议对用户，消费方，资源提供方也都带来了好处：用户不必再记忆数量繁多并且复杂的账号密码，隐私数据的安全性有了保障，并且可以跨应用分享自己的数据，使用户可以更加融入网络社交中；对于消费方，大大减少了对用户账号数据的维护工作量，并且可以通过接入到其他应用与其他应用交互提高用户的兴趣，同时减少繁琐的注册工作使用户的体验更加友好；对于资源提供方，可以加强其在社会化网络世界中的主要地位，增加用户流量。OAuth协议的普及使网络应用从传统的封闭式应用变成资源共享、互联的新型互联网网络格局，极大地促进了社会化网络以及Web2.0时代的发展。

OAuth协议从1.0到如今2.0，中间遇到了很多问题和漏洞，OAuth协议也一直不断完善规范中，总体来说，OAuth协议已经取得了显著地效果和广泛的应用，并且在未来拥有广大的应用前景。

## 本文主要研究工作：

本毕业论文题目为《基于OAuth 2.0的第三方用户信息系统的设计与实现》，因此，本文主要围绕着两个重点来讨论：一个是OAuth 2.0网络认证框架的研发；一个是用户中心的设计与实现，包括公司、项目组、用户、任务等几个基本概念的相互之间关联的实现。

其中，OAuth 2.0网络认证框架的研究主要包括OAuth 2.0协议的具体工作流程的研究以及其具体实现，并且实现开放平台，提供Open API，并提供第三方应用注册客户端，接入网络账号并提供不同的权限管理，满足开放社区用户注册和登录需求；用户中心的研发主要包括实现公司，项目组，用户，任务之间关联关系，三元审核，ISDE工作平台内部所有子业务系统账号系统的梳理，当用户中心的数据发生改变时能够实时向第三方应用推送通知，从而保持数据一致性，统一规范设计和研发用户信息数据库，用户信息的分组管理，用户的权限管理，用户状态管理模块用户认证。该项目要求实现高并发、大负载架构设计，最后能够基本稳定地为ISDE各子系统提供账号登录、权限管理、用户在线状态保持等服务。 能够支持并发1000人/秒的在线请求服务。

## 本论文的组织结构：

全文共分为 各章节：

第一章为绪论，主要介绍本课题的来源、意义以及研究背景，关于OAuth 2.0、开放平台的国内外研究现状，以及本论文的主要研究工作。

第二章为相关技术介绍，

# 相关技术介绍：

## PHP

PHP(Hypertext Preprocessor)语言是一门新兴的创建动态交互站点的强有力的服务器端脚本语言，拥有完整的面向对象编程特点，在当前的Web开发领域得到了广泛的运用。根据w3tech.com上2017年统计的数据PHP写的网站已经拥有超过80%的市场份额。

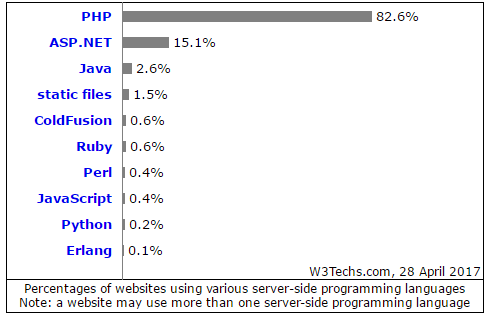


Figure 1-W3Tech上对于PHP的统计数据

PHP语言有非常多的优点：对于有类似C语言代码基础的人简单易学，上手较快，其弱类型的变量定义方式使代码简洁优雅，在项目管理，团队组建方面有其特有的优势。PHP拥有庞大的社区，大量的开源项目，合适的类库，功能较为丰富，性能足够好，可以进行热部署，项目部署比较简单，易修改，PHP具有较好的兼容性，，可以再不同的平台上运行（Windows，\*Unix，Mac OS），PHP与目前Web应用所使用的服务器（IIS，Apache，Nginx）所兼容，可以高效的运行于服务器端，并且PHP提供了广泛的数据库支持。

针对PHP的以上优点，本项目采用PHP作为主要开发语言。

## Laravel

Laravel是一款基于PHP语言的优雅的Web框架，其借鉴了许多框架以及设计技巧，比如Ruby on Rails、ASP.NET MVC等。它集成了许多有用的功能，这些功能是经常在Web应用中需要实现的，而Laravel集成了这些功能模块，增加了开发的便利，这些模块包括安全验证、路由、数据迁移、Session、缓存、邮件、Eloquent ORM等功能，在日常开发中，身份凭证验证几乎是必不可少的，而laravel框架已经继承了身份验证的模块，只需要Auth命令即可实现包括登录验证、获取登录用户信息、密码加密等功能；laravel提供了非常简洁的路由方式，所有的路由集中在route文件中，查看项目的路由可以非常方便简洁；数据库迁移在团队协作中可以非常方便的保持数据的统一；Eloquent ORM对象关系映射框架简化了数据库操作，并避免了SQL注入等可能存在的数据库漏洞；这些Laravel框架的特性大大提高了程序员的开发效率。对于PHP5.3以上的版本，可以使用Composer作为依赖管理工具，Composer借鉴了Maven、npm的设计理念，Composer安装依赖到项目的Vender目录，而不是全局，因此它仅仅是依赖管理工具。Composer可以通过命令行require安装，也可以在项目根目录下的composer.json文件里声明依赖，然后通过install命令下载或者update命令更新。Composer的出现方便了PHP开发者进行依赖管理，极大地提高了PHP的开发效率。

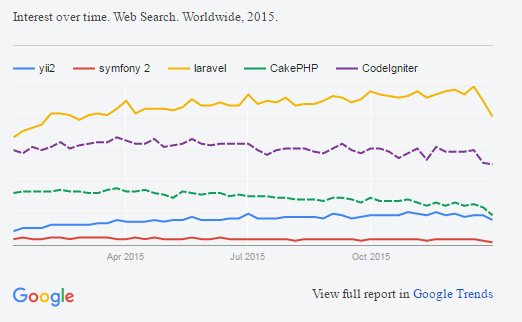


Figure 2－PHP框架市场占有情况

## Apache

Apache http server是一款开源的Web服务器，以其安全性、稳定性和跨平台性而被广泛的应用，是当前市场上最流行的服务器端软件之一。它支持多种服务器端语言，比如PHP，Python，Java，Perl等。Apache在当前的Web服务器领域的应用已经十分成熟，并且有着十分庞大的社区资源，不断有开发者为其开发新的特性，新的功能，修复缺陷。Apache的优点是速度快、性能稳定，可以作为代理服务器使用、配置简单（比如对PHP的支持配置）。

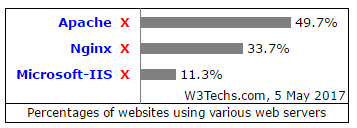


Figure -流行服务器市场占有情况

## HTTPS

HTTPS全称为超文本传输安全协议，通过提供了正在通信或者相关联网络上的安全身份验证，确保用户和站点之间的数据不能被获取，或者被第三方伪造，从而来保护隐私敏感数据。而HTTP，即超文本传输协议是不安全的，黑客会通过中间人攻击或者监听等手段获取网络传输过程中的敏感数据。在HTTPS中，客户端和服务器端通过对称加密来保证安全，但是协商对称加密算法时需要使用非对称加密来保证安全，但是非对称加密的过程是不安全的，中间人可以篡改公钥，因此客户端和服务器端使用数字证书签发机构颁发的数字证书来保证非对称加密过程的安全性，这样就可以协商出一个对称加密算法，客户端和服务器端通过对称加密算法来保证通信过程中的数据安全性。

前面我们用Apache作为本项目的服务器，而在项目中想要使用HTTPS是通过安装SSL证书来实现的。SSL全称为安全套接字层，是一种加密浏览器与服务器之间信息通信的标准安全技术，可以确保服务器和浏览器之间信息传递的安全性。我们需要在受信任的证书颁发机构（CA）处购买证书，然后将证书安装到Apache服务器，即可使用https来访问服务器端内容。

## OAuth 2.0

## 三元权限

## Ajax

## Redis

## RabbitMQ

## 单点登录

## 心跳检测

## 百度统计

# 网络账号系统的分析与设计

## 网络账号系统的需求分析

### 目标

本课题来自中国科学院软件研究所操作系统基础版研发支撑平台（ISDE）。在描述本网络账号系统的开发目标之前先引入应用集成的概念。随着公司企业的不断发展，公司使用的应用软件会不断增加，这些软件大多是来自世界各地的开源贡献者贡献的开源应用或者系统以及公司内部针对解决特定业务而开发的第三方应用。这些第三方应用大多都提供了开放平台和Open API对外提供服务。通过对这些应用的集成可以节省开发成本，满足新的业务需求。狭义来讲，本网络账号系统是针对当前实现的应用集成平台（Application Integration）来进行开发的，但随着之后应用规模、数量的不断扩大，网络账号系统也会应用于软件所的社区，数据资源管理等方面，本文主要讲述针对于应用集成平台（以下简称AI）的网络账号系统的开发。

本网络账号开发的目的主要有以下几点：

1. 基于OAuth 2.0授权协议的单点登录，支持应用集成平台AI，包括Gitlab、JIRA、mantis、TestLink等系统的登录认证。并且我们要实现针对于网络账号系统和AI平台的工作流系统。工作流系统的作用是“使在多个参与者之间按照某种预定义的规则传递文档、信息或任务的过程自动进行，从而实现某个预期的业务目标，或者促使此目标的实现”。在进行服务集成的时候工作流系统可将业务流程的任务关联到第三方工具的具体服务，通过业务流程的执行满足业务需求。因此我们需要实现网络账号系统与AI应用集成平台之间的预定义的文档、消息、任务等的自动传递。
2. 实现开放平台，开放简洁易用的Open API，向外提供丰富的接口以提供大量的信息，并提供接口测试工具使接入用户更加直接快捷的了解学习Open API和消息队列的调用和数据参数传递。
3. 网络账号系统作为用户中心平台，还需要管理公司内部所有人员的基本信息，群组关系以及任务分配。因此需要处理用户、群组、任务之间的相对关系，并且对三者的数据进行管理。

### 可行性分析

可行性分析的目的是为了对问题进行研究，以最小的代价在最短的时间内确定问题是否可解，以及分析在经济、技术、运行组织上可能存在问题。

#### 经济可行性

经济可行性分析是从经济角度出发，估算系统的成本效益，包括系统运行、维护成本，对社会经济的效益。

网络账号系统将各个独立的子系统的账号统一收敛到一起，避免了传统的账号系统存在的账号繁杂、难以记忆维护的情况；并对公司的成员、项目组、任务关系进行统一管理，大大的节约了时间、人力成本，并且规范了业务操作程序，避免出现差错，提高了安全性和可靠性，从而可以提高公司以及项目组的信息化和现代化管理水平。总体来说，系统收益远大于成本。

#### 技术可行性

技术可行性是根据系统的目标来衡量是否具备所需要的技术。

本系统在开发过程中使用Ubuntu 16.04作为操作系统，PhpStrom作为集成开发环境，使用PHP和Laravel作为开发语言和Web框架，使用MySQL作为数据库系统。现有技术完全可以承担开发任务。

#### 操作可行性

本系统是基于B/S架构的系统，所以操作不受操作系统的影响，易于使用，不用进行专门的培训，管理员用户需要对公司内部的组织结构，任务分配有清晰的概念，并能进行有效管理。

### 系统的功能需求

功能需求规定了必须在产品中实现的软件功能，用户利用这些功能来完成任务，满足业务需求。经过与软件所总体部领导以及老师进行讨论，结合他们的工作流程以及功能细节，确定了系统所需具备的基本功能以及对于不同身份的用户应该具有怎样的功能，以及确定功能之间的关系。

进过详细的分析与规划，我们最终确定网络账号系统应具有以下几点功能：

1. OAuth 2.0授权流程实现以及多系统之间的单点登录。实现OAuth 2.0协议中作为功能最完整、流程最严密的授权码模式（authorization code）在客户端、授权服务器、资源服务器之间的流程，客户端使用access\_token实现单点登录。
2. 开放平台提供Open API，并且允许第三方应用注册接入。提供开放平台，Open API列表，管理员可以添加、删除、修改接口组、接口、测试API并查看消息结果及格式；用户可以查看接口组列表、接口列表、测试API并查看消息结果及格式。提供客户端注册，第三方应用需要提供重定向的URL，并可以选择客户端权限。
3. 用户中心的设计与实现。用户中心管理主要包括对用户、项目组、任务之间的管理，并且当用户、项目组、任务等信息发生变化时，需要对第三方应用即客户端进行信息推送，客户端进行对应数据操作从而保持数据一致性。其中因为用户有管理员和普通用户两种角色，因此网络账号系统用户中心的功能主要围绕着管理员用户和普通用户两种角色分开讨论。管理员用户主要有系统管理、注册新用户、组织管理、任务管理、人员管理、项目组管理、客户端注册管理、开放平台信息管理等功能。上述功能均具有增加、删除、修改、查询等基本操作。普通用户须具有客户端注册、开放平台查看接口列表和接口组列表的基本功能，并且普通用户可以申请管理员，经过管理员审核通过后角色转为管理员用户。

### 系统的业务需求

#### 组织结构介绍

本网络账号系统自上到下分别有公司（此系统中只有一个公司，即软件所总体部）、项目组、成员。用户类型主要分为学生、职工、实习生、责任QA，除此之外还有系统管理员角色。组内成员角色主要分为负责人、QA、普通成员。

#### 角色需求分析

作为网络账号系统，本系统不同角色主要需要实现的业务流程有：管理员管理用户、权限、用户类型、组内角色、审核管理员申请、注册用户、组织管理、任务管理、项目组管理、客户端管理、开放平台管理等业务；项目负责人主要负责项目组人员管理、任务分配等业务；普通成员包括学生、实习生、职工等主要负责解决人物、实际开发。

### 系统的性能需求

对于本网络账号系统，主要有以下几点性能需求：

1. 主体数据库的数据加载速度必须远大于新数据的产生速度，不允许出现主体数据库数据更新不及时的现象发生；
2. 数据库应支持应用系统或者公司、项目组、用户、任务等数据的同步或异步更新；
3. 在支持用户并发业务数据的综合查询时间<1s；
4. 能够支持并发1000人/秒的在线请求服务；
5. 系统需要满足7×24小时的不间断运行，运行一段时间后，整体执行效率不衰减。
6. 系统网站首页打开速度3s以下，账号登录速度 8s以下。
7. 系统能在高于实际系统运行压力1倍的情况下，稳定的运行24小时。

### 系统的安全需求

本网络账号系统涉及到用户的网络账号以及项目组信息、任务信息、用户信息 私密数据，本系统为Web应用，访问用户为互联网上的所有用户，这样大量的访问增加了危险性，因此，本系统的安全需求非常重要。本系统的安全需求分析主要得出以下几点需求：

1. 对系统敏感数据（如用户密码、数据库连接信息等配置信息）需进行加密处理。
2. 实现系统的保密性，为不同的用户提供不同的访问权限，只有经过授权的用户才能读写网络账号系统的信息，而且必须防止信息的非法、非授权的泄漏，防止对系统信息的越权存取。
3. 服务器需要能进行网络周边防御，以最大限度降低外界攻击的危险，并能及时检测外界的非法攻击和访问。
4. 本系统针对软件所总体部内部员工使用，因此需要进行IP地址访问控制，要求只能从指定的IP、IP范围、DNS来访问本系统。
5. 保证网络传输过程中的安全，避免第三方非法截取数据或者伪造数据。

### 系统的接口需求

本网络账号系统提供了开放平台，开放Open API供第三方接入并返回对应数据，此系统提供标准的Json数据格式与数据库对接，对第三方接入的应用的数据请求进行规范处理，以便于数据库的数据操作处理。对于一个有好的开放平台来说，从业务角度来说，Open API应该能够对公司组织、项目组、成员进行统一管理，维护公司、项目组、成员的职能和公共秩序，并能进行数据采集和数据挖掘分析；从技术上角度来说，我们需要实现RESTful风格的API，RESTful风格的API具有轻量，充分利用HTTP本身的语义，不需要其他的消息协议，并且具有透明性，面向资源，一目了然，无状态，不用考虑上下文，不考虑当前的状态，极大地降低了复杂度的优点，而且我们需要逐步完善统一Open API的数据标准，这些标准不仅能应用与本系统，也能够用于第三方应用以及本系统未来的扩展，这样可以促进系统内部和系统之间数据的标准化和一致性，有利于数据资源的整合、跨应用的数据资源共享和数据分析利用，有利于系统未来的升级和维护。因此我们对于这些开放平台提供的Open API也提出了以下接口需求：

1. 接口要简洁明了，使第三方可以快捷的接入，并迅速学会使用；
2. 保证接口数据的安全性，第三方必须提供access\_token以及相应的参数信息才能获取数据，并且要避免非法SQL注入或其他攻击来截取数据；
3. 保证接口的性能，要求达到100人/秒的压力；

## 网络账号系统的总体设计

### 系统体系架构

网络账号系统的总体架构设计如下图所示：



Figure -网络账号系统总体架构设计图

### 系统功能模块结构

网络账号系统的功能模块结构图如下图所示：



Figure 5-网络账号系统功能模块结构图

## 网络账号系统的功能模块设计

### 基于OAuth 2.0的身份凭证认证框架：

本网络账号系统最基础的功能模块支撑就是基于OAuth 2.0协议的身份凭证认证框架，身份凭证认证框架提供了基础的账号密码认证，包括获取access\_token的OAuth 2.0协议流程，根据access\_token授权获取数据等基本功能。

其功能结构图如下图所示：



Figure 6-基于OAuth 2.0的认证模块功能结构图

基于OAuth 2.0的身份凭证认证模块的流程图如下图所示：



Figure 7-基于OAuth 2.0的身份认证模块流程图

其中，我们主要是用授权码模式来实现OAuth 2.0的授权流程，其流程图如下图所示：



Figure -授权码模式的授权流程

每一步的消息传递如下：

（A）当用户访问客户端时，客户端会将用户转向认证服务器的授权登录页面。

（B）在授权登录页面，用户可以登录以及选择是否给客户端授权。

（C）如果用户给客户端授权，认证服务器将用户重定向至客户端事先规定的附有授权码（Authorization Code）的URL（Redirection URL）。

（D）客户端请求附上授权码的重定向URL，向认证服务器申请Access Token。客户端在后台的服务器上完成这一步骤，对用户是不可见的。

（E）认证服务器在后台校验授权码和重定向URI的对应关系，确认一致后，通过回调URL向客户端发送授权令牌（Access Token）和更新令牌（Refresh Token）。

### 服务配置：

服务配置模块提供配置本系统各个功能模块的参数配置，将服务配置独立出来统一进行管理，可以提高项目迁移、服务修改的效率，其功能结构图如下图所示：



Figure -服务配置模块功能结构图

服务配置的流程图如下图所示：



Figure -服务配置流程图

### 组织管理：

### 项目组管理：

### 任务管理：

### 用户管理：

### 三元权限管理：

### 客户端管理：

### 开放平台管理：

### 系统消息队列管理：

## 网络账号系统的数据库设计

## 网络账号系统的安全设计

# 网络账号系统的实现与测试

# 总结与展望

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××,×××××××××××××××××××××××××

--------章与章之间插入分页符----------

**参考文献** (黑体小2号加粗居中)

[1]□王静康,张凤宝,夏淑倩等.论化工本科专业国际认证与国内认证的“实质性”.高等工程教育研究,2014,5:1-4

[2]□Stone J A, Howard L P. A simple technique for observing periodic nonlinearities in Michelson interferometers. Precision Engineering,1998,22(4):220-232

[3]□朱印红,袁衍明.Dreamweaver完美网页设计——技术入门篇.(第一版).北京:中国电力出版社,2006:19～20

[4]□Lewis S L. Physics and chemistry of the solar system.北京:北京大学出版社,2014.1～2

[5]□陈剑.上博简《民之父母》“而得既塞於四海矣”句解释[EB/OL］.简帛研究网站，http://www.bamboosilk.org/Wssf/2003/chenjian03.htm．2003-01-18

( 宋体小4号)

……

……

……

**附录** (黑体小2号加粗居中)

×××××××××××××××××××××××××××

( 宋体小4号)

……

……

……