# 绪论：

选题背景和意义：

2004年7月，美国费城首次提出建设基于WLAN标准的无线宽带城域网络，也叫“无线费城计划”，但令人遗憾的是，“无线费城计划”最终因商业模式不够明朗而以失败告终，尽管中途有几次“重启”，但最终项目受到种种挑战而不得不终止。今时今日，wifi已经进入的千家万户，每个公司内部有wifi，每个家庭内部有wifi，每个公共场所也有wifi，它已经成为了我们生活中必不可少的一部分。在一个公司内部，在一个公司内部，内部人员可以登陆wifi之后使用公司网络。当有外部人员想要使用wifi时，需要通过一些方法才能连接公司wifi，例如企业内部人员为此外来人员申请wifi接入密码，外来人员通过输入此密码来使用wifi。在输入密钥之后系统会进行认证，密钥错误则无法连接wifi。但是连接公司的wifi之后，公司内部网络有被攻击的风险，以及需要处理访问控制方面的问题。

所以，在这种背景下，如何设计、开发一个多级身份管理系统，来管理一个企业的wifi接入情况，并且具备灵活的登录、多重认证和多级安全策略，是当前需要解决的事情。如此，企业wifi的管理将会更加的透明，直观，网络系统以及数据也会更加的安全。

国内外研究现状：

多级身份管理目前国内的研究和应用集中在普通网络环境中的企业级解决方案,主要满足企业内部员工以及企业外部外来人员对于全生命周期身份管理和单点登录的需求。针对高敏感信息和高安全等级的特定网络环境的需求,提出了多级身份管理的概念。

在多重认证方面，大多数软件采用的在登陆时进行验证码验证功能。

在访问控制方面，逐渐出现了DAC、MAC、RBAC、ABAC等访问控制模型。

在基于WEB的用户认证管理系统的设计与实现文中，作者充分理解4A(帐号Account、认证Authentication、授权Authorization、审计Audit)理念，设计了基于Web的用户认证系统。基于Web的用户认证系统提供了统一的集中管理平台，集中管理用户帐号、集中登录认证、集中用户授权和集中操作审计。本论文阐述的基于Web的用户认证系统，提供了统一的集中管理平台解决了电台内部应用系统之间统一认证、单点登录、操作审计的问题，提高了电台内部网络的安全性和易用性。

现在国内有很多身份管理系统的网站，例如第三只眼，神州云动，泛微，天易成等，这些公司大多专注电脑管控、局域网管控、聊天管控、邮件管控、屏幕管控等功能，已经基本满足大多数企业的需求。

课题任务：

首先，在网上查阅文章，分析现有网络身份特点及身份管理基础框架以及包括的基本功能，对此进行分析，利用前端开发技术以及后段开发技术实现web应用，身份管理整合了用户身份信息保护和资源访问控制等诸多技术，具备灵活的登录、多重认证和多级安全策略。

然后，了解到当前的主流开发网站的语言是HTML5，据统计2013年全球将有10亿手机浏览器支持HTML5，同时HTML Web开发者数量将达到200万。毫无疑问，HTML5将成为未来5-10年内，移动互联网领域的主宰者。HTML 5开发领域的领军人物包括Sencha，Adobe，Appcelerator，appMobi及Facebook，亚马逊，Google三大巨头。不管我们是想开发出新型视频应用的开发商如Brightcover还是想开发新型音频应用的开发商如Soundcloud，不论是桌面应用还是移动应用，HTML 5都是创新的主旋律。在几年的时间里，Node逐渐发展成一个成熟的开发平台，吸引了许多开发者。有许多大型高流量网站都采用Node.JS进行开发，Node发展如此之快得益于社区的贡献，非常多的程序员为了壮大Node而贡献自己的一份力量。Node是这个时代非常火爆的后台语言。

接着，思考如何进行多重认证，如何实现多级安全策略，以及如何使系统变得更加安全。

最后，学习HTML，CSS，JS语言以及Node.js的基础知识，包括前端的页面开发基础，Bootstrap以及js的基本操作，最后学习node.js如何搭建服务器，如何连接mongodb进行数据库的操作最后确定使用基于角色的访问控制技术来实现多级安全策略以及访问控制。

本系统主要了企业wifi内部的多级身份管理功能。

管理员在连接的wifi重定向到系统登陆界面之后，输入账号密码进入系统，进入系统之后，会添加角色，添加权限，管理员可以给每个角色加上不同的权限，不同的权限可以访问不同的页面，也就是说不通的角色可访问的页面是不同的，一个角色可以绑定多个权限，之后就可以进入添加员工页面来添加员工，同样，会为员工绑定角色，一个员工绑定一个角色。当这个新建的员工也连接wifi登陆系统之后，就会从服务器获取自己的角色以及对应的访问页面的权限，每当访问系统中的某个页面都会进行一次权限的校验，员工可以为外来人员申请介入wifi 的密码，外来员工获取到这个密码之后，在登陆界面点击“我是外来人员”就进入的对应的输入密码的界面，输入之后服务器进行计算，来确定是否可以连接公司内部网络。连接之后管理员就可以查看当前连接Wi-Fi的具体情况以及日志。

本文共分为6章，基本为以下内容:

第一章主要介绍了选题背景及意义、国内外的研究现状以及发展态势、本课题的主要任务以及本文的结构。

第二章主要介绍了本课题所需要用到的主要的技术的基础知识，包括HTML,Node.js等，以及本多级身份管理系统的第三方平台以及开发工具等。

第三章详细地介绍了多级身份管理系统的需求，对系统的具体内容进行阐述，从系统的设计开始，详细介绍此系统的模块设计方案以及RBAC的实现方法，对系统的安全作出解释，以及如何实现多重认证、多级安全策略。

第四章主要介绍了登陆模块，管理员模块，员工模块，外来人员模块的主要实现方法。

第五章主要介绍了如何对系统进行的测试。

第六章主要对本课题的主要工作进行了总结，对目前的成果进行评价，以及对多级身份以及访问控制安全的一些展望。

# 第2章 课题相关理论基础及技术

2.1 HTML技术

HTML是一种超文本标记语言，它是一种规范，一种标准，是一种能被浏览器所识别的语言，一个网页本身是一个文件，在文件中编写代码，告诉浏览器如何显示其中的内容，首先浏览器会加载外部脚本(.js)和央视表文件(.css)，之后会解析并且执行该脚本代码。

2.2 Node.js 技术

近年来，随着智能手机与HTML5的不断普及，Javascript脚本语言的重要性也随之不断提升，而且由于最近几年许多Javascript引擎中均内置了一种编译器，叫做JTI，让Javascript引擎的处理速度得到了大幅度的提升，让Javascript的运行速度变得奇快无比，甚至逊色很多服务端开发语言。于是，很多工程师就提出了这样的一个想法，如果能够让服务器端跟客户端一样使用同一种语言来开发，岂不妙哉？

于是在2009年，美国人Ryan Dahl推出了一个遵循CommonJS标准的服务器端Javascript脚本语言开发框架，它的名字叫Node.js。在Node.js内部，运行的是V8 Javascript语言，该语言可以运行在服务器端，我们都知道js是单线程的，所以Node.js的一个非常重要的特性就是它可以通过单线程实现异步的操作，通常，提起异步处理，大家总是会想到服务器端的多线程处理方式，在Node中，我们通过时间环已基非阻塞I/O基址实现服务器端的异步处理。

Node.js修改了客户端到服务器端的连接方法，解决了这个问题，因为它并不为每个客户端连接创建了一个新的线程，而是为每个客户端连接触发一个在Node.js内部进行处理的事件，因此如果使用Node.js，可以同时处理多达几万个用户的客户端连接。

严格的说，Node.js是一个用于开发各种Web服务器的开发工具，在Node.js服务器中，运行的是高性能的V8 Javascript脚本语言，该语言是一种可以运行在服务器端的Javascript脚本语言。V8 Javascript引擎是由Google公司使用C++语言开发的一种高性能Javascript引擎，该引擎并不局限于在浏览器中运行，Node.js将它转用在了服务器中，并且为其提供了许多附加的具有各种不通用途的API。例如在一个服务器中，经常需要处理各种文件，对文件进行操作，还有可能对各种二进制数据进行操作，在Javascript，只具有非常有限的对二进制数据的处理功能，而Node.js所提供的Buffer类则提供了丰富的对二进制数据的处理能力。

另外，在V8 Javascript引擎内部使用一种全新的编译技术。这意味着开发者编写的Javasript脚本代码与开发者编写的地段的C语言具有非常相近的执行效率，而这也是Node.js服务器非常重要的一个特性。

2.3 MongoDB 数据库

MongoDB是一个基于分布式文件存储的数据库。我认为它属于非关系型数据库，是非关系型数据库单中功能最丰富，最像关系数据库的。它支持的数据结果非常松散，是类似JSON的格式，文档包含一个或者多个字段，每个字段包含特定数据类型的值，包括数组二进制数据和子文档，倾向于共享类似结构的文档被组织为集合。因此可以存储比较复杂的数据库。

主要特点：

1. 高性能：

MongoDB提供了高性能的数据持久型尤其是对嵌入式数据模型的支持减少了数据库系统上的I/O活动以及在索引方面支持更快的查询，并且可以包含来自嵌入式文档和数组的键。

1. 丰富的查询语言

MongoDB支持丰富的查询语言来支持读写操作以及数据聚合、文本搜索和地理空间查询。

1. 高可用性

MongoDB的复制工具，称为副本集，提供自动故障转移和数据冗余。副本集是一组保持相同的数据集，从而提供冗余和提高数据的可用性的MongoDB服务器。

1. 水平可伸缩性

作为核心功能的一部分，MongoDB提供了横向可伸缩性，分片在一组计算机集群分布数据。从3.4版本开始，MongoDB支持基于分片键创建数据区域，在一个平衡的集群中，MongoDB只将区域覆盖的读写写入区域的碎片。

2.3 Vue技术

Vue是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。其实，Vue在我的手里边就是用来开发单页面应用的一个工具，它可以让我的程序开发起来更加的方便，它会自动的响应数据的变化情况，并且根据用户在代码中遇险设定好的绑定关系，这在后边章节中介绍代码的时候着重讲解，简单来说就是数据变化，跟着它绑定的数据就会跟着变化。Vue通过组件，它把单页面应用拆分成各个组件，每一个组件都是一个单独的.vue文件，我们只需要在它们的父组件中先引入各个组件的URL，然后写好各个组件的标签在父组件的模块中，就像给函数传入参数一样，这个参数叫做组件的属性，然后再分别写好各种组件的实现，这就是Vue单页面应用的具体实现方法。

Virtual DOM：中文名称是虚拟DOM，简单来说，这就是一种可以预先通过Javascript进行各种计算，把最终的DOM操作计算出来并且优化，由于这个DOM操作属于预处理操作，并没有真实的操作DOM，所以叫做虚拟DOM，最后在计算完毕才将真正的DOM操作提交，将DOM操作变化反映到DOM树上。

2.4 LeanCloud实时通信功能

LeanCloud是行业领先的一站式后端云服务提供商，专注于开发者提供一流的工具、平台和服务。它提供了包括数据存储、文件存储与CDN、消息推送和实时通信在内的后端云服务，同时提供支撑后端代码的云引擎和云函数等开发工具，全面涵盖移动开发的需求；我们同时也提供了易于集成的全平台 SDK，支持 iOS、Android 应用和游戏开发，以及包括微信小程序在内的 web 开发。

LeanCloud实时通信功能是本系统所要使用的功能，实时通信在 SDK 里面分为两个层次：

* 一层是底层实时通信基础模块，负责与服务器匹配和授权，建立基础的连接和底层控制。这个层次只会派发一些基础的事件出来，你可以通过 SDK 监听这些事件。这个层面 SDK 会保证底层连接的稳定，包括断开重试，心跳连接等策略；
* 另一层是业务逻辑层，用户可以使用 SDK 建立不同的 Conversation（对话）。一个 Conversation 就是一个独立的通信单元，但 Conversation 间一般是无法通信的。当然你可以自己在业务逻辑层，通过派发自定义事件的方式来封装其他自定义的逻辑。当创建一个新 Conversation 之后，对应的服务器端就会自动生成这个 Conversation，除非你自行删除，否则该 Conversation 一直存在。但是用户如果没有连接，该房间不会占用服务器资源，只是存储的一个数据条目；

2.5 Sublime Text3

Sublime是我学习的前端开发之后最喜欢的开发工具以及软件。

Sublime Text 是一个代码编辑器，也是HTML和散文先进的文本编辑器。它支持多种编程语言的语法高亮、拥有优秀的代码自动完成功能，还拥有代码片段（Snippet）的功能，可以将常用的代码片段保存起来，在需要时随时调用。支持 VIM 模式，可以使用Vim模式下的多数命令。支持宏，简单地说就是把操作录制下来或者自己编写命令，然后播放刚才录制的操作或者命令。

Sublime还有编辑状态恢复的能力，即当你修改了一个文件，但没有保存，这时退出软件，软件不询问用户是否要保存的，因为无论是用户自发退出还是意外崩溃退出，下次启动软件后，之前的编辑状态都会被完整恢复，就像退出前时一样。

Sublime非常突出的功能就是它具有良好的扩展功能，官方的称呼是安装包。使用快捷键(ctrl+shift+p)就可以寻找自己想要的安装包，重启软件就可以使用该安装包。还有一个非常棒的特点就是它具有飞常强大的快捷键功能，可以选择整行，从光标处移动至尾行，合并行等。

这种体积小，运行速度快，文本功能强大，还支持变异功能以及具有Package Control大量插件的开发工具，实在是让人爱不释手。

2.6 本章小结

本章介绍了在本次课题中会使用到的主要技术，前端开发HTML的基本理论知识，后端开发Node.js的基本知识，以及会使用到的第三方平台和开发工具，为后续的多级身份管理系统的需求分析和设计开发做好铺垫。

# 第3章 多级身份管理系统的需求分析与设计

3.1 功能性需求：

本课题的主要任务是采用HTML加Node.js实现一个具有管理角色、管理权限、管理员工，以及为外来人员申请wifi登陆密码，外来人员连接WIFI等功能的多级身份管理系统。在本系统中，会设计三种不同的用户，管理员，内部员工以及外来人员。

3.1.1 系统内主要角色

1. 系统管理员

系统管理员在连接wifi登陆系统之后，可以进入他的功能列表界面，在系统中，他可以添加员工的角色，添加权限，给角色绑定任意的一种或多种权限，给员工绑定一种且只能一种权限。还可以查看当前连接wifi的情况。

1. 公司内部员工

公司内部员工具有多种角色，不同的员工具有相同或者不同的角色。例如：普通员工，前台人员等。他们的账号由管理员创建，可以修改密码。如果他们所属的角色具有为外来人员申请wifi密码的权限，那么他们就可以进入申请密码的页面为外来人员申请密码。

1. 外来人员

外来人员是那种不属于公司的人员，他们或许是其他公司派遣来学习的同事，或者是一些普通的人来到了公司办事情或者参观等。他们需要连接公司wifi的时候，就需要像拥有申请密码权限的公司内部员工获取为他们申请的密码，待密码输入之后服务器进行校验，通过就可以使用公司wifi。

3.1.2 管理系统主要功能模块

系统主要分为四大模块，分别是登陆模块，管理员模块，内部员工模块以及外来人员模块，一下便是对每个模块的具体介绍。

1. 登陆模块

任何管理系统皆需要登陆功能，在连接wifi之后，任何人都会被重定向到系统的登陆界面，不论是管理员还是外来人员都需要在此登陆才能使用网络。在登录框中输入账号以及密码之后，服务器对身份进行验证，验证成功则进入对应的系统页面，否则进行登陆失败提示。

基本顺序为：

1. 用户进入系统登陆界面
2. 在输入框中填写账号密码
3. 点击登录提交数据到服务器进行验证
4. 管理员模块

管理员在登陆系统之后，进入功能列表页面，具有5种基本功能。

1. 角色管理：因为只有建立了角色之后才能在新建员工的时候为员工匹配角色。角色指的是公司内部员工所属的不同种类，所以就拥有不同的角色。在角色管理模块中，管理员可以为角色匹配一种或者多种权限。管理员也可以对角色进行删除操作。
2. 权限管理：权限具有2个属性，一个是权限的名称，另外一个是该权限所对应的能访问的URL，一个权限只能绑定一个URL。管理员也可以对权限进行修改和删除操作。
3. 新建员工：这是管理员最基本也是最重要的功能，管理员输入员工的姓名以及他的密码，然后在下方为管理员选择他对应的角色，点击确认即可添加员工。
4. 员工管理：顾名思义，员工管理即可查看当前员工的信息，以及对员工的权限进行把控，例如员工是否可以登陆系统的权限，当点击员工禁用按钮的时候，该员工则不能登陆系统，也不能为外来人员申请wifi接入密码。
5. 登陆日志：次模块中，管理员将会看到2个表，一个是公司内部员工登陆状态表，第二个则是外来人员登陆状态表。管理员可以看到他们的登陆时间，已登录时长，如果是外来人员则可以看到是谁为他们申请的wif接入密码。
6. 内部员工模块

内部员工在连接wifi登陆系统之后，系统会判断他的角色，进而确定该员工对系统内部页面访问的权限。内部员工具有2个基本的功能。

1. 为外来人员申请wifi接入密码：首先内部员工点击申请wifi密码的按钮进入该页面，之后只需要填写外来人员姓名发送到服务器，然后服务器会返回接入密码到结果框中。内部员工把此密码给外来人员，外来人员通过输入密码便可连接wifi。
2. 修改密码：这是比较基本的功能，员工依次输入旧密码、新密码、确认新密码，待系统校验通过便修改成功。
3. 外来人员模块

外来人员在登陆界面点击”我是外来人员”链接，便可跳转到输入密码界面，在获取到公司内部员工为他申请的密码之后，把密码输入到输入框中，点击确认，待系统校验成功便可使用wifi访问网络。

3.2 非功能性需求

本次课题有一些非功能性需求，以下是基本的介绍：

1. 安全性

由于是本次课题的成果是以软件形式，更加详细来说是网站的模式存在，所以针对与网站的安全方面需要更加重视，包括接口安全，数据库信息安全，以及访问控制的安全。

1. 健壮性

在用户输入非法字符或者包含有恶意攻击的代码时，系统前端页面会自动识别，待数据传送带后台之后，服务器也会再次确认参数的规范程度。

1. 可维护性

系统采用Vue前端框架，没一个页面都是有不同的组件构成，当页面需要接入新的功能的时候，只需要再开发一个新的组件，添加到该页面对应的组件中去即可。在页面出现bug的时候，可以精确定位到模块中的某一行中。

1. 易使用性

良好的界面时一个优秀的软件所必不可少的部分，前端是最贴近用户的角色，项目中使用的Element-ui来提高页面与用户的交互能力，让用户在系统中操作功能的时候更加的顺畅，简单。

3.3 业务流程分析

多级身份管理系统的工作流程为：系统启动之后，连接wifi进入系统登陆界面，管理员输入账号密码进行登陆之后，可以进行系统后台的所有操作，包括添加员工、员工管理、角色管理、权限管理以及当前wifi连接情况统计。员工登陆系统之后，可进行为外来人员申请连接wifi的密码，以及账号的管理功能。外来人员在登陆页面进入外来人员专属登陆界面，输入公司内部员工为他申请的密码之后，也可以访问网络。本系统的业务流程图如图3.1所示：

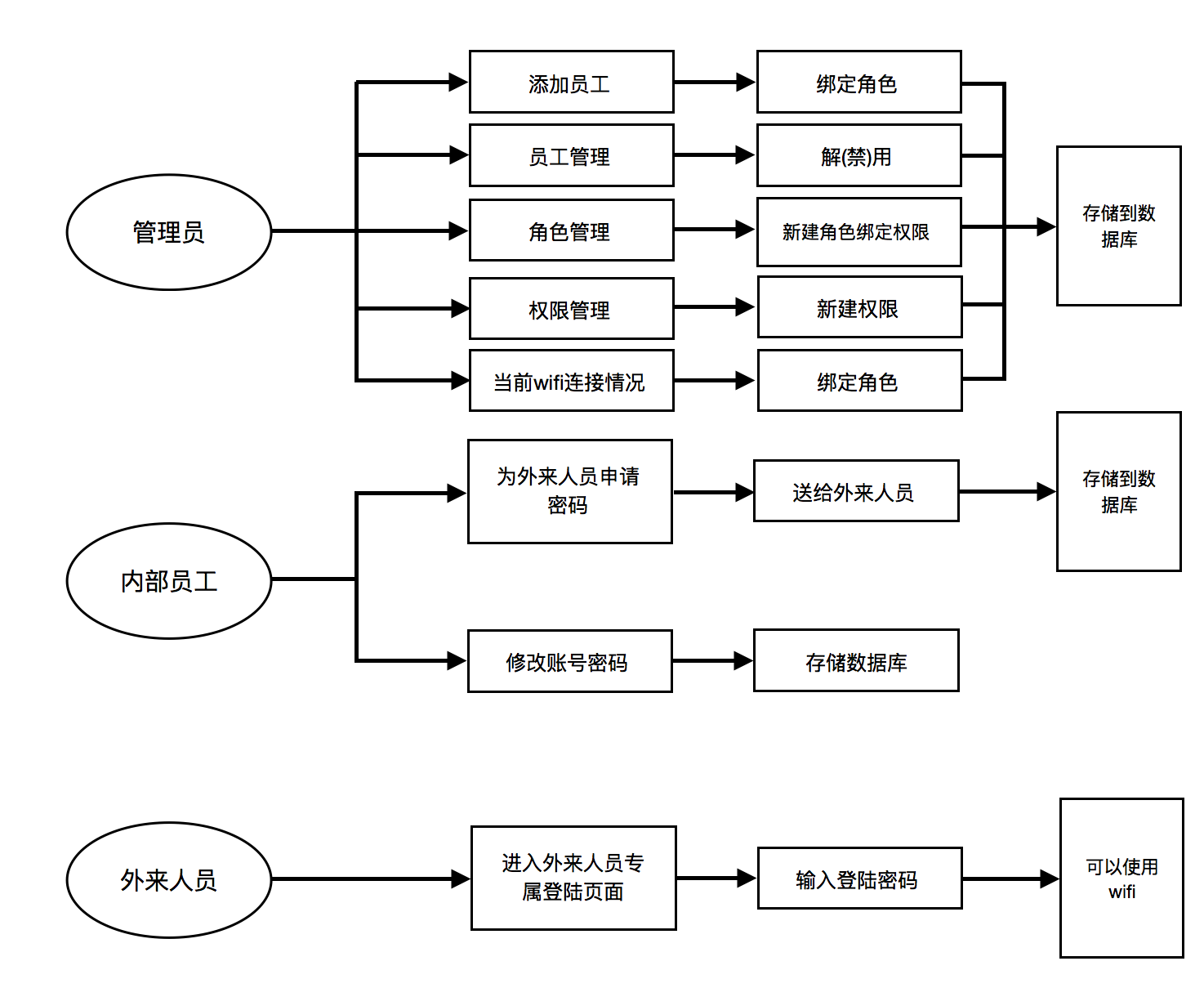


图 ‑1业务流程图

3.4 数据库设计

设计数据库，看似简单的事情，却是本次课题的一个非常重要的部分，在设计到多级身份访问策略的时候，我门采用的是基于角色的访问控制，简称RBAC。在RBAC中，权限与角色相关联，用户通过成为适当角色的成员而得到这些角色的权限。这就极大地简化了权限的管理。在一个组织中，角色是为了完成各种工作而创造，用户则依据它的责任和资格来被指派相应的角色，用户可以很容易地从一个角色被指派到另一个角色。角色可依新的需求和系统的合并而赋予新的权限，而权限也可根据需要而从某角色中回收。角色与角色的关系可以建立起来以囊括更广泛的客观情况。

为了实现访问控制，我需要在数据库中设计7张表，没有错，就是7张表。

表格 ‑1 管理员表

|  |  |  |  |
| --- | --- | --- | --- |
| 英文字段 | 中文含义 | 数据类型 | 备注 |
| \_id | 对象编码 | String | 数据库自动生成 |
| account | 用户名 | String | 不可改变 |
| password | 用户密码 | String | 可以改变 |
| level | 身份等级 | Number | 身份的等级 |

表格 ‑2 内部员工表

|  |  |  |  |
| --- | --- | --- | --- |
| 英文字段 | 中文含义 | 数据类型 | 备注 |
| \_id | 对象编码 | String | 数据库自动生成 |
| account | 用户名 | String | 不可改变 |
| password | 密码 | String | 可修改 |
| tel | 员工电话 | String | 用来获取验证码 |
| banner | 员工是否被禁用 | Boolean |  |
| onLine | 员工是否在线 | Boolean |  |
| Ips | 员工常登陆IP裂变 | Array | 根据此来判断是否要用多重认证 |
| startTime | 登陆时间 | Number |  |
| level | 身份等级 | Number | 身份的等级 |

表格 ‑3外来人员表

|  |  |  |  |
| --- | --- | --- | --- |
| 英文字段 | 中文含义 | 数据类型 | 备注 |
| \_id | 对象编码 | String |  |
| name | 外来人员姓名 | String | 外来人员姓名 |
| EncryptedName | 加密后的名字 | String | 加密后的姓名 |
| password | 申请到的密码 | String |  |
| stuffId | 为他申请密码的员工id | String |  |
| stuffName | 为他申请密码的员工姓名 | String |  |
| startTime | 开始连接时间 | Number |  |
| level | 身份等级 | Number |  |

为了实现RBAC，我需要添加以下的表：

表格 ‑4角色表

|  |  |  |  |
| --- | --- | --- | --- |
| 英文字段 | 中文含义 | 数据类型 | 备注 |
| \_id | 对象编码 | String |  |
| name | 角色名称 | String | 管理员所建 |

表格 ‑5权限表

|  |  |  |  |
| --- | --- | --- | --- |
| 英文字段 | 中文含义 | 数据类型 | 备注 |
| \_id | 对象编码 | String |  |
| name | 权限名称 | String |  |
| url | 此权限对应的路由 | String | 一个权限只有一个路由访问权限 |

表格 ‑6角色权限关系表

|  |  |  |  |
| --- | --- | --- | --- |
| 英文字段 | 中文含义 | 数据类型 | 备注 |
| \_id | 对象编码 | String |  |
| userId | 内部员工的对象编码 | String | 数据库中员工的\_id值 |
| roleId | 角色的对象编码 | String | 数据库中角色的\_id值 |

表格 ‑7角色权限关系

|  |  |  |  |
| --- | --- | --- | --- |
| 英文字段 | 中文含义 | 数据类型 | 备注 |
| \_id | 对象编码 | String |  |
| roleId | 角色的对象编码 | String | 数据库中角色的\_id值 |
| AccessId | 权限的对象编码 | String | 数据库中权限的\_id值 |

3.5 本章小结

本章节主要介绍了本次课题的功能性需求，包括系统内的主要角色、系统主要功能模块，接着简单介绍了非功能性需求包括安全性、健壮性、可维护性和易使用性，最后是系统的数据库表的设计，为后边的web应用的开发打好基础，之前刷知乎的时候，读到过一句话:”开发web工程的时候，准备工作是最重要的事情，否则当你面临麻烦的时候你甚至都会忘掉你之前所准备的大多数数据”。所以，准备工作至关重要。

# 第4章 多级身份管理系统的开发与实现

4.1搭建前后端开发环境

4.1.1系统开发环境以及辅助工具

本课题需要同时进行前端开发以及后段开发，所以前端开发以及后段开发都要进行配置。以下便是本课题所需要的具体开发工具以及语言:

1. 操作系统: MacOS 10.13.3
2. 前后端开发工具: Sublime Text3
3. 前端编程语言: HTML
4. 后端编程语言: Node.js
5. 数据库: MongoDB
6. 前端开发框架: Vue + webpack + Es6
7. 辅助第三方平台: Leancloud

4.1.2 Node.js环境的安装与配置

本课题我选择使用Node.js作为后台开发语言而不是java主要是因为想要突破自己，让自己在大学的最后一段时间再学习一门新的技能。很庆幸，我学习到了一些自己想要学习到的知识。

安装Node.js的话，首先打开浏览器访问Node.js的官网，进入官网之后，下载稳定版本，本课题使用的Node.js版本为8.1.3，下载完成后，双击刚刚下载的文件，按照默认步骤一路next就安装成功啦。然后打开终端，输入

node –v和npm –v

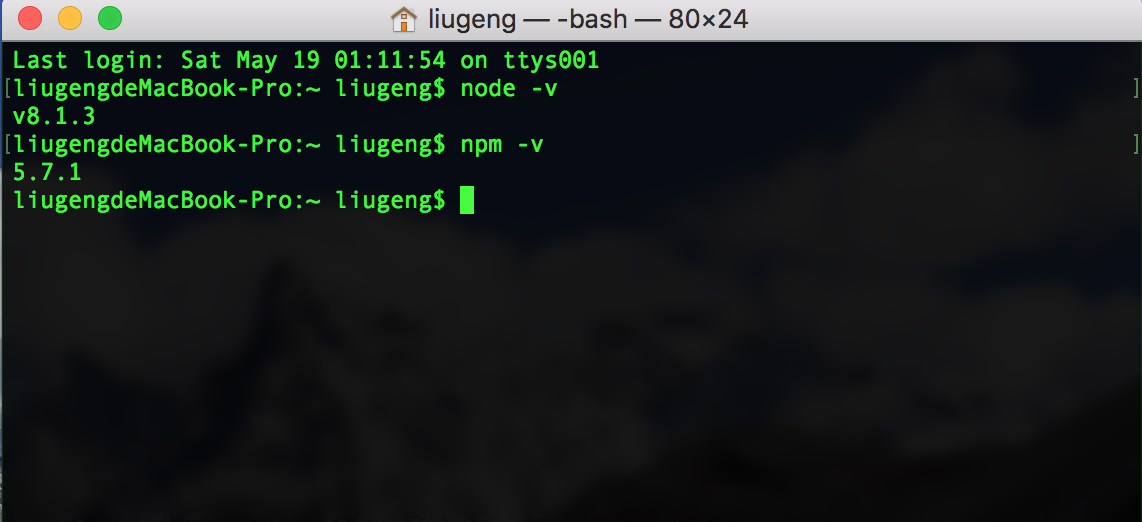


图 ‑1Node版本

如果出现了如图4-1一样的结果，便说明安装成功。大家会发现，我多打了一行命令;npm-v。

npm是什么？

npm是Node.js的包管理工具，npm为JavaScript开发者提供了一个分享代码的方式，同时也能够为开发者提供一个在他们项目中重用代码的方式。这些被重用的代码被称为包（packages），有时候也被称为模块（modules）。一个包通常包含了一个或者多个文件，同时也包含了一个定义了这个包元信息的文件"package.json"。在一个应用中，例如网站会依赖成百上千个包，这些包一般来说都比较小。通常情况下开发者创建一个包用于解决一个特定的问题。本课题中，很多前端开发便要使用到很多npm的功能来帮助我们提高开发的效率。

至此，本课题中的后台环境已经搭建完成，在全局作用域中都可以使用。

4.1.3前端开发工程的搭建

说到前端开发工程的搭建，我使用的是Vue，在第二章中我已经介绍过Vue的基础于概念，这里不再赘述。下面需要陈述的是如何使用Vue搭建我们想要的前端工程框架。

首先，我们需要在我们的硬盘空间中新建一个文件夹来存放我们的项目工程，因为本次课题属于毕业设计，所以我起名final，不仅意味着我大学生涯的结束，也意味着我大学学习的结束。

然后，打开终端，输入指令” cnpm install -g vue-cli”，enter，等待安装，这里就会有疑问，不是安装Vue吗？为什么安装的是vue-cli呢？这里解释一下，官方Vue项目模板的目的是提供有意义的，包含电池的开发工具设置，以便用户尽快开始使用实际的应用程序代码。然而，这些模板对于如何构建应用程序代码以及除了Vue.js之外还使用了哪些库而言并无特色。它还有一个比较大众的名字就是Vue脚手架，安装了vue-cli我们便可以非常快速的让Vue帮我们自动搭建好我们的工程目录在我们的final文件夹中。所有的我们后面开发需要饮用的第三方包或者是中间件都存在了项目中一个叫做node\_modules的文件夹中，这便是上边提到的可复用。

最终，我们前端的工程目录应该如图4-2所示：

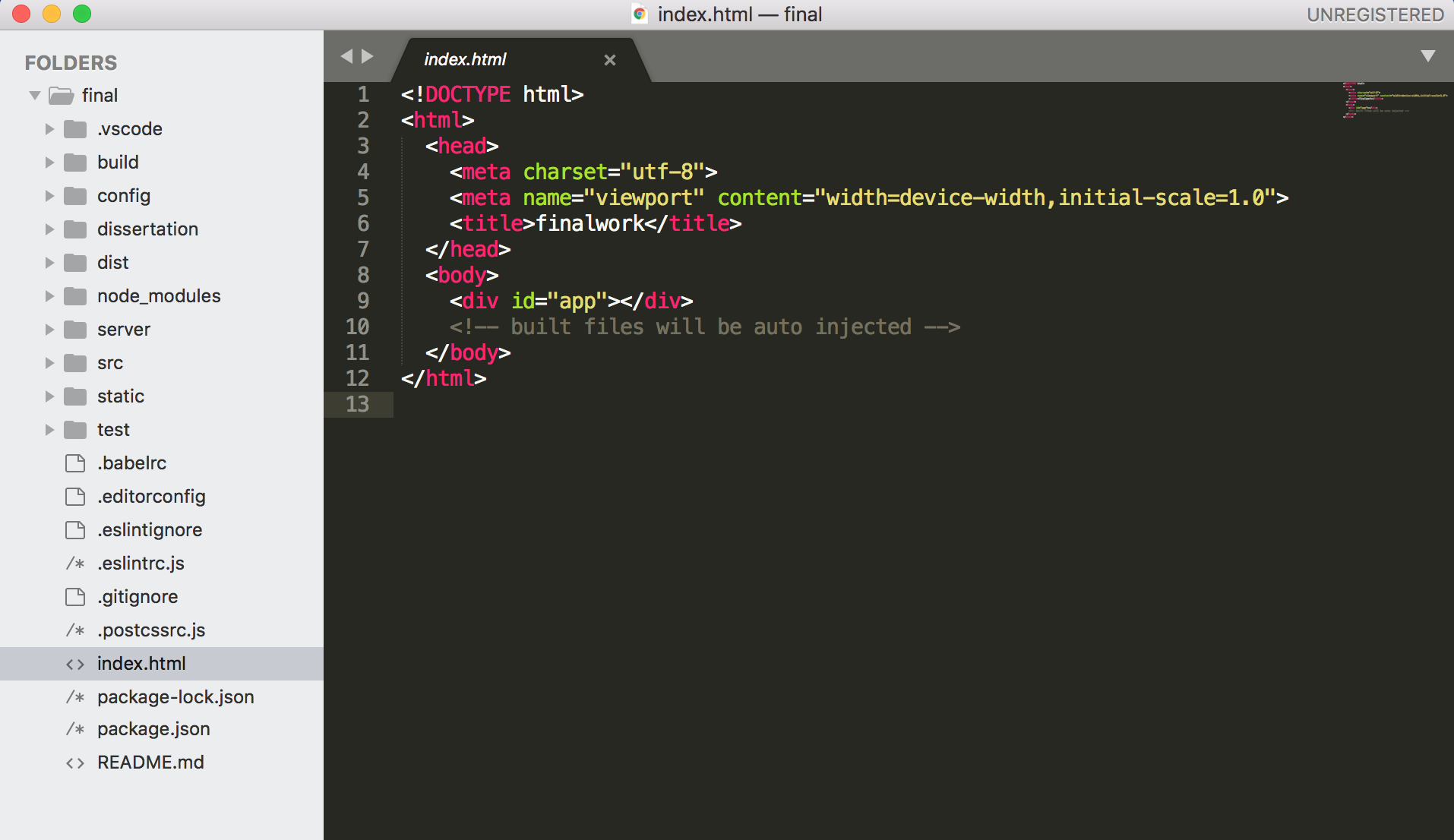


图 ‑2工程目录图

在我们准备开发的时候，我们必定要注重我们的代码开发习惯，很开心，有人帮我们搞定了这个事情，那就是ESLint，ESLint最初是由Nicholas C. Zakas于2013年6月创建的开源项目。它的目标是提供一个插件化的javascript代码检测工具。JavaScript 是一个动态的弱类型语言，在开发中比较容易出错。因为没有编译程序，为了寻找 JavaScript 代码错误通常需要在执行过程中不断调试。像 ESLint 这样的可以让程序员在编码的过程中发现问题而不是在执行的过程中。

ESLint 的初衷是为了让程序员可以创建自己的检测规则。ESLint 的所有规则都被设计成可插入的。ESLint 的默认规则与其他的插件并没有什么区别，规则本身和测试可以依赖于同样的模式。为了便于人们使用，ESLint 内置了一些规则，当然，你可以在使用过程中自定义规则。ESLint 使用 Node.js 编写，这样既可以有一个快速的运行环境的同时也便于安装。当然，本项目中并不需要我们手动安装ESLint，vue-cli已经帮我做好了一切。当我们想要检查我们的代码质量的时候，就打开终端，进入项目目录，执行”npm run dev”，enter，ESLint便开始了他的工作。如图4-3所示。

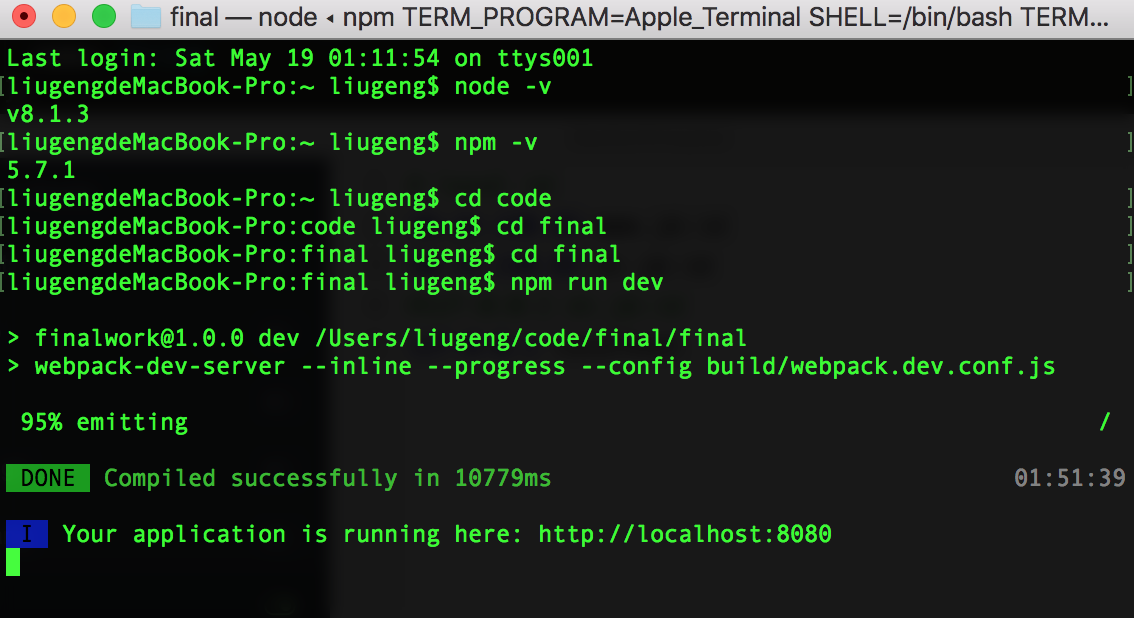


图 ‑3ESLint工作图

至此，我们的前端工程搭建完成，后段环境安装完成，接下来，就进入我们的模块开发环节。

4.2 登陆模块的设计与实现

之前介绍过vue的开发方式，每一个页面都是由组件构成的，也叫做自定义组件