



中山大學  
SUN YAT-SEN UNIVERSITY

# 本科生毕业论文（设计）

## Undergraduate Graduation Thesis (Design)

题目 Title: 基于 Angular 框架的  
产品及资源包管理 web 应用

院系  
School (Department): 数据科学与计算机学院

专业  
Major: 软件工程（移动信息工程）

学生姓名  
Student Name: 刘发斌

学号  
Student No.: 14353196

指导教师（职称）  
Supervisor (Title): 王变琴（高级工程师）

时间：二〇一八年五月四日  
Date: May 4th 2018

## 表一 毕业论文（设计）开题报告

### Form 1: Research Proposal of Graduation Thesis (Design)

论文(设计)题目:

Thesis (Design) Title: 基于 Angular 框架的产品版本及资源包管理 web 应用

(简述选题的目的、思路、方法、相关支持条件及进度安排等)

(Please briefly state the research objective, research methodology, research procedure and research schedule in this part.)

#### 选题目的:

时代在进步, 技术在发展, 我们需要改变自己的观念, 在摸索中不断尝试新的事物。本论文基于企业的产品线控制的需求, 致力于建立一个完善的资源包管理应用, 为产品版本的开发, 测试, 审核, 发布等流程的管理提出的一站式解决方案。技术实现上, 本应用基于目前较成熟的 angular 框架, 实现资源包控制线的一系列功能, 在实践中发掘目前前端技术上的优缺点, 改变前端的思维模式。

#### 思路:

该应用需要包括三方面的功能:

账户管理的功能包括登录, 注册, 注销, 更改个人信息, 基本满足了对于应用使用者的需求  
产品管理的功能包括修改产品信息, 添加某角色(产品开发者/测试员)的成员, 删除某角色的成员, 更换管理员, 新建产品(即提交新产品的信息和新建产品的第一个版本)。版本管理的功能包括新建版本(需要上传新版本的资源包), 版本流程的管理(测试, 审核, 发布), 版本回滚。最后还有成员管理, 这个功能其实包括在产品管理里面, 这里主要说明一下成员和用户的区别。两者之间的区别在于, 一个用户可以是多个成员, 甚至是多个不同产品的相同或者不同的成员, 而每个产品下都有三种成员: 产品管理者, 产品开发者, 产品测试员, 他们拥有不同的权限。

#### 方法:

- 选用合适的前端框架和服务器实现技术
- 搜索相关的资料进行学习
- 设计交互页面以及所需数据
- 设计数据库表格
- 编写接口文档
- 搭建项目，分模块实现功能
- 编写后台，按文档实现接口
- 前后端对接，调试改进

**相关支持条件:**

Windows 系统

安装配置 nodejs

安装 angular

**进度安排:**

根据上面的步骤进行安排:

- 1 月，搜索资料，设计数据库以及页面
- 2 月，搭建前端项目
- 3 月，实现后台接口
- 4 月，前后端对接，调试改进

Student Signature:

Date:

**指导教师意见:**

Comments from Supervisor:

1. 同意开题
  2. 修改后开题
  3. 重新开题
1. Approved( ) 2. Approved after Revision( ) 3. Disapproved( )

Supervisor Signature:

Date:

## 学术诚信声明

本人所呈交的毕业论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。本毕业论文的知识产权归属于培养单位。本人完全意识到本声明的法律结果由本人承担。

本人签名：

日期：

## Statement of Academic Integrity

I hereby acknowledge that the thesis submitted is a product of my own independent research under the supervision of my supervisor, and that all the data, statistics, pictures and materials are reliable and trustworthy, and that all the previous research and sources are appropriately marked in the thesis, and that the intellectual property of the thesis belongs to the school. I am fully aware of the legal effect of this statement.

Student Signature:

Date:

## 基于 Angular 框架的产品版本及资源包管理 web 应用

**[摘 要]** 本文实现了一个基于 Angular 框架的 web 应用，用于管理产品的版本从开发到发布的流程。提供了对产品名下的成员管理，产品版本的资源包上传下载，以及版本的流程管理和操控。该应用由 nodejs 提供后台支持，所有后台接口由 nodejs 实现。前端框架使用了 Angular 5，数据库使用关系型数据库 MySQL，最终实现了所有的功能，通过了体验测试。这是一次对于新的前端思想的体验和实现，说明了如今前端不再仅仅是页面，更可以是一个可以实现复杂功能的，只需要浏览器就可以运行的跨平台 web 应用。

**[关键词]** Angular；Nodejs；前端

## **A web application based on Angular frame to manage packages of products**

**[Abstract]** This article implements a web application based on the Angular framework, which is used to manage the version of the product from development to release process. It provides member management, product version uploading and downloading, version management and manipulation. The application is provided by nodejs with backstage support, and all back-end interfaces are implemented by nodejs. The front frame uses Angular 5, the database uses relational database MySQL, and finally realizes all the functions, through the experience test. This is the experience and implementation of a new front-end idea, which shows that the front end is no longer only a page, but also a cross platform web application that can be run with a browser.

**[Keywords]** Angular; Nodejs; Front-end

## 目录

<b>第一章</b>	<b>引言</b>	<b>2</b>
1.1	选题背景与意义	2
1.2	国内外研究现状和相关工作	2
1.3	应用的需求	3
1.4	本文的论文结构与章节安排	4
<b>第二章</b>	<b>开发工具介绍</b>	<b>5</b>
2.1	开发环境	5
2.2	开发工具 VS code	5
2.3	依赖包管理	5
2.4	框架	6
2.5	后端服务器	7
2.6	数据库	8
<b>第三章</b>	<b>研究方法</b>	<b>10</b>
3.1	应用功能设计	10
3.2	前端架构	11
3.3	界面设计	12
3.4	数据库设计	12
3.5	后端开发	14
<b>第四章</b>	<b>实验与结果</b>	<b>17</b>
4.1	login 和 register 模块	17
4.2	newProduct 和 newVersion 模块	17
4.3	数据展示模块	18
<b>第五章</b>	<b>总结与展望</b>	<b>22</b>
5.1	工作总结	22
5.2	研究展望	22
	<b>参考文献</b>	<b>23</b>

致谢 .....	25
----------	----



# 第一章 引言

## 1.1 选题背景与意义

Web 发展了几十个春秋，风起云涌，千变万化。Web 技术发展的速度让人感觉那几乎不是继承式的迭代，而是一次又一次的变革，一次又一次的创造。所谓前端就是 html+css 吗，可能以前是，但现在已经不再是这样了。随着前端技术的发展以及观念的变化，引领技术潮流的几大巨头也在不断地创新和发展，不断提出新的概念，企图超越别人，扩大影响力。到 2009 年之后，JavaScript 类库已经颇为成熟，各大类库也是相互吸收优点，不断完善并提高自身性能，然而功能上已经没有太多增加的势头。部分框架开始了思想上的转变，更加注重前端开发的组织和结构，条理性强了很多，如 YUI 等。ECMAScript 规范的争执，开启了浏览器引擎大战，各大厂商也趁机瓜分 IE 份额，Chrome 和 Firefox 在这场战役中取得大胜。jQuery 火了，成千上万的 JQ 插件让网页开发变得尤为轻松，而随之而来的也是页面的臃肿和性能调优的深入探索。因此，前端风向渐渐转向工程化和模块化，前端工程化开始普及，各公司开始推出自己的前端集成开发解决方案。Node.js 前后端分离的流行，中间层的出现改变了前后端的合作模式<sup>[1]</sup>。纵观前端技术发展，只从前端应用开发框架上来看，就先后经历了 DOM API、MVC、MVP、MVVM、Virtual DOM、MNV\* 阶段，逐步解决了前端开发效率、设计模式、DOM 交互性能的问题。

## 1.2 国内外研究现状和相关工作

在规范上，前端新标准和草案在不断更新，HTML、CSS、JavaScript 标准也在渐渐完善，经过大版本的更新稳定，目前前端三层结构实现已经形成了 HTML5、CSS3、EcmaScript 6+ 标准规范结合的阶段。2011 年 HTML5 的技术发展和推广都向前迈进了一大步，语义明确的标签体系、简洁明了的富媒体支持、本地数据的储存技术、canvas 等等各类技术被广泛应用。相反，flash 技术开始逐渐凋零。

在思维模式上，前端开发逐渐趋向工程化和模块化，模块化是一种处理复杂系统分解为代码结构更合理，可维护性更高的可管理的模块的方式。理想状态下我们只需要完成自己部分的核心业务逻辑代码，其他方面的依赖可以通过直接加载被人已经写好模块进行使用即可。在前端工程化上，几个派系相互争斗，产出了 AMD、CMD、UMD 等规范，也衍生了 SeaJS、RequireJS 等模块化工具。

在框架上，angular，vue，react 作为当前前端市场的三大主流框架，被绝大多数的开发人员所使用着。angular 是 Google 支持并开发维护的 JavaScript 框架，从 2009 年 AngularJS 诞生开始，Google 公司不断对这个框架于进行改进和升级，2017 年 11 月 1 日 Google 公司发布了 Angular 5，该版本可支持高新能、离线使用、面安装的 app 式体验，并借鉴了来自 Ionic、NativeScript 和 React Native 中的技术与思想，构建原生移动应用，能够横跨 Mac，Windows，Linux 平台，并可以自动生成和拆分代码，提高生产效率。除此之外，angular 还有简单强大了模板语法，快速创建 UI 视图，还有强大的项目构建工具 Angular-CLI 和测试工具 Karma。angular 是一个比较完善的前端 MVW(Model-View-Whatever) 框架，包含模板，数据双向绑定，路由，模块化，服务，依赖注入等所有功能，模板功能强大丰富，并且是声明式的，自带了丰富的 Angular 指令，AngularJS 通过指令扩展了 HTML，且通过表达式绑定数据到 HTML，而且它是以一个 JavaScript 文件形式发布的，可通过 script 标签添加到网页中。

就前端主流技术框架的发展而言，过去的几年里发展极快，在填补原有技术框架空白和不足的同时也渐渐趋于成熟。未来前端在已经趋向成熟的技术方向上面将会慢慢稳定下来，并进入技术迭代优化阶段，例如语言标准、前端框架等。但这并不代表前端领域技术就此稳定了，因为新的技术方向已经出现，并在等待着下一个风口的到来。不管未来如何，就前端应用开发方向来讲，MVVM、Virtual DOM 和同构的技术解决方案依然会延续发展一段时间。对于移动端应用，前端要重点发展的下一步可能就是 MNV 的原生 NativeView 开发，例如使用通用的 MNV 前端技术方案来降低移动端 Native 和前端 Web 交互的开发成本，让前端既可以通过 Native 编译开发出稳定的原生应用外壳，也可用来开发快速迭代、高性能的移动端 MNV\* 应用，最终形成一套移动端上高效率的前端开发生态体系。另一方面，新领域的 Web 化思路也会给前端带来新的技术革新和发展机遇，例如 Web VR (Virtual Reality, 虚拟现实)、物联网 (Physical Web, 顾名思义，就是将物体连入网络的一种理念) Web 化或者网站人工智能等，这些方向的开发者早已跃跃欲试，目前国外也能找到少数这样的应用站点<sup>[2]</sup>。经过近几年的发展，现代前端已经发展到跨端、跨界面的革新阶段，目前主流以基于 MVVM、Virtual DOM、移动端 MNV\* 思路和前后端同构技术进行开发的项目居多，实现的方向也多种多样。

### 1.3 应用的需求

当组织和机构变得庞大的时候，或许许多部门都在做着重复的事情，而这些事情只为自己的部门服务，这样将会导致工作效率的低下，还无法满足产品快速迭代的需求。为此，一个整合所有相同功能的，可扩展的，高效的管理系统是必要的。本论文左中解

决的是当产品需要更新资源包的时候，如众多客户端因为更新而需要替换的资源包，如何对这些资源包进行管理呢？这些资源包属于哪些产品，是哪个版本，这个版本更新了什么，资源包要在哪里下载，如果出错了要怎么控制回滚版本，这些都可以进行整合处理。加入有这样一个管理资源包及其版本的后台管理系统，那么将大大提高工作效率。因此，本论文基于企业的产品线控制的需求，致力于建立一个完善的资源包管理的一站式解决方案。技术实现上，本应用基于目前较成熟的 `angular` 框架，实现资源包控制线的一系列功能，在实践中发掘目前前端技术上的优缺点，改变前端的思维模式。

## 1.4 本文的论文结构与章节安排

本文共分为五章，各章节内容安排如下：

第一章引言。

第二章开发工具介绍。

第三章应用设计与实现。

第四章体验测试。

第五章是本文的最后一章，总结与展望。是对本文内容的整体性总结以及对未来工作的展望。

## 第二章 开发工具介绍

### 2.1 开发环境

操作系统：Windows 10

开发工具：VS code

后端支持：Nodejs

框架支持：Angular 5

数据库：MySQL

### 2.2 开发工具 VS code

VSCode 是微软出的一款轻量级代码编辑器，免费而且功能强大，对 JavaScript 和 NodeJS 的支持非常好，自带很多功能，例如代码格式化，代码智能提示补全、Emmet 插件等。它其实就是一款简单的代码编辑工具，跟 Visual Studio、WebStorm、Eclipse、myEclipse... 这些集成的开发环境并不是一个概念。但是它拥有极其强大的扩展性，只要安装对应的语言插件，就可以对该语言提供强大的支持。因此，选择这个开发工具能极大提高开发效率。

### 2.3 依赖包管理

本应用选择 npm(Node Package Manager) 来对依赖包进行管理，NPM 是随同 NodeJS 一起安装的包管理工具，能解决 NodeJS 代码部署上的很多问题，常见的功能有：

- 允许用户从 NPM 服务器下载别人编写的第三方包到本地使用。
- 允许用户从 NPM 服务器下载并安装别人编写的命令程序到本地使用。
- 允许用户将自己编写的包或命令程序上传到 NPM 服务器供别人使用。

因为本应用需要使用 nodejs 部署服务器，所以无需其他的包管理器，直接可以使用 npm 进行包管理，我们只需要在 package.json 文件上配置所需的依赖，然后 ‘npm install’ 即可安装所需的依赖包

## 2.4 框架

框架 (Framework) 是整个或部分系统的可重用设计, 表现为一组抽象构件及构件实例间交互的方法; 另一种定义认为, 框架是可被应用开发者定制的应用骨架。前者是从应用方面而后者是从目的方面给出的定义。

可以说, 一个框架是一个可复用的设计构件, 它规定了应用的体系结构, 阐明了整个设计、协作构件之间的依赖关系、责任分配和控制流程, 表现为一组抽象类以及其实例之间协作的方法, 它为构件复用提供了上下文 (Context) 关系。因此构件库的大规模重用也需要框架。

构件领域框架方法在很大程度上借鉴了硬件技术发展的成就, 它是构件技术、软件体系结构研究和应用软件开发三者发展结合的产物。在很多情况下, 框架通常以构件库的形式出现, 但构件库只是框架的一个重要部分。框架的关键还在于框架内对象间的交互模式和控制流模式。

框架比构件可定制性强。在某种程度上, 将构件和框架看成两个不同但彼此协作的技术或许更好。框架为构件提供重用的环境, 为构件处理错误、交换数据及激活操作提供了标准的方法。

应用框架的概念也很简单。它并不是包含构件应用程序的小片程序, 而是实现了某应用领域通用完备功能 (除去特殊应用的部分) 的底层服务。使用这种框架的编程人员可以在一个通用功能已经实现的基础上开始具体的系统开发。框架提供了所有应用期望的默认行为的类集合。具体的应用通过重写子类 (该子类属于框架的默认行为) 或组装对象来支持应用专用的行为。

软件系统发展到今天已经很复杂了, 特别是服务器端软件, 设计到的知识, 内容, 问题太多。在某些方面使用别人成熟的框架, 就相当于让别人帮你完成一些基础工作, 你只需要集中精力完成系统的业务逻辑设计。而且框架一般是成熟, 稳健的, 他可以处理系统很多细节问题, 比如, 事物处理, 安全性, 数据流控制等问题。还有框架一般都经过很多人使用, 所以结构很好, 所以扩展性也很好, 而且它是不断升级的, 你可以直接享受别人升级代码带来的好处。



图 2-1 MVVM 框架

本应用使用的是 2017 年 11 月 1 日 Google 公司发布的 Angular 5。angular<sup>[3]</sup> 作为当前最主流的三大前端框架之一，拥有及其完善的功能，而选择它的很重要的一个原因是，相比于 vue 和 react 这种轻量级的框架，它更适用于开发大型的，比较复杂的的企业级应用，它拥有完善的项目搭建脚手架，单元测试工具，这对于工程化，规范化的应用开发是十分有利的。同时，它的可扩展性很强，需要新增什么功能，就新建一个组件，只要将组件导入项目中即可，无需在项目中改动过多的代码当然，随着版本的更迭，angular 也不断吸收 vue，react 等优秀框架的特点，不断改进自己，使得开发人员使用起来更加方便和得心应手。

使用方法：npm install -g @angular/cli

## 2.5 后端服务器

后端服务器使用 express + nodejs 搭建

Node.js<sup>[4]</sup> 是一个基于 Chrome JavaScript 运行时建立的一个平台，简单的说它就是运行在服务端的 JavaScript。Node.js 是一个事件驱动 I/O 服务端 JavaScript 环境，基于 Google 的 V8 引擎，V8 引擎执行 Javascript 的速度非常快，性能非常好。Javascript 是一个事件驱动语言，Node 利用了这个优点，编写出可扩展性高的服务器。Node 采用了一个称为“事件循环(event loop)”的架构，使得编写可扩展性高的服务器变得既容易又安全。提高服务器性能的技巧有多种多样。Node 选择了一种既能提高性能，又能减低开发复杂度的架构。这是一个非常重要的特性。并发编程通常很复杂且布满地雷。Node 绕过了这些，但仍提供很好的性能。

Node 采用一系列“非阻塞”库来支持事件循环的方式。本质上就是为文件系统、数据库之类的资源提供接口。向文件系统发送一个请求时，无需等待硬盘（寻址并检索文件），硬盘准备好的时候非阻塞接口会通知 Node。该模型以可扩展的方式简化了对慢资源的访问，直观，易懂。Node.js 使用 Module 模块去划分不同的功能，以简化应用的开发。Modules 模块有点像 C++ 语言中的类库。每一个 Node.js 的类库都包含了十分丰富的各类函数，比如 http 模块就包含了和 http 功能相关的很多函数，可以帮助开发者很容易地对比如 http,tcp/udp 等进行操作，还可以很容易的创建 http 和 tcp/udp 的服务器。

在几年的时间里，Node.JS 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，此外，开发人员还可以使用它来开发一些快速移动 Web 框架。除了 Web 应用外，NodeJS 也被应用在许多方面，这些项目涉及到应用程序监控、媒体流、远程控制、桌面和移动应用等等。

Express 是一个基于 Node.js 平台的极简、灵活的 web 应用开发框架，它提供一系

列强大的特性，帮助你创建各种 Web 和移动设备应用，使用 Express 可以快速地搭建一个完整功能的网站。

Express 框架核心特性如下：

- 可以设置中间件来响应 HTTP 请求。
- 定义了路由表用于执行不同的 HTTP 请求动作。
- 可以通过向模板传递参数来动态渲染 HTML 页面。

## 2.6 数据库

数据库 (Database) 是按照数据结构来组织、存储和管理数据的建立在计算机存储设备上的仓库<sup>[5]</sup>。简单来说本身可视为电子化的文件柜——存储电子文件的处所，用户可以对文件中的数据进行新增、截取、更新、删除等操作。严格来说，数据库是长期储存在计算机内、有组织的、可共享的数据集合。数据库中的数据指的是以一定的数据模型组织、描述和储存在一起、具有尽可能小的冗余度、较高的数据独立性和易扩展性的特点并可在一定范围内为多个用户共享<sup>[6]</sup>。

数据库有以下几个特点<sup>[7]</sup>：

- 实现数据共享

数据共享包含所有用户可同时存取数据库中的数据，也包括用户可以用各种方式通过接口使用数据库，并提供数据共享。

- 减少数据的冗余度

同文件系统相比，由于数据库实现了数据共享，因而避免了用户各自建立应用文件。从而，减少了大量重复数据，减少了数据冗余，维护了数据一致性。

- 数据的独立性

数据的独立性，包括逻辑独立性和物理独立性。

逻辑独立性，是指数据库中数据的逻辑结构与用户的应用程序是相互独立的。数据的逻辑结构改变了，用户的应用程序可以不变。

物理独立性，是指数据库中数据的物理结构的变化不影响数据的逻辑结构。

- 数据实现集中控制

在文件管理方式中，数据处于一种分散的状态，不同的用户或同一用户在不同处理中其文件之间毫无关系。利用数据库可对数据进行集中控制和管理，并通过数据结构（数据模型）表示各种数据的组织以及数据间的联系。

- 数据一致性和可维护性，以确保数据的安全性和可靠性

- 安全性控制：以防止数据丢失、错误更新和越权使用；

- 完整性控制：保证数据的正确性、有效性和相容性；
- 并发控制：使在同一时间周期内，允许对数据实现多路（多用户）存取，又能防止用户之间的不正常交互作用。

- 故障恢复

由数据库管理系统提供一套方法，可及时发现故障和修复故障，从而防止数据被破坏。数据库系统能尽快恢复数据库系统运行时出现的故障，可能是物理上或是逻辑上的错误。比如对系统的误操作造成的数据错误等。

就目前而言，数据库分为关系型（传统型）数据库和非关系型数据库。当前主流的关系型数据库有 Oracle、DB2、Microsoft SQL Server、Microsoft Access、MySQL 等。非关系型数据库有 NoSql、Cloudant。

两者各有其优缺点，非关系型数据库简单易部署，成本低，查询速度较快，利于数据分割，储存格式是 key,value 形式、文档形式、图片形式等等，所以可以存储基础类型以及对象或者是集合等各种格式<sup>[8]</sup>；而传统的关系型数据库相比于前者，可以做到更加复杂的数据查询，如多表联合查询等，而且其对事务的支持使得对于安全性能很高的数据访问要求得以实现。

由于关系型数据库的各个表之间存在的较强的联系，因此不利于数据的分散，也就是难以将数据分割存储到不同的服务器中，这会导致在数据量太大的时候对服务器的负荷加大。



## 第三章 研究方法

### 3.1 应用功能设计

我们需要对复杂的开发过程进行组织，设计一个产品线系统<sup>[9]</sup>，它是 CMU/SEI 提出的产品开发组织方式，集中体现了软件复用思想。从上面的活动图可以看出，活动流

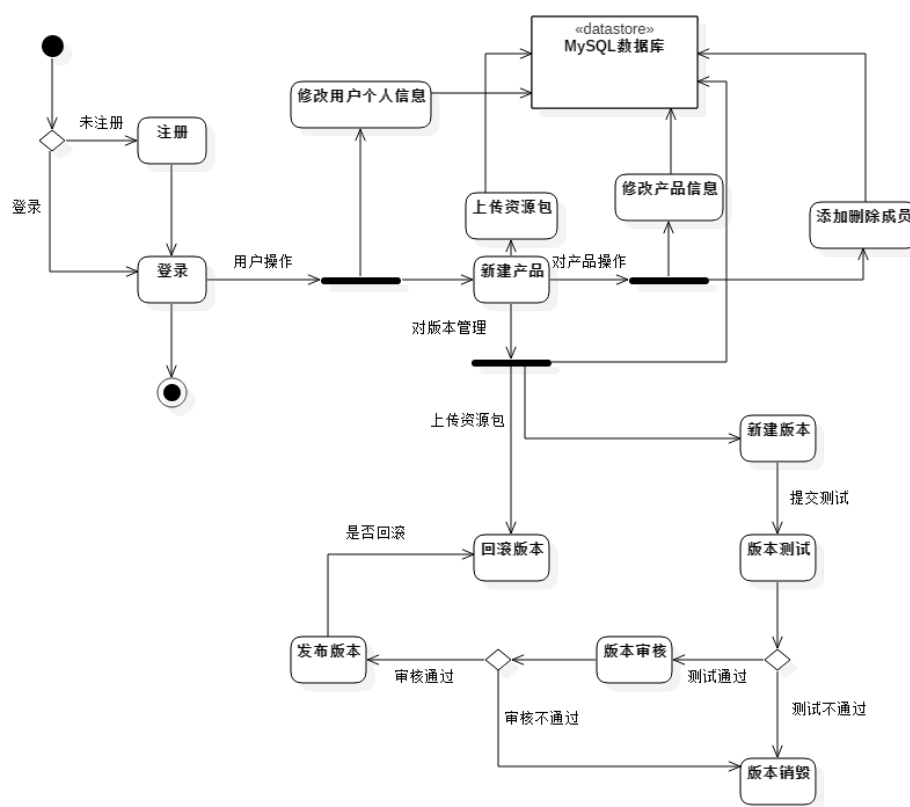


图 3-1 应用功能设计

程包含了三个方面：账户管理、产品管理（包括产品信息的修改和成员的增删）、对产品版本的管理

- 账户管理的功能包括登录，注册，注销，更改个人信息，基本满足了对于应用使用者的需求
- 产品管理的功能包括修改产品信息，添加某角色（产品开发者/测试员）的成员，删除某角色的成员，更换管理员，新建产品（即提交新产品的信息和新建产品的第一个版本）
- 版本管理的功能包括新建版本（需要上传新版本的资源包），版本流程的管理（测

试，审核，发布)，版本回滚

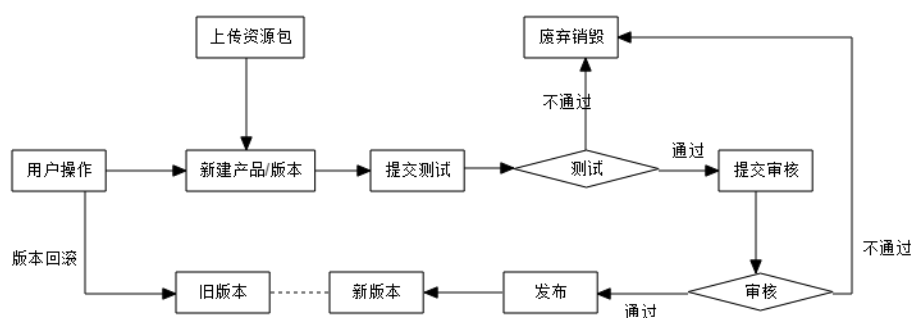


图 3-2 版本管理流程

- 最后还有成员管理，这个功能其实包括在产品管理里面，这里主要说明一下成员和用户的区别。两者之间的区别在于，一个用户可以是多个成员，甚至是多个不同产品的相同或者不同的成员，而每个产品下都有三种成员：产品管理者，产品开发者，产品测试员，他们拥有不同的权限所以成员其实跟用户和产品都有关系，是联系两个模型的重要枢纽

## 3.2 前端架构

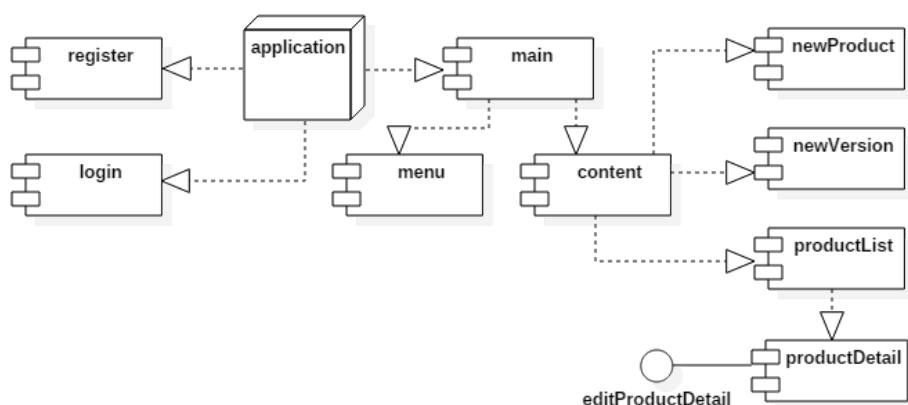


图 3-3 前端架构

angular 的优点是模块化和组件化，将每个页面封装成一个个组件，利用路由功能在不同的组件之间跳跃，实现页面的快速跳转。同时，利用模板语法和数据绑定，使得前后端开发分离，我们只需要从后台获得关键数据，即可在前台的 HTML 模板中填入数据，渲染页面。与传统 web 系统相区别，web 应用能为用户提供丰富的操作，能够随用户操作不断更新视图而不进行 url 跳转。ng 官方也声明它更适用于开发 CRUD 应用，

即数据操作比较多的应用，而非是游戏或图像处理类应用。为了实现这些，ng 引入了一些非常棒的特性，包括模板机制、数据绑定、模块、指令、依赖注入、路由。通过数据与模板的绑定，能够让我们摆脱繁琐的 DOM 操作，而将注意力集中在业务逻辑上。

上图的组件架构可以表示为：

- login
- register
- main
  - menu
  - content
    - \* newProduct
    - \* newVersion
    - \* productList
      - productDetail

### 3.3 界面设计

阿里的 ant design<sup>[10]</sup> 针对 angular 封装了对应的 UI 风格和许多实用的组件供开发者使用，让开发者不必花费过多的时间在 UI 的设计上面，让开发的周期变得更短。

使用方法：npm install ng-zorro-antd

- 本应用的主要界面包括顶部功能栏，左侧边导航栏，以及剩下的内容展示区
- 顶部栏包括搜索输入框，查看系统邮件的入口，新建产品功能按钮，登出按钮
- 左侧导航栏显示了当前登陆用户的信息，以及该用户参与的项目导航列表
- 内容展示区则是界面中剩下的部分，不管是产品列表还是产品成员的展示，或者是新建版本、新建产品的显示都在这块区域
- 其余界面包括登录、注册界面，两个界面均填满整个浏览器窗口

### 3.4 数据库设计

本应用使用 MySQL 数据库建立一系列相互联系的表。之所以如此选择，一是因为本应用的各表之间的联系比较紧密，需要经常在多表之间进行联合查询，二是 MySQL 支持事务功能，对数据进行 CRUD 的时候能保持原子性。下面是数据库中各表的设计信息：

- user 表

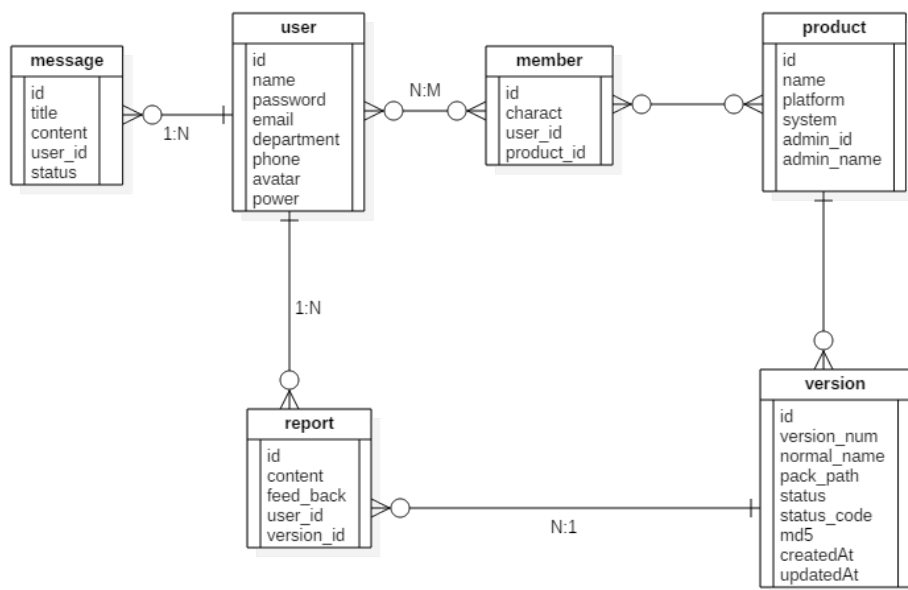


图 3-4 数据库设计

用户表里面包含用户的个人信息，比如姓名，账号密码，手机号，邮箱等，储存着本应用的所有账户的信息

- **product 表**

产品表里面包含产品的所有信息，包括所属平台，操作系统，管理员等。而产品的版本信息由 **version** 表提供，我们将两个表建立 1:N 的关系，表示一个产品可以有多个版本，这样就可以通过 join 等 MySQL 操作获取到对应的 **version** 表的数据。产品的成员信息由 **member** 表提供，实际上，**member** 表是表现出产品表 and 用户表之间的 N:M 关系的连接表，它将两个表的关系记录下来以方便查询，这样我们能很方便地将某个用户添加到某个产品的成员名下。

- **member 表**

如上面所说，该表是表现出产品表和用户表之间的 N:M 关系的连接表，因此它需要包括 **user** 和 **product** 两个表的外键 **user\_id** 和 **product\_id**，同时，它也有自己的属性 **charact** 来表明这个成员是属于哪个权限的，是开发者还是测试员。

- **version 表**

一个产品可以有多个版本，而 **version** 表记录着所有的版本，它包含版本号，当前状态，资源包的存储路径，文件的 md5 值，和 **product** 的外键 **product\_id**。

- **message 表**

这是存储向用户发送的系统提示邮件的数据表，记录着系统提示的 **title**，**content**，接收人的 **id**

- report 表

这是向每个 version 的测试期提供的功能，每个测试员需要对该版本进行测试并填写测试报告，所以 report 表和 version 表其实是 N:1 的关系，每个 version 都有多个 report。同时它还存储这提交人，提交日期，提交内容等数据，以给审核者提供参考，并可以保存审核员的批复意见。

由于前端数据尤其是离线数据可能因为很多的原因出现被更改甚至丢失的情况，因此在程序的设计过程中始终要牢记的一点就是保存在前端缓存中的数据是不可信的<sup>[11]</sup>。不能够根据其中的不可信数据做重要决策，并且将其不经验证直接保存在数据库之中<sup>[12]</sup>。

## 3.5 后端开发

后端服务器采用 express+nodejs 进行搭建。用 express 可以快速搭建一个基于 nodejs 的服务器，我们只需要把精力都放在接口实现上即可。将接口分门别类在对应的不同文件中，利用 express 的路由功能可以精确访问到接口。而要使服务器能连接数据库，我们需要额外的模块，我们可以使用基于 nodejs 的 mysql 模块，但是有一个热门的 orm (Object Relationship Model) 框架 Sequelize<sup>[13]</sup> 对于描述表之间的关系，以及多表查询方面有更加好的解决方案，同时 Sequelize 基于 Promise 实现异步流程控制，能更好地解决 nodejs 中数据库操作的回调地狱问题，能够更方便我们对数据库进行操作，因此我们选用 Sequelize 来对数据库进行操作。接下来对应用的三个主要方面的功能的后台流程实现进行简单介绍。

### 3.5.1 用户管理

账户的操作包括登录，注册，注销，更改个人信息，其工作流程如下：

- 用户注册

注册页面，输入用户信息，包括用户名和密码，提交；后台检索数据库数据，是否存在同一个用户，不存在则创建此用户；返回用户信息，跳转到应用主页。

- 用户登录

登录页面输入用户名和密码；后台检查用户名和密码是否正确；正确则返回信息并跳转到应用主页，而这个时候可以将比较常用的信息如用户名称，手机号等储存在本地储存中，这种存储方式是 Html5 提供的新的存储方式，区别于 session 和 cookie，它可以永久储存数据，而且容量也比较大<sup>[15]</sup>。

### 3.5.2 产品管理

产品管理流程包括修改产品信息，修改（增删）产品成员

- 修改产品信息：提交新的产品信息到后台，由后台更新数据库，需要提供产品 id 作为产品的唯一凭证
- 获取产品的成员列表：在 member 表中按 product\_id 进行搜索，并根据不同的 character 来区分不同的成员列表，将三个列表返回前台
- 添加成员：在用户列表中选择用户，提交用户 id=A 和产品 id=B，并提交成员角色信息（产品开发者 =2，产品测试员 =3）后台在 member 表上添加一条数据，记录提交的三个信息，这条成员信息表示的是 A 用户是 B 产品下的一个某角色的成员，如此就不需要在 user 和 product 表储存冗余的信息
- 删除成员：提交成员 id，在 member 表上根据 id 删除对应的数据
- 新建产品：这个功能其实包括了新建第一个产品版本，只是多了需要提交产品的相关信息这个步骤，因此关于新建第一个版本的实现流程和新建任一版本的基本相同，在下一小节进行阐述

### 3.5.3 版本管理

版本管理流程包括新建版本，版本从开发到发布的系列流程操控，版本回滚控制

- 文件上传功能：新建任何一个版本都避不过上传一个新的资源包的步骤，因此我们需要实现文件上传的功能，而 version 表上的每一条数据都与一个资源包相关。文件上传中间件 multer 模块<sup>[14]</sup> 可以很方便地实现文件上传功能，并保存文件的相关信息，当然，我们需要配合这个功能还需要做很多工作。
- 新建版本：后台从前端接收此版本的资源包，新的版本号，产品的唯一标识 (product\_id)，在 version 表上插入一条新数据。我们需要记录版本号，这是必然的，还需要记录文件的储存路径以便需要的时候可以提供下载，并且需要记录该版本的状态和状态码
- 版本回滚：在 version 表中删除发布的最新版本的信息及其资源包，那么我们搜索到的最新版本信息将是上一个发布版本的
- 流程控制：为了方便地存储版本从开发到发布的系列流程，并对其进行操控，我们需要状态和状态码，而改变版本的流程和状态，只需要改变其状态和状态码即可，当然，状态进入 failed 的需要删除服务器中的资源包，以避免占用多余的空间状态有三种：process 在执行，failed 已撤销，publish 已发布，对应版本可能的

三种状态状态码有 5 中：1-提交测试，10-测试失败，11-提交审核，110-审核失败，111-审核通过并发布，如下图

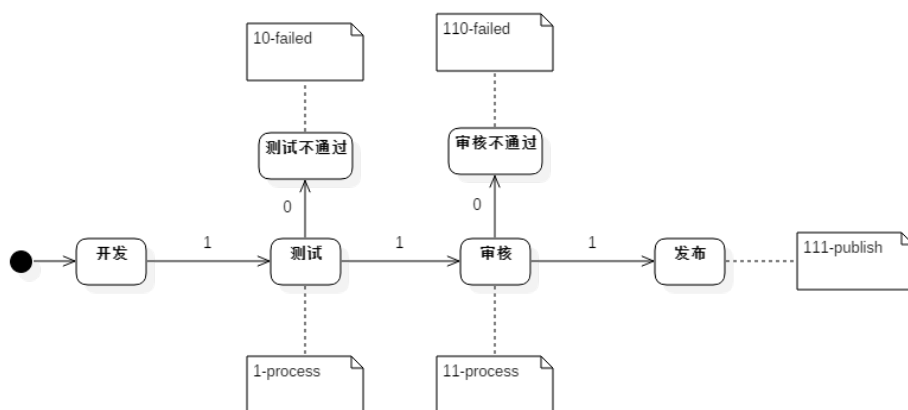


图 3-5 状态图示

## 第四章 实验与结果

### 4.1 login 和 register 模块

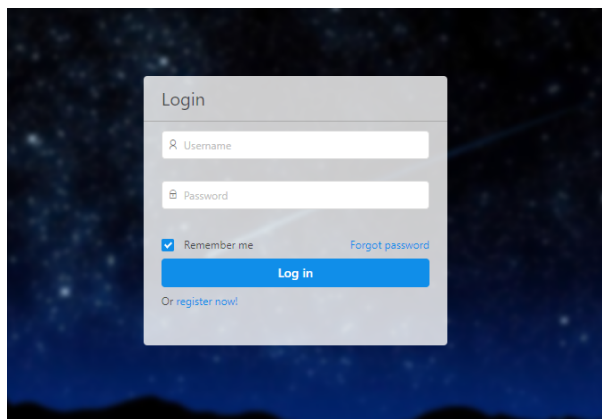


图 4-1 登录模块

\* UserName ? :

\* E-mail:

\* Password:

\* Confirm Password:

\* Phone Number: +86

☐ I have read the [agreement](#)

图 4-2 注册模块

### 4.2 newProduct 和 newVersion 模块



新建产品

\*

产品名称:

please input the product name

\*

项目管理员:

刘发斌

项目开发者:

刘发斌1 ×

刘发斌2 ×

项目测试员:

刘发斌2 ×

刘发斌3 ×

版本:

v1.0.0

\*

平台:

Mobile

PC

H5

\*

操作系统:

Android

IOS

Windows

Mac

Linux

\*

上传资源包:

Click to Upload

Please choose a package!

Submit

图 4-3 新建产品

新建版本

当前版本: 1.1.0

平台: Mobile

操作系统: Android

\*

新建版本:

1

2

3

\*

上传资源包:

Click to Upload

d3d9.rar

Submit

图 4-4 新建版本

## 4.3 数据展示模块

### 4.3.1 侧边栏

如图 4-5

### 4.3.2 产品列表

前端根据用户的 id 查询与用户相关的所有产品列表并返回



图 4-5 侧边栏



图 4-6 产品列表

### 4.3.3 产品详情

产品详情包括：产品信息、成员列表、版本流程、产品管理员的审核页面 系统信息列表显示的是跟用户相关的变动通知



图 4-7 产品详情

项目成员变动	↓ 下载最新版本
您被选为项目【王者荣耀2】开发者	
项目成员变动	↓ 下载最新版本
您被选为项目【王者荣耀2】测试员	
产品申请成功	↓ 下载最新版本
您申请的产品【王者荣耀2】通过审核	
项目成员变动	↓ 下载最新版本
您被选为项目【王者荣耀3】测试员	

图 4-8 系统信息列表

## 第五章 总结与展望

### 5.1 工作总结

本应用使用了基于 Angular 5 搭建了一个实现了资源包控制线的一系列功能的 web 应用，不但包括了前端架构的搭建，也涵盖了后台服务器的支持和数据库的操作，上线部署就能开箱即用。为达到较好的用户体验和高效的开发过程，本项目使用了一些比较成熟的依赖包，不管是 UI 上的 ng-zorro 或是 ORM 的 Sequelize 或是文件上传的 multer，可以说是建立在了前人的努力上。因此，本应用在用户界面上的体验比较好，而且保证了一系列功能的完整性和良好的交互体验，充分利用了前端近年的最新技术是一个很大的优点。但是，对于开发者的我来说，在享受着 angular 框架构建大型项目上的便利性的同时，也面临着接触完全全新的技术的不适应感，angular 是由 typescript 实现的，有一定的学习门槛，对于不熟悉的技术，我经常会遇到各种各样不同的问题，搭建前端时也有，实现后台时也有，这样或那样的问题总会导致我的代码比较凌乱，降低了代码的可维护性，甚至增大了出错的可能性。

### 5.2 研究展望

除了以上的个人技术问题之外，该项目还可以再继续改进，比如如何优化访问速度，如何优化运行的性能，如何保证数据的安全性，如何安全验证用户登录等等。因此，本项目仍有很大的发展空间，比如在界面上进一步优化，在代码上进一步精简，在数据库的操作上更加规范合理。当然，项目本身的可扩展性是很强大的，未来可以添加一个数据分析模块，后台管理员模块，模块化编程使得我们不需要改动太多的代码就可以新增一个功能。

## 参考文献

- [1] 阎王, 近几年前端技术盘点以及 2016 年技术发展方向 [EB/OL],<http://taobaofed.org/blog/2016/01/04/font-end-tech-inventory/>, 2018-4-20.
- [2] ouven, 聊聊未来的前端时代,<https://jixianqianduan.com/frontend-javascript/2016/12/11/future-front-end.html>, 2018-4-20.
- [3] Angular,Angular 中文网 [EB/OL],<https://www.angular.cn/>, 2018-4-20.
- [4] Nodejs,Nodejs 中文文档,<http://nodejs.cn/>, 2018-4-20.
- [5] 徐晖, 冯永兵, 肖传娥, 等. 基于 SQL 的数据存储和查询 [J]. 山西电子技术,2001,29(3):15-17.
- [6] 周屹李艳娟. 数据库原理及开发应用 (第二版): 清华大学出版社, 2013: 8
- [7] u012655815,数据库的几大最主要特点 [EB/OL],<https://blog.csdn.net/u012655815/article/details>
- [8] 王功明, 关永, 赵春江, 等. 面向对象数据库发展和研究 [J]. 计算机应用研究, 2006, 23(3):1-4.
- [9] 杨芙清, 梅宏, 李克勤. 软件复用与软件构件技术 [J]. 电子学报,1999,27(2):68-75.
- [10] Ant Design,Ant Design of Angular NG-ZORRO[EB/OL],<https://ng.ant.design>, 2018-4-20.
- [11] 张坤, 吕义正. HTML5 客户端存储的安全性探析 [J]. 黄山学院学报,2014, (05): 13-15.
- [12] Hanna S, Akhawe D, Saxena P, et al. (In)Secure Usage of New Client Side Primitives[J]. W2sp Web Security & Privacy, 2010.
- [13] Jonas Zhang,Sequelize Docs 中文版 <http://docs.sequelizejs.com/>, 2018-2-23.
- [14] npm,multer,<https://www.npmjs.com/package/multer>, 2018-4-20.

- [15] 冯永亮.HTML5 本地数据存储技术研究 [J]. 西安文理学院学报（自然科学版）,2014, (03): 66-69.

## 致谢

四年时间转眼即逝，青涩而美好的本科生活快告一段落了。回首这段时间，我不仅学习到了很多知识和技能，而且提高了分析和解决问题的能力与养成了一定的科学素养。虽然走过了一些弯路，但更加坚定我后来选择学术研究的道路，实在是获益良多。这一切与老师的教诲和同学们的帮助是分不开的，在此对他们表达诚挚的谢意。

首先要感谢的是我的指导老师王变琴老师。我作为一名本科生，缺少学术研究经验，不能很好地弄清所研究问题的重点、难点和热点，也很难分析自己的工作所能够达到的层次。王老师对整个研究领域有很好的理解，以其渊博的知识和敏锐的洞察力给了我非常有帮助的方向性指导。他严谨的治学态度与辛勤的工作方式也是我学习的榜样，在此向王老师致以崇高的敬意和衷心的感谢。

最后我要感谢我的家人，正是他们的无私的奉献和支持，我才有了不断拼搏的信息的勇气，才能取得现在的成果。

刘发斌

2018 年 5 月 4 日



毕业论文 (设计) 成绩评定记录

**Grading Sheet of the Graduation Thesis (Design)**

<p>指导教师评语</p> <p>Comments of Supervisor:</p> <p>论文主要工作是基于 Angular 5 框架设计和实现了一个资源包管理 web 应用，包括了前端架构的搭建，后台服务器的支持和数据库的操作等功能，并进行了体验测试，达到预期。论文达到综合能力训练要求。</p> <p>成绩评定</p> <p>Grade:</p> <p>指导教师签名</p> <p>Supervisor Signature: <span style="float: right;">Date:</span></p>		
<p>答辩小组或专业负责人意见</p> <p>Comments of the Defense Committee:</p>     <p>成绩评定</p> <p>Grade:</p> <p>签名: <span style="float: right;">Date:</span></p> <p>Signatures of Committee Members</p>		
<p>院系负责人意见</p> <p>Comments of the Academic Chief of School:</p>     <p>成绩评定</p> <p>Grade:</p> <p>签名 <span style="margin-left: 100px;">院系盖章</span></p> <p>Signature: <span style="margin-left: 100px;">Stamp:</span> <span style="float: right;">Date:</span></p>		