**编号**

JNDX_M

**本科生毕业设计（论文）**

**题目：** 基于智能匹配策略的高校内推网设计

物联网工程 学院 计算机科学与技术 专业

学 号 0304110107

学生姓名 高 帅

指导教师 蒋 敏 副教授

二〇一五年六月

# 设计总说明

本课题主要研究基于智能匹配策略的高校内推网系统的设计.旨在以优秀的职位匹配算法建立一个完善的应届生内推网系统，在节省求职者和内推者双方时间的同时解决公司和应届毕业生之间的双向需求问题.

本系统以学长或者学姐在本系统上发布招聘信息，学弟或者学妹投递简历，学长挑选最为满意的简历内推到公司指定部门，之后通知面试为主线.并着重于以组合推荐算法，主动为学弟智能匹配推荐较为符合要求的职位信息，并为推荐的职位进行匹配度的客观公正且符合逻辑的评分，以节省查找职位的宝贵时间.以精准而又直观的方式提升应聘双方的效率，从而更好的使本系统成为连接应届毕业生与企业之间的桥梁.

目前国内有很多的内推网站，且知名度很高，比如内推网，拉勾网等一系列内推系统，但这些内推网站基本都是针对于社会招聘，而社会招聘的流程基本趋于成熟，从目前的社会招聘来看，应聘人是直接面向需求部门的，针对性较强，应聘成功率较高，整体发展趋势趋于平稳上升期，未来的招聘模式必定以针对性招聘为主.另外对于推荐算法这部分国内外都有了很多的研究，其中较为知名的是基于内容的推荐，基于协同过滤的推荐，基于关联规则的推荐，基于效用的推荐，基于知识的推荐以及较为常用的组合推荐.而本系统最主要的就是使用到了组合推荐，日趋成熟的推荐算法也在随着领域的不同而不断的改进推荐模型.现阶段又结合大数据，完全性跟踪分析人机交互习惯，记录搜索关键词等突出手段来达到较高的推荐满意度.

虽然对于社会招聘的内部推荐已经相当成熟且竞争激烈，但对于应届毕业生这块却很少涉及，而经过在企业实习总结，企业中较为看中由本团队推荐过来的人才.目前校园招聘的流程主要为统一笔试，统一几轮面试，终面，谈offer.往往在笔试过程中刷掉了很多偏科的单方面领域的人才，而在面试环节又不是本应聘部门面试，不能够进行针对性面试，被刷的几率非常非常大.但内部推荐直接推荐应聘者到用人单位的指定岗位，省去笔试的大关卡，面试由本部门的人进行针对性面试，使企业招到更符合要求的人才，未来的招聘必定朝更有针对性，匹配度的方向发展，而本系统以坚持打通应聘者和内推人之间的流程为原则，基于目前社会招聘内推系统的成熟性来设计本系统.

完成本系统使用了目前较为流行的一些技术，如：webx3.0、Spring、Ibatis和组合推荐.Webx是阿里巴巴的一项开源技术，相对稳定且效率很高，设计过程中主要参考了webx框架指南的PDF资料.Spring是目前较为成熟的框架，其控制翻转和依赖注入以及切面在企业中已经使用的炉火纯青，而学习Spring中较为实用的材料主要为《Spring就是这么简单》入门书籍和《Spring源码深度解析》这种中端书籍.Ibatis是这些年较为流行的一款半自动化数据库映射框架，它和hibernate各有优点和缺点，但在本应用中使用Ibatis可以结合Ibatis小巧精悍且更加直观的操作sql语句的优点，更加轻量级的完成本系统与数据库的交互.学习Ibatis的参考资料为《ibatis in action》和中译版《ibatis实战》.另外组合推荐在个性化推荐和电子商务目前较为火爆，这里运用到计算求职者个人与职位的匹配度来进行推荐，设计过程中查看了很多个性化推荐的博客，总结了简历推荐的优秀算法，从而设计了本系统的组合推荐匹配算法.

关键词： 高校；内推；智能；组合推荐；

ABSTRACT

The main research is a system based on intelligent matching strategy to help undergraduates find good job by internal referral. We are aiming to set up a perfect internal recommended system on an outstanding algorithm with job match. This system can time for both save job seeker and pusher and build the two-way between the companies and the graduates at the same time.

Seniors can publish recruitment information on this system. Then undergraduates send their resumes. Seniors choose to push the most satisfactory resume to designated departments and notice interview. This is the main line of the system. We focus on a combination of recommendation algorithm which can recommend matching jobs for the undergraduates based on objective and logical matching score between the job requirements and the skill of the undergraduates and save valuable time. Our system will becomes the bridge of graduates and enterprises and enhance the efficiency of recruit procedure.

Currently there are a lot of internal referral sites with high visibility, such as www.neitui.me and www.lagou.com. But these internal referral sites are basically targeted at social recruitment, and the social recruitment process is basically mature. From the current point of view of social recruitment, candidates who are directly facing the demand sector, have high success rate. In the future, targeted recruitment will become an important way. There have been a lot of researches in the field of referral algorithm, such as content based recommendation, collaborative filtering based recommendation, association rules based recommended, utility based recommendation. Combined with big data, tracking and analysis interactive habit, recording keyword and searching keywords, recommendation algorithms achieve a higher satisfaction in the present.

While social recruitment of internal referral is already quite mature and highly competitive, but the recruitment of undergraduates are rarely involved. After the internship, enterprises are more fancy with the one who recommended by the own team. Currently the main campus recruitment process includes written examination, rounds of interviews, the final interview, offer discussion. Lots of talents with unbalanced skills are shut out. Moreover, the interview are not carried out by the candidates department and could be targetless. Thus the probability of push down is very big, even for those real talents. But internal referral directly recommends the suitable candidates to the employing department, eliminates the need for written test. Interviews are conducted by the department's targeted interviews and enable enterprises to hire people who meet the requirements well. Future hiring must be more targeted, matching degree of development. This system adhere to open up the flow of candidates and principles.

We use some of the more popular technology, such as: webx3.0, Spring, ibatis and combinations recommendation. Webx is an open source technology of Alibaba, it is relatively stable and high efficiency. Spring is a mature framework. It’s features like control flip and dependency injection and section is reached a pinnacle in the use of enterprise. Ibatis is more popular in these years, it is a semi-automated database mapping framework, it has advantages and disadvantages comparing with hibernate. However, in this application, Ibatis can combine the advantages of small and more intuitive operation of sql statement. It is more lightweight in interaction between the system and the database. Also the combination recommendation is more popular in Personalized Recommendation system and e-commerce. The system calculates the degree of matching of job seekers to job Recommended based on this algorithm. Inspired by a lot of personalized recommendation blog during the design process, I summarize those outstanding algorithm for resume, and then design this recommendation system for undergraduates upon combination recommended matching algorithm.

Keyword: Undergraduates; Recommendation; Intelligent; Recommended combination

目录

[第1章 绪论 1](#_Toc421536684)

[1.1 引言 1](#_Toc421536685)

[1.2 一个实用的高校内推系统 1](#_Toc421536686)

[1.3 实现本系统所需的资源 2](#_Toc421536687)

[第2章 高校内推系统概述 5](#_Toc421536688)

[2.1 内推系统的发展 5](#_Toc421536689)

[2.1.1 国内内推系统的诞生 5](#_Toc421536690)

[2.1.2 国际内推系统的发展情况 5](#_Toc421536691)

[2.1.3 中国内推系统的发展情况 5](#_Toc421536692)

[2.2 本课题的设计目标 6](#_Toc421536693)

[第3章 基于智能匹配的高校内推系统程序设计 7](#_Toc421536694)

[3.1 bootstrap概述 7](#_Toc421536695)

[3.1.1 什么是bootstrap 7](#_Toc421536696)

[3.1.2 booststrp在本系统中的应用 7](#_Toc421536697)

[3.2 Webx3.0的总体概述 7](#_Toc421536698)

[3.2.1 什么是webx 7](#_Toc421536699)

[3.2.2 webx的设计理念 7](#_Toc421536700)

[3.2.3 webx的层次 8](#_Toc421536701)

[3.2.4 本系统为什么要用Webx 9](#_Toc421536702)

[3.2.5 webx的优势 9](#_Toc421536703)

[3.3 服务层框架Spring 10](#_Toc421536704)

[3.3.1 Spring的框架特征 10](#_Toc421536705)

[3.3.2 Spring的七个模块 11](#_Toc421536706)

[3.3.3 Spring于本系统的重要性 13](#_Toc421536707)

[3.4 数据层的Ibatis 14](#_Toc421536708)

[3.4.1 Ibatis的设计理念 14](#_Toc421536709)

[3.4.2 本系统为什么要使用Ibatis 16](#_Toc421536710)

[ 简单 16](#_Toc421536711)

[第4章 智能组合推荐模块的研究 19](#_Toc421536712)

[4.1 智能推荐的简介 19](#_Toc421536713)

[4.2 智能推荐系统的发展历程 19](#_Toc421536714)

[4.3 智能推荐系统的基本框架 20](#_Toc421536715)

[4.3.1 基于内容的推荐 20](#_Toc421536716)

[4.3.2 基于协同过滤的推荐 21](#_Toc421536717)

[4.3.3 基于组合的推荐 21](#_Toc421536718)

[4.4 智能推荐系统的作用 22](#_Toc421536719)

[4.5 本系统的职位与简历匹配算法 23](#_Toc421536720)

[4.5.1 职位与简历的属性抽取 23](#_Toc421536721)

[4.5.2 职位权重系数设置方法： 23](#_Toc421536722)

[4.5.3 职位权重计算和有关属性权重设定 24](#_Toc421536723)

[4.5.4 匹配算法公式 26](#_Toc421536724)

[第5章 基于智能匹配策略的高校内推系统的设计 27](#_Toc421536725)

[5.1 界面设计 27](#_Toc421536726)

[5.1.1 首页设计 27](#_Toc421536727)

[5.1.2 个人信息界面设计 27](#_Toc421536728)

[5.2 实现用户注册 29](#_Toc421536729)

[5.3 实现用户登录 30](#_Toc421536730)

[5.4 实现职位浏览与投递 31](#_Toc421536731)

[5.5 实现发布内推 34](#_Toc421536732)

[5.6 实现简历编辑 36](#_Toc421536733)

[5.7 实现智能推荐 37](#_Toc421536734)

[5.8 实现接收简历 40](#_Toc421536735)

[第6章 小结 43](#_Toc421536736)

[6.1 系统功能概述 43](#_Toc421536737)

[6.2 系统存在的不足 43](#_Toc421536738)

[6.3 有待改进之处 43](#_Toc421536739)

[参考文献 45](#_Toc421536740)

[致谢 47](#_Toc421536741)

# 绪论

## 引言

基于智能匹配策略的高校内推系统是针对应届毕业生的找工作的内推网站.主要研究如何通过本系统使好的企业招到好的应届毕业生人才，好应届毕业生进入到好的企业.

目前的社会调研中，纵观应届生找工作的流程，一些企业直接与应届生交互的内推网呈现出一种巨大的需求.根据中国教育在线的统计，2014年全国高校毕业生规模创历史新高，达到727万人，其中江苏省高校毕业生近54.5万人，2014年成为“更难就业季”，而2015年高校毕业人数同比2014年又增加22万人，对应届毕业生提出了更大的就业挑战，而目前全国高校毕业生的就业率在70%上下波动，也就是有近225万人每年未能找到工作或面临失业.其中互联网行业占的比重较重，而在未来的几年，中国的高校毕业生人数，将以每年25万人左右的速度增长，就业压力也会随之加大，各大应届生求职网上的职位信息铺天盖地，应届生们往往对于简历都是浅目的的海投状态，目前社招的内推招聘形式，让求职人直接和公司联系起来，优化了很多环节，入职率更高，而应届生的内推在公司也是备受推荐的，各大优秀的公司为争夺人才不断推出新政策，学长内推学弟入职可拿丰厚的奖金，从企业和求职者之间行成一个良性的需求环，由此显示出未来企业和应届生将普遍使用内推网的趋势.

着眼于企业和应届生之间的良好交互，将是高校内推网在未来服务于企业和应届生的核心所在.因此，研究实现一个较为实用的高校内推系统就显得尤其的有意义.

## 一个实用的高校内推系统

根据本课题的研究需要，我以阿里巴巴这家公司的实习生内推流程为例，阿里巴巴公司每年的2到3月份是实习生招聘的内推时间，在3月中旬内推结束，进入后续环节.内推流程如下图1-1，而传统的求职招聘流程还是以求职者为主动因素，在招聘网站或者宣讲会上确定钟意职位，投递简历，等待通知.主动求职的同时却是被动获得信息，信息不能够对称，导致求职效率较低.具体流程图如下图1-2：



图 1-1 传统招聘流程图 图 1-2 内推招聘流程图

由于各大公司为了抢夺人才资源，实习生和应届生内推都会进行的比较早，而往往公司里已经就职的学长和大多数高校的毕业生与学弟们并没有直接的熟悉度，也并不知道学弟的技术能力如何，从而使双方都缺失内推机会，本系统专注于已入职学长和学弟之间的交互问题，以学校为基本点推广，增加同校学长和学弟的交互程度，从而达到双方内推成功的目的.

## 实现本系统所需的资源

本系统的实现主要借助于JetBrains公司所提供的集成编译器IDEA.IDEA 全称IntelliJ IDEA，是java语言开发的集成环境，IntelliJ在业界被公认为最好的java开发工具之一，尤其在智能代码助手、代码自动提示、重构、J2EE支持、Maven、JUnit、CVS整合、代码审查、 创新的GUI设计等方面的功能可以说是超常的.JetBrains这家公司总部位于捷克共和国的首都布拉格，开发人员以严谨著称的东欧程序员为主.目前IDEA也是深受国内开发者喜爱的编译器之一，用户群也大大超过了Eclipse，它在代码编写的智能程度令人惊叹.

下图1-3是IDEA14.0.3的集成编译环境和智能提示：



图1-3 Idea集成编译环境

下图1-4时IDEA所提供的调试界面：

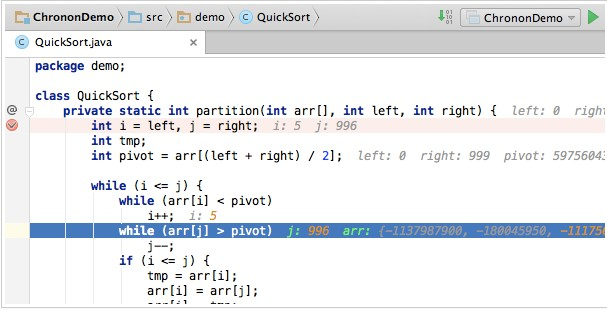


图1-4 Idea的调试环境

Idea的调试会在单行语句的后面显示出字段值，调试更加直观.

# 

# 高校内推系统概述

## 内推系统的发展

### 国内内推系统的诞生

目前国内的大型内推网有两家：

一家是内推网（http://www.neitui.me） 2013年4月28日上线第一个版本，这家网站创始人是黄小亮，曾在阿里巴巴和盛大网络工作，而内推网的灵感也是来自在阿里巴巴工作的时候感受到了阿里巴巴内部推荐的强大，另外认识到了传统招聘渠道的一些问题，所以决定做一个纯中介的招聘平台，理想的方式下是由团队主管或者架构师直接和应聘者对接，创办内推网起初只有两个人，一个人兼职做技术、产品、运营和测试等，而另外一个人全职做技术开发和维护等.

另外一家是拉勾招聘（http://www.lagou.com） 2013年7月20日上线，上线的最初是走的国际板的LinkedIn模式，希望能够做出一版“中国版的LinkedIn”但是上线三个月后效果并不理想，拉勾网最终决定回到招聘本身，做职位内推，也就是在对接双方需求的基础上，提供更具长远价值的服务.

在互联网的大潮下，一切都是可能被颠覆的，内推网的诞生就在一步一步颠覆传统的招聘行业.

### 国际内推系统的发展情况

内推在美国已经是一种较为流行的求职方式，其中google、apple、facebook、amazon和twitter这些知名互联网公司不仅倾向于向内推的应聘者提供橄榄枝，而且在应聘者入职成功后还会给推荐者一笔丰厚的奖金.目前较为出色的职业社交网站有LinkedIn和Classdoor. LinkedIn创建于2002年，2003年5月5日正式上线，目前已经成为全球最大的职业社交网站，致力于向全球职场人士提供沟通平台，并协助他们事半功倍，发挥所长.LinkedIn会员人数在世界范围内已经超过3亿，Linked有一个很大的优点，就是推荐信，求职者可以邀请朋友，教授，或者是公司职员帮助写推荐信，这会极大的增加档案的真实性，使技能与工作经历更受认可.并且可以帮助求职者扩展人脉网络，更利于公司的内部推荐.

### 中国内推系统的发展情况

如今中国的互联网大潮愈演愈烈，传统招聘网站在不断的被颠覆，很多时候传统的招聘网站都是HR发布招聘信息，应聘者投递简历，HR再挑出认为不错的简历录入到待面试系统，之后是部门之间选择简历之后再进行面试，在铺天盖地的海投中，自身的简历很难被直接重视，而内推网就是绕过了猎头和HR的服务理念，目前的发展状态下内推网截止到目前每天有9万+的人来这个网站寻找机会，有3万+的知名互联网名企入住了内推网.而拉勾网相对于内推网的发展速度来说不相上下，仅仅在2014年6月底拉勾网上传的简历数量达到了25万左右，而总的投递量也接近了300万份.

未来的招聘系统中将更加直接的通过内推系统使招聘部门和应聘者直接对接，略过hr这一步，招聘效率更快.而以后的招聘将会成为企业和应聘者之间的一种服务.

## 本课题的设计目标

本课题致力于设计一款针对高校应届生的内推网站，就目前而言，高校每年的应届生会进入到很多公司企业，但是进入到企业后和高校的联系就基本断了，学长对于母校、应届生和公司来说都是一笔丰厚的资源，公司希望在职员工可以推荐其优秀的学弟来公司实习继而留用，而学弟又倾向于有学长进行好公司的推荐.母校方面更在注重应届生的优秀就业率.本系统就是要把学长和学弟通过这个系统联系起来.通过学长的发布职位或者学弟的求职简历来建立联系，从而使学长对学弟的能力进行熟悉，并直接内推到公司部门，省去了在传统招聘系统中海投简历和不断跑路参加宣讲会的时间与精力.

# 基于智能匹配的高校内推系统程序设计

## bootstrap概述

### 什么是bootstrap

Bootstrap来自Twitter，是目前最受欢迎的前端框架，底层是基于 HTML、CSS、Javascript 的，它是一个CSS/HTML框架，并且提供了优雅的HTML和CSS规范，是由动态CSS语言less编写的，在jQuery的基础上进行了更为个性化和人性化的完善，形成了一套自己独有的风格，并且兼容大部分的JQuery插件，它的出现为所有开发者、所有场景提供了较为方便, 简洁灵活，更加快捷的开发模式.目前在很多网站运用颇为广泛，成为了Github上热门的开源项目，目前很多公司都包含NASA的MSNBC（微软全国广播公司）的breakoing news项目都使用了该框架.

### booststrp在本系统中的应用

本系统的基本样式使用的都是bootstrap，包括导航栏置顶：导航栏永远停留在页头部分并在页头固定，输入框：并非简单的输入框，输入框具有记忆功能，能够记住你上一次的输入信息，然后双击输入框就会弹出你上次输入的信息并进行选择，页面按钮可以自定义按钮的大小并设定按钮的颜色，按钮形状相对来说比较圆滑更加人性化.另外页面的布局是响应式的布局，也就是当页面处于不同的宽度或不同的长度的时候页面会呈现不一样的界面，这就是响应式布局.

## Webx3.0的总体概述

### 什么是webx

Webx是一套基于Java Servlet API的通用Web框架.它在Alibaba集团内部被广泛使用.

* 2001年，阿里巴巴内部开始使用Java Servlet作为WEB服务器端的技术，以取代原先的Apache HTTPD server和mod\_perl的组合.
* 2002年，选择Jakarta Turbine作为WEB框架，并开始在此之上进行扩展.
* 2003年，经过大约一年的扩展，框架开始成熟.开始逐渐被称为Webx 1.0.
* 2004年，借着淘宝网的第一次改版，正式推出了Webx 2.0.由于Turbine开源项目发展过于缓慢.Webx 2.0是从零开始完全重写的，仅管它仍然延续了Turbine的使用风格.
* 2004年11月，Webx 2.0和Spring框架整合.从那以后，Webx 2.0一直在进化，但没有作根本性的改动.
* 2010年，Webx 3.0发布.Webx 3.0抛弃了Webx 2.0中过时的、从Turbine中发展而来的Service框架，直接采用Spring作为其基础，并对Spring作了重大改进.Webx 3.0完全兼容Webx 2.0的代码，只需要修改配置文件就可完成升级.
* 2010年底，Webx 3.0向社会开源.

### webx的设计理念

Webx的设计理念有两点，一点是框架化，一点是层次化.应用框架确定了整个应用的结构，而且框架允许我们在不改变整体结构的基础上扩展功能，目前一些做的好的开源框架，都有一个共性 —— 它们并不是简单地实现Web应用所需要的功能（诸如Action、模板、表单验证等），而是把框架建立在另一个基础框架之上.这个基础框架的作用是：组装模块；提供扩展机制.建立在这种基础上的Web框架有很好的适应性和扩展性，可以应对Web应用不断变化和发展的需求.早期的Turbine是建立在Service基础上的，Webwork是建立在Xwork之上的，而如今的Spring MVC是建立在spring框架之上的.一个Web框架的好坏，往往不是由它所实现的具体功能的好坏决定的，而是由其所用的基础框架的好坏决定的.Webx建立在SpringExt的基础上 —— SpringExt是对Spring的扩展.Spring是当今主流的轻量级框架.SpringExt没有损失任何Spring的功能，但它能够提供比Spring自身更强大的扩展能力.

Webx的设计是层次化的，这也正好突出了模块的良好设计.

如图3-1模块B扩展了模块A，同时被模块C扩展.这样就形成了A、B、C三个层次.

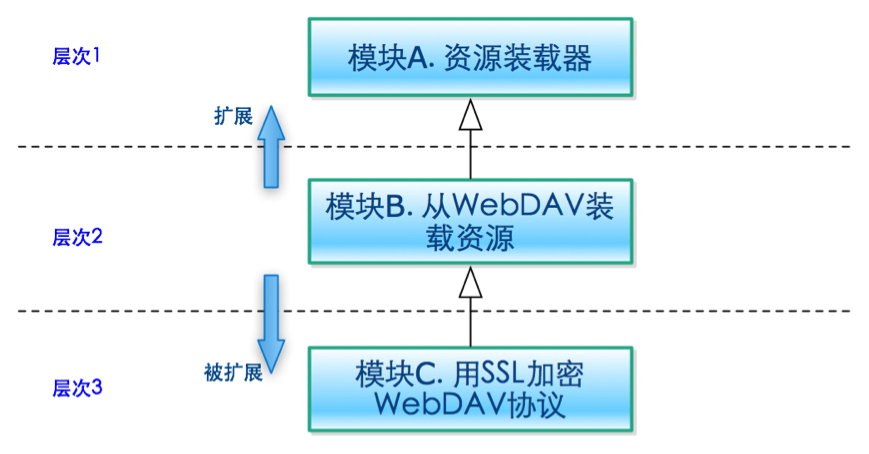


图3-1 Webx的层次化模块

图中层次之间有如下的关系：

上层定义规则，下层定义细节；（上层、下层也可称为内层、外层）

上层是抽象的，下层是具体的；

越上层，越稳定（越少改变）；越下层，越易变.

依赖倒转（Dependency Inversion）.下层（具体）依赖上层（抽象），而不是上层依赖下层.

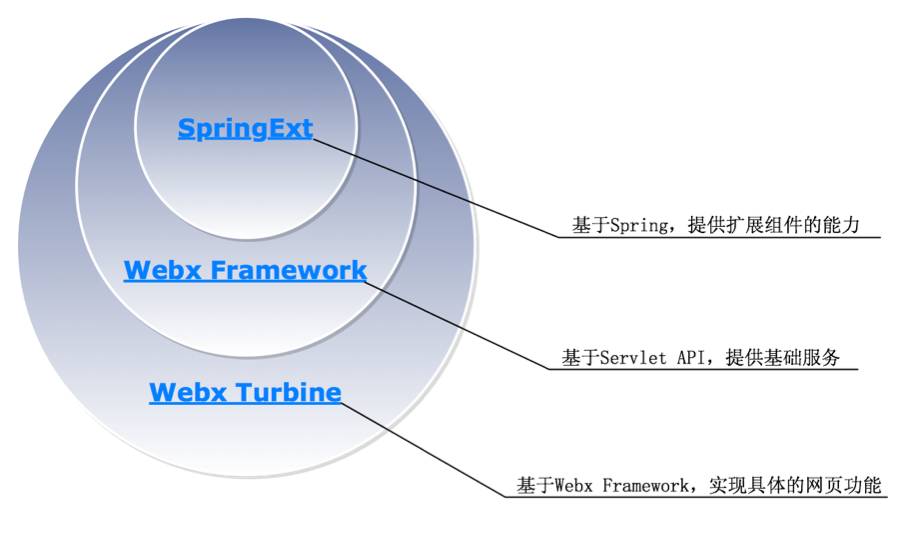
下层扩展上层时，不需要修改到上层的任何代码和配置.即符合开闭原则.层次化的设计，使软件中的每一个部分都可被增强或替换.

层次化是精心的设计而来的.设计这个层次化的组件，主要考虑了切分功能.每个组件专心做一件事.分析哪些会改变，哪些不会改变.不变部分固化在组件中，可能会改变的部分抽象成接口，以便扩展.考虑默认值和默认扩展.默认值和默认扩展应该是最安全、最常用的选择.对于默认值和默认扩展，用户在使用时不需要额外的配置.

Webx鼓励层次化的模块设计，而SpringExt提供了创建和配置层次化组件的机制.

### webx的层次

Webx的层次在于Webx如何处理页面、如何验证表单、如何渲染模板等等功能.事实上，这些是Webx最外层、非本质的功能.Webx框架不仅鼓励层次化设计，它本身也是层次化的.我既可以使用全部的Webx框架，也可以只使用部分的Webx框架.大体上，Webx框架可以划分成三个大层次，如图3-2所示.



**图3-2 Webx的三大层次**

1.SpringExt：基于Spring，提供扩展组件的能力.它是整个框架的基础.

2.Webx Framework：基于Servlet API，提供基础的服务，例如：初始化Spring、初始化日志、接收请求、错误处理、开发模式等.Webx Framework只和servlet及spring相，它不关心Web框架中常见的一些服务，例如Action处理、表单处理、模板渲染等.因此，事实上，我可以用Webx Framework来创建多种风格的Web框架.

3.Webx Turbine：基于Webx Framework，实现具体的网页功能，例如：Action处理、表单处理、模板渲染等.

### ****本系统为什么要用Webx****

**从性能上讲Webx是一款很成熟的技术，经受住了天猫双十一的压力考验，并且在阿里巴巴多个系统的调用中具有很强的解耦性，对性能是一个很大的提升；**

**从结构上讲webx的框架很牢固，是一款层次化很强，很清晰的框架，它以“组件”为单位，易于后期系统的扩展，并且比没有扩展的spring更易使用，有因为层次化很清晰而足够开发者扩展.**

**从适合度上来讲，在阿里巴巴中，几乎每一个系统都在使用webx，webx在阿里巴巴的系统中是无处不在的，对于这款技术的上手速度也是较快的，而且在公司内部资源的基础下，后期会有更好的维护性.**

**从代码操作性上来讲，webx相信约定胜于配置，配置文件过多往往影响代码的解耦性，webx在约定方面做得很好，相同的文件夹名直接对应相同的java逻辑处理类和页面展示模板，从而省去很多的配置.**

### ****webx的优势****

Webx是一种成熟可靠的技术，针对于阿里巴巴及属下网站而言，webx在阿里巴巴和淘宝网用了多年，目前也经受了多次双十一的高峰冲击，webx对于超大访问量的电子商务网站有很强的耐考验性.

另外webx在开放和扩展性方面也是很简介的，它对spring提供几乎无缝的支持，spring是当今主流的轻量级框架，webx3.0和springMVC一样，完全建立在spring框架之上，可以运用spring的所有特性，同时webx也对spring做了一些扩展，使spring bean不再是bean，而是升级为“组件”，一个组件可以扩展另外一个组件，也可以被其他组件扩展，这种机制造就了webx的非常好的扩展性，且比未经扩展的spring更易使用，其次webx被设计成多个层次，层次之间分界线很清晰，每个层次都足够开发和进行扩展，我们可以使用全部的webx也可以使用webx的任何一个层次.

## 服务层框架Spring

### ****Spring的框架特征****

Spring是一个开源框架，它由Rod johnson创建.它是为了解决企业应用开发的复杂性而创建的.Spring使用基本的JavaBean来完成以前只可能由EJB(sun的JavaEE服务器端组件模型)完成的事情.然而，Spring的用途不仅限于服务器端的开发.从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中得到不菲的好处.

Spring是一个轻量级的控制反转(IoC)和面向切面(AOP)的容器框架.它具有以下几个优点：

轻量——从大小与开销两方面而言Spring都是轻量的.完整的Spring框架可以在一个大小只有1MB多的JAR文件里发布.并且Spring所需的处理开销也是微不足道的.此外，Spring是非侵入式的：Spring应用中的对象不依赖于Spring的特定类.

控制反转——Spring通过一种称作控制反转（IoC）的技术促进了松耦合.当应用了控制翻转(IoC)，一个对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象.你可以认为IoC与JNDI(java命名与目录接口)相反，不是对象从容器中查找依赖，而是容器在对象初始化时不等对象请求就主动将依赖传递给它.

面向切面——Spring提供了面向切面编程的丰富支持，允许通过分离应用的业务逻辑与系统级服务进行内聚性的开发.应用对象只实现它们应该做的——完成业务逻辑——仅此而已.它们并不负责其它的系统级关注点，例如本系统中的日志或者是对于事务的支持.

容器——Spring包含并管理应用对象的配置和生命周期，在这个意义上它是一种容器，你可以配置你的每个bean如何被创建——基于一个可配置原型你的bean可以创建一个单独的实例或者每次需要时都生成一个新的实例，以及它们是如何相互关联的.然而，Spring与传统的重量级EJB容器具有很大的不同，EJB容器经常是非常庞大与笨重的，并且在应用中难以使用.

框架——Spring可以将简单的组件配置、组合成为复杂的应用.在Spring中，应用对象被声明式地组合，典型地是在一个XML文件里.Spring也提供了很多基础功能（事务管理、持久化框架集成等），将应用逻辑的开发留给了我们自身来处理.

所有Spring的这些特征能够使我们编写更干净、更可管理、并且更易于测试的代码.它们也为Spring中的各种模块提供了基础支持.

Spring适用于任何java应用,并提供了更多的轻量级的应用功能.目的是解决轻应用开发的复杂性.

### ****Spring的七个模块****

Spring框架由七个定义明确的模块组成，如图3-3

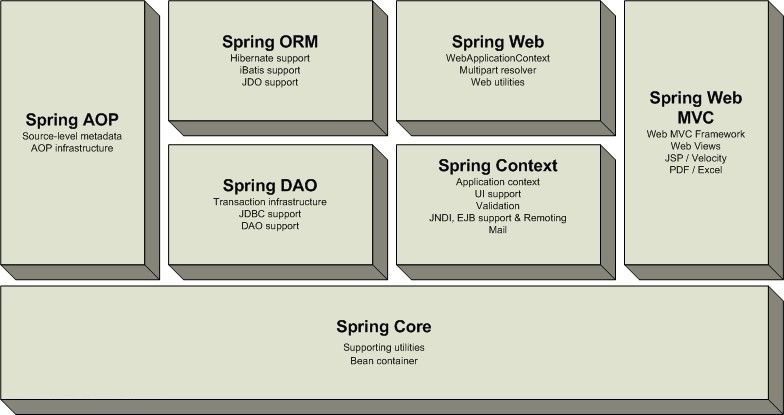


图3-3 Spring框架概览图

如果作为一个整体，这些模块为我们提供了开发企业应用所需的一切.但我们不必将应用完全基于Spring框架.一般开发者我们可以挑选适合自己应用的模块来进行使用.

如图上所示，所有的Spring模块都是在核心容器之上构建的.容器定义了Bean是如何创建、配置和管理的.当我们在开发一些应用时，我们会不知不觉的使用这些类.下面分别介绍一下Spring的七大模块.

* **核心容器**

这是Spring框架最基础的部分，它提供了依赖注入（DependencyInjection）特征来实现容器对Bean的管理.这里最基本的概念是BeanFactory，它是任何Spring应用的核心.BeanFactory是工厂模式的一个实现，它使用IoC将应用配置和依赖说明从实际的应用代码中分离出来.

* **应用上下文（Context）模块**

核心模块的BeanFactory使Spring成为一个容器，而上下文模块使它成为一个框架.这个模块扩展了BeanFactory的概念，增加了消息、事件传播以及验证的支持.另外，这个模块提供了许多企业服务，例如电子邮件、JNDI访问、EJB集成、远程以及时序调度（scheduling）服务.也包括了对模版框架例如Velocity和FreeMarker集成的支持.

* **Spring的AOP模块**

Spring在它的AOP模块中提供了对面向切面编程的丰富支持.这个模块是在Spring应用中实现切面编程的基础.为了确保Spring与其它AOP框架的互用性，Spring的AOP支持基于AOP联盟定义的API.AOP联盟是一个开源项目，它的目标是通过定义一组共同的接口和组件来促进AOP的使用以及不同的AOP实现之间的互用性.

* **JDBC抽象和DAO模块**

使用JDBC经常导致大量的重复代码，取得连接、创建语句、处理结果集，然后关闭连接.Spring的JDBC和DAO模块抽取了这些重复代码，因此我们可以保持数据库访问代码干净简洁，并且可以防止因关闭数据库资源失败而引起的问题.这个模块还在几种数据库服务器给出的错误消息之上建立了一个有意义的异常层.使我们不用再查找SQL错误消息的原因，异常报错消息非常明显！另外，这个模块还使用了Spring的AOP模块为Spring应用中的对象提供了事务管理服务.

* **对象/关系映射集成模块**

Spring提供了ORM模块.蛋挞并不试图实现自己的ORM解决方案，而是为几种流行的ORM框架提供了集成方案，包括Hibernate、JDO和iBATIS SQL映射（3.4节会有说明）.Spring的事务管理支持这些ORM框架中的每一个也包括JDBC.

* **Spring的Web模块**

Web上下文模块建立于应用上下文模块之上，提供了一个适合于Web应用的上下文.另外，这个模块还提供了一些面向服务支持.例如：实现文件上传的请求，它也提供了Spring和其它Web框架的集成，比如Struts、WebWork.

* **Spring的MVC框架**

Spring为构建Web应用提供了一个功能全面的MVC框架.虽然Spring可以很容易地与其它MVC框架集成，例如Struts，但Spring的MVC框架使用IoC对控制逻辑和业务对象提供了完全的分离.它也允许我们声明性地将请求参数绑定到自身的业务对象中.

* **Spring框架Web页面乱码问题**

在做java Web 项目时，乱码问题时常都会出现，解决方法也不尽相同，有简单也有复杂的；如果加入了Spring框架之后就不一样了，可以采用Spring框架自带的过滤器CharacterEncodingFilter，这样可以大大减轻了处理乱码的工作量，非常简单方便而且又容易理解，配置方式如下：在web.xml文件中filter的位置加上如下内容：

<filter>

<filter-name>encodingFilter</ filter-name >

<filter-class>

org.springframework.web.filter.CharacterEncodingFilter

</filter-class>

<init-param>

<param-name>encoding</param-name>

<param-value>UTF-8</param-value>

</init-param>

<init-param>

<param-name>forceEncoding</param-name>

<param-value>true</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>encodingFilter</filter-name>

<url-pattern>\*</url-pattern>

</filter-mapping>

### ****Spring于本系统的重要性****

**本系统使用的webx是基于Spring的，最主要的用到的Spring3.0的注解功能和MVC结构以及切面.**

注解配置可以充分利用 Java 的反射机制获取类结构信息，这些信息可以有效减少了很多工程的配置工作.如使用 JPA 注释配置 ORM 映射时，我们就不需要指定 PO 的属性名、类型等信息.而且如果关系表字段和 PO 属性名、类型都一致，甚至都不需要编写任务属性映射信息，因为这些信息都可以通过 Java 反射机制获取.另外注解和 Java 代码位于一个文件中，而 XML 配置采用独立的配置文件，大多数配置信息在程序开发完成后都不会调整，如果配置信息和 Java 代码放在一起，有助于增强程序的内聚性.而采用独立的 XML 配置文件，开发者在编写一个功能时，往往需要在程序文件和配置文件中不停切换，这种思维上的不断跳跃会严重降低开发效率.因此在本系统中选择使用注解配置.而注释配置在目前的使用率中有进一步流行的趋势.Spring 2.5 的一大增强就是引入了很多注释类，使得开发者已经可以使用注释配置完成大部分 XML 配置的功能.下面我将说明使用注释进行 Bean 定义和依赖注入的内容.

Spring 2.5 引入了@Autowired，@Resource注释，它可以对类成员变量、方法及构造函数进行标注，完成自动装配的工作.来看一下使用@Autowired进行成员变量自动注入的代码：

@Autowired  
private UserManager userManager;

如果想要对这个对象进行注入，还需要在xml中配置一段扫描的配置，如：

*<!--扫描类包以应用注解定义bean-->*

<context:component-scan base-package="com.jiangnan.\*"></context:component-scan>

在xml配置了这个标签后，spring可以自动去扫描base-pack下面或者子包下面的java文件，如果扫描到有@Component @Controller@Service等这些注解的类，则把这些类注册为bean.

可以注册为bean的类在类头都会加上@Component @Controller@Service其中之一个字段，如：

@Component  
public interface UserManager {}

@Component @Controller@Service这三个字段分别代表不同的注册含义这里不再多讲.

**这样配置完后，在启动spring容器的时候，spring会自动扫描base-back下面或者子包下面的java文件.然后把对应的类注册为bean，当这个对象被调用的时候，就会自动向这些对象注入bean，实现方式特别简单灵活.**

## [****数据层的Ibatis****](#_Toc137881443)

### [****Ibatis的设计理念****](#_Toc137881444)

iBATIS是一个混合式的解决方案（hybrid solution），借鉴了多种操作关系数据库的方法的理念.

iBATIS是一种data mapper.也就是一个映射层，在对象和数据库间传递数据，并保持两者与映射层本身相独立.

Martin很好地区分了数据映射（Data Mapping）和元数据映射（Metadata Mapping），后者正是O/RM工具的依据，这种工具将数据库的表和列映射到应用程序中的类和字段（field），也就是说它将数据库的元数据映射到类的元数据.图3-4显示了类和数据库表的O/R映射的情形.在这种情况下，类的每个字段映射到了表中的一个相应的列.

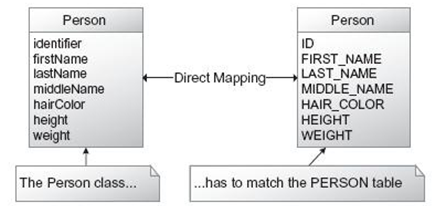


图3-4 O/RM的全自动数据映射

iBATIS则与之不同，它不是直接在类与数据表或字段与列之间进行关联，而是把SQL语句的参数（parameter）和返回结果（result）映射至类.iBATIS是处于类和数据表之间的一个中间层，这使得它在类和数据表之间进行映射时更加灵活，而不需要数据库模型或对象模型（object model）的任何修改.这个中间层实际上就是SQL，它使得iBATIS能够更好地分离数据库和对象模型的设计，这样就相对减少了两者间的耦合.图3-5说明了iBATIS如何使用SQL映射数据.

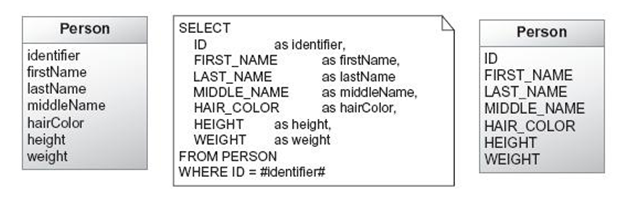


图3-5 Ibatis的sql映射

从图3-5可以看到，iBATIS的映射层正是SQL.开发者只需要关心SQL，iBATIS会为您处理类的属性和数据表的列之间的映射.同时也为了消除与其它各种映射方式引起的混淆，通常称这种Data Mapper为SQL Mapper.

SQL映射是ibatis的一个重要设计理念，任何SQL语句都可看作是一组输入和输出.输入的值是参数（parameter），通常出现在WHERE子句中.输出的值则是出现在SELECT子句中的列.图3-6 Ibatis输入和输出 描述了这种观点.

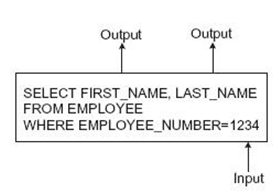


图3-6 Ibatis输入和输出

这种方式的优势在于SQL语句给开发人员带来了很大的灵活性.我们可以轻松地操作数据使之与对象模型匹配而无需修改后台的数据库设计.此外，开发人员可以使用内置的数据库函数或存储过程来返回多个不同的表或结果，SQL的强大能力变得信手拈来.

iBATIS使用一个简单的XML描述文件来映射SQL语句的输入和输出.图3-7显示了一个描述文件的示例.

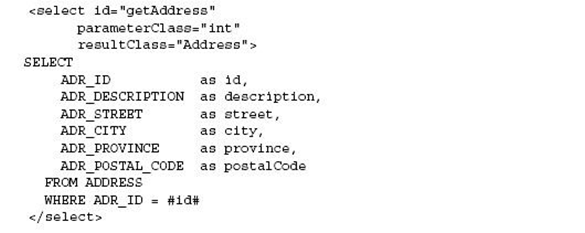


图3-7 ibatis描述文件示例

此处可以看到一条SELECT SQL语句，它返回的是地址信息.根据<select>元素我们可以了解它接受一个整型数作为参数，也就是WHERE子句中标记为#id#的部分.我们也可了解结果为Address类型的一个实例，这里假定Address类包含了与SELECT子句中每一列的别名名称相同的属性.例如，别名为id的列将会映射至Address类的id属性.这就是映射一条接受整型参数、返回Address对象的SQL语句所需的全部了.执行这条语句使用的Java代码是：

Address address = (Address) sqlMap.queryForObject("getAddress", new Integer(5));

我们的SQL映射方式可移植性很强，可以应用到任何特性完整的编程语言.

### [****本系统为什么要使用Ibatis****](#_Toc137881445)

本系统使用ibatis作为系统和底层数据库之间的半自动化映射主要考虑到ibatis的如下优点：

### 简单

iBATIS是当前公认的最简单的持久层框架之一.简单是iBATIS团队设计目标的核心，其重要性几乎要超过其它任何方面.它的简单是通过它构建的基础来达到的,iBATIS对于数据库管理员和SQL开发人员来说也是简单的.几乎任何拥有SQL编程经验的开发人员都很容易理解iBATIS的配置文件.

* **生产力**

一个好框架要考虑的第一要旨是使开发人员更具生产力.通常框架会处理一些通用任务，减少重复性的编码，解决复杂的架构问题.

* **性能**

事实上，在较低层次上来看，任何框架都会存在或多或少的性能损失.不过在当今应用程序开发中，更为重要的是，如何从数据库中获取数据，何时获取它，以及获取的频率.例如，从数据库中获取分页后的列表数据能显著地提升应用程序的性能，因为这样就避免了一次加载过多的数据.类似的，使用像延迟加载（lazy load）这样的特性可以避免在给定的用例下加载不必要的数据.另一方面，如果我们确定需要加载复杂的对象属性，而这些属性来自于多个数据表，那么使用单条SQL来加载数据也可以极大地改善性能.iBATIS提供了多种性能优化策略，通常要以简单的方式来配置iBATIS，但它的性能会像JDBC一样好，甚至更好.另一个需要考虑的地方是，并不是所有的JDBC代码都能编写得很好.JDBC是一个很直接面对数据库的API，要想正确使用需要注意很多地方，而且需要重复写很多代码.导致其性能甚至不如iBATIS.

* **分离关注点**

在典型的JDBC代码中，有时会看到数据库资源如连接、结果集散布在程序各个层中.在一些糟糕的程序中，我们会看到数据库连接、语句出现在表现层.这是最糟糕的设计.目前的程序开发，应用程序分层的是很重要的.我们看到了应用程序如何在较高层次上分层，持久层是如何处于中间层次的.iBATIS提供了此种分层的支持，它会管理所有数据持久相关的资源，如数据库连接，语句和结果集等.它提供了数据库无关的接口和API，帮助应用程序中的其它层能够与任何数据持久相关的资源保持独立.使用iBATIS，我们面对的是真正的对象，而不是任意的结果集.iBATIS使保持良好的分层变成一件容易的事.

* **分工**

DBA（数据库管理员）们一般都很重视他们负责的数据库，多数不会让其它人编写SQL语句.还有的人很擅长编写SQL，其他人都希望由他来写SQL.在此处，因为SQL语句和应用程序代码分离得非常清楚，SQL开发人员可以按其固有的方式进行开发，而不用担心什么字符串的拼接.所以即使现在我来同时开发java代码和SQL，后期如果需要优化数据库性能,可以让DBA直接来改的SQL而不用关心应用的问题.如果使用JDBC就没这么简单了，因为SQL往往包含在一连串的字符串拼接中，或者是由遍历和条件动态动态生成.使用O/RM会更糟糕，我们必须运行程序，然后在日志中输出语句，即使找到了，也不能做任何事情.iBATIS使得开发者和DBA都可以自由地开发、查看、修改SQL语句.

* **可移植性**

iBATIS的可移植性是很强的.这得益于其相对简单的设计，它可以实现于几乎任何语言和平台上.iBATIS支持三种最流行的开发平台：Java，Ruby和C#.

目前对于不同的平台都能够很好的兼容，更重要的是，其*概念*和*方式都*是可移植性很强的.这样我们所有应用程序的设计可以保持一致.对于语言和应用程序的类型来说，iBATIS比任何其它框架支持得都多.在本程序中一致性非常重要，所以iBATIS是很好的选择.

* **开源和可信度**

事实上，iBATIS是免费的，开源的软件.开源软件最大的优势之一是可信且免费.开源代码可以直接查看源码的设计，以及对于后续的优化.还包含在遇到问题时如何快速的定位排查，另外对于本设计不需要付费，也不会涉嫌使用盗版，免去不少的麻烦.

# 智能组合推荐模块的研究

## 智能推荐的简介

互联网的出现和普及给用户带来了大量的信息，随着互联网在生活中的不断渗透，互联网信息对生活的狂风乱炸，各种信息数量和种类层不穷，虽然网络满足了用户在信息时代对信息的需求，但随着网络的迅速发展而带来的网上信息量的大幅增长，使得用户在面对大量信息时无法从中获得对自己真正有用的那部分信息，在浏览网页或者购买物品时会浪费大量的时间在一些不关紧要的页面上，从而不能精确定位到用户真正在意的问题，导致浪费大量的时候，又无法起到很好的反馈效果.对信息的使用效率反而降低了，这就是所谓的信息超载（information overload）问题.

解决信息超载问题一个非常有潜力的办法是推荐系统，它是根据用户的信息需求、兴趣等，将用户感兴趣的信息、产品等推荐给用户的个性化信息推荐系统.和搜索引擎相比推荐系统通过研究用户的兴趣偏好，进行个性化计算，由系统发现用户的兴趣点，从而引导用户发现自己的信息需求.一个好的推荐系统不仅能为用户提供个性化的服务，还能和用户之间建立密切关系，让用户对推荐产生依赖.智能推荐的特点就是根据用户的兴趣特点和行为，向用户推荐用户感兴趣的信息和商品，无论在任何场景下用户都会接触到智能推荐，智能推荐又叫个性化推荐，目前在电子商务方面用的比较多，比如淘宝而言，每个人进入淘宝首页看到的商品信息是不一样的，个性化推荐系统是建立在海量数据挖掘基础上的一种高级商务智能平台，以帮助电子商务网站为其顾客购物提供完全个性化的决策支持和信息服务.

推荐系统现已广泛应用于很多领域，其中最典型并具有良好的发展和应用前景的领域就是电子商务领域.同时学术界对推荐系统的研究热度一直很高，逐步形成了一门独立的学科.

## 智能推荐系统的发展历程

1995年3月，卡耐基.梅隆大学的RobertArmstrong等人在美国人工智能协会上提出了个性化导航系统Web Watcher;斯坦福大学的 MarkoBalabanovic等人在同一会议上推出了个性化推荐系统LIRA；

1995年8月，麻省理工学院的Henry Lieberman在国际人工智能联合大会（IJCAI）上提出了个性化导航智能体Letizia；

1996年， Yahoo 推出了个性化入口My Yahoo；

1997年，AT&T实验室提出了基于协作过滤的个性化推荐系统PHOAKS和Referral Web;

1999年，德国Dresden技术大学的Tanja Joerding实现了个性化电子商务原形系统TELLIM；

2000年，NEC研究院的Kurt等人为搜索引擎CiteSeer等加了个性化推荐功能；

2001年，纽约大学的Gediminas Adoavicius和Alexander Tuzhilin实现了个性化电子商务网站的用户建模系统1：1Pro；

2001年，IBM公司在其电子商务平台Websphere中增加了个性化功能，以便商家开发个性化电子商务网站.

2003年，Google开创了AdWards盈利模式，通过用户搜索的关键词来提供相关的广告.AdWords的点击率很高，是Google广告收入的主要来源.2007年3月开始，Google为AdWords添加了个性化元素.不仅仅关注单次搜索的关键词，而是对用户一段时间内的搜索历史进行记录和分析，据此了解用户的喜好和需求，更为精确地呈现相关的广告内容.

2007年，雅虎推出了SmartAds广告方案.雅虎掌握了海量的用户信息，如用户的性别、年龄、收入水平、地理位置以及生活方式等，再加上对用户搜索、浏览行为的记录，使得雅虎可以为用户呈现个性化的横幅广告.

2009年，Overstock（美国著名的网上零售商）开始运用ChoiceStream公司制作的个性化横幅广告方案，在一些高流量的网站上投放产品广告. Overstock在运行这项个性化横幅广告的初期就取得了惊人的成果，公司称：“广告的点击率是以前的两倍，伴随而来的销售增长也高达20%至30%.”

2009年7月，国内首个推荐系统科研团队北京半分点信息科技有限公司成立，该团队专注于推荐引擎技术与解决方案，在其推荐引擎技术与数据平台上汇集了国内外百余家知名电子商务网站与资讯类网站，并通过这些B2C网站每天为数以千万计的消费者提供实时智能的商品推荐.

2011年9月，百度世界大会2011上，李彦宏将推荐引擎与云计算、搜索引擎并列为未来互联网重要战略规划以及发展方向.百度新首页将逐步实现个性化，智能地推荐出用户喜欢的网站和经常使用的APP.

## 智能推荐系统的基本框架

推荐系统有3个重要的模块：用户建模模块、推荐对象建模模块、推荐算法模块.推荐系统把用户模型中兴趣需求信息和推荐对象模型中的特征信息匹配，同时使用相应的推荐算法进行计算筛选，找到用户可能感兴趣的推荐对象，然后推荐给用户.本系统主要运用了以下三个方面的算法

### ****基于内容的推荐****

基于内容的推荐（Content-based Recommendation）是信息过滤技术的延续与发展，它是建立在项目的内容信息上作出推荐的，而不需要依据用户对项目的评价意见，更多地需要用机器学习的方法从关于内容的特征描述的事例中得到用户的兴趣资料.在基于内容的推荐系统中，项目或对象是通过相关的特征的属性来定义，系统基于用户评价对象 的特征，学习用户的兴趣，考察用户资料与待预测项目的相匹配程度.用户的资料模型取决于所用学习方法，常用的有决策树、神经网络和基于向量的表示方法等.基于内容的用户资料是需要有用户的历史数据，用户资料模型可能随着用户的偏好改变而发生变化.

基于内容推荐方法的优点是：

* 不需要其它用户的数据只要有个人部分信息就足够了.
* 能为具有特殊兴趣爱好的用户进行推荐.
* 能推荐新的或不是很流行的项目，没有新项目问题.
* 通过列出推荐项目的内容特征，可以解释为什么推荐那些项目.
* 已有比较好的技术，如关于分类学习方面的技术已相当成熟.

缺点是要求内容能容易抽取成有意义的特征，要求特征内容有良好的结构性，并且用户的口味必须能够用内容特征形式来表达，不能显式地得到其它用户的判断情况.

### ****基于协同过滤的推荐****

协同过滤推荐 （Collaborative Filtering Recommendation）技术是推荐系统中应用最早和最为成功的技术之一.它一般采用最近邻技术，利用用户的历史喜好信息计算用户之间的距离，然后利用目标用户的最近邻居用户对商品评价的加权评价值来预测目标用户对特定商品的喜好程度，系统从而根据这一喜好程度来对目标用户进行推荐.协同过滤最大优点是对推荐对象没有特殊的要求，能处理非结构化的复杂对象，如音乐、电影.

协同过滤是基于这样的假设：为一用户找到他真正感兴趣的内容的好方法是首先找到与此用户有相似兴趣的其他用户，然后将他们感兴趣的内容推荐给此用户.其基本思想非常易于理解，在日常生活中，我们往往会利用好朋友的推荐来进行一些选择.协同过滤正是把这一思想运用到电子商务推荐系统中来，基于其他用户对某一内容的评价来向目标用户进行推荐.

基于协同过滤的推荐系统可以说是从用户的角度来进行相应推荐的，而且是自动的即用户获得的推荐是系统从购买模式或浏览行为等隐式获得的，不需要用户努力地找到适合自己兴趣的推荐信息，如填写一些调查表格等.

和基于内容的过滤方法相比，协同过滤具有如下的优点：

* 能够过滤难以进行机器自动内容分析的信息，如艺术品，音乐等.
* 共享其他人的经验，避免了内容分析的不完全和不精确，并且能够基于一些复杂的，难以表述的概念（如信息质量、个人品味）进行过滤.
* 有推荐新信息的能力.可以发现内容上完全不相似的信息，用户对推荐信息的内容事先是预料不到的.这也是协同过滤和基于内容的过滤一个较大的差别，基于内容的过滤推荐很多都是用户本来就熟悉的内容，而协同过滤可以发现用户潜在的但自己尚未发现的兴趣偏好.
* 能够有效的使用其他相似用户的反馈信息，较少用户的反馈量，加快个性化学习的速度.

虽然协同过滤作为一种典型的推荐技术有其相当的应用，但协同过滤仍有许多的问题需要解决.

### 基于组合的推荐

由于各种推荐方法都有优缺点，所以在实际中，组合推荐（Hybrid Recommendation）经常被采用.研究和应用最多的是内容推荐和协同过滤推荐的组合.最简单的做法就是分别用基于内容的方法和协同过滤推荐方法 去产生一个推荐预测结果，然后用某方法组合其结果.尽管从理论上有很多种推荐组合方法，但在某一具体问题中并不见得都有效，组合推荐一个最重要原则就是通 过组合后要能避免或弥补各自推荐技术的弱点.

在组合方式上，有研究人员提出了七种组合思路：

* 加权（Weight）：加权多种推荐技术结果.
* 变换（Switch）：根据问题背景和实际情况或要求决定变换采用不同的推荐技术.
* 混合（Mixed）：同时采用多种推荐技术给出多种推荐结果为用户提供参考.
* 特征组合（Feature combination）：组合来自不同推荐数据源的特征被另一种推荐算法所采用.
* 层叠（Cascade）：先用一种推荐技术产生一种粗糙的推荐结果，第二种推荐技术在此推荐结果的基础上进一步作出更精确的推荐.
* 特征扩充（Feature augmentation）：一种技术产生附加的特征信息嵌入到另一种推荐技术的特征输入中.
* 元级别（Meta-level）：用一种推荐方法产生的模型作为另一种推荐方法的输入.

## 智能推荐系统的作用

个性化推荐的最大的优点在于，它能收集用户特征资料并根据用户特征，如兴趣偏好，为用户主动作出个性化的推荐.而且，系统给出的推荐是可以实时更新的，即当系统中的商品库或用户特征库发生改变时，给出的推荐序列会自动改变.这就大大提高了电子商务活动的简便性和有效性，同时也提高了企业的服务水平.

总体说来，一个成功的个性化推荐系统的作用主要表现在以下三个方面：

* 将电子商务网站的浏览者转变为购买者：电子商务系统的访问者在浏览过程中经常并没有购买欲望，个性化推荐系统能够向用户推荐他们感兴趣的商品，从而促成购买过程.
* 提高电子商务网站的交叉销售能力：个性化推荐系统在用户购买过程中向用户提供其他有价值的商品推荐，用户能够从系统提供的推荐列表中购买自己确实需要但在购买过程中没有想到的商品，从而有效提高电子商务系统的交叉销售.
* 提高客户对电子商务网站的忠诚度：与传统的商务模式相比，电子商务系统使得用户拥有越来越多的选择，用户更换商家极其方便，只需要点击一两次鼠标就可以在不同的电子商务系统之间跳转.个性化推荐系统分析用户的购买习惯，根据用户需求向用户提供有价值的商品推荐.如果推荐系统的推荐质量很高，那么用户会对该推荐系统产生依赖.因此，个性化推荐系统不仅能够为用户提供个性化的推荐服务，而且能与用户建立长期稳定的关系，从而有效保留客户，提高客户的忠诚度，防止客户流失.

个性化推荐系统具有良好的发展和应用前景.目前，几乎所有的大型电子商务系统，如Amazon、eBay等不同程度的使用了各种形式的推荐系统.国内方面，知名购物网站麦包包、凡客诚品、库巴网、红孩子等都率先选择了本土最先进的百分点推荐引擎系统构建个性化推荐服务系统.在日趋激烈的竞争环境下，个性化推荐系统能有效的保留客户，提高电子商务系统的服务能力.成功的推荐系统会带来巨大的效益. 另一方面，各种提供个性化服务的Web站点也需要推荐系统的大力支持，国内推荐系统领航者百分点科技就Web站点个性化内容推荐方面也做出了贡献，在信息爆棚的今天，实施个性化推荐势在必行.

## 本系统的职位与简历匹配算法

### 职位与简历的属性抽取

本算法设计前，我分别对内推网、拉勾网、智联招聘、前程无忧等极大招聘网站的求职者简历格式和职位信息发布格式进行了整理，并结合本设计的特殊性和应届生与内推者特有的属性列举出将要用于匹配推荐算法的字段：

表4-1 简历属性

|  |  |
| --- | --- |
| 字段（属性） | 标识符 |
| 学校 | W1 |
| 求职意向(根据投递职位排序) | W2 |
| 学历 | W3 |
| 专业 | W4 |
| 期望工作地 | W5 |
| 期望薪酬 | W6 |

表4-2 职位属性

|  |  |
| --- | --- |
| 字段（属性） | 标识符 |
| 内推人母校 | T1 |
| 工作地点 | T2 |
| 薪资待遇 | T3 |
| 学历要求 | T4 |
| 专业要求 | T5 |
| 公司规模 | T6 |
| 发布日期 | T7 |
| 职位投递次数前三 | T8 |

### 职位权重系数设置方法：

在职位匹配推荐算法时，需要为有关的属性特征分配权重系数，以便计算一份简历与相关职位的匹配程度，本系统主要使用了两个方案：

1. 用户在求职招聘网站浏览投递职位的时候，他们的投递信息会被记录到数据库，对于相似的职位名称，本系统可以进行习惯统计，提取应聘者较为习惯投递的职位名称属性，用于设置职位与简历的属性权重的设置，这是基于内容和兴趣的推荐，该方法根据“用户习惯”得来，用户在浏览系统后收集用户在系统内的操作作为反馈，系统便可对这些反馈数据进行统计和处理.
2. 利用目前较为准确的调查问卷，通过设置一些关于职位和简历的问题给目标群体填写，然后获取相关的权重数据，根据问卷网上的调查问卷对应届毕业生抽样的调查，统计数据.

### 职位权重计算和有关属性权重设定

1. 浏览职位时您最关注的是：设置权重为Vn.

表4-3 权重Vn表

|  |  |  |
| --- | --- | --- |
| 字段（属性） | 标识符 | 权值 |
| 内推人母校和自己相同 | T1 | 0.32 |
| 工作地点 | | T2 | 0.12 |
| 薪资待遇 | T3 | 0.14 |
| 学历要求 | T4 | 0.10 |
| 公司规模 | T6 | 0.12 |
| 职位投递次数前三 | T8 | 0.20 |

1. 以下几项您关注程序的先后顺序是？设置权重为Sn.

表4-4 权重Sn表

|  |  |  |
| --- | --- | --- |
| 字段（属性） | 标识符 | 权值 |
| 内推人母校和自己相同 | T1 | 0.25 |
| 工作地点 | | T2 | 0.13 |
| 薪资待遇 | T3 | 0.14 |
| 学历要求 | T4 | 0.10 |
| 专业要求 | T5 | 0.12 |
| 公司规模 | T6 | 0.10 |
| 发布日期 | T7 | 0.07 |
| 职位投递次数前三 | T8 | 0.19 |

1. 所在学校匹配系数Cn.与求职者相同母校的会更加的利于内推，所以需要对所在学校这个属性分配一个权值系数.主要分为同一学校、不同学校，权值分配如下表示：

表4-5 匹配系数表Cn

|  |  |
| --- | --- |
| 是否在同一学校 | 权重 |
| 同一学校 | 1.0 |
| 不同学校 | | 0.5 |

1. 工作地点的匹配系数Rn.符合求职者期望的工作地点的职位会比较受到求职者的欢迎，所以需要对工作地点这个属性分配一个权值系数.主要分为同一地点和不同地点，权值分配如下表示：

表4-6 匹配系数表Rn

|  |  |
| --- | --- |
| 期望工作地方是否相同 | 权值 |
| 同一地点 | 1.0 |
| 不同地点 | | 0.5 |

1. 薪酬待遇的匹配系数Pn.求职者都希望企业给自己的待遇越高越好.薪酬待遇的匹配系数将按期望薪酬相对于公司薪酬的几个阶段来分配权值.

表4-7 薪酬待遇系数Pn

|  |  |
| --- | --- |
| 薪酬所占阶段 | 权值 |
| 期望薪酬最大值<公司薪酬最小值 | 1.0 |
| 公司最小薪酬在期望薪酬之间 | 0.8 |
| 期望薪酬在公司薪酬之间 | | 0.6 |
| 公司最大薪酬在期望薪酬之间 | | 0.4 |
| 期望薪酬最小值>公司薪酬最大值 | | 0.2 |

1. 公司规模的匹配系数Gn.公司规模与职位稳定性有一定的关系.假定求职者都期望进入大公司工作，为大公司分配较高的权值.

表4-8 公司规模匹配系数Gn

|  |  |
| --- | --- |
| 公司规模 | 权值 |
| 1-50 | 0.2 |
| 50-100 | 0.5 |
| 100-500 | 0.8 |
| 500以上 | 1.0 |

1. 发布日期匹配系数Zn.发布日期是反映企业的招聘信息新鲜程度的一个属性，所以求职者会希望系统推荐给自己的职位信息是最近发布的，而且时间越近越好.

表4-9 匹配系数Zn表

|  |  |
| --- | --- |
| 发布日期 | 权值 |
| 近三天 | 1.0 |
| 近一周 | 0.8 |
| 近两周 | 0.5 |
| 近一个月 | 0.2 |

1. 职位投递次数前三 Jn.职位投递前三反映了求职者的简历投递情况，根据协同推荐，对于相同的职位，按照求职者投递的次数多少来排序并分配权值.

表4-10 匹配系数Jn表

|  |  |
| --- | --- |
| 职位匹配 | 权值 |
| 与第一匹配 | 1.0 |
| 与第二匹配 | 0.8 |
| 与第三匹配 | 0.5 |

### 

### 匹配算法公式

 (4-1)

# 基于智能匹配策略的高校内推系统的设计

## 界面设计

### 首页设计



图5-1 首页设计

该界面是用户未登录时的主界面，在webx中整个页面是一个大的layout. 页面的最上方是固定的导航栏，右侧是登陆框，右下方是热门学校. 最下方是底部页脚。这些页面都是control. 左上方是搜索框以及热门职位，左下方是一些职位信息，职位信息包含职位名称，公司名称，公司所在地和薪资范围，这两个页面是一个screen页面，用户通过左上方内嵌在首页的登录框登录. 基于webx的机制，在本页面可以记住用户所输入的密码. 登录失败时会有弹窗提示，登录成功后会在当前页跳转并展示个人信息.参见5.3小节.

如果用户尚未注册，会提示该用户不存在，需要注册，点击注册按钮会跳转到注册页面，注册页面会在5.2小节进行介绍.

### 个人信息界面设计



图5-2 个人信息页面

进入个人信息页面时默认会展示个人信息栏目，此页面左侧固定为几个栏目页面，分为个人信息，发布内推，简历编辑，智能推荐，投递记录，接受简历，发布记录这几个栏目.这个几个栏目根据角色不同展示不同，求职者只展示个人信息，简历编辑，智能推荐和投递记录.而推荐者展示个人信息，发布内推，接受简历和发布记录四个栏目，右侧为具体栏目对应的内容，个人信息页面如上图所示包含用户的真实姓名，联系邮箱，母校名称，自我介绍， 联系QQ以及对于推荐条数的设置. 填写完成后点击保存按钮，即可保存成功，如果保存失败会提示相关错误信息.

## 实现用户注册

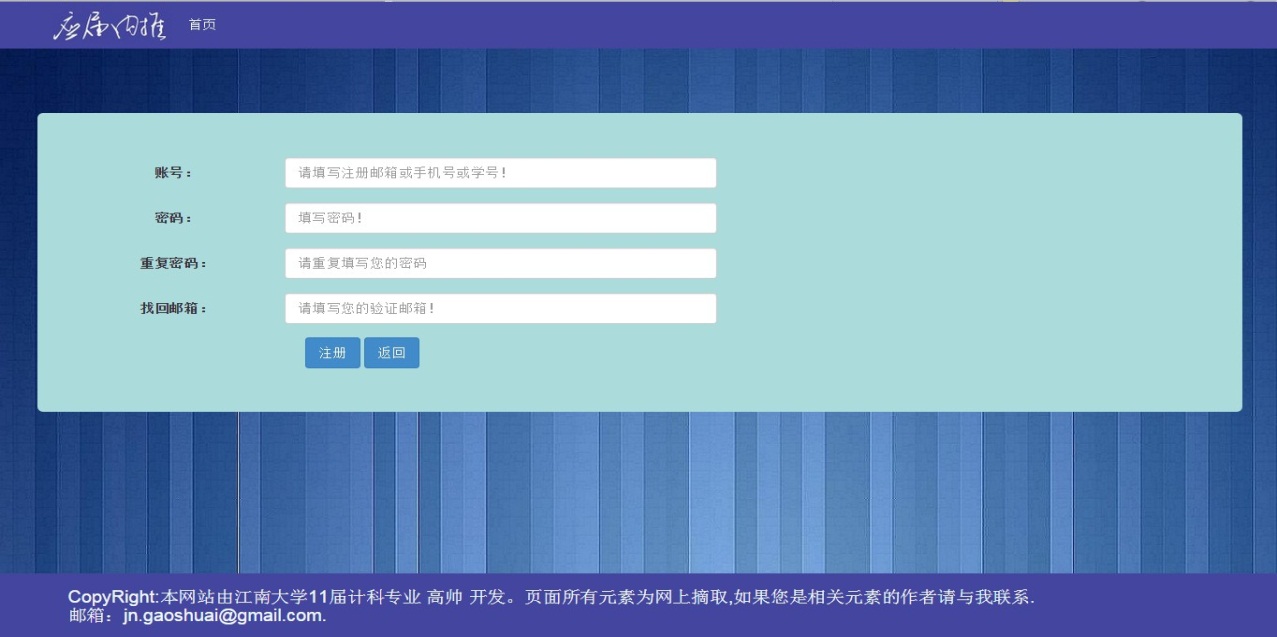


图5-3 注册页面

用户注册，用户注册时填写注册账号，密码，并填写找回密码邮箱.实现过程如下：

public void doRegister( @Param("userId") String userId,

@Param("password") String password,

@Param("rePassword") String rePassword,

@Param("email") String email,

Context context) {

if (!password.equals(rePassword)) {

context.put("errorMessage", "两次密码不相同，请重新输入！");

context.put("result", "failed");

return;

}

if (!checkEmail(email)) {

context.put("errorMessage", "邮箱格式不正确，请重新输入！");

context.put("result", "failed");

return;

}

User user = userManager.getUserByUserId(userId);

if (user != null) {

context.put("errorMessage", "该账号已存在，请换一个账号名注册哦！");

context.put("result", "failed");

} else {

Integer flag = userManager.register(userId, password, email);//数据库update的时候返回的是受影响的行数，如果行数大于0就表示成功了

if (flag > 0) {

context.put("result", "success");

} else {

context.put("errorMessage", "注册失败，请稍后再试！");

context.put("result", "failed");

}

}

}

Webx根据@Param获得提交的字段数据，首先对两次输入的密码是否相同进行判断，将错误信息put到页面，然后返回，页面提示密码不相同. 之后再检查输入的邮箱格式是否正确，如果不正确也同样会提示，第三点根据用户输入的账号查找数据库，如果能够获得到对象，则说明该账号是已经被注册过的，返回错误，提示该账号已经被注册了，否则插入数据库，数据库使用的是ibatis的半自动化，插入数据库时该语句返回的是数据库受影响的行数，如果大于0说明插入成功了，如果不是，说明数据库可能暂时出现了问题，需要稍等一会再注册.注册成功后返回首页.

## 实现用户登录

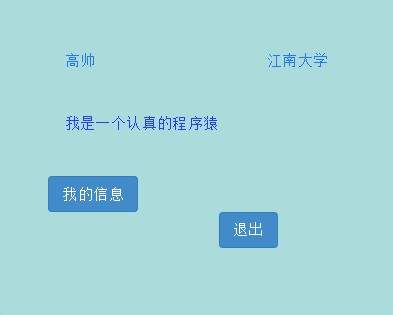


图5-4 登录框 图5-5 登录后

用户填写账号和密码后登陆，如果输入的账户不存在，或者密码不对，则会有相应提示，用户登录成功后会在首页中原来登录的区域会展示为当前登录后这个页面，当前页面会展示姓名，大学，和个人介绍. 点击我的信息页面进入个人信息主页. 点击退出，退出当前账号. 具体实现代码为：

public void doLogin(@FormGroup("login")

@FormField(name = "loginError", group = "login")

@Param("userId") String userId,

@Param("password") String password,

Navigator navigator,

HttpSession session,

ParameterParser params,

Context context) throws Exception {

User user = userManager.login(userId, password);

(user != null) {

//在session中创建SessionUser对象

SessionUser sessionUser =

(SessionUser) session.getAttribute(WebConstant.SESSION\_USER\_SESSION\_KEY);

if (sessionUser == null || sessionUser.hasLoggedIn()) {

sessionUser = new SessionUser(WebConstant.ACCESS\_REALM);

}

sessionUser.upgrade(userId);

session.setAttribute(WebConstant.SESSION\_USER\_SESSION\_KEY,sessionUser);

navigator.redirectTo(WebConstant.LOGIN\_RETURN\_DEFAULT\_LINK);

} else {

navigator.redirectTo(WebConstant.LOGIN\_RETURN\_DEFAULT\_LINK)

.withParameter("loginResult","failed")

.withParameter("errorMessage","用户名或密码不对！请重新登录!");

}

}

首先登录时先根据用户名和密码查询一下数据库，如果能够找到当前对象，说明该对象存在，如果找不到说明不存在.该对象存在时将该对象根据session\_key放入httpSession中，用户未退出本系统或未退出浏览器时Session中一直保存着该用户数据，直到该用户退出系统或者关闭浏览器，session中的数据将会被删除.

## 实现职位浏览与投递



图5-6 职位浏览

根据webx的规则，分页查询职位信息，目前默认每页展示10个职位信息.

if (searchText == null) {

jobs = jobManager.getJobListWithPageSize(startRow);

sumResult = jobManager.getJobList().size();

} else {

jobs = jobManager.queryJobsBySearchText(searchText, startRow);

sumResult = jobs.size();

}

如果没有执行搜索，则直接分页查询职位信息.如果是搜索的则根据搜索关键字分页查询职位信息.



图5-7 职位投递

首页可点击职位信息进入职位详细信息页面，职位信息页面展示内推人，薪资，发帖时间，公司名称，公司地点以及内推人所在高校.在职位信息中展示职位的具体要求，在职位要求摘要中展示公司较为吸引人的一些摘要信息，比如公司具有健身房. 点击发送简历按钮会将个人自己的个人简历发送给职位发布人.

public void doSendResume(@Param("id") Integer id,

HttpSession session,

Navigator navigator,

Context context) throws Exception {

SessionUser sessionUser=

(SessionUser) session.getAttribute(WebConstant.SESSION\_USER\_SESSION\_KEY);

if(sessionUser==null){

context.put("result","failed");

context.put("errorMessage","你还没有登录，先到首页登录再投递该职位哦！");

navigator.redirectTo(WebConstant.LOGIN\_RETURN\_DEFAULT\_LINK);

return ;

}

DeliveryPost deliveryPost =

jobManager.getCheckSendInfoByJobIdAndUserId(id,sessionUser.getUserId());

System.out.println("deliveryPost="+deliveryPost);

if(deliveryPost!=null){

context.put("result","failed");

context.put("errorMessage","你已经投递过该职位啦，相同职位职能投递一次，赶快去看看其他职位吧.");

return ;

}

Integer flag= jobManager.addDeliveryRecord(sessionUser.getUserId(),id,DELIVER);//数据库update的时候返回的是受影响的行数，如果行数大于0就表示成功了

if(flag>0){

context.put("result", "success");

context.put("errorMessage", "投递成功啦，请静待学长联系你吧！");

}else{

context.put("result", "failed");

context.put("errorMessage", "投递失败，请稍后再试下哦，亲！");

}

}

如果投递职位时还没有登录会提示先登录，如果已经投递过该职位将不能重复投递相同的职位，满足以上要求后才可以正确发送简历.

## 实现发布内推

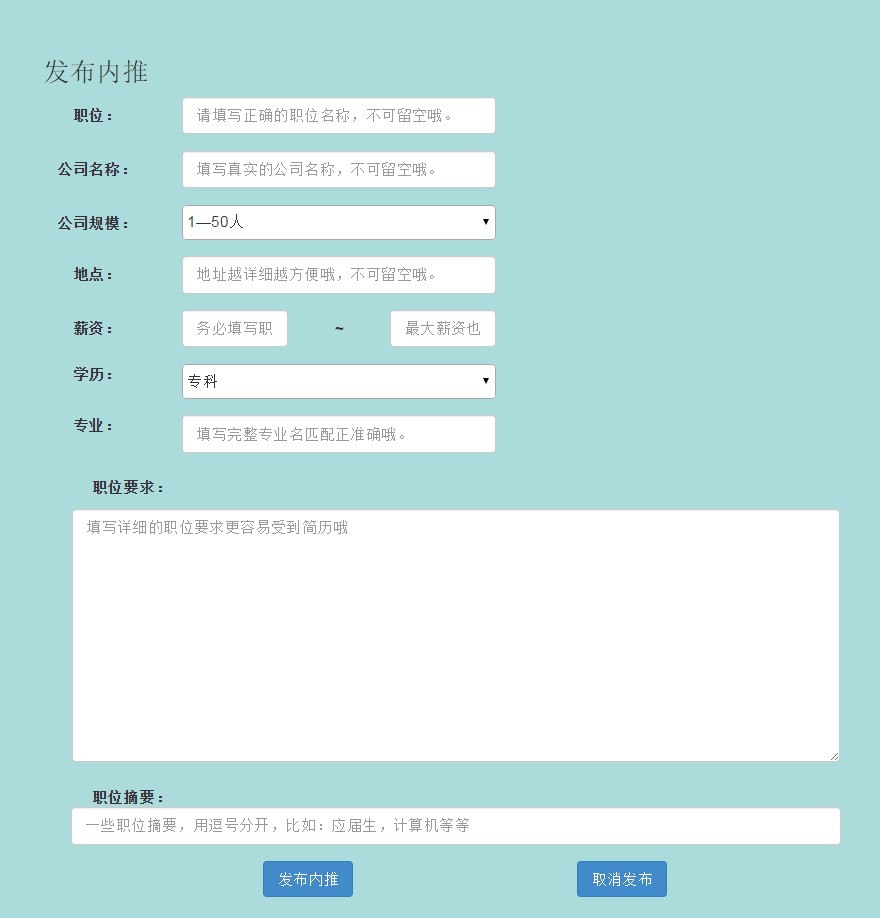


图5-8 发布内推

Job job = new Job();

job.setUserId(userId);  
job.setJobName(jobName);  
job.setCompany(company);  
job.setMinMoney(minMoney);  
job.setMaxMoney(maxMoney);  
job.setAddress(address);  
job.setJobReq(jobReq);  
job.setReqSummary(reqSummary);  
job.setUniversity(user.getUniversity());  
job.setEduBackground(eduBackground);  
job.setSpecialty(specialty);  
Integer jobId = jobManager.postJob(job);

if(jobId<0){  
context.put("result", "failed");

return;

}  
Integer flag=jobManager.addDeliveryRecord(sessionUser.getUserId(),jobId,POST);

发布内推页面需要填写职位名称，公司，薪资，地点，职位要求和职位摘要以及学历要求.

且必填字段不能为空，每个字段会有验证，如果为空会做出验证并提示失败.验证通过后将职位信息记录到数据库.

## 实现简历编辑



图5-9 简历编辑

Resume resume = new Resume();

resume.setUserId(userId);  
resume.setName(name);  
resume.setPhone(phone);  
resume.setEmail(email);  
resume.setBirthday(birthday);  
resume.setCity(city);  
resume.setMinMoney(minMoney);  
resume.setMaxMoney(maxMoney);  
resume.setExperience(experience);  
resume.setEduBackground(eduBackground);  
resume.setUniversity(university);  
resume.setSpecialty(specialty);  
Integer flag = userManager.editResume(resume);

简历编辑需要填写个人信息的若干字段，其中必填字段有姓名，手机，邮箱，出生年月，所在城市，经历和学历.这些字段会有相关的不为空验证，根据对象将个人信息set到对象的字段中，然后操作数据库，修改个人简历信息，个人简历保存时分为第一次创建和已有简历修改，当数据库中不存在个人简历时会直接创建一份个人简历信息，当数据库中已经存在当前登录者的个人简历时，会执行update个人简历而不是insert数据库.

## 实现智能推荐



图5-10 智能推荐

public void execute(HttpSession session,Context context) {

SessionUser sessionUser = (SessionUser)

session.getAttribute(WebConstant.*SESSION\_USER\_SESSION\_KEY*);

ResumeDO resumeDOInfo =

userManager.getUserResumeByUserId(sessionUser.getUserId());

UserDO userDO=userManager.getUserByUserId(sessionUser.getUserId());

List<AiResultDO<JobDO>> aiResultDOLists =

jobManager.calculationMatching(resumeDOInfo).subList(0, userDO.getAiNums());

context.put("aiResultLists", aiResultDOLists);

}

@Override

public List<AiResultDO<JobDO>> calculationMatching(ResumeDO resumeDO) {

List<AiResultDO<JobDO>> aiResultDOList = new ArrayList<AiResultDO<JobDO>>();

List<JobDO> jobDOList = getJobList();

DecimalFormat df = new DecimalFormat("######0.00");

for (JobDO jobDO : jobDOList) {

Double recommendPoint = calculationRecommendPoint(jobDO, resumeDO);

AiResultDO<JobDO> aiResultDO = new AiResultDO<JobDO>();

aiResultDO.setData(jobDO);

aiResultDO.setAiNums(df.format(recommendPoint));

aiResultDOList.add(aiResultDO);

}

Collections.sort(aiResultDOList, new Comparator<AiResultDO>() {

public int compare(AiResultDO arg0, AiResultDO arg1) {

return arg1.getAiNums().compareTo(arg0.getAiNums());

}

});

return aiResultDOList;

}

public Double calculationRecommendPoint(JobDO jobDO, ResumeDO resumeDO) {

List<ProbabilityListDO> probabilityList = getProbabilityList();

MatchingPointDO matchingPointDO = getMatchingPointDO(jobDO, resumeDO);

if (probabilityList == null) {

System.out.println("probabilityList is null");

return null;

}

ProbabilityListDO mostAttention = probabilityList.get(0);//最关注的属性分值

ProbabilityListDO mostAttentionOrder = probabilityList.get(1);//最关注的属性顺序分值

if (matchingPointDO == null) {

System.out.println("matchingPointDO is null");

return null;

}

System.out.println("matchingPointDO:" + matchingPointDO);

Double pointOfMostAttention = mostAttention.getUniversity() \*

matchingPointDO.getUniversityPoint()

+ mostAttention.getAddress() \* matchingPointDO.getAddressPoint()

+ mostAttention.getMoney() \* matchingPointDO.getMoneyPoint()

+ mostAttention.getEduBackground()

+ mostAttention.getCompanySize() \* matchingPointDO.getCompanySizePoint()

+ mostAttention.getDeliveryThree() \* matchingPointDO.getDeliveryThreePoint();

Double pointOfMostAttentionOrder = mostAttentionOrder.getUniversity()

\* matchingPointDO.getUniversityPoint()

+ mostAttentionOrder.getAddress() \* matchingPointDO.getAddressPoint()

+ mostAttentionOrder.getMoney() \* matchingPointDO.getMoneyPoint()

+ mostAttentionOrder.getEduBackground()

+ mostAttentionOrder.getSpecialty()

+ mostAttentionOrder.getCompanySize() \* matchingPointDO.getCompanySizePoint()

+ mostAttentionOrder.getPostTime() \* matchingPointDO.getPostTimePoint()

+ mostAttentionOrder.getDeliveryThree() \* matchingPointDO.getDeliveryThreePoint();

return (pointOfMostAttention + pointOfMostAttentionOrder) / 2;

}

推荐算法会根据用户的投递行为，是否和内推人在同一所大学，学历以及专业和职位要求来计算推荐指数. 基于组合的推荐结合了用户的投递习惯，以及用户个人信息的内容，协同这两个方面对用户适合的职位进行推荐，详细推荐算法可细读4.5节本系统的职位与简历匹配算法.

## 实现接收简历

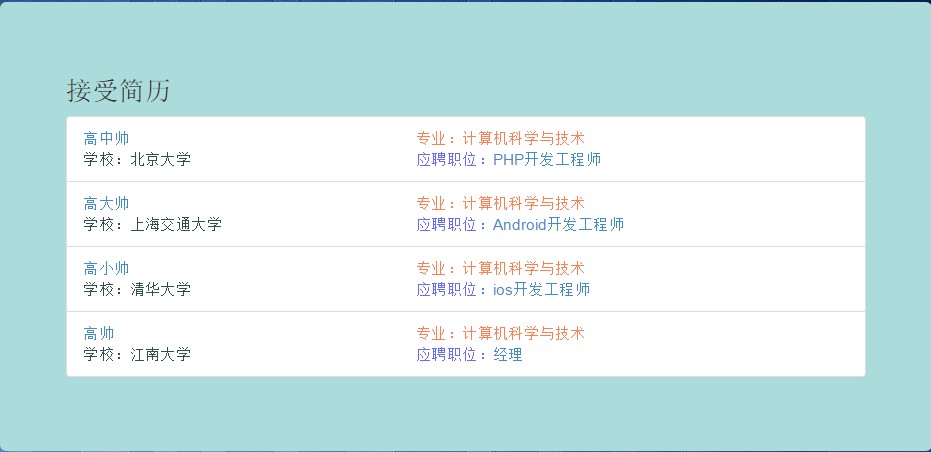


图5-11 接受简历

public void execute( HttpSession session,Context context) {

SessionUser sessionUser =

(SessionUser) session.getAttribute(WebConstant.SESSION\_USER\_SESSION\_KEY);

List<Job> jobs = jobManager.getJobInfoByUserId(sessionUser.getUserId());

List deliveryPosts = new ArrayList();

for (Job job : jobs) {

List deliveryPost = jobManager.getJobListByJobId(job.getId(), DELIVERY);

deliveryPosts.addAll(deliveryPost);

}

context.put("deliveryPosts", deliveryPosts);

}

首先根据用户id查询到该用户发布的职位信息，然后根据职位信息在投递记录表中获取这些职位有哪些投递记录，进行展示.

# 

# 小结

## 系统功能概述

本系统是一个基于智能匹配策略的内推网系统.致力于搭建应届毕业生和企业已就职学长之间的桥梁，形成一个常态化的学长内推学弟的内推系统. 本系统主要以内推人和应聘人所在学校为维度，学长发布完内部推荐职位后，需有相应的应聘人投递简历，内推人查看应聘人简历后决定是否内推，相同学校的应聘人和内推人将有很大的机会相互认识熟悉并且成功内推.

## 系统存在的不足

由于自己的技术能力有限，因此这个系统存在诸多的不足或者说是有待完善的地方.以下根据当前主流的内推网系统列出我的这个系统的一些不足之处：

1. 目前界面的交互和视觉较差.
2. 用户个人简历只能是自主填写而不能够上传附件.

## 有待改进之处

1. 页面在视觉上仍有很大的提升余地，以及在交互上并不是很简洁.
2. 用户个人简历以附件word或者pdf文件存储.

# 

# 参考文献

1. 埃克尔. java编程思想（Think in java）[M]. 机械工业出版社，2007
2. 李刚. java疯狂讲义[M]. 电子工业出版社，2008,322-345
3. 陈雄华，林开雄. Spring3.0就这么简单[M]. 人民邮电出版社，2013.
4. 沃尔斯，布雷登巴赫. Spring in Action[M]. 人民邮电出版社，2008,31-216
5. 贝让 等著，叶俊 等译. IBATIS实战[M]. 人民邮电出版社，2008
6. 赫佳. Spring源码深度解析[M]. 人民邮电出版社，2013,231-244
7. 刘锦峰，朱少强. 招聘网站职位与简历的双向匹配推荐算法[J]. 时代经贸. 2014（3）
8. 江志祥. 智能简历解析系统的研究与实现[D]. 北京邮电大学，2009
9. 李煊，汪晓岩 庄镇泉. 基于关联规则挖掘的个性化智能推荐服务[J]. 计算机工程与应用，2002
10. 姚奇富. 基于web访问挖掘的个性化智能推荐服务[J]. 图书情报工作， 2006(5)
11. Clive Begin,Brandon Goodin,Larry Meadors. ibatis in action[M]. Manning. 2007(1):1-384
12. Joni Hersch. Education Match and Job Match[J]. The MIT Press. 1991(1):140-144
13. John Robst. Career Mobility, Job Match, and Overeducation[J]. Eastern Economic Journal. 1995(4): 539-550
14. Curtis J. Simon and John T. Warner Matchmaker, Matchmaker: The Effect of Old Boy Networks on Job Match Quality, Earnings, and Tenure [J]. The University of Chicago Press. 1992(3):306-330
15. Dept. of Electr. & Comput. Eng. An intelligent Web recommendation engine based on fuzzy approximate reasoning[J]. IEEE International Conference on. 2003(2):25-28

# 

# 致谢

首先要感谢我的导师蒋敏老师.蒋老师是一个特别特别认真负责的好老师，在课题设计的过程当中，蒋老师认真的为我们规划毕业设计时间点，帮我们耐心审查开题报告和检查论文翻译，提出其中描述不清晰的点，指导我们认真按时完成毕业设计，为毕业设计的后期留下充足的时间，在我的毕设中给出很多专业的建议和指导.另外也要感谢江南大学物联网学院顾德老师在写作格式问题中提供的帮助，毕业设计的格式是我的一大头痛之处，看了顾德老师的视频，真的对于格式方面的调整节省了不少的时间，再次对蒋老师和顾德老师表示感谢.

其次要感谢我的现实习公司的师兄，公司中每天有很多的事情，有时间都忙到很晚很晚，毕业设计是在平时晚上的空余时间中拼凑时间来完成的，界面不是很美观，交互也不是特别的简洁，还需要做很多的修改，平时公司的事情很多，师兄知道我在做毕设，分配给我的任务就相对比较少些，所以可以在晚上空出大多数的时间来做毕业设计.

最后要感谢大学四年里我所遇到的好老师好学长们，从你们身上我学到了一个物联网专业弱校向外界展示的不屈与努力，也是因为你们，我才能够有耐力在实验室一行一行的书写我的大学，其实我一直都是一只it小小鸟不断的努力朝着自己的方向飞. 感谢大学四年里和我一起相处的所有同学们，尤其是我的室友，我是一个睡觉很容易受打扰的人，经过一段时间磨合，晚上大家都开始早睡，是你们陪我成长、陪我走过大学四年美丽的光阴.

真诚地感谢你们！