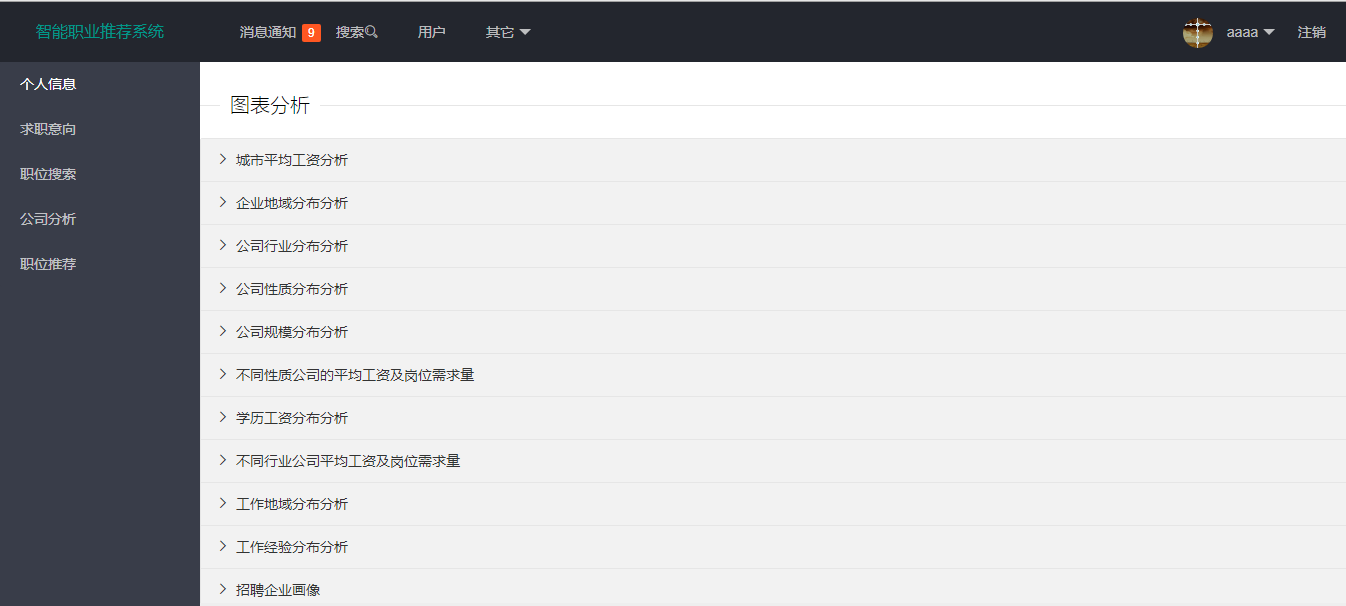
# 页面

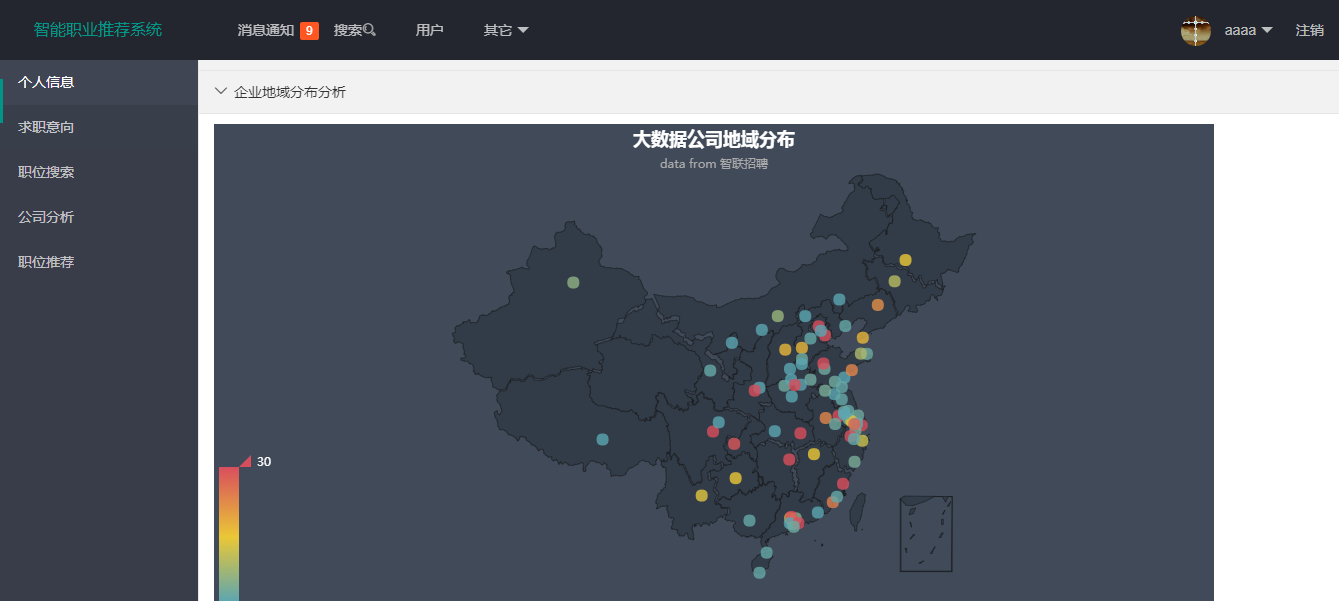
页面总体由登陆注册，个人信息查看修改，求职意向查看修改，职位搜索，工作和公司的详情查看，公司相关分析，职位相关分析，职位推荐组成。

## 登录注册



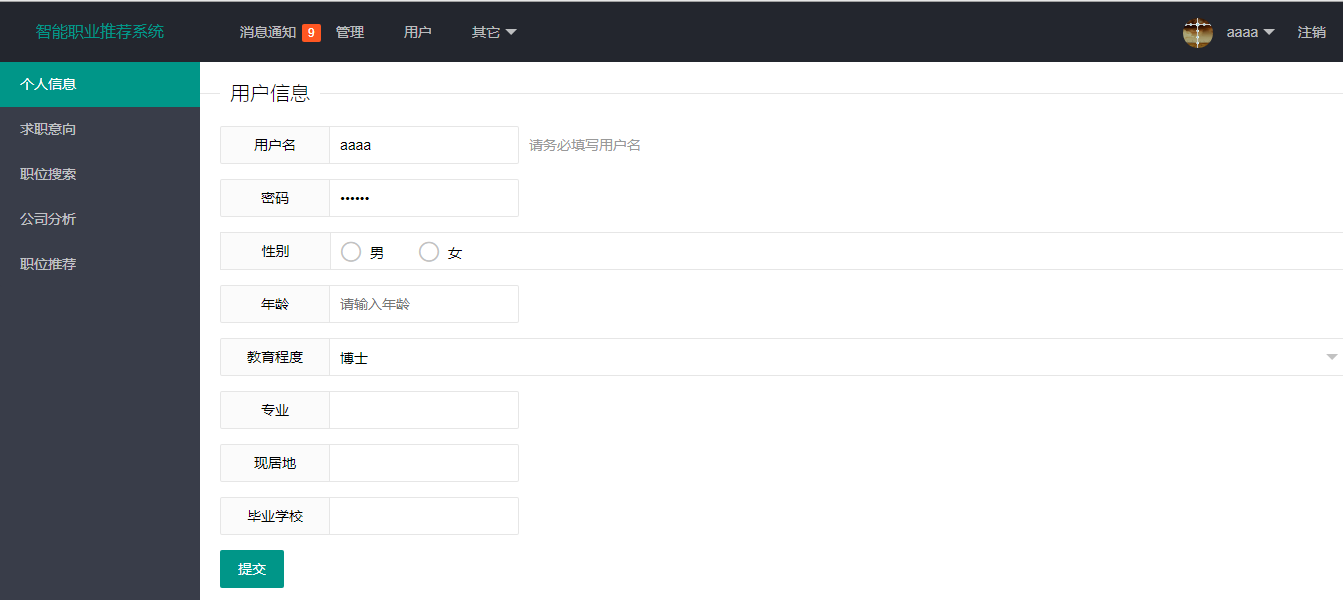
## 系统首页





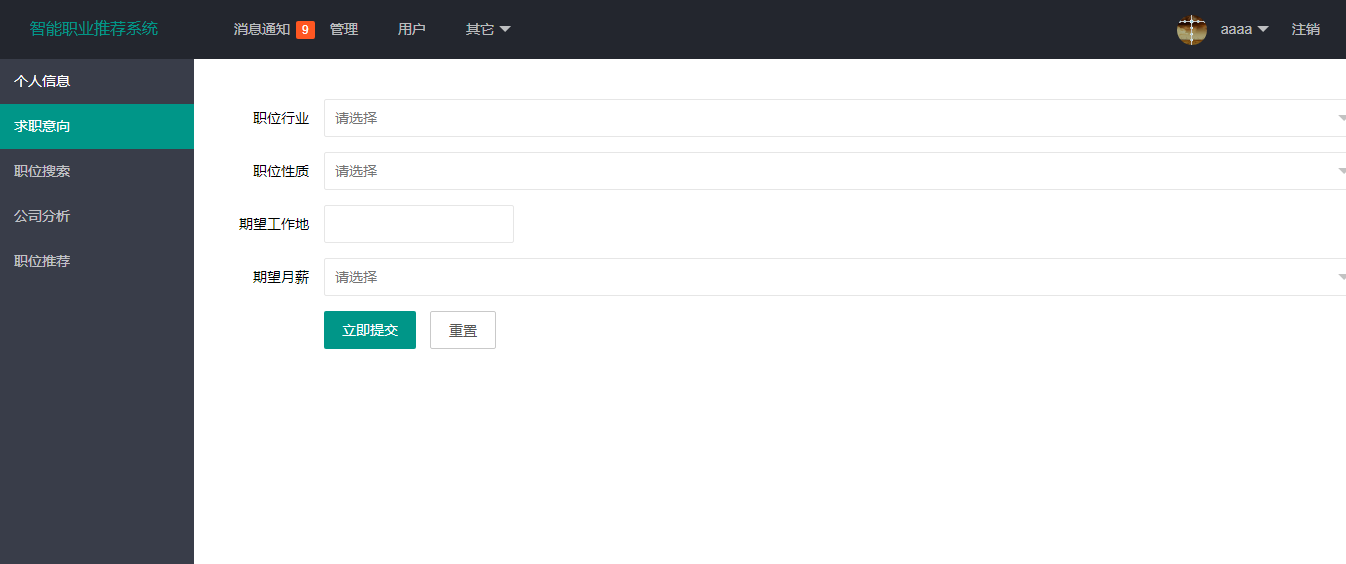
系统可以通过点击不同的分析展示不同的分析结果，详细的相关分析已经在《招聘数据分析》中展示。

## 用户信息



用户可以通过填写个人相关信息，增加之后职位推荐的精确性。

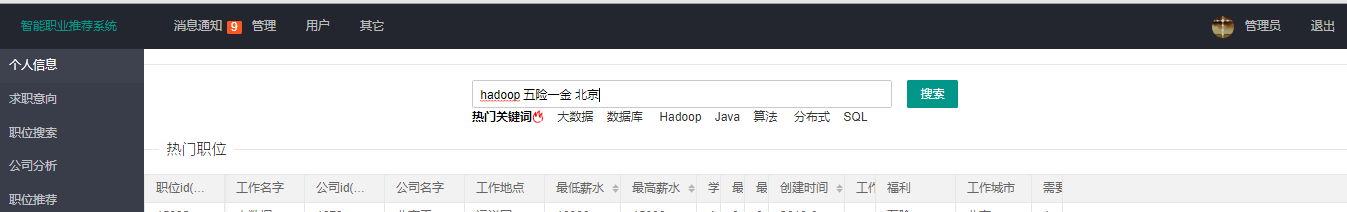
## 用户求职意向



用户可以通过填写求职意向相关信息，增加之后职位推荐的精确性。

## 职位搜索

用户可以输入想要查询的信息，系统经过相关匹配，返回最符合条件的数据并排序，如图以hadoop这个职位进行匹配。



返回结果均为相关结果，而且对工作名字里面出现了的关键字进行了标注。



点击对应的id可以查看详细信息





## 职位推荐

系统通过用户的个人信息和求职愿意和其浏览收藏记录以及他人的相似情况做出推荐结果。

每个用户完善信息 然后点职位推荐，职位推荐判断如果没打分就会打分，就是说每个新用户要点一次职位推荐，等待后才能得到推荐结果，同时系统还采用系统基于lucene的文本相似度计算的基于内容的推荐方法，推荐结果受到用户行为的影响，推荐结果占比按用户的近期不同职位浏览点击记录以及收藏记录占比组成，推荐出的结果还会尽量满足用户的求职意愿例如薪水、工作地、工作性质的需求以及匹配用户的个人信息例如用户学历等，推荐出最适合用户的工作，由此增加增加推荐的精确度。

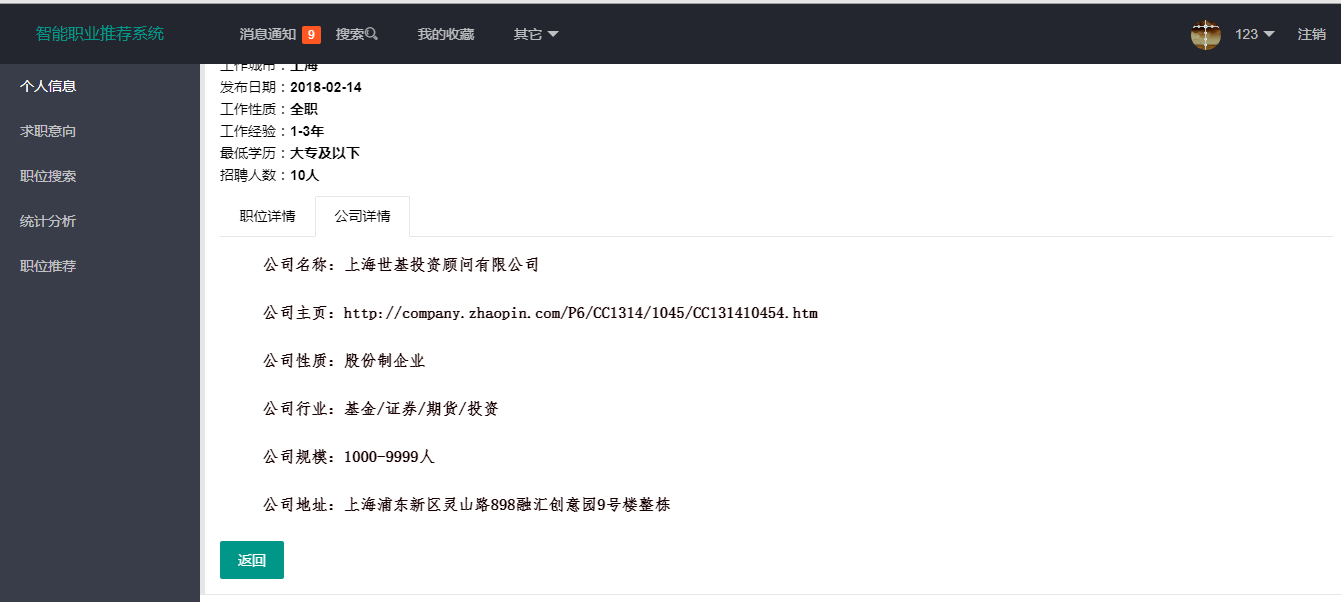




## 职位及公司详情

通过点击对于的职位名字可以获得职位和公司的详细信息，并可以收藏以便后续再次浏览。





# 部署

由于采用开源平台搭建系统，不便于您下载，故只提供web项目代码和部分测试数据

本次需要的软件有：

jdk1.7以上

tomcat7以上（至少两个，一个用于提供solr服务，一个用于提供web服务）

solr

redis

mysql5.5以上

maven3

# 安装方法

1. 在mysql中导入recommend\_system\_v1.0.sql，生成备份数据。
2. 使用maven将recommend\_system-web源码打包成war，部署到8080端口的tomcat。
3. 将solr部署到8081端口的tomcat。
4. 将solr缓存清空，清空操作可见<http://blog.csdn.net/lbf5210/article/details/51207043>。
5. 手动导入索引，输入localhost:8081/search/importall，观察tomcat日志查看是否导入成功。
6. 启动redis，并配置redis所在的服务器ip，并关闭相应端口的防火墙。
7. 浏览器输入localhost:8080进入系统，注册并登录即可以体检系统，您也可以使用测试用户aaa，密码666666。

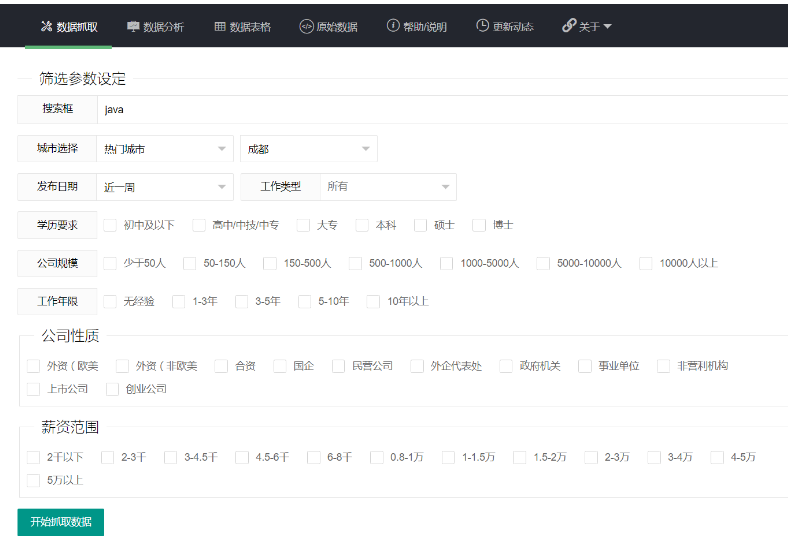
# 展望与总结

由于时间以及团队人数的限制，系统存在一些漏洞，并且有许多可以改进的地方，下面是我们自己考虑可以并且能够加入的内容。

## 用户自主选择爬取什么样的数据

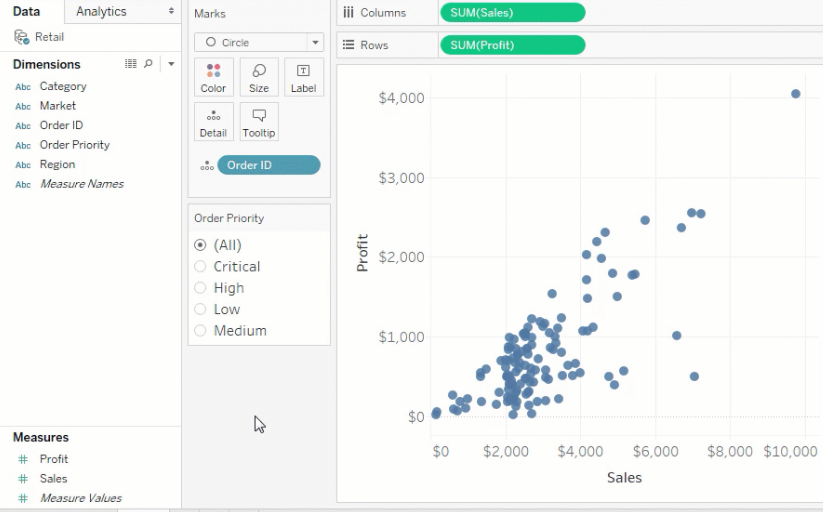
此功能暂时在开发中，本系统代码暂未展示。

用户可以通过自主选择参数，可以筛选自己想看的职位，并且对其进行数据分析。



## 用户拖拽字段自主进行数据分析

我们可以尝试接入Tableau，实现用户自主数据分析，增加系统的灵活性，系统展望如图。



## 推荐模块的细节优化

当用户输入需求以后，应该查询得到列表中有级别区分 比如Top3：最大化显示Top10等。

当用户查看了推荐的职位以后，一键跳转到实际的招聘网页，方便用户。

## 总结

系统演示采用的是单机架构，实际上的项目是采用hadoop+hive+hbase+sqoop+storm+kafka +mahout+flume分布式架构，可以应对大量的数据并且保证一定的性能，由于搭建对应的系统十分复杂，不方便供人们使用，本次项目就未主要描述相关配置，如果读者想实现大数据的架构，可以采用腾讯云阿里云等云服务，这样可以让自己更加专注于业务核心；同时系统还有很多不足之处，在未来我们会继续改进，让系统更加人性化，更加健壮。