

**软件工程课程设计**

**实验报告**

**实验名称：高考志愿填报推荐系统设计**

**班 级：2014级软件工程3班**

**姓 名：uknowho**

**学 号：0x091395**

**指导教师：辛动军**

2016.11.22

目录：

[1 系统任务概述： 2](#_Toc470203851)

[相关解释： 3](#_Toc470203852)

[2 系统需求分析 4](#_Toc470203853)

[2.1系统任务及特点： 4](#_Toc470203854)

[2.2 功能模型： 4](#_Toc470203855)

[2.2.1推荐系统： 4](#_Toc470203856)

[2.2.2对比系统： 7](#_Toc470203857)

[2.2.3交互系统 10](#_Toc470203858)

[2.3对象模型 16](#_Toc470203859)

[2.3.1用户类 16](#_Toc470203860)

[2.3.2推荐类 17](#_Toc470203861)

[2.3.3学校类 17](#_Toc470203862)

[2.3.4对比类 18](#_Toc470203863)

[2.3.5网络类 18](#_Toc470203864)

[2.3.6图形化类 19](#_Toc470203865)

[2.3.7交互类 19](#_Toc470203866)

[2.4数据模型 20](#_Toc470203867)

[2.5动态模型 21](#_Toc470203868)

[2.5.1交互图 21](#_Toc470203869)

[3 问题及总结 26](#_Toc470203870)

## 1 系统任务概述：

随着教育水平日渐提高，越来越多的人接受到了受教育的权利。并且越来越多的人步入了理想的大学殿堂。然而对于高考结束的同学来说，志愿填报随即成为高考之后又一重大人生抉择。尽管有一部分高中毕业生早已对自己理想的大学进行了研究和选择，可是忙于学习，了解到的内容不乏有些片面。那么对于其他大部分毕业生来说，并没有提前做好功课。其中最主要的原因可能是因为无法对自己的分数持有十足的把握。也许提前考虑了某些理想的大学，也花了很久的时间去了解。但是往往分数却达不入校标准或有被学校进行调剂的后患。心情落差大，也就没有之前那么高昂的兴趣去了解其他符合自己当前分数线的学校。通过调查我们了解到，对大学不了解的原因主要有以下几种：

面对成绩心情差，志愿随便就填了。

不知道从哪儿去了解，志愿填报参考书上也不够简明。

网上对学校的评论反差大，自己也拿不准。

无法在短时间内对多所学校进行了解。

不知道自己喜欢什么。

想去离海近的地方，却不知道那里哪些学校适合自己。

……

针对上述问题，我们进行了分析。大概总结出填报志愿困难的一些要素：

无法对院校进行全面的分析；

不能对院校进行客观的描述和评论；

不能根据自己的真实情况快速获取推荐；

无法根据自己的对周围环境的偏好进行推荐；

于是对于总结出来的几点问题要求。决定出开发面向于高中毕业生或在接下来的一段时间内即将进行高考，填报志愿的同学们的一款软件。该软件提供的服务有：

1. 根据学生的专业喜好进行推荐；
2. 根据学生的真实分数进行推荐；
3. 根据学生的环境偏好进行推荐；
4. 横向对比符合要求的院校；
5. 纵向对比相似院校；
6. 院校详细分析；
7. 消费状况分析；
8. 交通是否便利；

### 相关解释：

* 横向对比：对筛选出的院校如下项对比进行：

1. 五年内学校录取分数/平均录取分数/是否有降分录取情况；
2. 院校王牌专业，录取人数，录取率；
3. 就业率排名前10的专业；
4. 转专业要求高低；
5. 学校男女比例；
6. 录取生来源地分布统计；

* 纵向对比：

1. 学校伊始；
2. 是否远离城市中心
3. 优秀毕业人/代表性毕业人；
4. 共开设专业数；

* 院校详细分析（近五年）：

1. 各专业师资力量；
2. 各专业就业方向；
3. 各专业就业率；
4. 各专业考研率/考研录取率；
5. 图书馆简述，藏书量/特有功能等等；

## 2 系统需求分析

### 2.1系统任务及特点：

该软件旨在向需要做志愿填报的一批同学提供服务：根据自己的要求进行相关介绍及推荐。该系统具有分析全面，对比清晰，推荐合理，数据量大且数据真实，包含第一批本科，第二批本科，第三批本科及其他大专院校，惠及不同分数段的同学。

### 2.2 功能模型：

#### 2.2.1推荐系统：

推荐系统可以根据用户的真实分数，民族，地域/环境偏好，专业倾向等信息进行院校推荐。推荐原则是优先一本院校，其次二本，依次类推。推荐服务流程如图2.1及用例各角色用例图如图。

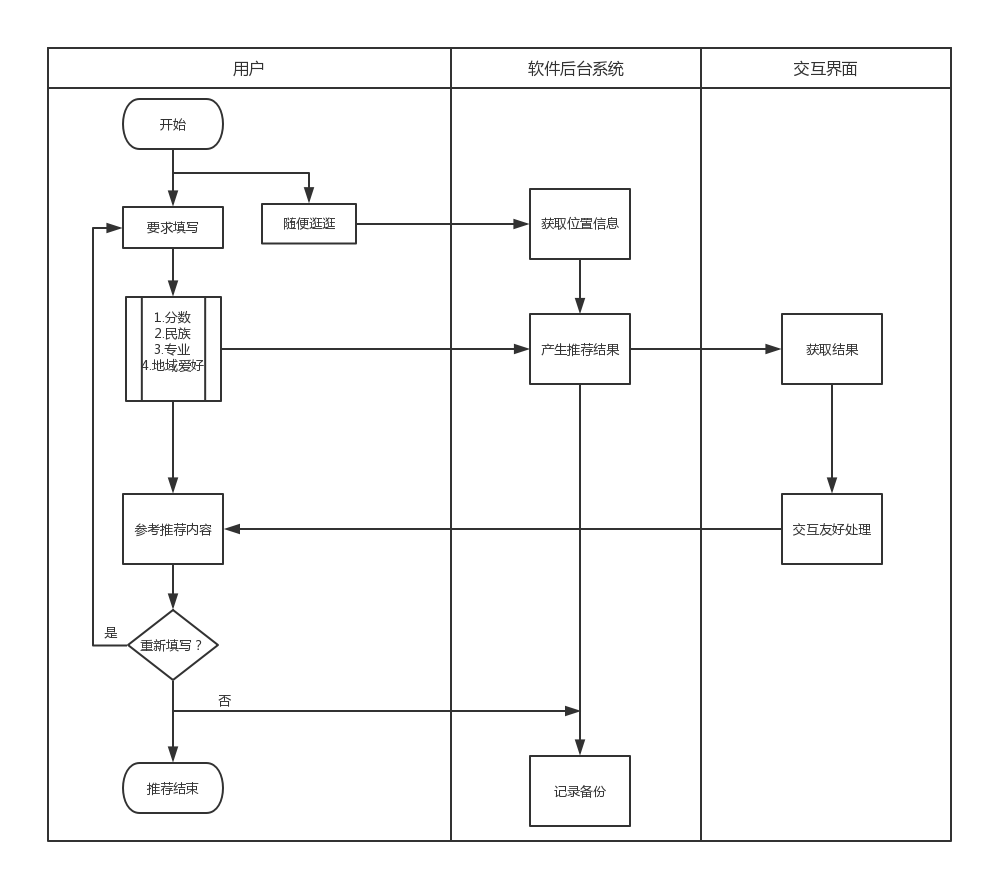


图2.1推荐流程图

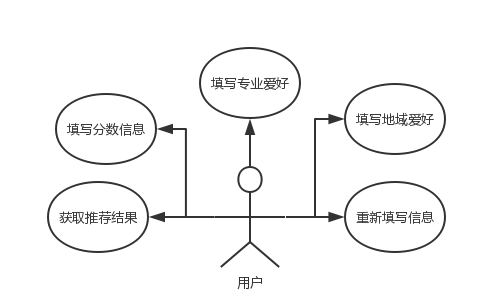


图2.2用户用例

表2.1用户用例描述

|  |  |
| --- | --- |
| 填写分数 | 输入框，用于 填写用户的真实分数 |
| 填写地域爱好 | 下拉框，省/市选取，用于填写用户的低于爱好 |
| 填写专业爱好 | 输入框，用于填写用户的低于爱好，若没有该专业应给予用户适当提示 |
| 获取推荐结果 | 按钮，点击获取由推荐系统产生的推荐结果，显示在交互系统中 |
| 重新填写 | 清空输入，重新填写 |

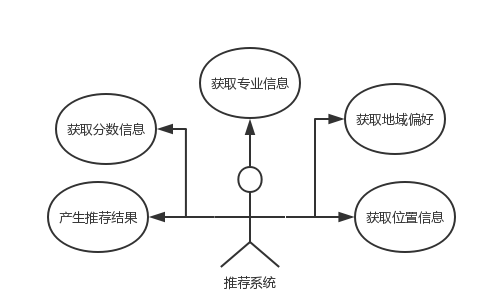


图2.3推荐系统用例

表2.2推荐系统用例描述

|  |  |
| --- | --- |
| 获取专业信息 | 获取用户选择的专业 |
| 获取分数信息 | 获取用户填写的分数 |
| 获取地域偏好 | 获取用户选择的地域 |
| 获取地理位置 | 获取用户当前的地理位置 |
| 产生推荐结果 | 根据推荐算法产生推荐结果，并将结果传递给交互系统进行处理 |

推荐系统还可以根据网络类获取的网络数据向用户展示更多信息，比如人数统计，专业人数，学生生源分布等信息。

#### 2.2.2对比系统：

通过推荐系统产生的推荐结果可能不唯一，那么在用户进行选择时有必要看出不同结果之间的对比状况。于此，推荐系统根据以下几个方面对筛选出的结果进行对比。

1. 横向对比
2. 纵向对比
3. 并将结果以图表的形式输出到交互系统。对比系统的流程图如图2.4：

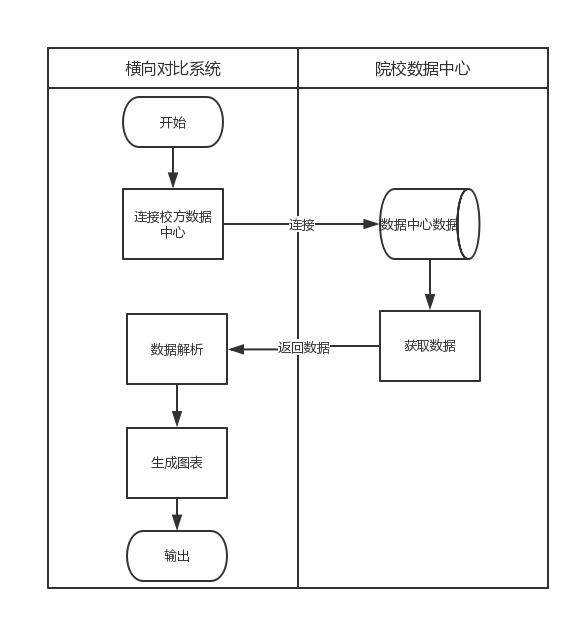


图2.4对比系统流程图

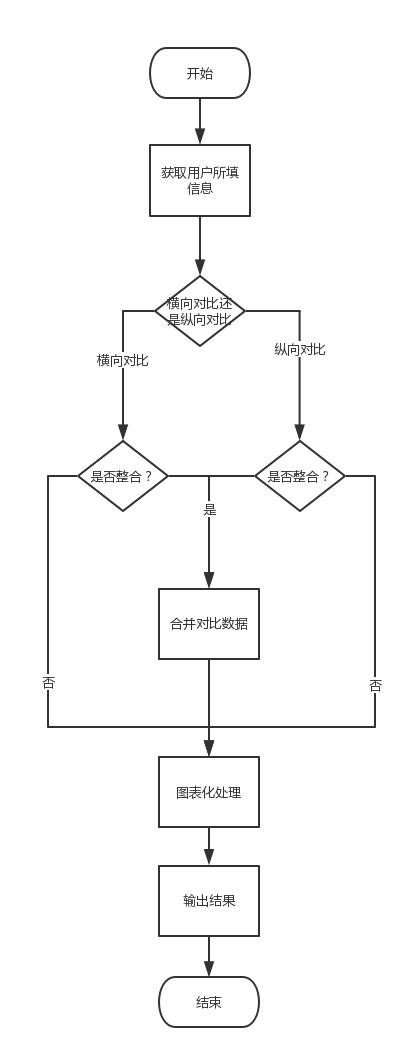
横向对比与纵向对比具体内容请参考“系统任务概述”章节。横向对比的流程图如图2.5：

图2.5横向对比流程图

纵向对比与横向对比的流程图一致。连接校方相应的数据库后希望能够尽可能多的获取校方公开的数据，而不仅仅限与横向与纵向对比的内容。因为在软件服务的过程中只有对比系统会连接两次数据库。其他功能的服务数据都是基于上述过程获取到的数据。数据格式可以使xml格式或json格式，方便数据的解析。生成对比图表的大概样式如图：

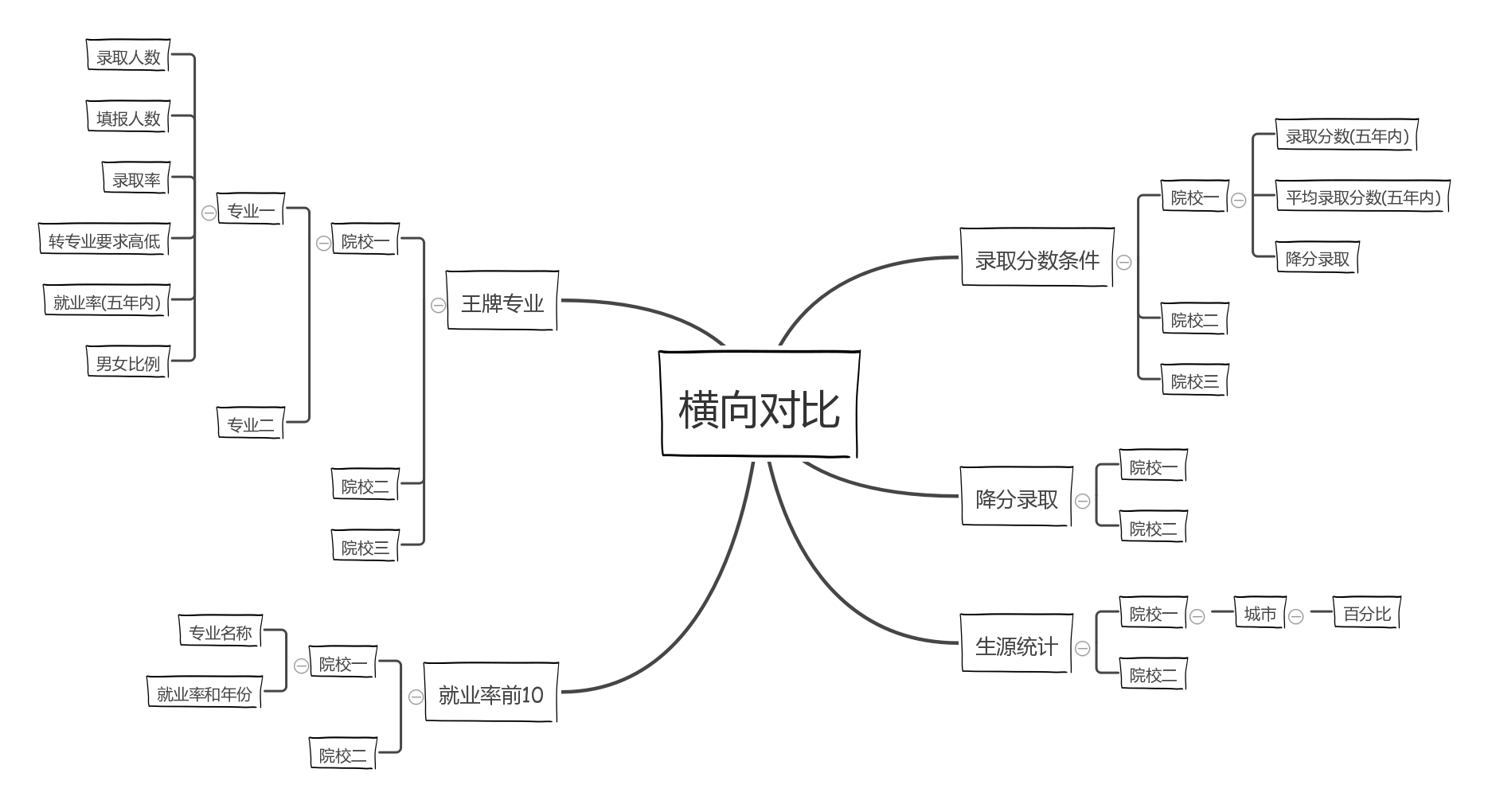
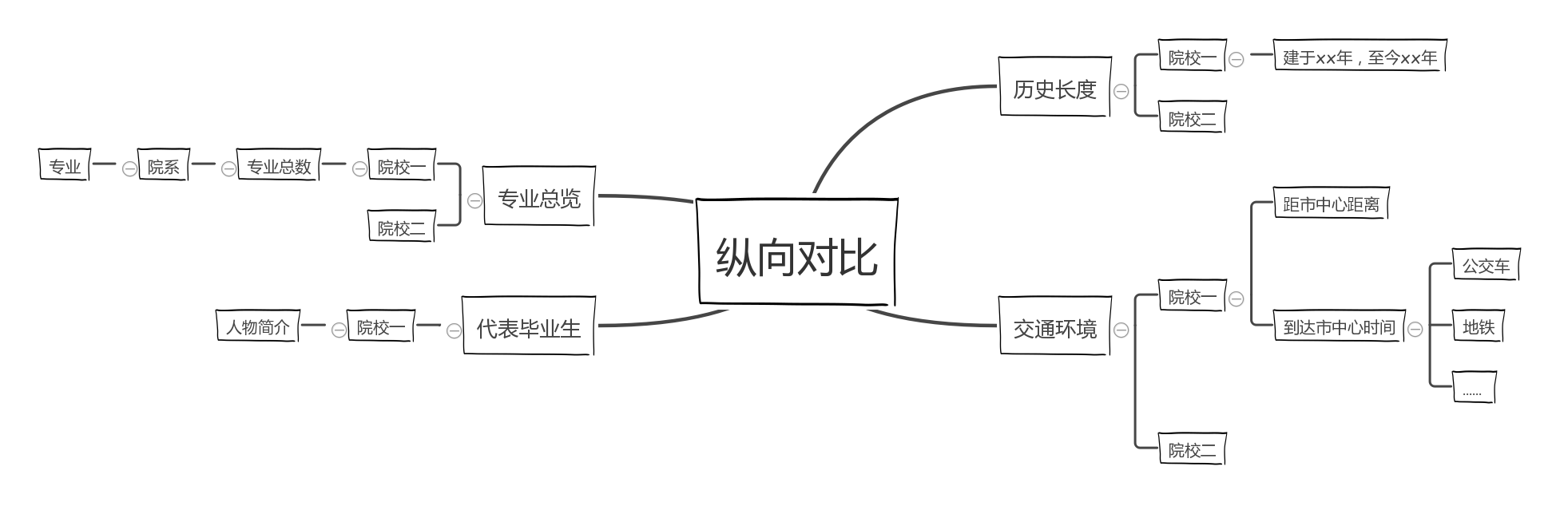


图2.6横向对比图

横向对比与纵向对比的展现方式主要以展开图为主。上图描述了每一层及应包含的标题内容，具体内容可以使列表，条形图，柱状图，饼状图。以最容易表达的方式进行选择。

图2.7纵向对比图

#### 2.2.3交互系统

交互系统是主要与用户进行交互的系统，为用户提供便利的接口方便实现软件的各项功能。交互系统设计的原则是一切从用户角度出发。交互系统提供的服务有：

1. 注册
2. 登录
3. 游客模式
4. 信息录入
5. 获取结果
6. 发帖
7. 评论
8. 收藏
9. 意见反馈

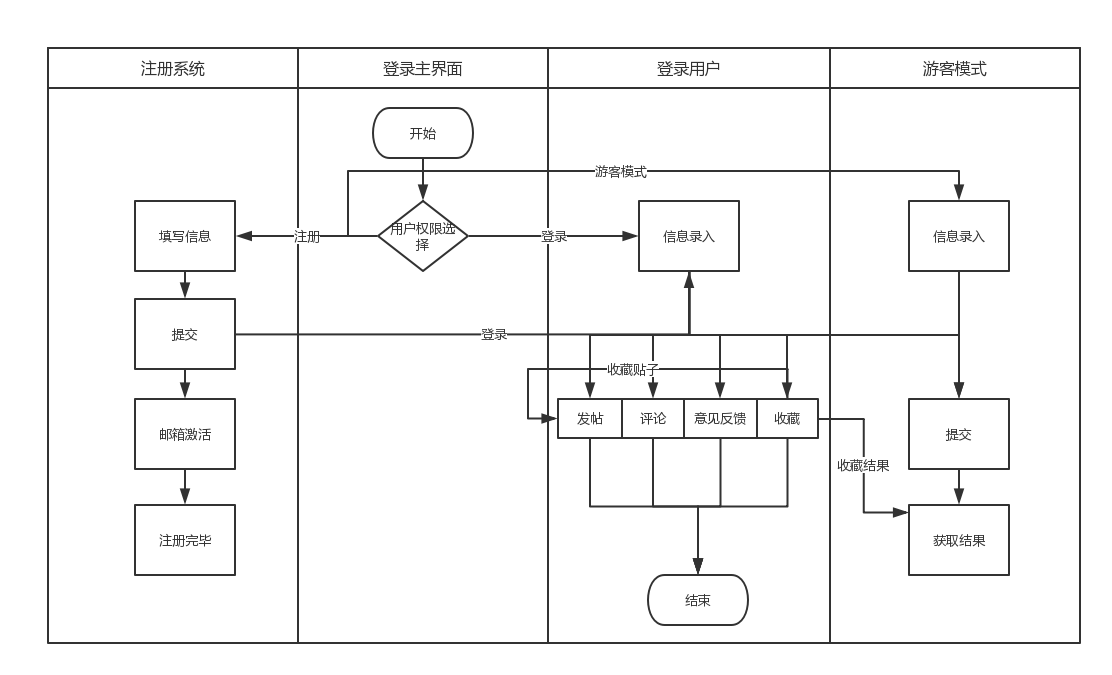
其中仅当用户登录后拥有发帖，评论，意见反馈功能。信息录入即录入自己的分数，专业爱好，地域爱好等信息。发帖可以向软件使用的其他用户提出援助问题，或者发表心得见解以及建议。收藏可以对喜欢的贴或者对推荐结果进行收藏。意见反馈可以向开发者邮箱发送使用者的意见和建议，以改善用户体验。交互系统的流程图如下：

图2.8交互系统流程图

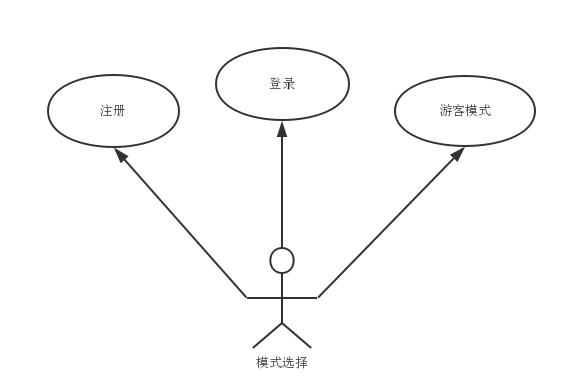
进入初始界面后，要求用户选择浏览模式。

图2.9模式选择用例图

表2.3模式选择用例描述

|  |  |
| --- | --- |
| 注册 | 跳转到填写用户信息页面 |
| 登录 | 弹出输入框，输入用户名密码。或者提供第三方联合登录。微博，微信，QQ |
| 游客模式 | 进入主界面，但不具有评论等功能 |

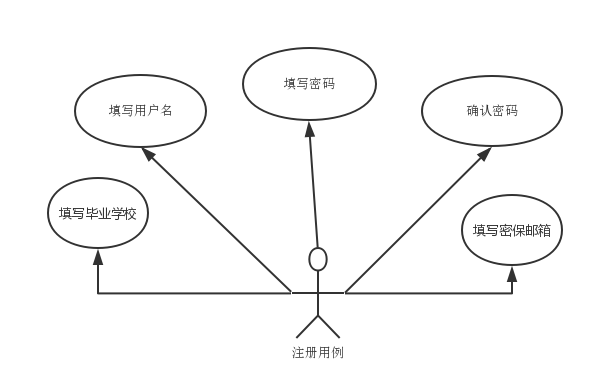


图2.10注册用例

表2.4注册用例描述

|  |  |
| --- | --- |
| 填写用户名 | 输入用户名，限制除英文字母/中文/数字以外的其他字符的输入 |
| 填写密码 | 用户采用弱口令时弹出对话框给予提示 |
| 确认密码 | 在输入一遍密码 |
| 填写毕业学校 | 下拉框 省/市/县/学校进行选择，对未收录的学校提供反馈链接 |
| 填写密保邮箱 | 用于激活账号，更改密码 |

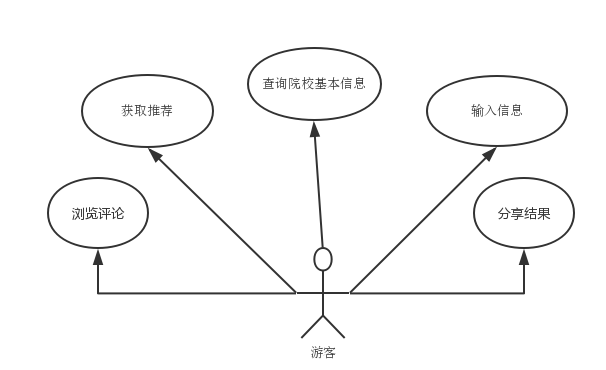


图2.11游客用例图

表2.5游客用例描述

|  |  |
| --- | --- |
| 输入信息 | 输入推荐系统所需信息 |
| 获取推荐 | 获取推荐系统生成的结果 |
| 查询院校基本信息 | 学校的地理位置，建校历史，优秀校友，专业排名，就业情况 |
| 浏览评论 | 浏览注册用户发表文章及文章后其他注册用户的评论 |
| 分享结果 | 将自己的推荐结果分享至QQ好友，微信好友，微博，朋友圈 |

登录用户除了具有游客所拥有的所有功能之外，还具有自己特殊功能，如发表文章等。

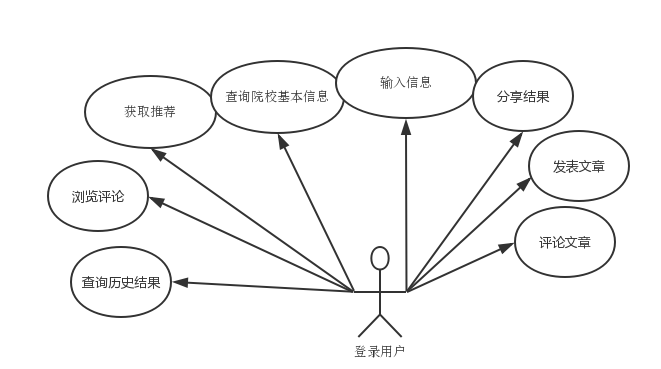


图2.12登录用户用例

表2.6登录用户用例描述

|  |  |
| --- | --- |
| 输入信息 | 输入推荐系统所需信息 |
| 获取推荐 | 获取推荐系统生成的结果 |
| 查询院校基本信息 | 学校的地理位置，建校历史，优秀校友，专业排名，就业情况 |
| 浏览评论 | 浏览注册用户发表文章及文章后其他注册用户的评论 |
| 分享结果 | 将自己的推荐结果分享至QQ好友，微信好友，微博，朋友圈 |
| 发表文章 | 用户具有自己的空间，能够发表文章，且对文章具有修改，删除操作，且支持Markdwon书写 |
| 评论文章 | 在浏览其它用户的文章时可以进行字数为200字以内的评论 |
| 查询历史结果 | 可以获取自己之前多次使用推荐功能输入数据并产生的结果，且可修改数据并重新进行推荐 |

### 2.3对象模型

对象模型是为了方便编码工作者工作提前对系统应该存在的类及类的属性和方法进行粗略总结的一种体现。根据前面所提到的用例，功能需求，需求分析，等描述中可以抽象出一些类，并且在此基础上通过逻辑关系延伸出其他类。

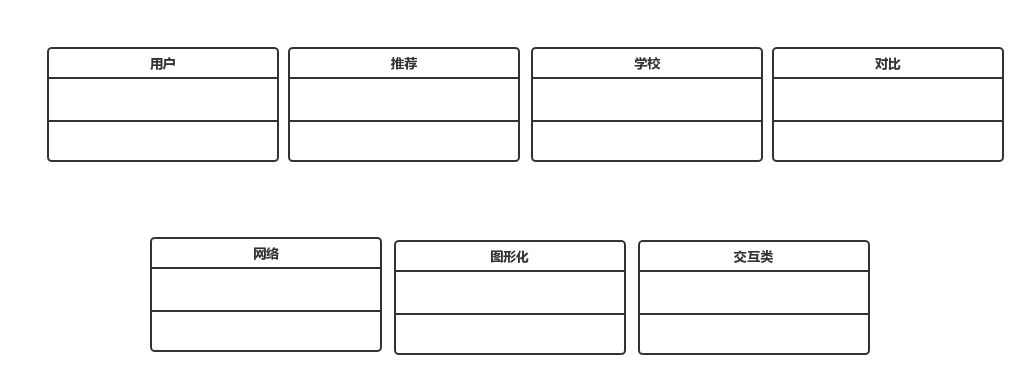


图2.13对象模型

#### 2.3.1用户类

定义的模型都需要通过代码实现，采用面向对象的设计思想，将实体抽象成各种各样类，并定义其属性，能够更有利的完成编码工作。

表2.7用户类

|  |  |  |
| --- | --- | --- |
| 属性 | 用途 | 类型 |
| 用户名 | String |
| 密码 | String |
| 确认密码 | String |
| 游客标识 | Int |
| 邮箱 | String |
| 毕业学校 | String |
| 真实分数 | Int |
| 专业喜好（编号） | Int |
| 地域喜好（编号） | Int |
| ID | Int |
|  | | |
| 方法 | 用途 | 返回值类型 |
| 获取用户名 | String |
| 获取密码 | String |
| 设置ID | Void |
| 方法 | 设置毕业学校 | Void |
| 设置分数 | Void |
| 设置专业（编号） | Void |
| 设置地域（编号） | Void |
| 设置游客标识 | Void |
| 获取确认密码 | Void |
| 设置邮箱 | Void |
| 用户名检查 | void |
| 密码核对 | void |
| 用途 | 返回值 |
| 发送激活邮件 | void |
| 获取重置密码邮件信息 | String |
| 更改密码 | void |

#### 2.3.2推荐类

表2.8推荐类

|  |  |  |
| --- | --- | --- |
| 方法 | 用途 | 返回值 |
| 获取用户分数 | Void |
| 获取用户所选地域（编号） | Void |
| 获取用户所选专业（编号） | Void |
| 推荐 | 待定 |

获取了推荐系统所需要的数据后进行算法推荐。返回值应该是能够进行图形化格式处理的数据格式。这方面了解的内容还不够多，暂时以待定代替。

#### 2.3.3学校类

表2.9学校类

|  |  |  |
| --- | --- | --- |
| 属性 | 用途 | 类型 |
| 学校名称 | String |
| 学校编号 | Int |
|  | | |
| 方法 | 用途 | 返回值 |
| 获取xxx | 待定 |

学校的信息通过与高校的校园网对接，由网络爬虫爬取相关的数据。获取xxx具体应该与学校提供的数据与爬虫爬去的数据为准。

#### 2.3.4对比类

表2.10对比类

|  |  |  |
| --- | --- | --- |
| 方法 | 用途 | 返回值 |
| 获取推荐结果 | Void |
| 生成对比 | 待定 |

对比类是对比系统中的一部分，而不是全部，目前我能够想到的对比类需要实现的只有这两个方法。该类应该是便于扩展的。并且与推荐类类似，有些结果需要生成图像化数据，方便用户直观的浏览出各个结果之间的差异，所以需要将结果返回给图形化类，因此某些返回值依然用待定替代。

#### 2.3.5网络类

网络类主要用来在各个高校的校园网中爬取信息和内容。例如学校人数，各个年级人数，开设什么院系，什么专业，生源分布等等一系列可以公开的信息。即推荐系统所需的第三方信息的信息来源由网络类负责，并通过统一的数据格式发送。

表2.11网络类

|  |  |  |
| --- | --- | --- |
| 方法 | 用途 | 返回值 |
| 获取学生数 | Int |
| 获取学院 | 自定义 |
| 获取专业 | 自定义 |
| 获取生源信息 | 自定义 |
| 获取学校基本信息 | 自定义 |
| 方法 | 获取优秀毕业生 | 自定义 |
| 获取平均录取分 | Int |
| 获取最低录取分 | Int |
| 获取最高录取分 | Int |
| 获取专业录取率 | Int |
| 获取专业就业率 | Int |
| 获取专业考研率 | Int |

因为整个项目的最初目的只是为了给高中毕业生提供一种使用简单，目的明确，能够清晰的了解到自己最关注的问题，所以并没有获取跟此功能没有太大关系的内容。如果后期有相关需求的话，可以获取更多信息，提供更加丰富的功能。

#### 2.3.6图形化类

数据通过推荐系统的处理，将结果发送给图形化类，负责将数据以图形的方式向用户提供更加爱直观的数据分析结果。图形化类产生的数据样式包括饼状图，柱形图，折线图，地图分布图等形式。提供选项让用户选择什么样的形式展现数据。

同样的，对于多种推荐结果用户更倾向于将两组或多组数据进行对比，从而做出自己的决定。那么图形化类也应该提供另一种图形化方式将两组或多组数据做对之后的图形展现出来。基本类的方法，属性如下表：

**·单图形化类**

表2.12单图形化类

|  |  |  |
| --- | --- | --- |
| 方法 | 用途 | 返回值 |
| 获取录取分数数据 | Int |
| 获取填报专业数据 | Void |
| 获取生源地数据 | Void |
| 生成饼状图 | Void |
| 生成扇形图 | Void |
| 生成条形图 | Void |
| 生成折线图 | Void |
| 生成地图分布图 | Void |

·**对比图图形化**

表2.13对比图形化

|  |  |  |
| --- | --- | --- |
| 方法 | 用途 | 返回值 |
| 生成饼状对比图 | Void |
| 生成折线对比图 | Void |
| 生成地图分布对比图 | Void |
| 生成柱形对比图 | Void |
| 生成扇形对比图 | Void |

#### 2.3.7交互类

交互类是与用户接触的界面，为用户提供模式选择（注册用户还是游客）和注册功能。在用户进入主界面后则为用户提供输入数据及产生结果的方法。

**·注册类**

表2.14注册类

|  |  |  |
| --- | --- | --- |
| 方法 | 用途 | 返回值 |
| 获取用户名 | String |
| 获取密码 | String |
| 获取确认密码 | String |
| 密码核对 | Boolean |
| 获取邮件地址 | String |
| 发送激活邮件 | Boolean |

**·界面类**

界面类主要用来展示，不应该具有业务处理功能。下表只列出主要组件。

表2.15界面类

|  |  |  |
| --- | --- | --- |
| 控件 | 类型 | 用途 |
| 输入框 | 输入分数 |
| 选项框 | 选择省/市 |
| 选项框 | 选择院系/专业 |
| 控件 | 按钮 | 清除输入 |
| 按钮 | 提交数据 |

主要功能的组件基本上是上表所述的这些。但界面类应该提供更加丰富的内容以提高用户的使用体验。并且队友注册用户，应有关于自己的页面，可以管理自己的文章，设置其他信息等等。关于界面类的更详细的内容应该在详细设计部分有所体现。

### 2.4数据模型

从互联网中获取的数据需要进行存储，能够更方便的进行其他业务操作。因为获取的数据量很庞大，且类型复杂。因此不适合使用关系型数据库进行存储。转而应该向数据类型灵活的MongoDB或者其他类似的数据库靠拢。因为对其他数据库类型的了解不是很深入，这里暂时可以选用MongoDB。关于MongoDB的简单陈述：

**MongoDB比RDBMS的优势**

* 架构：MongoDB是文档型数据库，其中一个集合保存不同的不同的文件。字段的数量，内容和该文件的大小可以是不同于从一个文件复制到另一个。
* 一个单一的对象，结构相对清晰
* 没有复杂的连接
* 深查询能力。 MongoDB支持动态查询使用基于文档的查询语言，如SQL几乎一样强大的文件
* 调优
* 易于规模化：MongoDB是易于扩展
* 不需要数据库对象的应用程序对象转换/映射
* 使用内部存储器存储（窗口）工作组，从而实现更快的数据存取

**为什么要使用MongoDB**

* JSON风格文件的形式，面向文档存储：数据存储
* 对任何属性可索引
* 复制和高可用性
* 自动分片
* 丰富的查询
* 快速就地更新
* MongoDB的专业技术支持

**应该在哪里使用MongoDB？**

* 大数据
* 内容管理和交付
* 移动和社交基础设施
* 用户数据管理
* 数据平台

### 2.5动态模型

动态模型能够更加清晰的描述类与类之间消息的传递过程与消息交互的先后关系。

#### 2.5.1交互图

交互图又叫时序图，显示对象之间的关系, 强调对象之间消息的时间顺序, 显示对象之间的交互;通过游客模式交互图，注册用户交互图，和对比数据交互同来描述系统主要功能。

**·游客模式交互图**

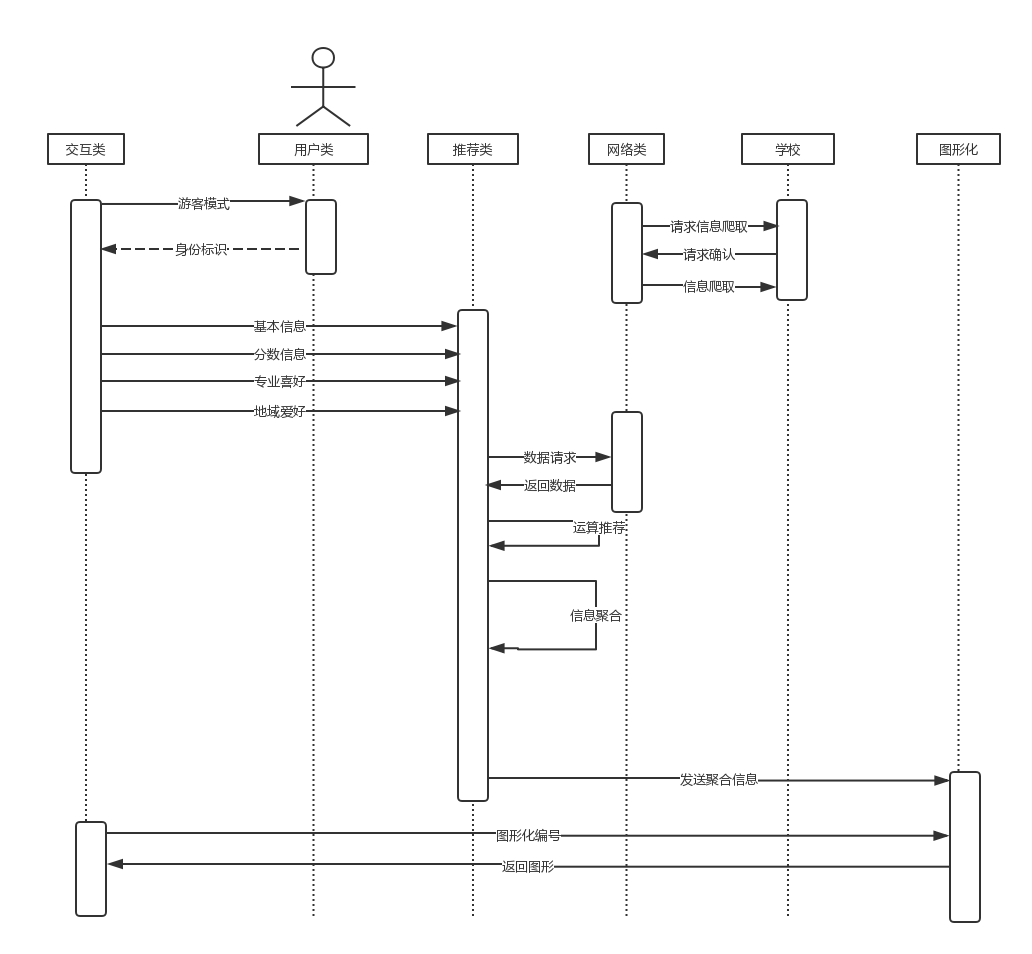


图2.14游客模式时序图

* 1. 用户首先在登录/注册界面选择游客模式；
  2. 通过用户类获取游客标识符，用于唯一区别用户；
  3. 网络类请求各大学校爬取数据；
  4. 学校统一爬取，返回确认信息；
  5. 网络类开始爬取内容；
  6. 用户基本信息，如当前位置等发送给推荐类；
  7. 发送用户填写分数；
  8. 发送用户填写专业；
  9. 获取用户填写地理位置；
  10. 推荐类请求网络类爬取的数据；
  11. 网络类返回内容；
  12. 根据用户信息和推荐算法进行推荐；
  13. 将结果进行格式化处理后发送给图形化进行处理；
  14. 图形化类获取用户选取的图形号；
  15. 根据图形号绘制对应图形并返回；

**·注册用户交互图**

* 1. 用户首先在登录/注册界面选择注册；
  2. 填写用户名；
  3. 填写密码
  4. 输入确认密码
  5. 键入邮箱
  6. 信息核对
  7. 网络类请求各大学校爬取数据；
  8. 学校统一爬取，返回确认信息；
  9. 网络类开始爬取内容；
  10. 存储用户信息
  11. 发送用户基本信息，如当前位置等发送给推荐类；
  12. 发送用户填写分数；
  13. 发送用户填写专业；
  14. 获取用户填写地理位置；
  15. 推荐类请求网络类爬取的数据；
  16. 网络类返回内容；
  17. 根据用户信息和推荐算法进行推荐；
  18. 将结果进行格式化处理后发送给图形化进行处理；
  19. 图形化类获取用户选取的图形号；
  20. 根据图形号绘制对应图形并返回；

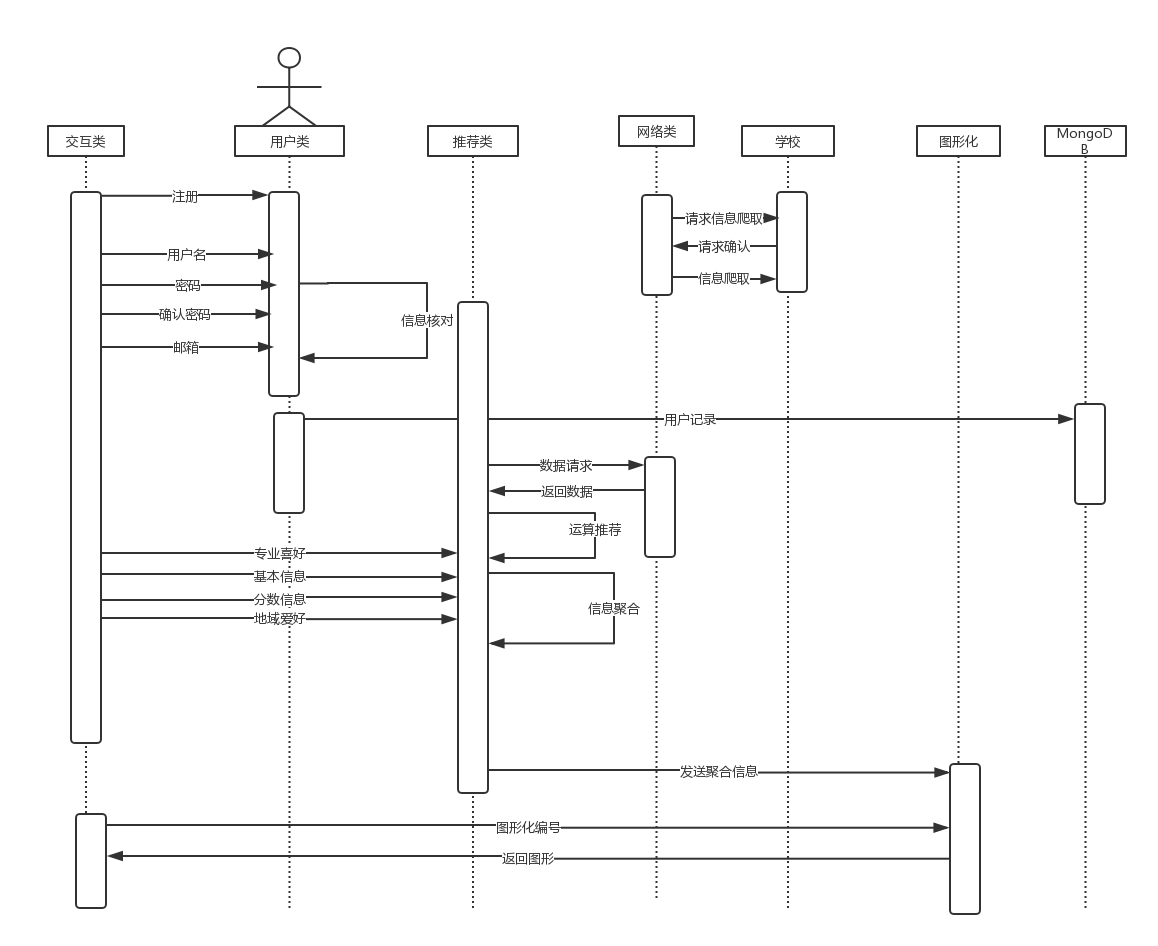


图2.15注册用户交互图

**·对比数据交互图**

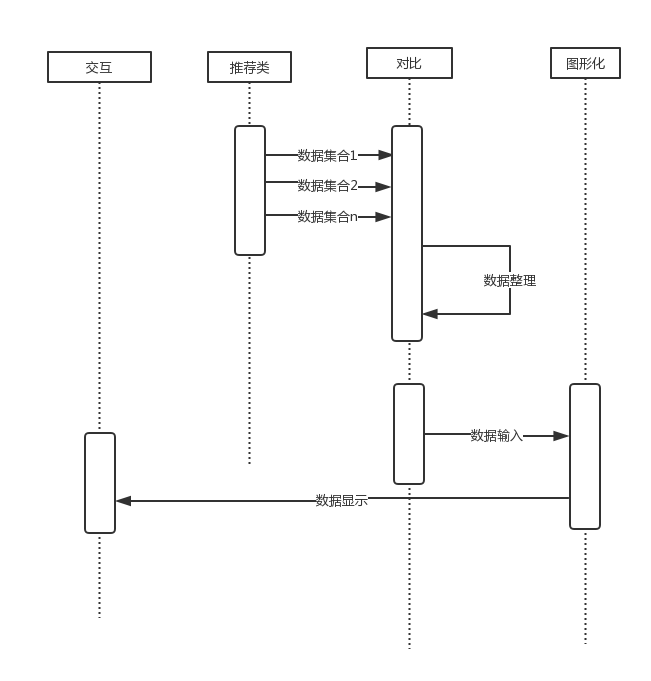


图2.16对比过程时序图

## 3 问题及总结

上述就是关于志愿填报推荐系统的一个简单的设计摘要。描述了该系统应具有的一些主要功能。当然，该系统涉及的数据量及其庞大，并且在处理这种量级的数据时并没有了解到相关的技术。这是这次作业中的一个不足之处，按照老师的建议应该是可以融入大数据的一些技术和云计算的相关知识。

这份设计书只是提供的只是这个志愿填报推荐系统应该具有的一些核心功能，但是需求总是在变化的，用户只想通过最简单的，最快捷的，最廉价的，最方便的途径去获取对自己最有价值的信息。然而在设计软件或系统时，应该提供比用户所能想到的更多的内容，并且以最简单易用的方式呈现。乔布斯说，其实用户自己不知道自己真正想要的是什么。同样的，如果在该系统之仅仅提供其他软件已经存在的成熟的功能，或者说把其他软件上的一部分功能做了整合。这并不会为本系统创造很多价值。所以我把我能够想得到的功能点尽可能多的阐述出来，因而这份报告存在着诸多的不足。

例如，并没有系统的详细设计，包括系统编码，界面设计，系统之间的细节，应该采用的技术点，算法要求等等都没有进行描述。

关于这个系统的平台最初的设想是在Web上以网站的形式呈现，但后来又觉得移动时代手机是学生都配备的设备，但很少有人会通过手机上的浏览器去登录功能性网站，多浏览的是知识性的页面，如百度百科，维基百科之类的。取而代之的是大家更习惯于点开手机上的App去满足一些功能性的需求。在手机App里，又有几乎全部的学生群体在使用腾讯旗下的微信，微信的功能也是越来越丰富，从最开始的聊天交友，慢慢延伸到交易平台，从微信公众平台又推出了一项叫做“小程序”的功能。所以一直没有定位到该去在哪一个平台去试水。老师的建议是从网站做起。但是很突出的问题就是现在学生使用电脑的机会远远比手机，平板设备要少的多。在用户群体上就没有很大的优势。

然后我希望这款软件不仅仅能够成为每当新一届毕业生毕业后能够在填报志愿时获益的一款软件，更能在进入自己满意的学校之后产生别的变化，比如作为新生更想接触在学校时间更长一些的学长学姐们，为自己以后在学校生活中提前做一些准备。同样地，作为老一届的同学们可能希望了解一些新鲜变化。虽然像百度贴吧，QQ群组，微信好友，甚至课程表软件也会提供类似的功能，但是都不能够保证软件使用者的身份信息。多多少少存在一些安全隐患问题。再者，更希望它不仅能够连接本校同学的人脉关系，跟能够跟同省市的其他大学进行信息交流和学术研讨。希望大家能通过它找到一些志同道合的伙伴，不论是一起复习考研，一起开发程序，一起创业都提供一个良好的平台。