|  |  |
| --- | --- |
| 职来网  Jobs Come | 摘要  基于大数据，给你精简的小数据。看到的是：最好的职位，更好的未来！，  华南理工大学软件学院几位本科生 |

目录

[1.总体设计 2](#_Toc11834101)

[1.1概念术语描述 2](#_Toc11834102)

[1.2.基本设计描述 3](#_Toc11834103)

[1.3主要界面流程描述 4](#_Toc11834104)

[1.4模块列表 4](#_Toc11834105)

[2.数据库模块设计： 4](#_Toc11834106)

[2.1数据库环境说明： 4](#_Toc11834107)

[2.2数据库命名规则 4](#_Toc11834108)

[2.3逻辑设计： 5](#_Toc11834109)

[2.4物理设计 5](#_Toc11834110)

[2.5安全性设计 7](#_Toc11834111)

[2.6数据库管理与维护说明 7](#_Toc11834112)

[3. 其他模块设计： 7](#_Toc11834113)

[爬虫与清洗模块 7](#_Toc11834114)

[可视化模块： 8](#_Toc11834115)

[4.程序设计 9](#_Toc11834116)

[4.1数据库模块设计 9](#_Toc11834117)

[4.2可视化模块设计 11](#_Toc11834118)

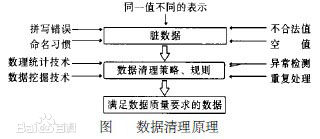
[4.3爬虫与清洗模块设计 11](#_Toc11834119)

[4.开发规划 11](#_Toc11834120)

## 1.总体设计

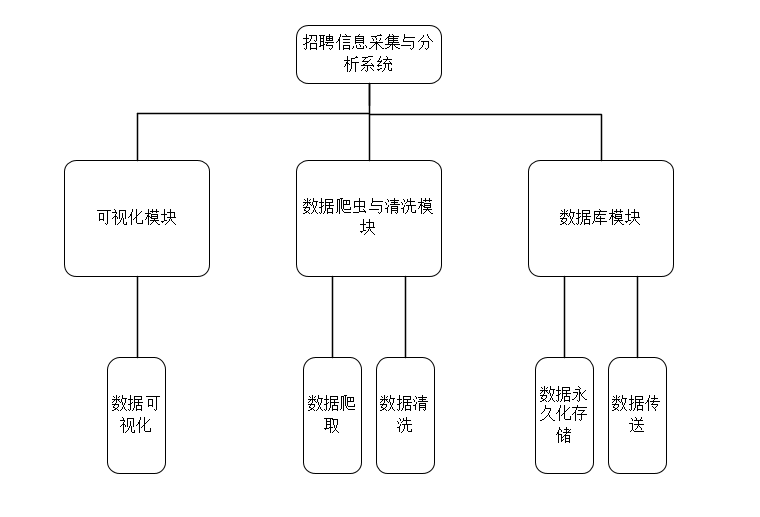
### 1.1概念术语描述

网络爬虫：是一个自动提取网页的程序，它为搜索引擎从万维网上下载网页，是搜索引擎的重要组成。传统爬虫从一个或若干初始网页的URL开始，获得初始网页上的URL，在抓取网页的过程中，不断从当前页面上抽取新的URL放入队列,直到满足系统的一定停止条件。然后，它将根据一定的搜索策略从队列中选择下一步要抓取的网页URL，并重复上述过程，直到达到系统的某一条件时停止。另外，所有被爬虫抓取的网页将会被系统存贮，进行一定的分析、过滤，并建立索引，以便之后的查询和检索

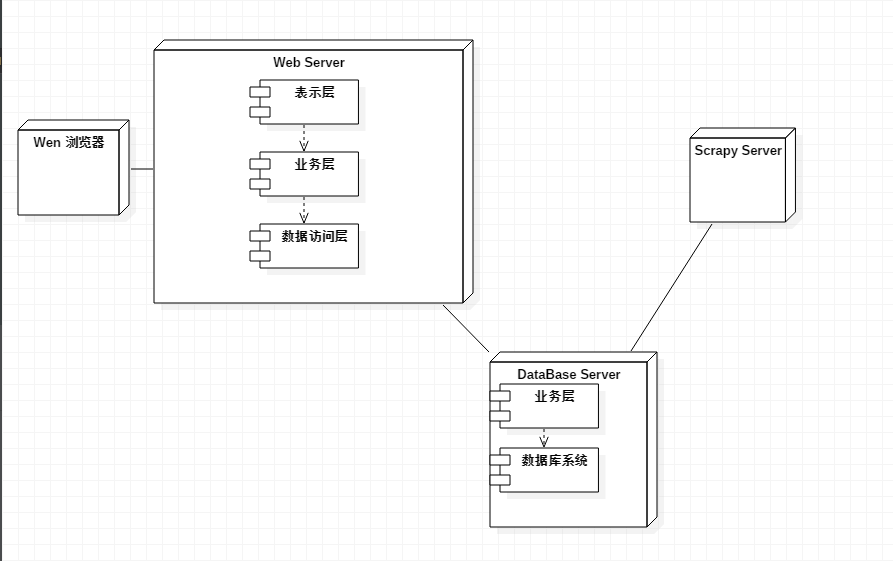


### 1.2.基本设计描述

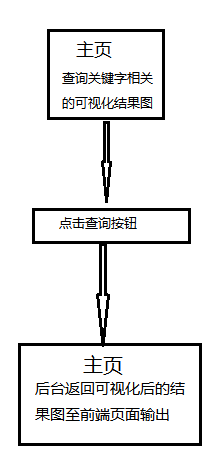
#### 1.2.1系统总体逻辑结构图



#### 1.2.2系统部署结构图



### 1.3主要界面流程描述



### 1.4模块列表

|  |  |  |
| --- | --- | --- |
| 模块名称 | 功能 | 备注 |
| ScapyandClean | 网络爬虫与清洗 |  |
| DataStorage | 数据存储 |  |
| Visualization | 数据可视化 |  |

## 2.数据库模块设计：

### 2.1数据库环境说明：

Mysql 32位数据库

### 2.2数据库命名规则

（1）完整并且清楚的说明本数据库的命名规则。

（2）如果本数据库的命名规则与机构的标准不完全一致的话，请作出解释。

1. 所有命名采用26个字母（区分大小写），数字0~9以及下划线\_，共63个字符。不能出现其他字符。

2. 不要使用计算机术语，尽量使用面向公司业务的术语。

3. 实体 ( 表、字段 ) 应该根据他们所代表的实体意义来命名。

3.1 表命名：tbl\_TableName

3.2 字段命名: 表名（不包括前缀）+ ”\_” +属性名称

4. 实体名应反映实体真实意义，即使采用缩写，其表示意义也应尽量清楚明白。

5. 列命名字母全大写。

6. 不以数字和下划线开头。

### 2.3逻辑设计：

需求说明：爬取招聘信息，可以展示相关数据分析。

#### 2.3.1数据爬取

爬取数据，生成职位信息表。

#### 2.3.2数据可视化

从职位信息表进行提取，根据不同职位分成多个职位信息表，进行可视化。

### 2.4物理设计

#### 2.4.1表汇总

|  |  |
| --- | --- |
| 表名 | 功能说明 |
| tbl\_AllPositionInfo | 全部职位信息 |
| tbl\_CompanyInfo | 公司信息 |
| tbl\_PositionInfo | 简化的职位信息（根据ALLPOSITIONINFO提取各类型的职位信息） |
| tbl\_PositionType | 职位类型 |

#### 2.4.2 tbl\_AllPositionInfo

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | | tbl\_AllPositionInfo | | | | |
| 列名 | 数据类型 | | 数据长度 | 空/非空 | 约束条件 | 说明 |
| ID | INT | | 4 | N | Primary | 表记录ID |
| COMPANY | VARCHAR | | 60 | N |  | 公司名 |
| WORKPLACEPROVINCE | VARCHAR | | 30 | N |  | 工作地点-省份 |
| WORKPLACECITY | VARCHAR | | 30 | N |  | 工作地点-城市 |
| DIPLOMA | VARCHAR | | 16 | N |  | 文凭要求 |
| NUMBER | SMALLINT | | 2 | N |  | 招聘人数 |
| DATE | INT | | 4 | N |  | 发布时间 |
| SUPPLEMENT | VARCHAR | | 1000 | Y |  | 职位信息 |
| POSITIONTYPE | VARCHAR | | 30 | N |  | 职位名称 |
| KEYWORD | VARCHAR | | 50 | Y |  | 关键字 |
| EXPERIENCE | INT | | 4 | N |  | 工作经验 |
| SALARY | INT | | 4 | N |  | 薪水 |
| 补充说明 | 【传送数据规定】  （1）自增id无需传送（2）公司名用string（3）工作地点使用列表（第一个为省份，第二个为城市），统一带省和市后缀字，北京市等直辖市格式为（北京省-北京市）澳门香港台湾为（澳门省-澳门市）（4）文凭要求传过来的关键字，（大专以下，大专，本科，硕士，博士），（5）招聘人数使用int，如果是“若干”等字体，使用数字0.（6）data使用int，存储unix时间戳（unix时间戳是从1970年1月1日(UTC/GMT的午夜)开始所经过的秒数，），都是有函数直接转换的。（7）职位信息，string即可,如果没有的话就用“无”代替。（8）职位名称，string（9）关键字，用string，多个关键字中间用-分隔（如 本科-工程师-男）。（10）工作经验，int，以年为单位，如果数据是3-5年，那就取区间平均数，即四年；如果是实习，用数字0代替，如果没有要求，用-1代替。  【数据库说明】（1）用id建立主索引testtbl\_AllPositionInfo  创建表格语句：CREATE TABLE testtbl\_AllPositionInfo(  ID INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,/\*主键字段，数值自增，唯一标识，非空\*/  COMPONY VARCHAR(60) NOT NULL,  WORKPLACEPROVINCE VARCHAR(30) NOT NULL,  WORKPLACECITY VARCHAR(30) NOT NULL,  DIPLOMA VARCHAR(16) NOT NULL,  NUMBER SMALLINT NOT NULL,  DATE INT NOT NULL,  SUPPLEMENT VARCHAR(1000) NOT NULL,  POSITIONTYPE VARCHAR(30) NOT NULL,  KEYWORD VARCHAR(50) NOT NULL,  EXPERIENCE INT NOT NULL,  SALARY INT NOT NULL  ) | | | | | |

#### 2.4.3 tbl\_CompanyInfo

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | | tbl\_CompanyInfo | | | | |
| 列名 | 数据类型 | | 数据长度 | 空/非空 | 约束条件 | 说明 |
| COMPANYNAME | VARCHAR | | 60 | N | primary key | 公司名 |
| COMPANYTYPE | VARCHAR | | 30 | N |  | 公司类型 |
| COMPANYSCALE | INT | | 4 | N |  | 公司规模 |
| COMPANYWELFARE | TEXT | | 64kb | Y |  | 公司福利 |
| 补充说明 | 用COMPANYNAME建立索引 | | | | | |

#### 2.4.4 tbl\_PositionInfo

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | | tbl\_PositionInfo | | | | |
| 列名 | 数据类型 | | 数据长度 | 空/非空 | 约束条件 | 说明 |
| ID | INT | | 4 | N | Primary | 表记录ID |
| COMPONY | VARCHAR | | 60 | N |  | 公司名 |
| WORKPLACE | VARCHAR | | 30 | N |  | 工作地点 |
| DIPLOMA | VARCHAR | | 16 | N |  | 文凭要求 |
| NUMBER | SMALLINT | | 2 | N |  | 招聘人数 |
| DATE | INT | | 4 | N |  | 发布时间 |
| EXPERIENCE | INT | | 4 | N |  | 工作经验 |
| 补充说明 | 用id建立主索引，DATE建立辅助索引 | | | | | |

#### 2.4.5 tbl\_CreatedTable

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | | tbl\_CreatedTable | | | | |
| 列名 | 数据类型 | | 数据长度 | 空/非空 | 约束条件 | 说明 |
| ID | INT | | 4 | N | primary key | 表记录ID |
| TABLENAME | VARCHAR | | 60 | N |  | 数据创建的表，不包括2.4.2-2.4.5这四个表 |
| 补充说明 | 用ID建立索引  建表语句：  CREATE TABLE testtbl\_CreatedTable(  ID INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,/\*主键字段，数值自增，唯一标识，非空\*/  TABLENAME VARCHAR(60) NOT NULL); | | | | | |

#### **2.4.6 tbl\_CompanyRecuitInfo**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | | False\_dyanamic\_CompanyRecuitInfo | | | | |
| 列名 | 数据类型 | | 数据长度 | 空/非空 | 约束条件 | 说明 |
| WORKPLACEPROVINCE | VARCHAR | | 30 | N |  | 工作地点-省份 |
| WORKPLACECITY | VARCHAR | | 30 | N |  | 工作地点-城市 |
| DIPLOMA | VARCHAR | | 16 | N |  | 文凭要求 |
| NUMBER | SMALLINT | | 2 | N |  | 招聘人数 |
| DATE | INT | | 4 | N |  | 发布时间 |
| POSITIONTYPE | VARCHAR | | 30 | N |  | 职位名称 |
| KEYWORD | VARCHAR | | 50 | N |  | 关键字 |
| EXPERIENCE | INT | | 4 | N |  | 工作经验 |
| SALARY | INT | | 4 | N |  | 薪水 |
| 补充说明 | 注意：并不存在这样的表格，而是动态建表（表名为公司名）  数据库说明】（1）用id建立主索引testtbl\_AllPositionInfo  创建表格语句：CREATE TABLE 公司名(  ID INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,/\*主键字段，数值自增，唯一标识，非空\*/  WORKPLACEPROVINCE VARCHAR(30) NOT NULL,  WORKPLACECITY VARCHAR(30) NOT NULL,  DIPLOMA VARCHAR(16) NOT NULL,  NUMBER SMALLINT NOT NULL,  DATE INT NOT NULL,  POSITIONTYPE VARCHAR(30) NOT NULL,  KEYWORD VARCHAR(50) NOT NULL,  EXPERIENCE INT NOT NULL,  SALARY INT NOT NULL,  ) | | | | | |

### 2.5安全性设计

#### 2.5.1角色与权限

|  |  |  |
| --- | --- | --- |
| 角色 | 可以访问的表与列 | 操作权限 |
| 中间数据处理存储 | tbl\_AllPositionInfo  tbl\_CompanyInfo  tbl\_PositionInfo  tbl\_PositionType | 读/写 |
| 数据可视化模块 | tbl\_PositionInfo  tbl\_PositionType | 只读 |

### 2.6数据库管理与维护说明

数据库，日志实时更新，管理人员定期检查数据库查看日志

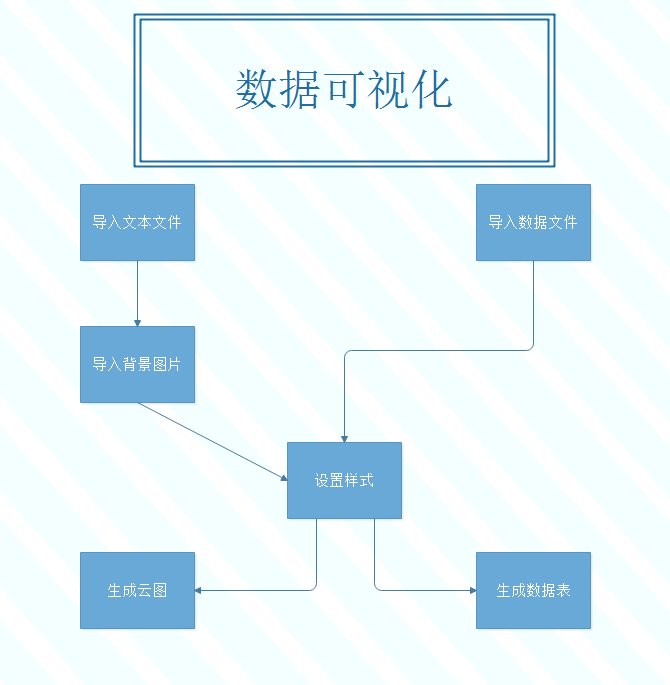
## 3. 其他模块设计：

### 爬虫与清洗模块

采用scrapy框架对选定的招聘网页的软件行业职位数据进行爬取，页面内容提取手段采用xpath解析，爬取列表页并且对列表页进行处理分裂出详情页，然后下载详情页进行处理提取详情页数据。对页面动态爬取，使得每次爬取可以提取到最新更新的信息，对于一些脏数据进行过滤，并对结果进行整理，规范化和持久化，最终输出到数据库进行保存。由于网页会经常更新，所以爬虫需要定时连续抓取，获取实时而不是过期信息。爬虫应遵循抓取协议，为了增加可扩展性去支持新的数据格式，爬虫尽量设计成模块化的形式。

难点：（1）职位信息清洗，做归类；为可视化做准备；（2）重复数据的剔除，同家公司在不同招聘网站同时发布招聘信息，应做剔除。

### 可视化模块：



## 4.程序设计

### 4.1数据库模块设计

#### 4.1.1数据库插入设计说明

##### 4.1.1.1算法及流程



##### 4.1.1.2数据结构

#### 4.1.2数据库可视化模块数据更新

##### 4.1.2.1算法及流程



##### 4.1.2.2数据结构

#### **4.1.3 数据库去重逻辑**

##### 4.1.3.1算法及流程



##### **4.1.3.2数据结构**

### 4.2可视化模块设计

可视化内容：（1）相关岗位近几年平均工资——反映趋势；（2）招聘人数趋势；（3）不同城市工资对比；（4）不同工作经验（包括实习）的需求以及工资等

#### 4.2.1 可视化选项

##### 4.2.1.1某岗位近几月的招聘人数趋势

逻辑：

1. 计算所需月第一天时间戳begintime，最后一天时间戳 endtime；
2. Select sum(NUMBER) from testtbl\_AllPositionInfo where date between $begintime and $endtime and positiontype like’%$(the input of user)%’

说明：将 $begintime 和 $endtime替换；并将$(the input of user) 替换成用户的输入。

Example:Select sum(NUMBER) from testtbl\_AllPositionInfo where date between 1561000000 and 1562000000 and positiontype like '%java%';

##### 4.2.1.2某岗位（近一月）不同城市工资对比

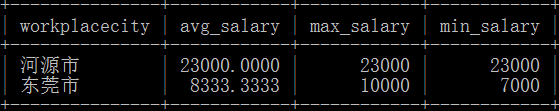
1. 计算所需月第一天时间戳begintime，最后一天时间戳 endtime；
2. sql 语句：

Select workplacecity, avg(salary) as avg\_salary, max(salary) as max\_salary,min(salary) as min\_salary from testtbl\_AllPositionInfo where date between $begintime and $endtime and salary>0 and positiontype like '%$(the input of user)%' group by workplacecity order by avg\_salary desc;

说明：查询结果根据平均工资大小降序排列，并且并且剔除工资面议等情况的数据（也就是没有具体工资额数据），在业务层对用户进行说明。

Example:

Select workplacecity, avg(salary) as avg\_salary, max(salary) as max\_salary,min(salary) as min\_salary from testtbl\_AllPositionInfo where date between 1561000000 and 1562000000 and salary >0 and positiontype like '%java%' group by workplacecity order by avg\_salary desc;



##### **4.2.1.3某岗位近几月的平均工资**

1. 计算所需月第一天时间戳begintime，最后一天时间戳 endtime；
2. sql 语句：

Select avg(salary) as avg\_salary from testtbl\_AllPositionInfo where date between $begintime and $endtime and salary >0 and positiontype like '%$(the input of user)%' ;

说明：查询结果为时间戳之间的平均工资，并且剔除了没有说明具体工资的岗位，在业务层对用户进行说明。

Example:

Select avg(salary) as avg\_salary from testtbl\_AllPositionInfo where date between 1561000000 and 1562000000 and salary >0 and positiontype like '%工程师%' ;

##### **4.2.1.4 某岗位近几个月招聘人数不同城市需求量对比**

逻辑：

1. 计算所需月第一天时间戳begintime，最后一天时间戳 endtime；
2. Select workplacecity,sum(NUMBER) as total\_number from testtbl\_AllPositionInfo where date between $begintime and $endtime and positiontype like’%$(the input of user)%’ group by workplacecity order by total\_numberdesc;

说明：将 $begintime 和 $endtime替换；并将$(the input of user) 替换成用户的输入。

Example:Select workplacecity,sum(NUMBER)as total\_number from testtbl\_AllPositionInfo where date between 1561000000 and 1562000000 and positiontype like '%工程师%' group by workplacecity order by total\_number desc;

### 4.3爬虫与清洗模块设计