

计算机专业词典系统的设计与开发

数学与信息学院 计算机科学与技术专业
105032016063 雷梦豪 指导教师：林岭

【摘要】计算机学科词典是一款能够使用户高效的查找计算机行业的专业词汇、缩写等信息的系统。本词典以 Java Web 为基础，通过 Python 实现爬虫以此获取到基础数据，从专业网站爬取获得的数据，让数据的可信度大大提高，同时也节约了通过人力来录入词典信息的大量时间，也在极大程度上减轻了开发的压力。本篇论文会从开发者的角度，叙述网站开发的各个重要步骤。

【重点】计算机专业缩写查询；Java Web；Python

目录

1. 引言	4
1.1 词典开发背景	4
1.2 设计期望	4
1.3 词典价值	4
1.4 开发基础	4
2. 系统需求	5
2.1 需求分析	5
2.2 用例图	5
2.2.1 用户用例图	5
2.2.2 管理员用例图	6
2.2.3 用例规约	6
2.3 功能需求	9
2.3.1 用户端功能	10
2.3.2 管理员端功能	11
2.3.3 获取词典数据功能	11
2.3 E-R 图	11
3. 词典设计	12
3.1 总体设计	12
3.1.1 词典分析	12
3.1.2 用户操作流程	12
3.1.3 管理员操作流程	14
3.1.4 词典设计	14
3.2 词典详细设计	15
3.2.1 表	15
3.2.2 用户功能	17
3.2.3 词汇功能	18
3.2.4 反馈功能	19
3.2.5 爬虫	20
4. 系统实现	20
4.1 开发环境	20
4.2 获取词典数据	20
4.2.1 获取词汇基本信息	20
4.2.2 通过 specificmeaning 爬取 lemmasummary	21
4.2.3 获取 Priority	23
4.2.4 将数据存进词典	24
4.3 词典功能	25
4.3.1 查询功能	25
4.3.2 登录功能	25
4.3.3 新增词条功能	26
4.3.4 查无反馈功能	26
4.3.5 处理反馈功能	27
4.3.6 维护词汇表功能	28
5. 系统测试	29
5.1 测试目标	29
5.2 测试类型	29

5.3 测试对象.....	30
5.4 测试环境.....	30
5.5 测试用例.....	30
5.5.1 词典首页功能.....	30
5.5.2 词典登录功能.....	30
5.5.3 新增词汇功能.....	31
5.5.4 查无反馈功能.....	32
5.5.5 处理反馈功能.....	32
5.5.6 维护词汇表功能.....	33
5.5.7 修改密码功能.....	33
5.6 结论.....	34
6. 系统评价.....	34
6.1 词典优势.....	34
6.2 词典不足.....	34
7. 总结.....	34
8. 致谢.....	34

1. 引言

1.1 词典开发背景

目前的高新产业可以说是蓬勃发展, 生机盎然。在当今时代, 计算机专业已然成为当今时代发展的重要推动力之一。计算机的应用早已深深融入到各个领域之中, 人们应用计算机的频率也是日益增加。问题是在学习以及使用的过程中, 会遇到一些让你百思不得其解的计算机专业词汇缩写^[1]。举个例子, 计算机词汇在计算机这一学科范畴内有其特定的含义, 不能从字面直接翻译, 而其真实意义可能与生活中这个单词的含义大相径庭。更不用说一些缩写, 即使知道词汇的表面意思, 也不能更好的理解词汇的内容意义。相对于一般人, 从事计算机的人有更大的概率遇到陌生或者不熟的计算机缩写, 因此也更容易因为计算机专业词汇而感到困扰。只有更好的了解学习了这些词汇, 才可以更高效地学习计算机的更多知识。要加深人们对计算机词汇的理解, 让大家可以更加便捷地学习计算机以及加深专业词汇的印象, 需要开发一个计算机词典系统让用户可以更加便利的进行词汇查询, 可以大大的节约使用者的时间, 提高学习和工作的效率。

需要应用计算机的地方日益增加, 同时从事计算机方面的人数也在不断上涨, 然而现有的可供查找计算机领域英语缩写的软件几乎没有, 只有一系列的普通词典软件, 这些产品仅能告诉我们单词的一般含义, 没办法给出专业名词的专业翻译。综上所述, 开发一个计算机词典系统来方便广大的计算机使用者的工作与学习确实迫在眉睫并且产品的未来非常光明。

1.2 设计期望

要开发出一款计算机词典系统, 让广大的计算机用户在接触那些无法直译或者无法理解的计算机词汇时, 可以在第一时间获得它的更多信息, 能更加便捷地学习到该词汇的词义, 通过这种方式可以让用户更加深刻的理解词汇同时为用户提供便利。本系统不是一款晦涩难懂的系统, 其目的是让用户使用起来方便快捷, 一目了然。此外, 用户随时可以通过反馈系统向管理员反馈系统的问题或者将新词汇加入本系统, 让广大的用户都参与到本产品的创作中来, 以此提高本产品的与用户的交互, 同时对系统数据的遗漏进行补充。

1.3 词典价值

本系统可让用户更便捷地查询计算机专有名词的详细信息, 通过本系统用户可以查询到缩写短语的完整释义, 包括词汇的中文意义以及所查单词的具体内容等, 让用户可以更方便的使用和了解计算机, 在学习过程中接触计算机词汇时可以及时加深了解。本系统能更好的提高对计算机的兴趣, 加速用户学习计算机的效率, 同时潜移默化地提升用户的计算机英语阅读能力^[2]。

1.4 开发基础

Python 是一种很流行的开源编程语言, 可以在各种领域中用于编写独立的程序和脚本^[3]。Python 语言是一门开发效率高的脚本语言^[4]。Python 语言具备功能完备的标准库, 语法简单, 开源实现, 语言层次高, 只关心求解的问题, 体现了计算机解决问题的计算机思维^[5]。正所谓尺有所短, 寸有所长, 不存在最好的语言。Python 在有着巨大优点的同时也存在一些瑕疵, 其中影响最直接的问题就是执行效率相对较低, Python 和 C 语言存在很大区别, C 语言只有在程序编译好之后才会运行程序, 而 Python 则是在运行时才会进行编译, 并且每执行一次都要进行编译^[6]。Python 语言也有运行速度较慢、缩进格式要求严格等缺点^[7]。

用 Java 技术来解决相关 Web 互联网领域的技术统称 Java Web 技术^[8]。Java Web 的运用范围非常广, 目前绝大部分网站的搭建都是通过 Java Web 来实现的。通过 Java Web 来制作网站有非常多优势。首先, Java Web 的普适性很强, 基本上市场主流的浏览器都适用; 此外, Java 由 Sun 公司开发, 属于开源程序语言, 所以其他开发者运用 Java 做开发不需要花费大量代价, 极大的减少了不必要的支出^[9]。有利必有弊, Java Web 的上手难度相对较高, 对于开发人员的要求相应的也更高^[10]。

2. 系统需求

2.1 需求分析

系统需要有两个部分的功能，其一是词典系统的自身功能，包括词条搜索，用户密码修改，用户进行反馈等等。此外就是获取词汇的爬虫功能，这一功能主要贡献就是获取数据资源，给词典添加基础数据。

搜索计算机词汇以及缩写等属于系统的自身功能，用户键入所查词汇的各类信息均可得到对应的完整的词汇描述，以及它的意义等，此功能不需要登录即可使用，只需要进入网页，进行简单的搜索即可。此外，因为有了反馈等功能，才可以让本系统有机会得到进一步的改进，让用户可以很方便地加入到词典的修订中来，同时要求管理员对使用者的反馈做出处理，完成系统的改进，其中具备如下几方面功能，报错反馈，也就是用户发现所查词条地具体说明有错误，即可进行报错；还需要查无反馈，当用户发现某词条在本词典中并没有相应的说明，那么用户即可通过此功能将信息发送至管理员，后续管理员再对该词条进行添加处理；用户还应该可以自行添加词条，有的用户愿意参与到词典的丰富当中来，所以说这个功能非常关键，用户添加词条，管理员进行检查，无误则加入词典之中，有误则再进行编辑或者移除处理。以上的反馈部分必须要用户登录才可以进行，所以我们需要检测用户是否登入，这一点相当关键，只有进行了登入的用户才可以反馈，防止垃圾信息过多，给管理员造成不必要的困扰。用户可以通过首页单击个人信息进行修改个人密码，这是每个系统都应该具备的基本功能。

再有就是管理员端需要具备的功能，具体的包括对词典中数据进行维护，包括 CURD 这种相对基础的操作，比如编辑词条，或者移除词条这一类的行为，并且词汇表在页面中应该通过表格形式展示出来，这样相对来说会更加美观；处理用户的反馈也是管理员端应该具备的功能，通过简单的操作即可对用户的反馈进行编辑、通过、移除这样的操作，不应该生涩难懂，要清晰明了，让管理员一目了然，自如的对系统进行管理和维护。

爬虫功能顾名思义就是从网络把的计算机相关缩写等信息爬取下来，然后再通过获取的信息再从相对专业的网站上进一步爬取对该词汇的描述，这种方法可以有效的准确的获取我们需要的词汇内容吗，同时节约了大量人力物力，摆脱了传统的手工输入这种效率不高并且错误率不低的模式。

2.2 用例图

2.2.1 用户用例图

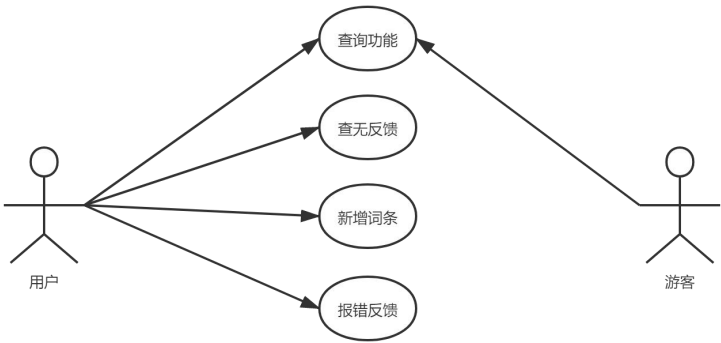


图 2-1 用户用例图

用户端需求如下：

- (1) 游客进入词典只能使用搜索功能查找词条，直接在输入框输入，点击查询即可；
- (2) 用户在使用时，若发现查找到的信息有谬误，点击页面的报错即可进行报错，将错误信息发送给管理员端，等待管理员处理；

(3) 用户能录入新词汇的信息加入到词典当中,需要信息完整并且要提交给管理员进行处理后才可加入词典;

(4) 用户在输入词汇时,若没有搜索到想要的结果,点击反馈,可以提交遗漏的信息至管理员端,等待管理员处理。

2.2.2 管理员用例图

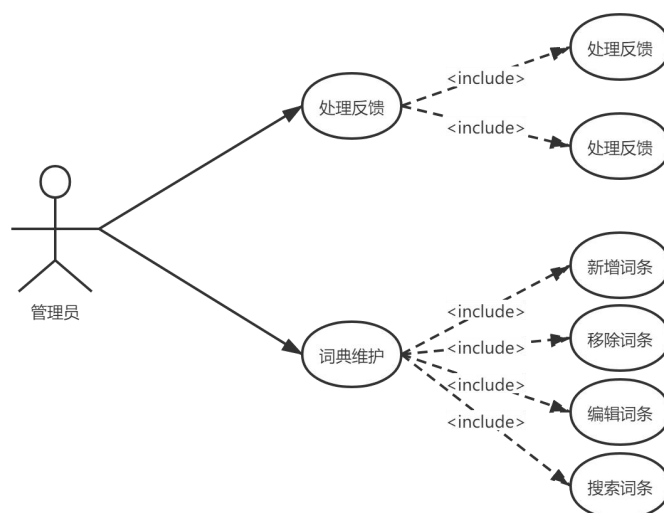


图 2-2 管理员用例图

管理员端需求如下:

- (1) 对使用者提交的信息进行处理,人工选择将数据加入数据库或是移除该反馈信息;
- (2) 能对数据库进行维护,也就是说可以在管理端查找、修改、删除、增添数据库数据。

2.2.3 用例规约

登录词典用例规约

用例编号	CD-1
用例名称	登陆词典
用例描述	词典验证用户身份合法后允许登录系统
参与者	管理员、用户
前置条件	进入登录页面
后置条件	参与者登陆成功,进入主页面
涉众利益	1、用户可以正常使用登陆后才可以使用的反馈报错等功能 2、管理员可以正常进行词典维护和反馈信息处理
基本路径	1、打开系统 2、进入登陆页面 3、选择参与者身份并键入用户名密码 4、参与者提出登录请求 5、系统检验参与者信息充分与否 6、系统检查参与者信息是否正确 7、参与者登陆成功,进入系统
拓展点	1、参与者信息不充分,返回登录界面

	2、参与者信息不正确，返回登陆界面
非功能需求	页面简洁精美，不需要过多华而不实的东西

首页功能用例规约

用例编号	CD-2
用例名称	首页功能
用例描述	参与者通过首页进行查询和报错
参与者	用户、游客
前置条件	进入查询页面
后置条件	参与者成功查询
涉众利益	参与者能从本词典中获取自己需要的词条信息
基本路径	1、打开系统 2、参与者输入关键字 3、参与者单击查询 4、显示结果 5、参与者点击报错 6、检查用户是否处于登录状态 7、参与者填写报错信息 8、检查信息完整性 9、参与者点击提交 10、提交成功
拓展点	1、发现参与者未登录，返回登录界面 2、参与者报错信息填写不完整，提示填写完整
非功能需求	实现模糊查询并且展示界面不会引起不适

新增词汇用例规约

用例编号	CD-3
用例名称	反馈信息
用例描述	参与者进行新增词汇
参与者	用户
前置条件	进入首页
后置条件	参与者新增词汇信息提交成功
涉众利益	1、用户可以实现新增词汇 2、游客不能新增词汇 3、管理员可以管理用户提交的信息
基本路径	1、进入系统 2、参与者登陆系统 3、点击新增词汇 4、检验参与者是否是已登录状态 5、参与者输入完整信息 6、检验信息完整性 7、参与者点击提交 8、提交成功
拓展点	1、发现参与者没有登陆，进入登陆页面 2、输入信息不完整，提示输入完整

非功能需求	用弹出编辑框展示并且提示要一目了然
-------	-------------------

查无反馈用例规约

用例编号	CD-4
用例名称	查无反馈
用例描述	参与者进行查无反馈
参与者	用户
前置条件	进入首页
后置条件	参与者反馈成功
涉众利益	1、用户可以实现查无反馈 2、游客不能进行查无反馈 3、管理员可以管理用户提交的信息
基本路径	1、进入系统 2、参与者登录系统 3、参与者点击查无反馈 4、检验参与者是否是已登录状态 5、参与者输入缺少的信息 6、检验是否输入 7、参与者点击提交 8、提交成功
拓展点	1、发现参与者没有登陆，进入登陆页面 2、没有输入信息，提示输入完整
非功能需求	用弹出编辑框展示并且提示要一目了然

维护词汇表用例规约

用例编号	CD-5
用例名称	维护词典
用例描述	参与者进行词典数据的维护
参与者	管理员
前置条件	进入管理员端
后置条件	维护词典成功
涉众利益	参与者可以进入并快捷管理词典
基本路径	1、进入系统 2、登录管理员端 3、查找词典 4、增添词条 5、移除词条 6、编辑词条
拓展点	1、未登录管理员端，无法维护
非功能需求	页面清晰明了，便于操作

处理反馈用例规约

用例编号	CD-6
用例名称	处理反馈
用例描述	参与者进行反馈数据管理
参与者	管理员

前置条件	进入管理员端
后置条件	处理反馈信息成功
涉众利益	参与者可以对反馈信息进行编辑、移除等处理
基本路径	1、进入系统 2、登录管理员端 3、查看用户反馈信息 4、编辑反馈信息 5、同意反馈信息 6、移除反馈信息
拓展点	1、未登录管理员端，无法维护
非功能需求	页面结构整齐，便于管理

修改密码用例规约

用例编号	CD-7
用例名称	修改密码
用例描述	参与者修改密码
参与者	用户
前置条件	进入首页
后置条件	参与者修改成功
涉众利益	参与者可以成功修改密码
基本路径	1、进入系统 2、登录系统 3、点击个人信息 4、检验登录状态 5、输入现有密码 6、输入新密码 7、确认新密码 8、单击提交 9、检测现有密码正确性 10、检测新密码和确认新密码一致性 11、完成更改
拓展点	1、检测未登录，进入登录界面 2、现有密码输入错误，提示重新输入 3、新密码和确认新密码不一致，提示重新输入
非功能需求	方便整洁，不设置障碍，并且同时保证用户记得密码

2.3 功能需求

系统具备两个方面的功能。其一是产品本身的开发。以及从专业网站爬取数据存入词汇库中形成 Vocabulary 表。产品的自身包括注册，登入，用户密码的修改等用户信息的相关功能；其次是查询功能，具体表现为使用者使用词典搜索词条和管理员后台处理数据时对数据库进行搜索等；还有用户对于词条谬误的报告，用户添加信息，管理员处理使用者的报告，管理员对词汇表进行操作的功能。Python 的工作是爬取网页上的信息并将其写入数据库中，也就是获取数据，让词典中有数据可供查询，也是为了 Java Web 的后续功能打基础，Java Web 部分需要对数据库中的信息进行增删改查，那么要求就是数据库中必须要有数据，此外，数据库的维护普通的用户是没有权限的，只有管理员才有资格维护数据库。User 表中的数据由游客通过注册来添加，Admin 表会由程序员直接将数据添加进数据库，不设置管理

员注册页面。**Feedback** 表中的数据由已经完成登入的使用者来添加提交，也就是报错和反馈，提交过后由管理员来处理反馈的数据，处理方式包括通过和移除。**Vocabulary** 表则是存储查询数据使用的表。

2.3.1 用户端功能

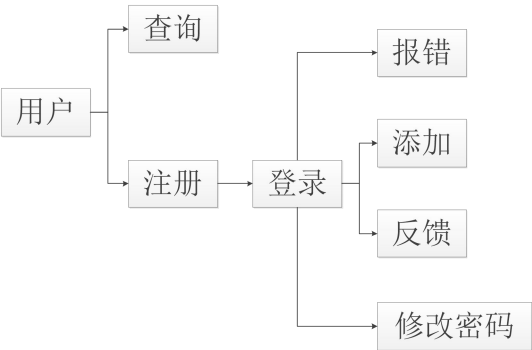


图 2-3 用户端功能图

- (1) 查询：用户可以通过本词典系统，来查询计算机专业英语词汇；用户可以通过本词典系统，来查询计算机专业英语词汇；用户通过输入所查词汇的缩写，英文全称或者中文全称的一部分后点击查询可以得到于所查内容相符或相近的结果，通过点击词汇，在结果框显示返回的结果。如结果有误，可以通过报错功能将错误上传以供管理员审核处理，报错功能需登入才可以完成。
- (2) 注册：用户可通过本系统的登陆界面进入注册界面，输入用户名和密码之后可以实现注册，获得个人账号。不登陆只能使用查询功能，以此来规范网络文明。
- (3) 登入：使用者可以通过登入界面进入个人账号，登陆后方可使用除查询外的其余功能。
- (4) 添加：用户在登入之后可以对词典进行完善，也就是说在使用本产品过程中如遇查询无果的情况，用户可以手动添加数据，成为本产品开发者的一员，完善本系统，类似于百度百科的人工添加词条。添加信息会发送到管理员端，管理员对该添加信息进行处理。
- (5) 反馈：使用者如遇系统 bug 或者所查词汇有问题并且自身无法解决的话可以通过反馈功能将问题反馈给管理员，管理员将会对反馈信息进行处理。
- (6) 密码修改：登陆后通过点击欢迎你字样打开修改密码模块，输入旧密码和新密码，如旧密码正确并且新密码符合即可完成密码的修改。
- (7) 报错：用户登入后操作中如找到数据有误等问题即可进行报错，将错误信息发送给管理员以供修正。

2.3.2 管理员端功能

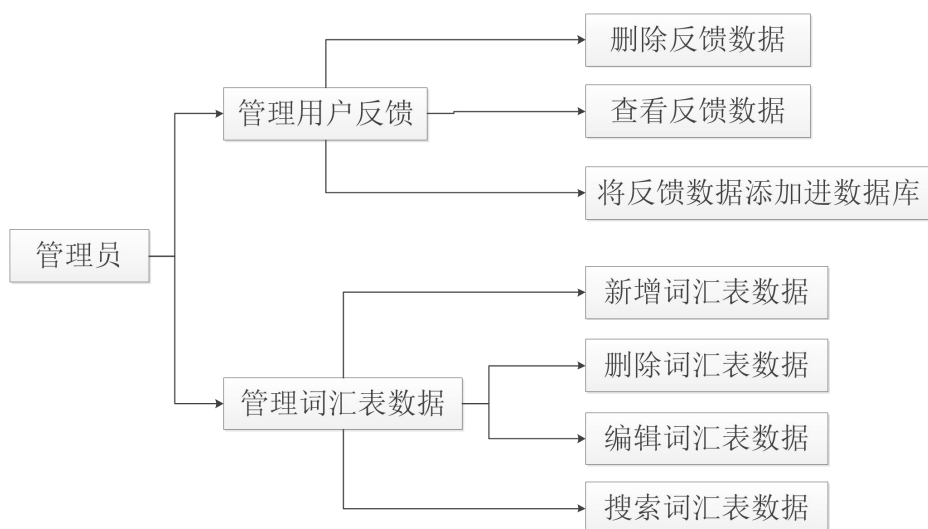


图 2-4 管理员端功能图

(1) 反馈：管理员在管理员端可以收到来自用户的报错反馈，添加反馈等信息，管理员审核并对其进行操作，如信息正确并且具有价值，那么管理员能直接该反馈直接添加进数据库，也就是词汇表之中；如数据不具有价值或者数据错误那么管理员能对该反馈信息进行移除处理。

(2) 管理词汇表数据：管理员可以在管理员端进行数据库的维护，包括对于词汇表的增添，删除，更新，查询。

2.3.3 获取词典数据功能

本词典数据的获取有两种方式。首先就是通过 Python 爬虫对相对权威的网站进行爬取收集基础数据，也就是缩略词，全称以及中文意义，然后通过基础数据进行词汇概述的爬取；其次就是词典对用户是开放的，用户可以参与到词典的完善之中来，作为是使用者的同时又是词典设计的参与者，通过这种方式也可以获取一定的词汇信息，让词典更加成熟。

2.3 E-R 图

对于本系统，数据库中共需要 6 个表，User 表，Admin 表，Feedback 表，Vocabulary 表，Qrecord 表和 Frecord 表。

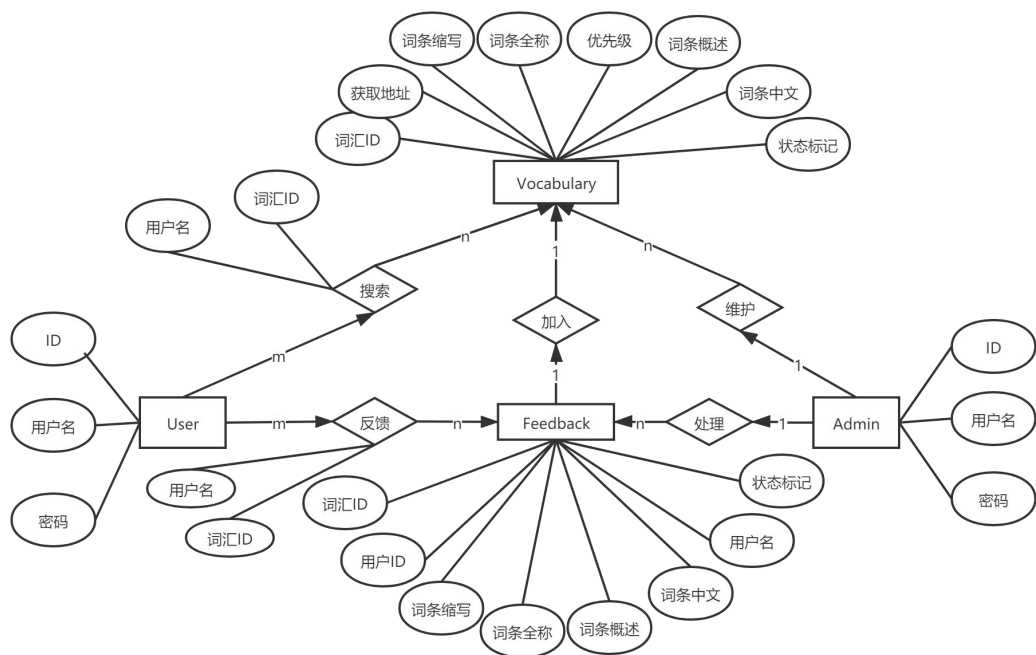


图 2-5 E-R 图

用户和管理员均具备 ID、用户名、密码三个属性。用户可提交完整反馈至管理员处以供处理，反馈具有词条缩写等具体属性。用户可以查询词汇表，管理员可以维护词汇表，词汇表具有词条全称等属性。

3. 词典设计

3.1 总体设计

3.1.1 词典分析

本词典是基于 Java Web 开发的一个网站，分为用户端和管理员端两个部分，用户可以在该系统实现对计算机词汇或者缩写的查找。该产品可以有效地提升人们使用计算机的效率，作为一款词典，它的功能让人一目了然，没有复杂或者难以理解的地方^[11]，可以做到让用户无门槛使用。

3.1.2 用户操作流程

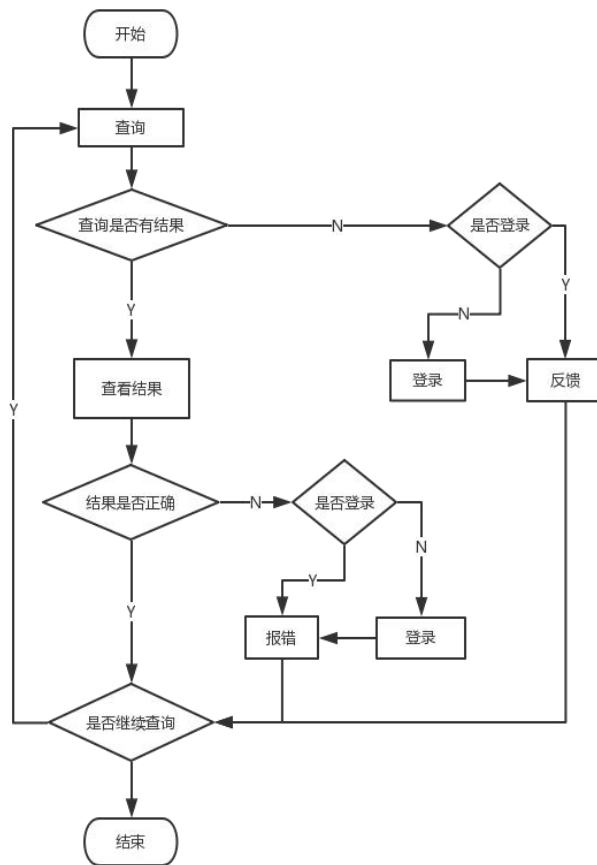


图 3-1 用户操作流程

游客和用户可以在页面进行词条搜索，并且进入词条查看其详细信息。但是只有已登入用户有权向管理员发送报错以及其他的反馈信息，系统会对用户状态进行检查，如未登录则跳转至登陆页面，完成后用户可以继续进行报错反馈等操作，完成查询等事件之后结束。

3.1.3 管理员操作流程

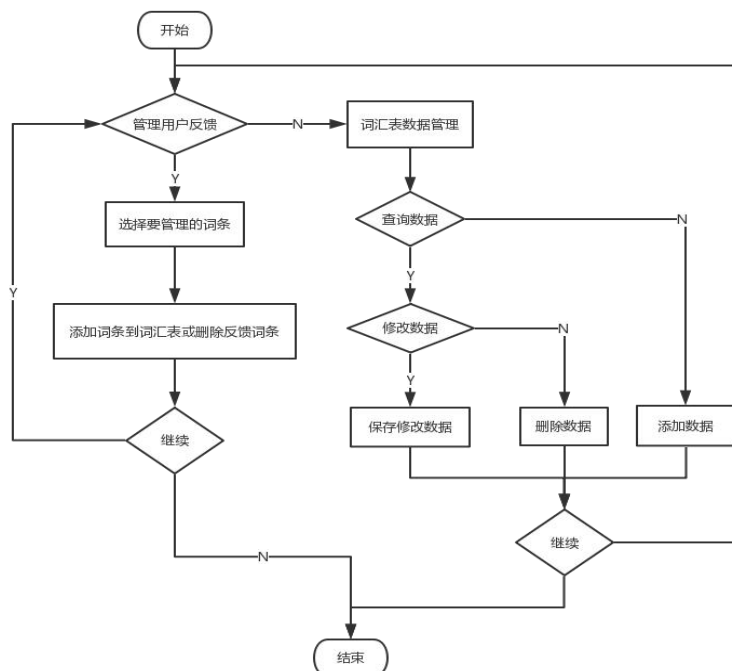


图 3-2 管理员操作流程

管理员在进图管理员端之后可以对用户提交的反馈信息进行处理，包括对其进行编辑，或者是检查无误直接加入词典之中，如果发现信息存在巨大问题，那管理员有权直接将其移除。管理员可以对词典进行维护，其中包括搜索词条，移除词条，新增词条，编辑词条。

3.1.4 词典设计

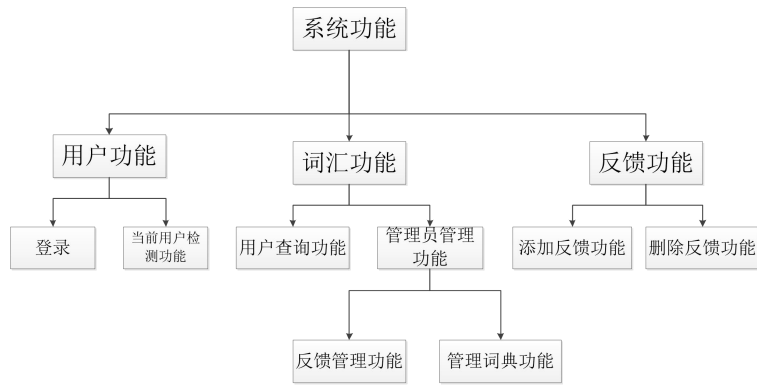


图 3-3 系统功能图

系统共分为三个功能模块，分别是用户功能，词汇功能以及反馈功能。

用户功能可以分为两个大类，一是登录功能，顾名思义可以实现用户的登录注册功能，用户是否登录的检查功能主要作用就是在用户进行反馈和报错操作的时候识别用户是否已经处于登陆状态，如果没有登录则无法继续进行反馈和报错。

词汇功能则是由用户端和管理员端两个部分组成；用户搜索词条时，输入值通过 `search` 方法在词汇表中搜索结果，查询所得的结果通过 `SearchServlet` 最终按 json 文本格式显示在网页中。管理员功能则再细分为两个部分：分别是对于反馈信息进行处理的功能以及对于词典的维护功能，用户通过 `userAdd` 方法提交新增词汇到管理员端，管理员则通过 `saveVocabulary` 方法来将用户新增词汇添加进词典之中。管理员可以在后台对词典的信息进行维护，包括基础的增添、移除、更改、查找。管理员对系统进行维护可通过 `search` 方法找到需要修改的数据，再调用 `saveVocabulary` 方法来完成修改或者 `delete` 方法来完成移除数据。

反馈功能部分包括用户的反馈和管理员对反馈的管理。使用过程中，如果用户发现词典信息存在谬误，可以进行报错，如果查询结果不存在，用户可以主动添加数据，通过 `UserAdd` 方法将反馈信息传递给后台，以供管理员处理。管理员进入后台之后，通过 `getFeedback` 方法将用户提交的数据交由管理员处理，管理员人工处理数据，检查无误的信息通过 `addToVocabulary` 加入词典之中，同时将 `Feedback` 表之中的该条记录进行移除，如果管理员在处理信息过程中发现用户提交的数据不足以加入词典，则可通过 `deleteFeedback` 方法将其移出 `Feedback` 表。

3.2 词典详细设计

本系统采用 MVC 框架进行开发，通过业务逻辑、数据、还有页面显示分离的方法来组织代码，其中 M 指的是 Model，在本系统中就是 bean 即是 Model；V 是 View，即本系统的前端；C 是 Controller，也是主要展示部分，也就是本系统的 `servlet`，用于接收请求，连接前后端。

3.2.1 表

User 表：记录用户的数据，也是实现用户登录等功能的基础。

表 3-1 User 表

字段名	数据类型	长度	是否为空	主外键	备注
id	int	11	否	主键	
username	<u>varchar</u>	255	否		
password	<u>varchar</u>	255	否		

Admin 表：记录管理员数据，也是管理员得以处理反馈等信息的基础。

表 3-2 Admin 表

字段名	数据类型	长度	是否为空	主外键	备注
id	int	11	否	主键	
username	<u>varchar</u>	255	否		
password	<u>varchar</u>	255	否		

Feedback 表：用户进行报错和添加词条等功能的基础。

表 3-3 Feedback 表

字段名	数据类型	长度	是否为空	主外键	备注
f_id	int	11	否	主键	反馈表的 id
v_id	int	11			用户反馈报错的词汇的 id
f_abbreviation	<u>varchar</u>	255			
f_fullwords	<u>varchar</u>	255			
f_specificmeaning	<u>varchar</u>	255			
f_lemmasummary	<u>varchar</u>	2255			
f_username	<u>varchar</u>	255			
f_remark	<u>varchar</u>	255			add: 标记该记录为用户新增词汇 notfound: 标记该记录为用户反馈查询不到 error: 标记为该记录存在错误

Vocabulary 表：查询以及管理员维护等功能的基础。

表 3-4 Vocabulary 表

字段名	数据类型	长度	是否为空	主外键	备注
id	int	11	否	主键	
abbreviation	<u>varchar</u>	255			
fullwords	<u>varchar</u>	255			
specificmeaning	<u>varchar</u>	255			
lemmasummary	<u>varchar</u>	2550			
address	<u>varchar</u>	225			
priority	<u>bigint</u>	50			
remark	<u>varchar</u>	255			若该记录为用户新增或反馈，则记录用户名

Qrecord 表：负责存留查询记录。

表 3-5 Qrecord 表

字段名	数据类型	长度	是否为空	主外键	备注
id	int	11	否	主键	
q-username	varchar	255			
v-id	int	255			查询的词汇 id

Frecord 表：负责存留反馈记录。

表 3-6 Frecord 表

字段名	数据类型	长度	是否为空	主外键	备注
id	int	11	否	主键	
f-username	varchar	255			
v-id	int	255			反馈的词汇 id

3.2.2 用户功能

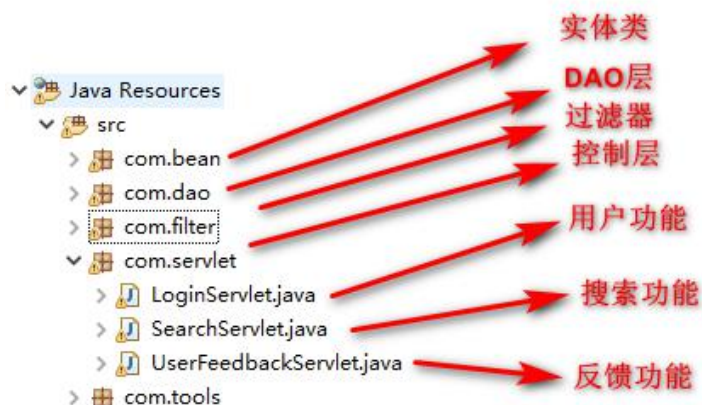


图 3-4 用户功能模块

用户登录功能包括用户以及管理员的登入登出、密码的更改以及确认用户是否登录等功能。

(1) userLogin

用本方法实现普通用户的登入，通过 isValid 方法来检查用户信息的有效性，检查若得到确认有效的结论，则得到成功登入的提示并更新页面。相反的，如果信息无效，那么用户会得到反馈来提示哪里输入有误。

(2) AdminLogin

用本方法实现管理员的登入，通过 isValid 方法检查检查管理员信息的有效性。

(3) checkStatus

用本方法实现对用户登入状态的检查，通过值判断用户登入与否。对于已完成登入的使用者，则其可以进行添加词汇等一系列的操作，将信息提交给管理员。如果是没有登入的使用者，则会弹窗要求其登入后在进行其他功能的操作。

(4) modifyPwd

用本方法实现更改密码，通过 modifyPwd 方法实现使用者密码的更新。

(5) userExit

用本方法实现登出词典，将词典还原到没有使用者登入的状态。

(6) userRegister

用本方法实现账户创建，通过 saveUser 方法实现把用户录入的 username 和 password 数据添加进 User 表之中。

3.2.3 词汇功能

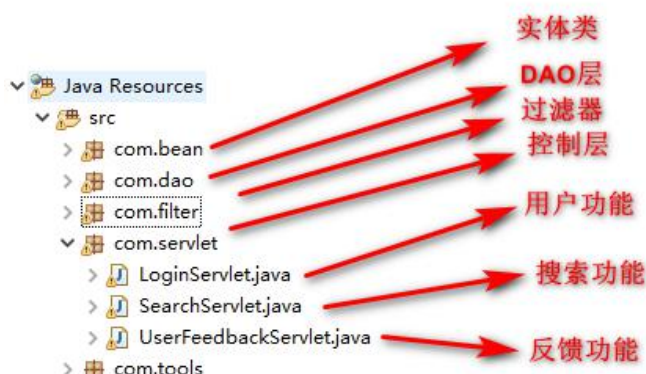


图 3-5 词汇搜索模块

词典中的信息主要但不是全部都来自于爬虫，用户也可以主动将信息录入词典。其实词汇部分的功能主要就是主页的搜索、管理员对于词典内容的维护所需要的功能，也就是增添词条，移除词条，搜索词条和更新词条，主要依赖于对 Vocabulary 表的 CURD。

(1) getSearch

用本方法实现搜索功能。使用者输入的数据在词典中进行查找，最终得到匹配的信息。双端的使用者对于词典的搜索都是由此方法实现的，通过 search 方法找到与输入相匹配的信息并且将 json 数据传回网页。

(2) adminModify

管理员用本方法实现在后台更新词典内容，通过 modify 方法来更新数据库的词条信息，获取布尔类型的结果，以此告知用户完成更新或者更新时出现错误。

(3) adminAdd

管理员用本方法实现现在后台新增词典内容，通过 saveVocabulary 方法来增添 Vocabulary 表中的词条信息。

(4) adminDelete

管理员用本方法实现现在后台移除词典内容，通过 delete 方法移除相应的数据。

3.2.4 反馈功能

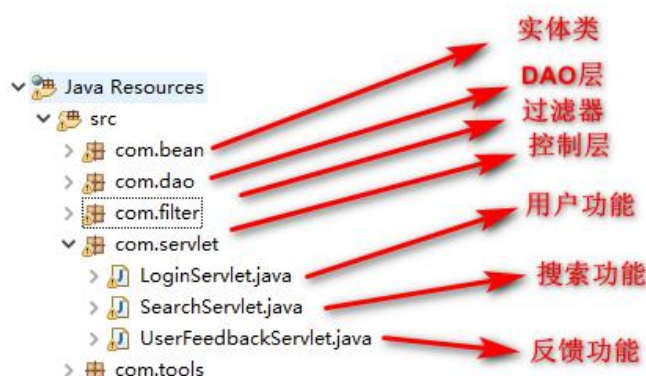


图 3-6 反馈模块

反馈功能其实就是通过使用者使用过程中的实际操作来对系统进行错误查找或者遗漏查找，然后将问题通过本功能发送交由管理员处理，管理员负责检查反馈数据，然后对其进行相应的处理。本功能是为任何一款词典都应该具备的，因为在当今这个数据爆炸的时代，没有任何一款词典能说自身的词汇是完备的，只有通过使用者的不断添加和查错，才能让词典更加充实，与此同时又使用户亲身加入词典的设计，提高了产品的亲和力，使产品更具前景。

(1) userAdd

用该方法实现增添词典词条内容功能，用户主动编写词典数据并通过 UserAdd 方法将记录加入 Feedback 表中提交给管理员处理。

(2) notFound

用本方法实现反馈词典遗漏信息的功能，使用者在搜索过程中，无法查到想要了解的词条，则可使用查无反馈功能，将遗漏的信息通过 UserAdd 方法写入 Feedback 表提交给管理员，然后管理员对使用者提交的数据进行处理。

(3) postError

通过爬虫获取基础词汇的方式有利有弊，好处自不必多说，缺点同样存在，那就是获取到的信息完整程度以及正确率无法达到百分之百，那么用户操作的过程中，一旦发现给出的词条信息不正确，可以通过 UserAdd 方法报错，将词条的准确数据加入 Feedback 表，然后提交给管理员处理。

(4) getFeedbackData

用本方法实现展示用户提交的信息，在登入后台之后，管理员可以查看反馈信息，通过 getFeedback 方法得到 Feedback 表中的全部信息，将 json 数据以表格形式展示在页面中。

(5) delete

用本方法实现对 Feedback 表中数据的移除。管理员在对反馈信息进行处理时，若发现某反馈数据错误，也就是无法加入词典，通过调用 deleteFeedback 方法将该反馈从 Feedback 表里移除。

3.2.5 爬虫



图 3-6 获取词典基础数据模块

这是获取词典基础数据的最主要渠道^[12]，通过爬虫添加数据，形成词典雏形，是后期的查询、增加、删除、更新等功能的数据基础。共包含 CrawData.py、CheckNull.py、Download.py、CrawPriority.py、Write.py、Craw.py 这六个类。CrawData 类通过 write 方法调用 Downloader 类中的 download 方法获取到页面的 abbreviation 和 fullwords 以及 specificmeaning 这三个数据并写入 Vocabulary 表中，Spider 类通过 crawl 方法调用 CheckNull 类中的 checkNull 方法判断 lemmasummary 是否为空，如果是，则该词条需要进一步的获取数据，调用 Download 类中的 download 方法，CrawPriority 类中的 getThePriority 方法得到词汇的 lemmasummary、address 和 priority，最后调用 Write 类包含的 write 方法将对应信息存储进词典数据库。到此即完成了词典基础数据的爬取。

4. 系统实现

4.1 开发环境

操作系统：Windows 10 专业版

开发平台：Eclipse IDE EE 4.14.0、Java SE 8.0、PyCharm、Tomcat 8.5、5.7.29 MySQL Community Server

4.2 获取词典数据

4.2.1 获取词汇基本信息

百度百科中相应网站包含大量本系统所需的缩写和短语。通过爬虫获取该页面的 abbreviation、fullwords、specificmeaning 数据并写入词典数据库中。

百度百科网页结构如下：

```

▼<div class="body-wrapper">
  <div class="before-content">
  </div>
  ▼<div class="content-wrapper">
    ▼<div class="content">
      <div class="personal-content">
      </div>
      ▼<div class="main-content">
        <div class="top-tool ">...</div>
        <div style="width:0;height:0;clear:both"></div>
        <dl class="lemmaWgt-lemmaTitle lemmaWgt-lemmaTitle">...</dl>
        <div class="promotion-declaration">
        </div>
        <div class="lemma-summary" label-module="lemmaSummary">...</div>
        <div class="lemmaWgt-promotion-leadPV8tn">
        </div>
        <div class="configModuleBanner">
        </div>
        <div class="basic-info cmn-clearfix">...</div>
        <div class="lemmaWgt-lemmaCatalog">...</div>
        <div class="anchor-list ">...</div>
        <div class="para-title level-2" label-module="para-title">...</div>
        <div class="para" label-module="para">使用Ctrl+F可以实现职责 在网页上快速查找文字! </div>
        <div class="anchor-list ">...</div>
        <div class="para-title level-2" label-module="para-title">...</div>
        <div class="para" label-module="para">A</div>
        <div class="para" label-module="para">(Active-matrix)主动矩阵</div>
        <div class="para" label-module="para">...</div>
      </div>
    </div>
  </div>
</div>

```

图 4-1 网页结构展示 1

查看网站源码，在 class 为 body-wrapper 下找到 para-title level-2 样式的 class，遍历源码寻找其同级类的 div 标签节点，下载所含数据。

初步下载得到的只是一串字符，只有将其分割为 abbreviation、fullwords 和 specificmeaning 三段才能写入数据库之中，通过正则表达式识别“()”，以“()”为界，“()”之前的是 abbreviation，“()”之中的为 fullwords，“()”之后的 specificmeaning，然后将三列数据写入 Vocabulary 表中。

```

class CrawlData(object):
    def __init__(self):
        print("CrawlTheData创建成功")

    def write(self):
        resp = request.urlopen("https://baike.baidu.com/item/" + quote("词根词源"))
        html_doc = resp.read().decode("utf-8")

        soup = bs(html_doc, "html.parser")
        db = pymysql.connect("localhost", "root", "root", "db_sys", use_unicode=True, charset="utf8")
        cursor = db.cursor()
        for content in soup.find('div', class_="main-content").find('div', class_="level-2").find_next_siblings('div', class_="para"):
            print(content.get_text())

            content = content.get_text()
            abb = re.match(r'[^()]*?(?=\\()', content)
            if abb is not None:
                abbreviation = abb.group().strip()
                abbreviation = abb
                #print(abbreviation)
            fullwords = re.search('(\\(.?\\))', content)
            #print(fullwords)
            p = re.compile(r'[\u4e00-\u9fa5]')
            res = re.findall(p, content)
            specificmeaning = ''.join(res)

            sql = "INSERT INTO t_vocabulary(abbreviation,fullwords,specificmeaning) \
                VALUES ('%s', '%s', '%s')"% \
                (abbreviation, fullwords, specificmeaning)

            try:
                cursor.execute(sql)
                db.commit()
                print("爬入数据成功")
            except:
                db.rollback()
                print("爬入数据失败")
        db.close()

```

图 4-2 爬取 abbreviation、fullwords 和 specificmeaning 代码

4.2.2 通过 specificmeaning 爬取 lemmasummary

先对 lemmasummary 进行判空，读取其他列数据，优先通过 specificmeaning 进行爬取，如果无果再通过 fullwords 和 abbreviation 进行 lemmasummary 的爬取。


```

class CheckNull(object):
    def checkNull(self):
        data=[]
        db=pymysql.connect("localhost","root","root","db_sys",use_unicode=True,charset="utf8")
        cursor=db.cursor()
        sql="select id,abbreviation,fullwords,specificmeaning from t_vocabulary where lemmasummary is null or lemmasummary="
        #sql="select id,abbreviation,fullwords,specificmeaning from t_vocabulary where priority is null"
        try:
            cursor.execute(sql)
            results=cursor.fetchall()
            #print(results)
            for row in results:
                id=row[0]
                abbreviation = row[1]
                fullwords = row[2]
                specificmeaning = row[3]

                word={"id":id,"abbreviation":abbreviation,"fullwords":fullwords,"specificmeaning":specificmeaning}
                #print(word)
                data.append(word)

            #print(data)
            return data
        except:
            print ("Error: unable to fetch data")
            return None
        db.close()

if __name__=="__main__":
    obj_readdata=CheckNull()
    print(obj_readdata.checkNull())

```

图 4-3 判空代码

Downloader 类是网页下载器，首先通过判断待爬网页的有效性来决定是否对网站进行进一步的下载，因为百科在页面有误时，将转至其专门的报错网页，所以下载器也需要对网页进行额外判断，当检测到其 url 为报错网页时，报告爬取失败并返回空值。

```

class Downloader(object):
    def download(self,url):
        if url is None:
            return None

        response=urllib.request.urlopen(url)
        #if response.getcode()!=200:##判断是否请求成功
        #    return None
        url=response.geturl()
        if url == "https://baike.baidu.com/error.html":
            print("爬取失败")
            return None
        #print(url)
        return response.read().decode("utf-8")

```

图 4-4 网页下载器代码

百科词条是由广大网友进行编写的，导致其内容结构不尽相同，但是通过阅读网站源码，我们需要的数据大概率会在 class 为 lemma-summary 的标签为 div 的子节点中，通过遍历和下载，即可获取其中数据，将其加入词典。

```

<div class="body-wrapper">
  <div class="before-content"></div>
  <div class="content-wrapper">
    <div class="content">
      <div class="personal-content">
      </div>
      <div class="main-content">
        <a class="posterFlag expert-icon" href="javascript:;" title="权威专家认证词条"></a>
        <div class="top-tool"></div>
        <div style="width:0;height:0;clear:both"></div>
        <dl class="lemma>gt-lemmaTitle lemma>gt-lemmaTitle"></dl>
        <div class="authorityListPrompt"></div>
        <div class="promotion-declaration">
        </div>
      <div class="lemma-summary" label-module="lemmaSummary">
        <div class="para" label-module="para">
          “人工神经网络 (Artificial Neural Network, 即ANN)，是20世纪80年代以来人工智能领域兴起的研究热点。它从信息处理角度对人脑神经网络进行抽象，建立某种简单模型，按不同的连接方式组成不同的网络。在工程与学术界也常直接简称为神经网络或类神经网络。神经网络是一种运算模型，由大量的节点（或称神经元）之间相互联接构成。每个节点代表一种特定的输出函数，称为激励函数（activation function）。每两个节点间的连接都代表一个对于通过该连接信号的加权值，称之为权重，这相当于人工神经网络的记忆。网络的输出则依网络的连接方式，权重值和激励函数的不同而不同。而网络自身通常都是对自然界某种算法或者函数的逼近，也可能是对一种逻辑策略的表达。”
        </div>
        <div class="para" label-module="para"></div>
      </div>
    </div>
  </div>
</div>

```

图 4-5 网页结构展示 2

```

def crawl(self):
    data=self.read.readData(1)
    #print(data)
    for row in data:
        id=row["id"]
        abbreviation = row["abbreviation"]
        fullwords = row["fullwords"]
        specificmeaning = row["specificmeaning"]
        #print ("abbreviation=%s,fullwords=%s,age=%s" % (abbreviation, fullwords, specificmeaning ))
        url="https://baike.baidu.com/item/"+quote(specificmeaning)
        html_doc=self.downloader.download(url)
        #try:
        if html_doc:
            #print(html_doc)
            soup = bs(html_doc,"html.parser")
            try:
                text=soup.find("div",class_="Lemma-summary").get_text().strip()
            except:
                try:
                    text=soup.find("div",class_="main-content").find("div",class_="para").get_text().strip()
                except:
                    continue
            priority=self.priority.getThePriority(specificmeaning)
            if text:
                #for tag in soup.find_all("div",class_="lemma-summary"):
                lemmasummary=text
                print(lemmasummary)#测试

                self.write.write(id,lemmasummary, url,priority)
            elif soup.find("div",class_="main-content").find("div",class_="para"):
                lemmasummary=soup.find("div",class_="main-content").find("div",class_="para").get_text().strip()
                #print(lemmasummary)#测试
                self.write.write(id,lemmasummary, url,priority)
        #except:
        #    print("crawl failed")

```

图 4-6 爬取 lemmasummary 代码

4.2.3 获取 Priority

百科的查询结果展示的先后排序是通过众多因素来决定的，这些因素大部分都是没有太大差异的，无非是网页质量，词条密度，网页权重等等，但是有一个因素至关重要并且不同词条的差异极大，那就是词条搜索次数，也就是词条点击量。本词典要优化查询，让用户可以更快的找到自己需要的词条，那么对于查询结果的排序显得极为重要，我们可以通过解析各个词条的被查询次数，来判断词条的优先级，从而给用户更好的体验。

对搜索网页源码进行查看，发现搜索次数作为时刻变动的数据会存储在 data-op 里，通过对 data-op 数据进行进制转换我们可以得到该关键字被搜索的次数，将此数据存入数据库 Vocabulary 表的 Priority 列中即可完成词条优先级的下载。

```

▼<div id="content_left">
  ▶<div class="result-op c-container xpath-log" srcid="1547" id="1" tpl="bk_polysemy" mu="https://baike.baidu.com/item/数据字典/1270246" data-op="{ 'y': 'CBFFF77A' }" data-click="{ 'p1': '1', 'rsv_bdr': '0', 'fm': 'albk', 'rsv_stl': '', 'p5': '1' }">...</div>
  ▼<div class="result_c-container" id="2" srcid="1599" tpl="se_com_default" data-click="{ 'rsv_bdr': '0', 'p5': '2' }">
    ▼<h3 class="t">
      ▶<a data-click="{
        'F': '778317EA',
        'F1': '9073F1E4',
        'F2': '4CA60D68',
        'F3': '54E5243F',
        'T': '1586082941',
        'y': 'FBEFFAD7'
      }" href="http://www.baidu.com/link?
ur1=juQZfSna4H88kXsVz4mC2QFMV1I5uH1vNwMdYqEgMTa_PocDafmP8ABzU7KFnQs3BqMBfcF9F0UzVvtCIji0-xy4W5rRtH4ZrsfarkyYU1S" target="_blank">...</a>
    </h3>

```

图 4-7 网页结构展示 3

```

class CrawlThePriority(object):
    def __init__(self):
        print("爬取priority")

    def getThePriority(self,word):
        try:
            headers = {
                'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
                'Accept-Encoding': 'gzip, deflate, compress',
                'Accept-Language': 'en-us;q=0.5,en;q=0.3',
                'Cache-Control': 'max-age=0',
                'Connection': 'keep-alive',
                'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:22.0) Gecko/20100101 Firefox/22.0'
            }
            url = 'http://www.baidu.com.cn/s?wd=' + word
            html = requests.get(url=url,headers=headers)
            soup = bs(html.content,"html.parser")
            result = soup.find("div",id="content_left").find("div",class_="xpath-log").get("data-op")
            s = eval(result)
            priority = int(eval(result)['y'],16)
            #print(int(s["y"],16))
            return priority
        except:
            return 1#爬取不到默认为1

```

图 4-8 获取 priority 代码

4.2.4 将数据存进词典

Write 方法负责将方才爬取的数据通过 ID 对应存入词典，至此词典基础数据获取部分内容基本完成。

```

class Write(object):
    def write(self,id,lemma,address,priority):
        db = pymysql.connect("localhost","root","root","db_sys",use_unicode=True, charset="utf8")
        cursor = db.cursor()
        sql="UPDATE t_vocabulary set lemmasummary='%s',address='%s',priority= '%d' \
        where id='%d'" % (lemma,address,priority,id)
        try:
            cursor.execute(sql)
            db.commit()
            print("插入数据成功")
        except:
            db.rollback()
            print("插入数据失败")
        db.close()

```

图 4-9 写入数据库代码

4.3 词典功能

4.3.1 查询功能

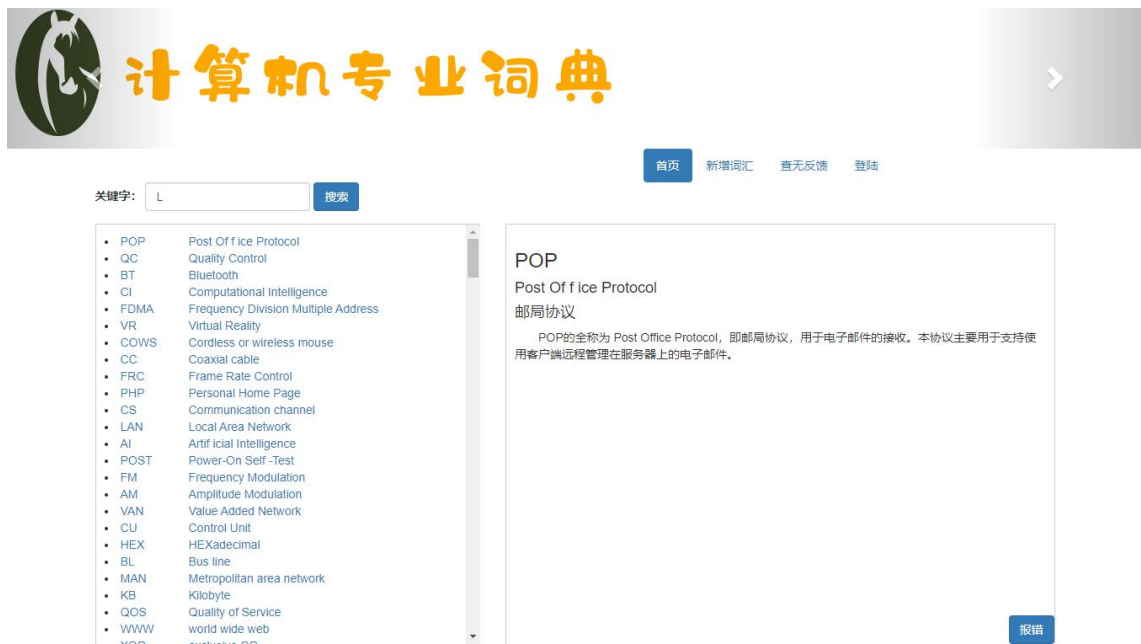


图 4-10 词典查询页

实现词汇的模糊搜索，系统会识别输入并进行额外处理，包括大小写转换，判断是否存在遗漏输入并获取相似结果，最后通过匹配度和词汇 Priority 值相结合对词条进行排序并在页面展示。令表单 ID 为 SearchInput，搜索按钮为 SearchButton，通过 jQuery 的 val 方法得到输入框数据，然后对其进行判空处理，合格则调用 post 方法发送和数据 action 至后端来检索数据。SearchServlet 收到数据和 action 请求，创建数组用于存储检索后符合的词汇；得到的词汇通过 json 格式回传到查询页面。通过 callback 方法完成对于检索数据的获得。

4.3.2 登录功能

分别为用户和管理员创建选择框，单击事件写入选其一则清除另一个的选中状态。

为登录创建 onclick 事件，即在得到用户名以及密码的表单数据之后对其进行判空，如果存在空，那么在表单下方弹出不能存在空的字样。若表单数据正常，则判断其选择框的选项，为用户则调用 userDAO 类中的方法判断登录信息的正确与否，如果选中的是管理员，那么则由 adminDAO 中的方法来进行判断。



用户登录

☐ 管理员 ☒ 用户

[没有账号? 去注册](#)

图 4-11 登录

4.3.3 新增词条功能

为 addButton 添加 onclick 事件，得到四个表单数据，分别对其进行判空操作，若存在空则显示不可存在空的字样，若所有表单都有数据，调用 ajax 方法将数据写入 Feedback 表中，同时令其 remark 的值为 add，再通过 UserAdd 方法回传是否添加成功。

[首页](#) [新增词汇](#) [查无反馈](#) [欢迎您, 1](#) [退出](#)

新增词汇

缩略词:

DFS

完整英文词汇:

Depth-First-Search

中文解释:

深度优先搜索

专业名词含义:

深度优先搜索是一种在开发爬虫早期使用较多的方法。它的目的是要达到被搜索结构的叶结点(即那些不包含任何超链的HTML文件)。在一个HTML文件中，当一个超链被选择后，被链接的HTML文件将执行深度优先搜索，即在搜索其余的超链结果之前必须先完整地搜索单独的一条链。深度优先搜索沿着HTML文件上的超链走到不能再深入为止，然后返回到某一个HTML文件，再继续选择该HTML文件中的其他超链。当不再有其他超链可选择时，说明搜索已经结束。 [1]

图 4-12 增添词条

4.3.4 查无反馈功能

为 postNotFoundButton 添加 onclick 事件，得到三个表单数据，单独对缩写的表单数据进行判空操作，若存在空则显示不可存在空的字样，若存在数据，调用 ajax 方法将数据写入 Feedback 表中，同时令其 remark 的值为 notfound，再通过 UserAdd 方法回传是否添加成功。

查不到? 反馈给管理员吧!

查不到:

要查询的缩略词:

DFS

完整英文词汇:

Depth-First-Search

中文解释:

深度优先搜索

重置提交

图 4-13 查无反馈

4.3.5 处理反馈功能

进入管理员端时，发送 `getFeedbackData` 这个行为至 `Servlet` 以展示 `Feedback` 表中的信息，通过 `getFeedback` 方法来获得表中信息，将其存放于泛型变量 `feedbacks` 中，最后以 `json` 的格式传回页面。灵活运用 `bootstrap-table` 表格组件给页面的 `feedbackTable` 表格增添各类获取到的信息。

给✔添加 `onclick` 事件，读取本条记录信息，若检测到 `remark` 值为 `add`，通过 `saveVocabulary` 方法把当前的数据写入 `Vocabulary` 表中，然后使用 `deleteFeedback` 方法在 `Feedback` 表中删除本行数据；若 `remark` 值为 `error`，通过 `modify` 方法更改其内容，然后使用 `deleteFeedback` 方法在 `Feedback` 表中删除本行数据。

给✖添加 `onclick` 事件，通过 `deleteFeedback` 方法在 `Feedback` 表里将本行数据移除。

给✎添加 `onclick` 事件，读取本条记录信息并将其显示在子窗口中以供编写。给确认添加 `onclick` 事件，得到各个表单之中的信息，将执行 `addToVocabulary` 行为的请求传入 `Servlet` 中的 `addToVocabulary` 方法，读取本条记录信息，若检测到 `remark` 值为 `add`，通过 `saveVocabulary` 方法把当前的数据写入 `Vocabulary` 表中，然后使用 `deleteFeedback` 方法在 `Feedback` 表中删除本行数据；若 `remark` 值为 `error`，通过 `modify` 方法更改其内容，然后使用 `deleteFeedback` 方法在 `Feedback` 表中删除本行数据。

用户反馈管理 数据管理 1 退出

用户反馈管理								
ID	V_ID	Abbreviation	fullWords	SpecificMeaning	LemmaSummary	Username	Remark	edit
17	126	DFS	Depth-First-Search	深度优先搜索	深度优先搜索是一种在开发爬虫早期使用较多的方法。它的目的是要达到被搜索结构的叶结点(即那些不包含任何超链的HTML文件)。在一个HTML文件中，当一个超链被选择后，被链接的HTML文件将执行深度优先搜索，即在搜索其余的超链结果之前必须先完整地搜索单独的一条链。深度优先搜索沿着HTML文件上的超链走到不能再深入为止，然后返回到某一个HTML文件，再继续选择该HTML文件中的其他超链。当不再有其他超链可选择时，说明搜索已经结束。	1	0	✎✔✖

图 4-14 管理反馈

编辑

v_id: 126

Abbreviation: DFS

FullWords: Depth-First-Search

SpecificMeaning: 深度优先搜索

LemmaSummary:

深度优先搜索是一种在开发爬虫早期使用较多的方法。它的目的是要达到被搜索结构的叶结点(即那些不包含任何超链的HTML文件)。在一个HTML文件中, 当一个超链被选择后, 被链接的HTML文件将执行深度优先搜索, 即在搜索其余的超链结果之前必须先完整地搜索单独的一条链。深度优先搜索沿着HTML文件上的超链走到不能再深入为止, 然后返回到某一个HTML文

Username: 1

Remark: 0


取消 确认

图 4-15 编辑反馈

4.3.6 维护词汇表功能

令表单 ID 为 InfoInput，搜索按钮为 getInfoButton，通过 jQuery 的 val 方法得到输入框数据，然后对其进行判空处理，合格则调用 post 方法发送和数据 action 至后端来检索数据。SearchServlet 收到数据和 action 请求，创建数组用于存储检索后符合的词汇；得到的词汇通过 json 格式回传到查询页面。通过 callback 方法完成对于检索数据的获得。灵活运用 bootstrap-table 表格组件给页面的 feedbackTable 表格增添各类获取到的信息。

给 adminAdd 添加 onclick 事件，跳出子窗口，要求编写表单值，然后对其进行判空，合格则经过给 add 添加 onclick 事件，从而得到表单内容，通过 saveVocabulary 方法把当前的词条信息加入 Vocabulary 表中。

给  添加 onclick 事件，读取本条记录信息并将其显示在子窗口中以供编写。给 modify 添加 onclick 事件，得到表单内容，对其进行除了备注一栏的其余内容包括 id 进行判空，若存在值为空则通过红色字样告知使用者缺少何值。通过 action 请求和表单数据调用 modify 方法完成对于词汇表的维护。


给  添加 onclick 事件，得到本记录的数据以及页面表格中的记录，调用 delete 方法将其进行移除处理并刷新表格。



图 4-16 管理查询

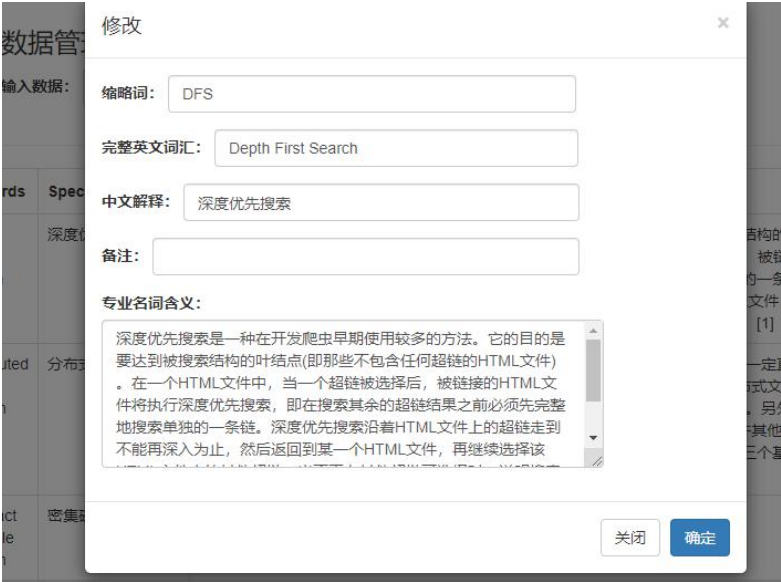


图 4-17 管理修改

5. 系统测试

5.1 测试目标

分别登入词典的双端,运用因果图法进行测试,执行用户端和管理员端的全部功能,判断与预期值是否相一致。

5.2 测试类型

人工测试。

5.3 测试对象

词典查询功能,用户新增词条功能,报错功能,登入注册功能,反馈功能,管理员处理反馈功能,管理员维护词典功能。

5.4 测试环境

Google Chrome 76.0.3809.87, Tomcat 8.5。

5.5 测试用例

5.5.1 词典首页功能

表 5-1 首页用例表

用例编号	用例名称	前置条件	重要级	优先级	测试阶段	测试步骤	预期结果	测试结果	编写人
001	通过缩写查询	位于首页	高	高	系统测试	输入缩写并搜索	正常显示结果	正常	雷梦豪
002	通过中文查询	位于首页	高	高	系统测试	输入中文并搜索	正常显示结果	正常	雷梦豪
003	通过全称查询	位于首页	高	高	系统测试	输入全称并搜索	正常显示结果	正常	雷梦豪
004	进入词条	进行查询操作	高	高	系统测试	单击查询结果	正常显示结果	正常	雷梦豪
005	游客能否报错	在游客状态进行报错	高	高	系统测试	单击报错	无法报错	正常	雷梦豪
006	用户能否报错	用户状态进行报错	高	高	系统测试	单击报错	成功进入页面	正常	雷梦豪
007	能否提交	提交报错	高	高	系统测试	登入并单击报错提交	成功发送	正常	雷梦豪

5.5.2 词典登录功能

表 5-2 登录用例表

用例编号	用例名称	前置条件	重要级	优先级	测试阶段	测试步骤	预期结果	测试结果	编写人
001	能否登入	在首页点击并选择用户登录	高	高	系统测试	对应输入已注册的账号和密码	成功登入	正常	雷梦豪
002	不输入密码登入	在首页点击并选择用户登录	高	高	系统测试	不输入密码	无法登入	正常	雷梦豪
003	不输入账号登入	在首页点击并选择用户登录	高	高	系统测试	不输入账号	无法登入	正常	雷梦豪
004	输入不正确密码登入	在首页点击并选择用户登录	高	高	系统测试	输入不正确的密码	无法登入	正常	雷梦豪
005	管理员登入	在首页点击并选择管理登录	高	高	系统测试	对应输入已在 Admin 表中的账号和密码	成功登入	正常	雷梦豪

5.5.3 新增词汇功能

表 5-3 新增词汇用例表

用例编号	用例名称	前置条件	重要级	优先级	测试阶段	测试步骤	预期结果	测试结果	编写人
001	进入新增词汇	位于首页	高	高	系统测试	点击新增词汇	进入新增词汇	正常	雷梦豪

002	无数据提交	位于首页，进入新增词汇，无数据	高	高	系统测试	单击提交	提示加入数据	正常	雷梦豪
003	缺少数据之一提交	位于首页，进入新增词汇，缺少一栏数据	高	高	系统测试	单击提交	提示缺少数据	正常	雷梦豪
004	正常提交	位于首页，进入新增词汇，数据完整	高	高	系统测试	单击提交	成功	正常	雷梦豪

5.5.4 查无反馈功能

表 5-4 查无反馈用例表

用例编号	用例名称	前置条件	重要级	优先级	测试阶段	测试步骤	预期结果	测试结果	编写人
001	进入查无反馈	位于首页	高	高	系统测试	点击查无反馈	进入查无反馈	正常	雷梦豪
002	无数据提交	位于首页，进入查无反馈，无数据	高	高	系统测试	单击提交	提示加入数据	正常	雷梦豪
003	仅提交词汇缩写信息	位于首页，进入新增词汇，输入缩写信息	高	高	系统测试	单击提交	成功	正常	雷梦豪
004	正常提交	位于首页，进入新增词汇，数据完整	高	高	系统测试	单击提交	成功	正常	雷梦豪

5.5.5 处理反馈功能



表 5-5 处理反馈用例表

用例编号	用例名称	前置条件	重要级	优先级	测试阶段	测试步骤	预期结果	测试结果	编写人
001	进入查无反馈	位于管理员端处理反馈页面	高	高	系统测试	查看反馈	正常查看	正常	雷梦豪
002	通过反馈	位于管理员端处理反馈页面	高	高	系统测试	单击 	成功加入	正常	雷梦豪
003	编辑反馈	位于管理员端处理反馈页面	高	高	系统测试	单击 	进入编辑	正常	雷梦豪
004	移除反馈	位于管理员端处理反馈页面	高	高	系统测试	单击 	成功移除	正常	雷梦豪

5.5.6 维护词汇表功能

表 5-6 维护词汇表用例表

用例编号	用例名称	前置条件	重要级	优先级	测试阶段	测试步骤	预期结果	测试结果	编写人
001	进入词典维护	位于管理员端维护词典	高	高	系统	搜索词汇, 查看词汇	正常查看	正常	雷梦豪

		页面			测试				
002	添加词汇	位于管理员 端维护词典 页面	高	高	系统 测试	单击添加	成功 进入 添加	正 常	雷梦 豪
003	编辑词条	位于管理员 端维护词典 页面	高	高	系统 测试	单击 	进入 编辑	正 常	雷梦 豪
004	移除词条	位于管理员 端维护词典 页面	高	高	系统 测试	单击 	成功 移除	正 常	雷梦 豪

5.5.7 修改密码功能

表 5-7 修改密码用例表

用例 编号	用例名称	前置条件	重 要 级	优 先 级	测试阶 段	测试步骤	预期结 果	测 试 结 果	编写 人
001	登陆系统	位于管理员 端维护词典 页面	高	高	系统测 试	登陆系统	正常登 录	正 常	雷梦 豪
002	进入修改 密码页面	位于管理员 端维护词典 页面	高	高	系统测 试	单击用户信息	成功开 始修改	正 常	雷梦 豪
003	完善信息	位于管理员 端维护词典 页面	高	高	系统测 试	输入正确的当 前密码、新密 码以及确认密 码	正确输 入	正 常	雷梦 豪
004	点击提交	位于管理员 端维护词典 页面	高	高	系统测 试	单击提交按钮	修改成 功	正 常	雷梦 豪

5.6 结论

项目的基础功能基本实现并且词典运行相对稳定。此外，系统问题修改及时，在预定日期之内完成词典开发和测试，可以上线进一步测试。

6. 系统评价

6.1 词典优势

本系统可让用户更便捷地查询计算机专有名词的详细信息，通过本系统用户可以查询到缩写短语的完整释义，包括词汇的中文意义以及所查单词的具体内容等，让用户可以更方便的使用和了解计算机，在学习过程中接触计算机词汇时可以及时加深了解。本系统能更好的提高对计算机的兴趣，加速用户学习计算机的效率，同时潜移默化地提升用户的计算机英语阅读能力。

6.2 词典不足

词典无法实现数据实时更新，Python 获取词条效率不算高并且成功率也不算很高^[13]。系统尚未引入语音搜索以及扫码登录功能，就现有的系统来说还可以进行优化。本词典由于种种原因没有开发移动端，使用起来还是有所不便^[14]。

7. 总结

毕业设计的课题让我收获良多，可以说为我的大学生活画上圆满句号的同时为我打开了一扇新的大门，对于计算机，我们在大学课程中所写的仅仅是基础，还无法将我们武装成为行业大牛。在进行毕业设计的过程中，因为需求，去学习了 Java Web 开发以及 Python 语言这门开发效率高的脚本语言^[15]。这对于我来说是一次巨大的突破，第一次独自一人负责一个项目，因为今年的特殊情况第一次在没有人的教导的情况下自主学习两种对我来说几乎全新的语言，在这个过程中，更加深刻的认识到大学科目的作用，期间遇到无数问题，从配置环境到数据库设计再到系统的开发都出现了许许多多的问题，特殊情况下资源不足，面对这些对自己而言非常棘手的“拦路虎”，只能自己慢慢摸索尝试解决，在这样一个过程中，我不光明白了自身知识的匮乏，更知道了面对问题，面对未知，我应该怎么去面对，因为问题迟早要处理，不然它会永远存在。计算机的未来一片光明，这条路没有尽头，我们的空间还很大，在前进的路上如逆水行舟，不进则退，不加油就会被取代，只有不断的努力才能得以进步。

8. 致谢

此次毕业设计和论文均在毕业指导老师林岭老师的悉心教导下设计并编写，林岭老师在今年全球疫情这种特殊的情况下依旧花费大量时间编写文章指导我们毕业论文应该如何编写及其细节。林岭老师期间多次与我们进行视频会议，在会议中与我们探讨项目的各个方面以及可以改进的部分。不管多忙老师都会给我们解疑答惑，在我们进步的道路上为我们披荆斩棘，可以说为了我们能从毕业设计中学习到更多东西费尽心力。再次感谢林岭老师的耐心和细心，以及为我们这群大小孩所做的一切。同时还要感谢过去四年中教导过我的老师以及给我提过建议和帮助过我的亲朋好友。

参考文献

- [1] 鲁特兹, M. Python 学习手册[M]. 机械工业出版社, 2011.
- [2] 黄志艳. 如何提高学生计算机专业英语学习的兴趣[J]. 科技创新导报, 2011 (30): 210-211.

- [3] 肖建. Python 编程基础[M]. 清华大学出版社, 2003.
- [4] 秦昶, 史晓楠. 《程序设计基础》课程教学改革实践——以 Python 语言为教学语言[J]. 软件导刊(教育技术), 2018, v. 17; No. 158(02): 76-77.
- [5] 刘卫国, 奎晓燕. Python 语言程序设计教学体系建设[J]. 计算机教育, No. 272(08): 13-17.
- [6] 王亚萍. 关于《Python 程序设计》课程改革的几点思考[J]. 智库时代, 2019, 191(23): 190-191.
- [7] 黄宏博. 以 Python 语言作为高校程序设计课程主语言的探讨[J]. 亚太教育, 2015(26): 285.
- [8] 郑千忠, 孙晓芳. Java Web 开发课堂实录[M]. 清华大学出版社, 2016.
- [9] 甘文丽, 王岚. 基于轻量级 Java EE 框架的高校招投标管理系统[J]. 实验室研究与探索, 2012, 31(11): 182-185.
- [10] 蔡金华. 基于 Java Web 的开发技术[J]. 电子技术与软件工程, 2019, 000(006): P. 53-54.
- [11] 吕海熊. SQL 语言到自然语言自动翻译的研究[D]. 北京理工大学.
- [12] 王世纯, 许新华, 张洪春, et al. 个体差异对 Python 爬虫获取教育大数据的影响研究[J]. 中国教育信息化, 000(9): P. 79-81.
- [13] 李明铭. 基于 Hadoop 的网络聚焦爬虫抓取策略和解析方法研究[D]. 武汉理工大学.
- [14] 李红. 移动终端学习资源平台应用程序开发[J]. 电子技术与软件工程, 000(020): P. 44-44.
- [15] 朱贇. Python 语言的 Web 开发应用[J]. 电脑知识与技术, 2017, 013(032): 95-96.

Design And Development Of Computer Dictionary System

LEI Meng-hao 105032016063 Advisor: LIN Ling

Major in Computer Science and Technology College of Mathematics and Informatics

【Abstract】 Dictionary system of Computer Science is a system that enables users to search professional vocabulary, abbreviations and other information of the computer industry efficiently. This dictionary is based on Java Web, and uses Python to implement crawler to obtain the basic data. The crawler can get the data from the professional website, which greatly improves the reliability of the data. At the same time, it also saves a lot of time to enter the dictionary information by manpower, and greatly reduces the development pressure. This paper will describe the important steps of website development from the perspective of developers.

【Keywords】 AbbreviationQuery Of Computer Science; Java web; python