
英语单词速查系统设计与实现

数学与信息学院 计算机科学与技术专业

105032016027 汪金木 指导教师：林岭

【摘要】随着计算机技术的不断发展，人们使用计算机不再只是单纯地浏览消息、发送邮件、聊天与游玩，更多的是为了达到学习的目的。设计一个英语单词速查系统，进而为用户学习英语提供一定的便利，同时还能够尽可能的满足用户的实际学习需求。在对英语单词速查系统设计与开发的过程中，运用的是C/S模式，本文基于Python编程语言的功能特点以及爬虫技术设计并实现一套英语单词速查系统，并将数据存储到SQL Server中。系统基于C#语言进行开发，有用户管理、词库管理、单词的搜索、修改、删除及增加等。在系统设计的进程中，本着可维护性、有效性、实用性的原则，因此运用三层体系架构来进行开发，不但减少了代码的重复实用，而且在提高开发效率方面有着积极作用，便于系统后期的管理与维护。另外，为了以后增加相应的功能，还提供了相应的接口。

【关键词】C/S；英语学习；三层体系结构

目录

1. 引言	- 3 -
1.1 系统设计背景	- 3 -
1.2 系统设计意义	- 3 -
1.3 系统贡献	- 3 -
1.4 开发语言 Python 与 C#	- 3 -
1.5 数据库理论	- 3 -
2. 系统需求	- 4 -
2.1 需求说明	- 4 -
2.2 用例图	- 4 -
2.2.1 功能模块图	- 4 -
2.2.2 系统功能	- 4 -
2.3 功能需求	- 7 -
3. 系统设计	- 7 -
3.1 可行性研究	- 7 -
3.1.1 经济可行性	- 7 -
3.1.2 技术可行性	- 7 -
3.1.3 操作可行性	- 7 -
3.2 现行业务系统描述	- 8 -
3.3 单词爬取流程分析	- 7 -
3.5 系统功能设计	- 8 -
3.6 数据库设计	- 9 -
4. 系统详细设计与实现	- 11 -
4.1 系统架构设计	- 11 -
4.2 搜索引擎抓取工作过程设计	- 12 -
4.3 爬行策略设计	- 13 -
4.4 系统实现	- 14 -
4.4.1 登陆首页	- 14 -
4.4.2 单词速查界面	- 14 -
4.4.3 单词复习测试页面	- 15 -
4.4.4 主代码展示	- 15 -
5. 系统测试	- 17 -
5.1 测试方法	- 17 -
5.2 测试环境	- 18 -
5.3 测试对象	- 18 -
5.4 测试用例	- 18 -
5.4.1 登录页面测试用例	- 18 -
5.4.2 我的收藏测试用例	- 18 -
5.4.3 词库管理测试用例	- 18 -
5.4.4 翻译功能测试用例	- 18 -
5.5 测试结果	- 18 -
6. 结 论	19
6.1 系统优点与不足	19
6.2 用户体验调查分析	19
7. 结论与系统扩展	19
致谢	19
参考文献	20

1. 引言

1.1 系统设计背景

单词记忆学习就是借助于计算机展开的相关单词学习，通过对自身所学内容的选择，对学习情况进行测试。随着计算机的普及应用，单词助记系统的设计与开发将会更加具有优越性，知识是无穷无尽的，学习知识具有交互性、时效性、共享性特点。这不但学习者提供了丰富的资源，还对学习内容进行了一个合理的管理，给自学人员的学习带来更大的便捷性，无形中提高了学习的效率节约了学习的资源。自学系统的侧重点就是集中在词库的管理设计方面，词库管理尤为关键，词库的好与坏，将会直接影响到系统的运作，给自学人员的学习将会带来一定的影响。对此，本设计的设计与开发可以满足自学者的自我学习管理，为其提供良好的学习条件。

1.2 系统设计意义

英语单词速查系统利用现有的技术，通过计算机从网上爬取丰富的单词资源为学习者建立一个知识储备库，并且还可通过自学的自我所需进行人性化的单词添加、删除、修改等操作。英语单词速查系统为自学者们提供了一个 24 小时的学习机会，克服了时空限制，随时随地，想学的就学；从根本上弥补了资源方面的欠缺，使得将所有英语知识融为一体。学生通过该系统可以自主、高效的去学习。

1.3 系统贡献

本文研究核心就是构建在上文所述背景及意义基础上，对网络数据信息爬取和存储，用户可以通过搜索，使用户的学习更加方便，可以满足用户的学习需求，对提高用户的学习效率提升发挥出了积极作用。

1.4 开发语言 Python 与 C#

Python 作为典型的计算机程序设计语言，具有语言简洁、跨平台以及功能强大及丰富的特征。是一个高层次的解释性、编译性、互动性和面向对象的脚本语言。在采用 Python 来对英语单词速查系统设计与开发的进程中，能够直接对其标准库进行引用，一定程度的提升了开发的效率。另外，采用该语言来开发系统是相对简单的，通俗易懂，能够运用最简单的代码来对最复杂的程序进行开发。相对于其他类型的开发语言来说，Python 最明显的特征便是自动化求解与抽象化解决方案。但 Python 的缺点也很明显，就是运行速度很慢。Python 语言最初是用来编写 shell，伴随着版本的更新换代，Python 越来越多的被用于独立的、大型项目开发。

C# (C sharp) 是对这一问题的解决方案。C# 在系统开发领域也得到了较为广泛的运用。特别是针对于基于 .NET 平台的应用程序来说，其编写效率是非常高的。使用 C# 语言结构可以有效的转化为 XML 计算机服务，并且可以在任何计算机系统上进行操作，是一种非常有效实用的编程语言。C# 语言不但对 C/C++ 语言的功能进行了继承，还可以更高效的开发程序。C# 与 C/C++ 因为存在着继承关系，所以他们在很多相似的地方，熟悉类似语言的开发者可以很快的熟悉 C# 语言。

C# 是可以自动管理的，它适合于绝大多数应用程序的开发，但是需注意的是，由于受到应用程序接口兼容问题的影响，导致一些应用程序的开发仍然需要运用 C++ 语言。

通常来说，一般 C# 运用以下两个策略来对问题进行解决：其一，允许有限制地使用纯指针；其二，内置对组件对象模型与基于 Windows 的 API 的支持。在 C# 中，所有的 COM 对象都能够自动生成，意味着无需其他类型的接口。

1.5 数据库理论

Microsoft SQL Server 2005 作为一种客户端/服务器结构的关系数据库管理系统，相对于其他类型的数据库管理系统来说，在性能方面有着明显的优势。另外，还提供了更可靠、更安全的存储方式，因此在高性能、高可用的数据应用程序开发期间，采用 SQL Server 2005 是最合适的选择。

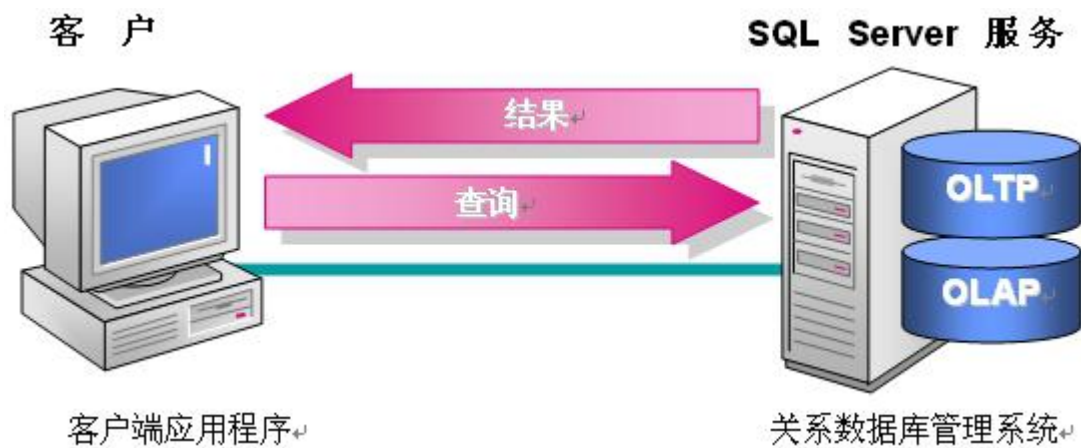


图 1-1 SQL Server 简介

2. 系统需求

2.1 需求说明

在英语单词速查系统中主要为两部分，首先是数据库数据的网络爬虫，另一方面是整个系统的各个功能。网络爬虫是将网络上的英语单词及中文解释整合在一个系统上，节省了时间。提高了效率。基础功能包含：用户管理、词库管理、每日一贴、单词复习、我的收藏、单词测试、翻译、单词导入。实现了单词的查找、学习与记录，实现了单词一体化的操作，方便使用者的学习。

2.2 用例图

2.2.1 功能模块图

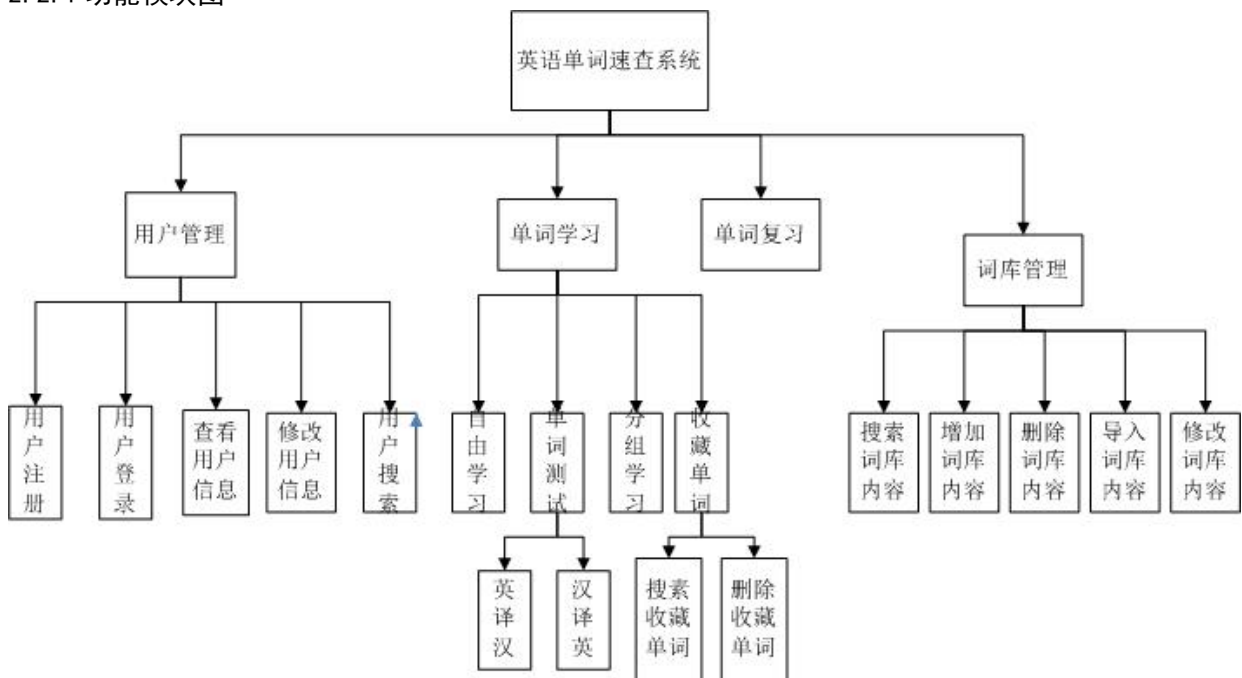


图 2-1 功能模块图

2.2.2 系统功能

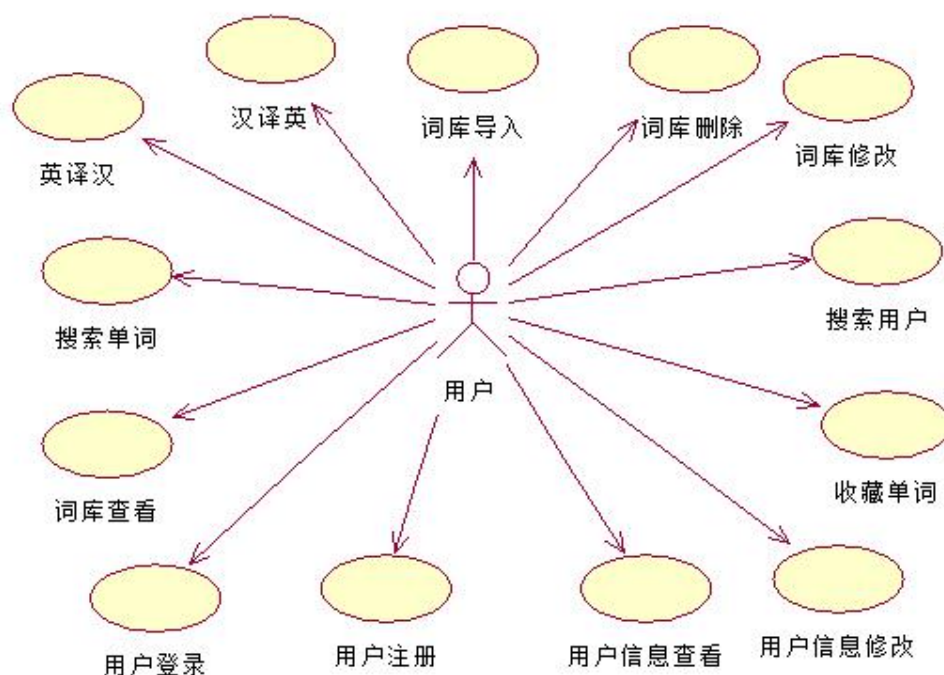


图 2-2 系统功能图

用户系统需具备以下功能：

- (1) 登录的用户在登录时，若首次登录，需要注册，注册后可直接登陆；
- (2) 登录的用户可以进行单词学习，可进行自由学习、单词测试、分组学习、收藏单词；
- (3) 登录用户可进行词库管理，包括单词的搜索、添加、删除、修改等。

表 2-1 用户注册用例规约

用例名称：	用户注册	
用例 ID：		
角色：	普通用户	
用例说明：	该用例主要功能是用户注册用户信息，用于登录	
前置条件：	用户打开软件，点击“注册”键	
基本事件流：	参与者动作	系统响应
	1. 用户打开软件，点击“注册”键； 3.填写注册信息，点击“保存键”；	2.系统响应点击事件，进入注册页面； 4.系统响应判断点击事件：‘退出’则退回注册页面；‘保存’则注册成功
备选流：		
扩展点：		
后置条件：	完成用户注册	

表 2-2 用户登录用例规约

用例名称:	用户登录	
用例 ID:		
角色:	普通用户	
用例说明:	该用例主要功能是用户登录	
前置条件:	用户打开软件, 输入账号密码, 点击“登录键”	
基本事件流:	参与者动作	系统响应
	1. 用户打开软件, 输入账号密码, 点击“登录键”;	2. 系统响应点击事件, 完成信息审核, 登录系统;
备选流:		
扩展点:		
后置条件:	完成用户登录	

表 2-3 用户搜索用例规约

用例名称:	用户搜索	
用例 ID:		
角色:	普通用户	
用例说明:	该用例主要功能是搜索用户	
前置条件:	用户登录, 进入用户搜索界面;	
基本事件流:	参与者动作	系统响应
	1. 用户登录, 进入用户搜索界面; 2. 填写需要搜索的用户名, 点击“搜索”键	2. 系统响应点击事件, 进入搜索用户界面; 4. 系统响应点击事件, 完成搜索用户功能;
备选流:		
扩展点:		
后置条件:	完成用户搜索	

表 2-4 用户搜索单词用例规约

用例名称:	用户搜索单词	
用例 ID:		
角色:	普通用户	
用例说明:	该用例主要功能是搜索单词	
前置条件:	用户登录, 进入翻译界面;	
基本事件流:	参与者动作	系统响应
	1. 用户登录, 进入翻译界面; 3. 填写需要搜索的单词, 点击“英译中”键; 5. 填写需要搜索的汉语, 点击“中译英”键	2. 系统响应点击事件, 进入翻译界面; 4. 系统响应点击事件, 完成搜索功能; 6. 系统响应点击事件, 完成搜索功能;
备选流:		
扩展点:		
后置条件:	完成用户搜索单词	

表 2-5 用户单词导入用例规约

用例名称:	用户单词导入	
用例 ID:		
角色:	普通用户	
用例说明:	该用例主要功能是导入新单词	
前置条件:	用户登录, 进入词库管理界面;	
基本事件流:	参与者动作	系统响应
	1. 用户登录, 进入词库管理界面; 3. 点击“新增单词”键; 5. 填写单词信息, 点击“添加”键, 或者“关闭键”;	2. 系统响应点击事件, 进入词库管理界面; 4. 系统响应点击事件, 进入添加单词界面; 6. 系统响应点击事件, “添加”完成单词的添加, “关闭”退出操作返回词库管理界面;
备选流:		
扩展点:		
后置条件:	用户实现单词的导入	

2.3 功能需求

系统设计与开发主要就是分为两个部分。其一, 实现系统基本功能; 其二, 数据库词汇表数据的爬取写入。第一部分主要就是由用户管理、词库管理、单词的搜索、添加、删除、修改功能组成。本系统设计, 主要涉及到以下几个表, 包括: 单词表、用户表及储藏表, 而系统的功能主要是使用到单词表和储藏表。Python 程序方面主要就是针对于词汇表数据加以维护, 并借助于 Python 程序, 将相关数据信息录入到表中, 只有完成这一步骤之后, 才能够实现单词表的操作。用户表数据来源由用户注册。数据库中单词表的数据来源于 python 程序爬取获得。

3. 系统设计

3.1 可行性研究

该阶段通过查询资料, 咨询相关人员。完成了系统的初期目标设定, 对系统进行分析和调研, 依照研究结果制定科学系统设计与开发方案, 并从经济、技术、操作可行性的角度进行深入的探究和分析。

3.1.1 经济可行性

系统设计与开发过程中, 需要参照相关资料才能够顺利展开, 这就需要对资料进行收集, 资料的获取主要就是通过网上直接获得, 开发成本相对来说偏低, 且开发编程环境是免费, 因此只需要在计算机上安装好相应的开发编程环境即可。对于该系统后期的维护与系统功能开展的费用也较低, 进而从经济的层面来分析, 系统的设计与实现是在可控的范围的。

3.1.2 技术可行性

技术可行性着重考量的问题, 就是当前技术条件能否满足系统的开发要求, 软件及硬件配置能够真正满足系统开发需求等。目前的 python 和 c# 技术已经相当的成熟稳定, 网络上也有相关的案例进行学习, 且该技术在课堂上有学习过, 因此对于开发的技术方面上是能够顺利开展的。所以, 在技术的角度上来看, 该系统可以实现的。

3.1.3 操作可行性

当前已经进入了信息化网络时代, 计算机的使用已经是十分普及了, 而对于本系统中, 界面简洁友好, 易于操作。所以在操作的角度上来看, 该系统是可行性的。

通过以上内容的分析, 了解到系统的设计与开发, 从经济、技术、操作方面都具有可行性。

3.2 现行业务系统描述

现阶段，我国教育模式始终还是以传统教学模式为主，也就是老师为主体，向学生传授相关专业知识。但此学习模式已经难以满足学习者完成自我学习需求，难以掌握自己的学习情况，无形中会减弱学生的学习热情，甚至还会影响学习效率。与此同时，英语学习方式也有创新，比如视频、音频广播等，尽管此方式可以弥补传统学习方法存在的欠缺，但却难以真正的做到提高学习效率，实现学生的自主学习。

3.3 单词爬取流程分析

运用数据挖掘技术来抓取网络信息，并将其存储到数据库，其次运用程序访问这些数据，实现一个单词速查系统。

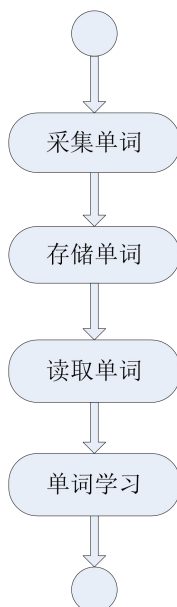


图 3-1 单词查询流程图

3.5 系统功能设计

首先，因英语单词速查系统的设计与开所面向的是特定群体，所以系统设计应该由管理人员进行专门设计，应该根据特定群体身份权限进行具体设定，避免英语词库内容出现管理方面的问题；其次，针对于系统使用用户群体而言，英语词库内容应该尽量清晰、充足、分类合理，这样才能够更加方便用户的学习；此外，系统界面设计还应该美观、可操作。

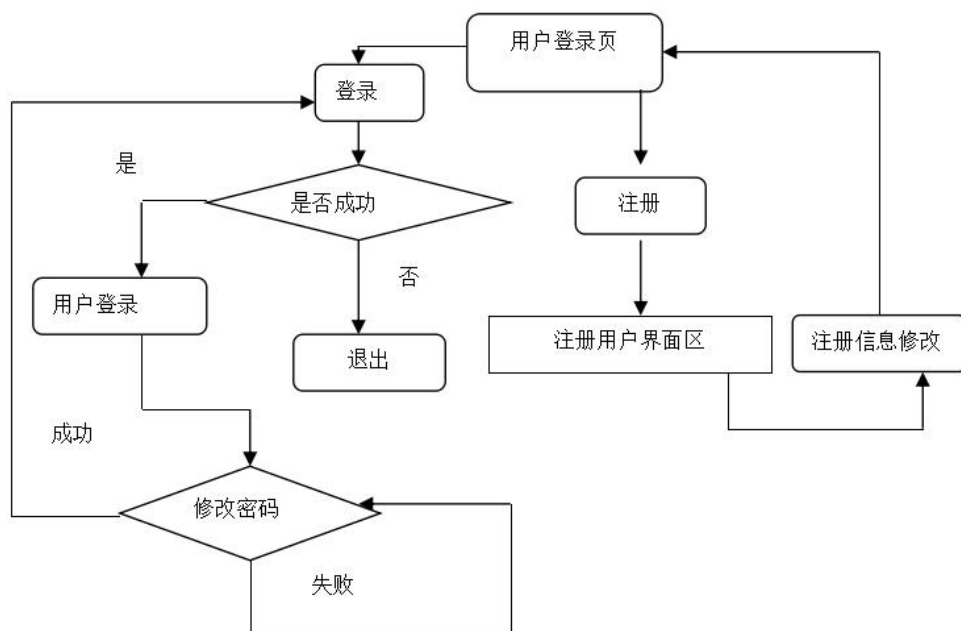


图 3-2 E-R 图

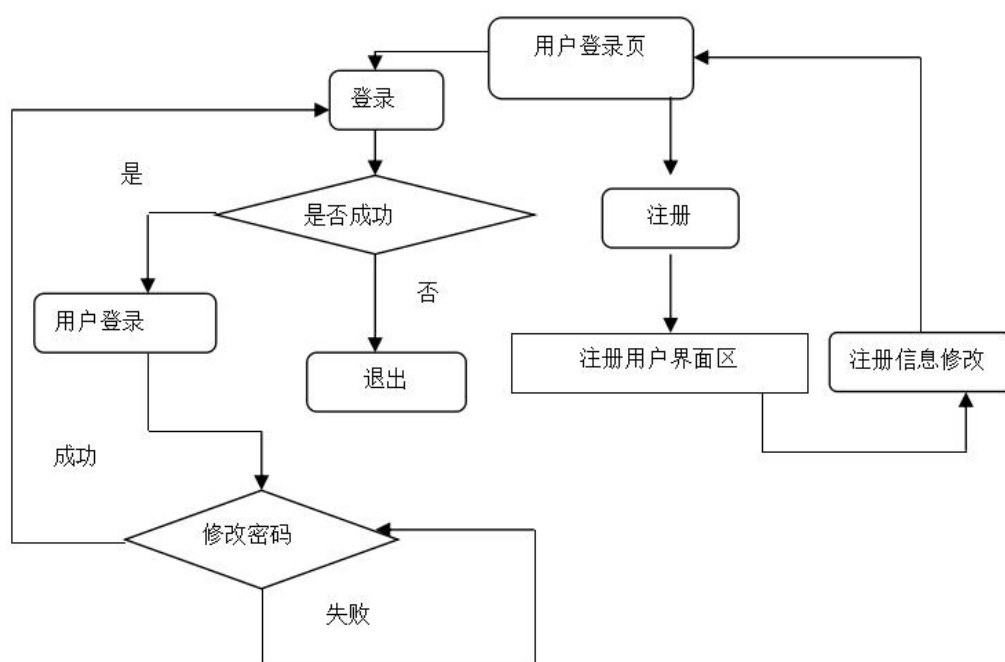


图 3-3 登录流程图

3.6 数据库设计

数据库是一种可以存储数据并对数据进行操作的工具。数据库实现了数据共享、减少数据的冗余度、数据的独立性、数据可以实现集中控制、数据保持一致性和维护性、故障修复。在整个系统中，数据库的作用不可替代，系统的运行效率、质量都与数据库有着密切的关系，对此，合理的、科学的来设计数据库是非常有必要的。根据系统设计的关系模型有以下用户，单词，单词收藏。

用户（用户ID, 真实姓名, 用户名, 登录密码, 性别, 出生日期, 用户类型, 邮箱, 用户状态, 联系电话, 身份证号）-----具体补齐，主键底下加横线

单词（单词ID, 翻译, 音标, 单词, 英文语言, 中文语音, 缩写, 分组, 单词类型）

单词翻译（用户名, 单词ID, 单词名称, 单词ID, 单词翻译ID）

表 3-6-1 用户表 (User)

字段名	类型	长度	是否为空	主键	说明
userID	int	11	否	√	用户 ID
UserName	varchar	50	是		用户姓名
PassWord	varchar	50	是		登录密码
IdCard	varchar	50	是		身份证号
Sex	varchar	50	是		性别
BirthDay	datetime		是		出生日期
RealName	int	50	是		真实名称
Email	varchar	50	是		邮箱
Telephone	varchar	50	是		联系电话
Remark	varchar	50	是		用户状态
UserType	int	11	是		用户类型

表 3-6-2 单词表 (Word)

字段名	类型	长度	是否为空	主键	说明
WordID	int	11	否	√	单词 ID

Word	varchar	500	是	单词
Translate	varchar	500	是	翻译
YinBiao	varchar	50	是	音标
Type	varchar	50	是	单词类型
Fenzu	varchar	50	是	分组
Zhongwenyuyin	varchar	590	是	中文语音
Yingwenyuyan	varchar	580	是	英文语言
Suoxie	varchar	50	是	缩写

表 3-6-3 单词翻译表 (Storage)

字段名	类型	长度	是否为空	主键	说明
Storage	int	11	否	√	单词翻译 ID
UserID	int	11	是		用户 ID
UserName	varchar	500	是		用户名称
WordID	int	11	是		单词 ID
WordName	varchar	50	是		单词名称

根据对系统的思路分析，画出数据库E-R模型图如图所示。

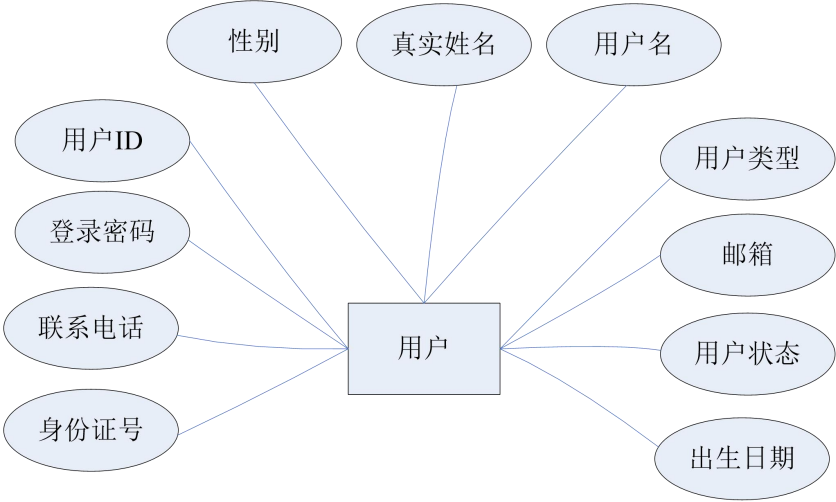


图 3-4 用户实体及属性

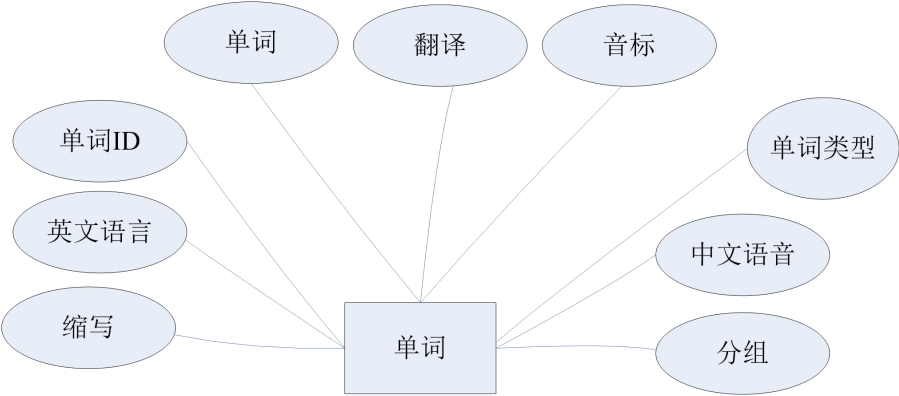


图 3-5 单词实体及属性

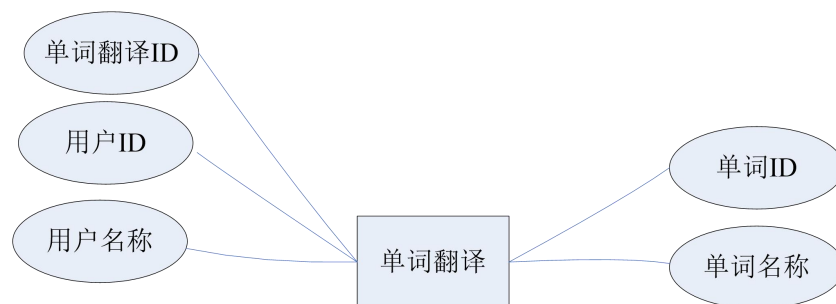


图 3-6 单词翻译实体及属性

4. 系统详细设计与实现

4.1 系统架构设计

逻辑层(中间层、应用服务器)是整个系统设计的重点和难点。针对于逻辑层来说,其核心功能是对数据请求进行封装。假如应用服务器很繁忙时,则应用服务器可根据易于管理原则来对业务进行分配,通过协调的方式以实现正常运转。

Provider 技术在本系统中得到了广泛的使用,对 DB 中数据才操作都被抽象为独立的接口式业务逻辑。前端代码只需要调用接口的方式,便可得到反馈。

针对于 SPIDER 体系结构来说,是由三个分支构成的。其一,任务执行端,该分支与一个网站有着密切的关系,在线程执行下载网页任务的过程中,在实际操作的过程中,首先需要对 URL 进行识别,然后通过分析页面的方式便可得到已知和新发现的站点 URL,在此背景下便可将得到的 URL 对列保存到数据库当中,这便是断点继传的根本所在。

synchronized 在 SPIDER 体系结构中的协调通信处于核心地位,对于系统处理速度的加快有着重要意义。另外,任务调度端在 SPIDER 体系结构中同样处于重要的地位,与任务执行端的关系是尤为密切的,只有在相互合作、协调的情况下,才能够发挥出其作用与价值。两者的 URL 队列中存储内容存在着一定的差异,其中一个存放的是站点的 URL,另一个存放的是站内 URL,可对其进行增加、查询、删除、修改等相关操作。假如简单的将其存储到内存中,则很容易导致出现内存空间不足的情况;假如直接将数据存储到数据库,由于数据量是非常大的,因此其效率必然偏低。

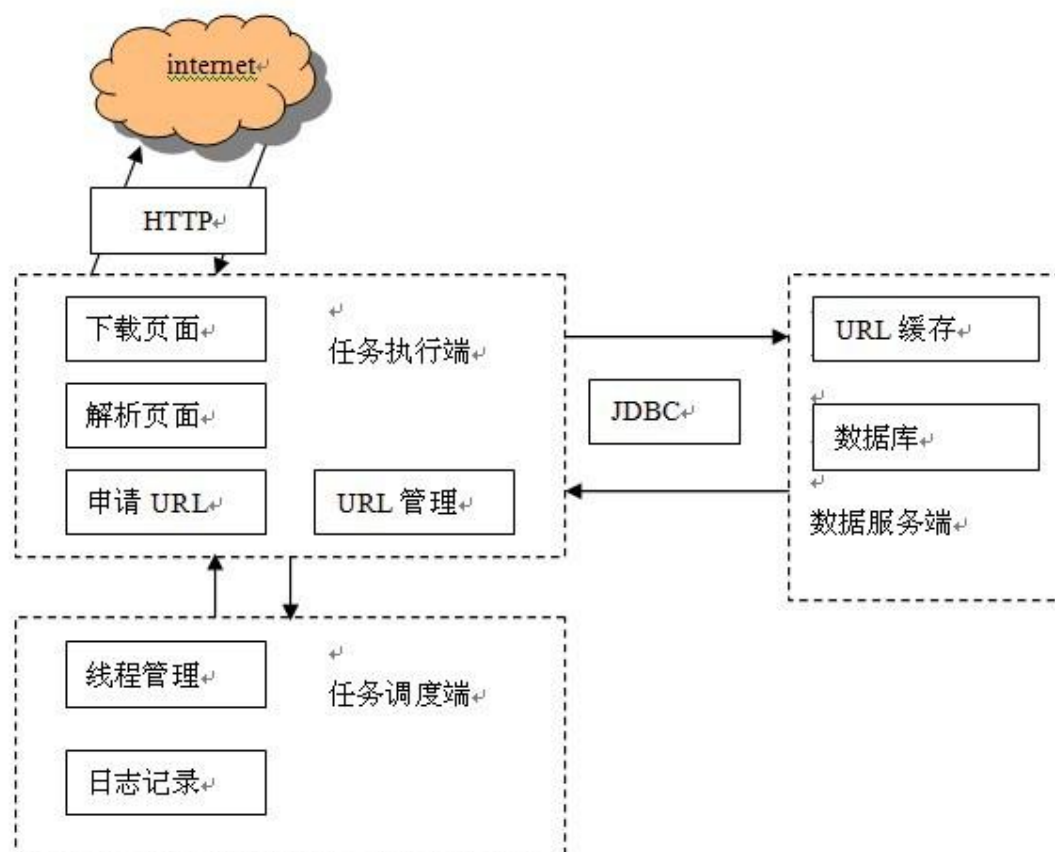


图 4-1 SPIDER 体系结构

4.2 搜索引擎抓取工作过程设计

在搜索引擎抓取工作进行的过程中，主要是由四个步骤构成的：

第一步：将初始 URL 加入到 URL 等待队列；

第二步：对爬虫线程进行创建并启动；

第三步：获取任务 URL，并对网页中全部的页面进行分析，得到 URL，通过仔细分析后假如属于相对地址，则将其变为绝对地址，并将站外 URL 去掉。假如 URL 无法解析或错误的 URL，则需要来对地址是否下载过进行判断，如果没有则放到队列，如果有则不必放入队列；

第四步：继续执行第三步，直到结束条件停止。

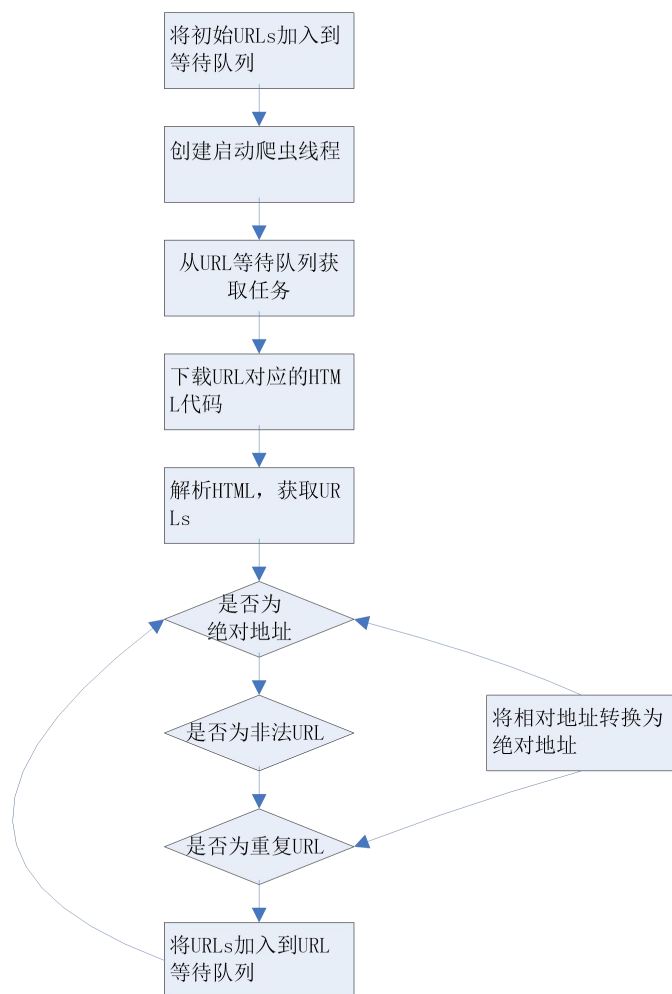


图 4-2 URL 工作原理

4.3 爬行策略设计

系统想要一次对网站中的所有页面进行访问，并对所包含的数据进行获取，那么最合理、最有效的算法便是广度优先算法，算法的特征便就是能够尽可能的覆盖更多的结点。在实际操作的过程中，首先需对 URL 进行获取，然后对相关 HTML 代码进行下载，便能够对网页中的全部 URLs 进行收集与整理，并一层一层的获取，最终达到目标。

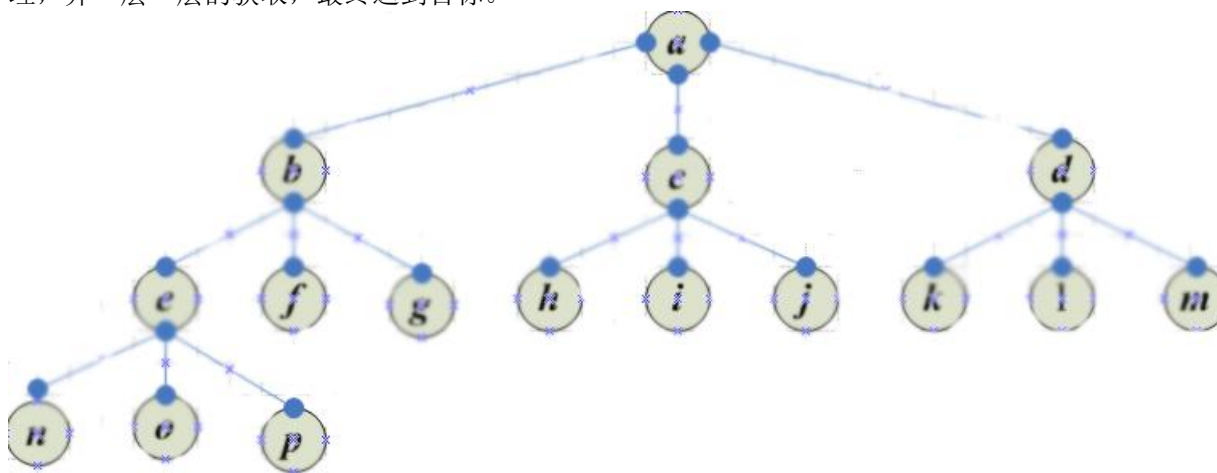


图 4-3 广度优先算法遍历

其中，广度优先算法的遍历具体如图 4-3 所示，a 代表的是开始的 URL，获取后便可得到 3 个 URLs，用 b、c、d 代替。然后依次采用这种算法，b 与 c 分别对应的是 e f g 和 h i j。将同一层次的 URLs

全部解析完毕后，便会进入到下一次来继续依次解析，直到所有的 URLs 都解析完毕。

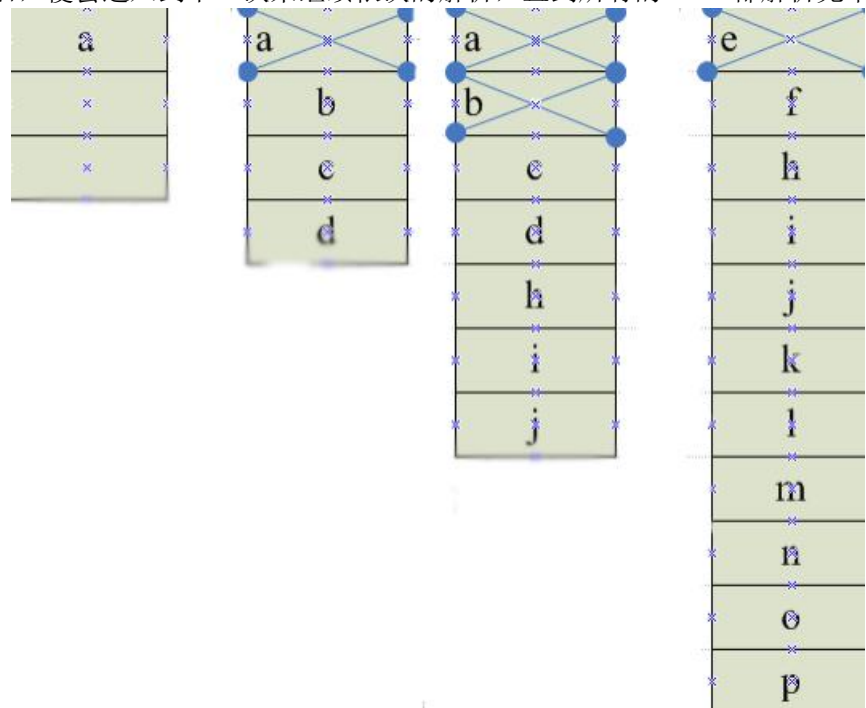


图 4-4 广度优先算法的等待队列

关于广度优先算法的等待队列，具体如图 4-4 所示，代表的是各个结点在不同阶段的存储状态。具体来说，第一列方框代表的是初始的 URL，可看出此时 a 加入到等待队列。第二列方框则说明 b c d 已经进入到队列中，来对对应的 URLs 进行获取的同时，将 a 删除。第三列方框则是解析 b 获得 e f g，同时将 b 的 URL 删除。第四列方框则是对 e 对应的 URLs :n o p 进行获取，同时将 e 删除。

4.4 系统实现

4.4.1 登陆首页



图 4-5 登录界面

4.4.2 单词速查界面

通过将采集的单词进行整理和设置，用户可以查询单词。

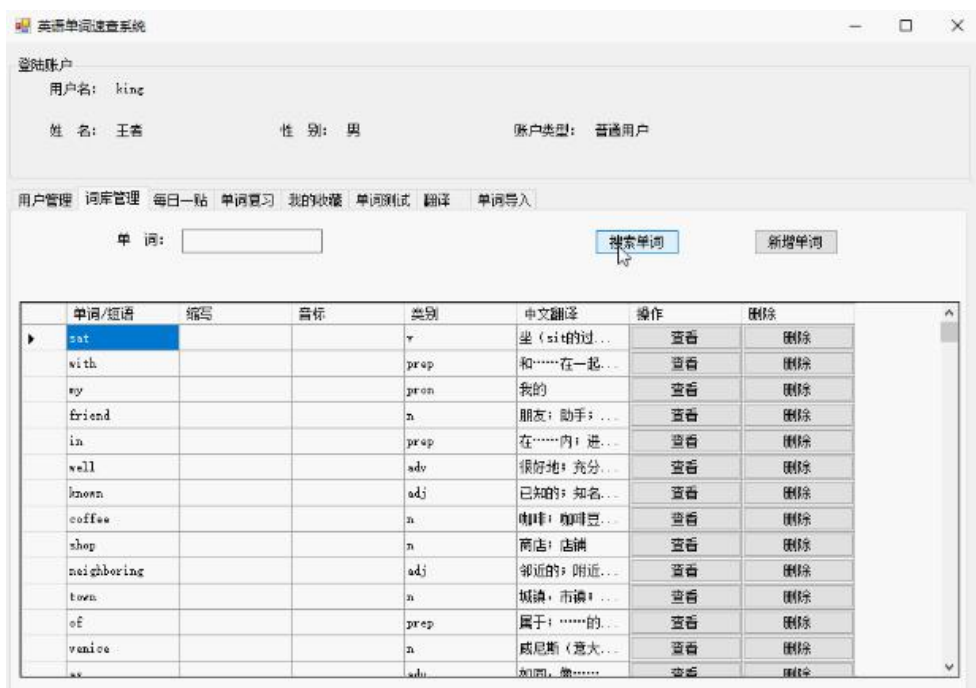


图 4-6 主界面

4.4.3 单词复习测试页面

内设 10 个单词，可进行复习测试。



图 4-7 测试界面

4.4.4 主代码展示

爬虫代码展示

```

0 def get_word(word):
1     #获得页面
2     pagehtml=get_page(word)
3     selector = etree.HTML(pagehtml.decode('utf-8'))
4     #单词释义
5     chitiao=get_chitiao(selector)
6     #单词音标及读音
7     yingbiao=get_yingbiao(selector)
8     #例句
9     liju=get_liju(selector)
0     return "%s\t%s\t%s\t%s"%(word,yingbiao,chitiao,liju)
1
2     filename='5.txt'
3     f=open(filename,"r")
4     words=f.readlines()
5     f.close()
6     filename2='5_jieguo.txt'
7     f=open(filename2,"wb")
8     i=0
9     for word in words:
0         time.sleep(0.2)
1         print(word.rstrip(),i)
2         word_line=get_word(word.rstrip())
3         f.write("%s\n"%(word_line))
4         i=i+1
5     f.close()

```

用户登录界面

```

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        if (txtName.Text == string.Empty || txtPass.Text == string.Empty)
        {
            MessageBox.Show("请输入用户名和密码!");
            return;
        }
        DataSet ds = MainConfig.GetRecord("select * from [user] where UserName='"+txtName.Text+"' and PassWord='"+txtPass.Text+"'");
        if (ds != null)
        {
            if (ds.Tables[0].Rows.Count > 0)
            {
                MainConfig.Usersds = ds;
                MainWindow mainwindow = new MainWindow();
                mainwindow.Show();
                this.Hide();
            }
            else
            {
                MessageBox.Show("登录失败!");
                return;
            }
        }
        else
        {
            MessageBox.Show("登录失败, 数据库操作失败!");
            return;
        }
    }
}

```

用户信息管理


```

private void MainWindow_Load(object sender, EventArgs e)
{
    if (MainConfig.Usersds != null)
    {
        DataRow dr = MainConfig.Usersds.Tables[0].Rows[0];
        this.lblUserName.Text = dr["UserName"].ToString();
        lblRealName.Text = dr["RealName"].ToString();
        lblSex.Text = dr["Sex"].ToString();
        lblType.Text = dr["usertype"].ToString() == "1" ? "管理员" : "普通用户";
        if (dr["UserType"].ToString() == "1")
        {
            tabControl1.TabPages.Remove(tabUser);
            tabControl1.TabPages.Remove(tabPage2);
        }
        else
        {
            InitUser();
            InitWord();
        }
    }
}

```

单词搜索功能

```

private void dataWord_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    string action = dataWord.Columns[e.ColumnIndex].Name; //操作类型
    if (action == "wordcaozuo")
    {
        string id = dataWord.Rows[e.RowIndex].Cells[7].Value.ToString();
        MainWindow.wordID = id;
        WordDetail detail = new WordDetail();
        detail.Show();
    }
    if (action == "wordshanchu")
    {
        if (MessageBox.Show("确认删除?") == DialogResult.OK)
        {
            string id = dataWord.Rows[e.RowIndex].Cells[7].Value.ToString();
            string sql = "delete [word] where id='" + id + "'";
            MainConfig.Excute(sql);
            DataSet ds = MainConfig.GetRecord("select * from [User]");
            dataWord.AutoGenerateColumns = false;
            dataWord.AllowUserToAddRows = false;
            dataWord.DataSource = ds.Tables[0].DefaultView;
        }
    }
}

```

单词小测试功能

```

private void button10_Click(object sender, EventArgs e)
{
    button10.Text = "开始测试...";
    button10.Enabled = false;

    DataSet ds = MainConfig.GetRecord("select top 1 * from word order by newid() desc");
    if (ds != null)
    {
        if (ds.Tables[0].Rows.Count > 0)
        {
            tempDC = ds.Tables[0].Rows[0]["word"].ToString();
            this.lblCs.Text = ds.Tables[0].Rows[0]["translate"].ToString();
        }
    }
}

```

5. 系统测试

5.1 测试方法

在登录用户，使用系统功能，执行指定操作，测试各项的功能。

5.2 测试环境

visual_studio_community_2017

5.3 测试对象

用户管理、词库管理、每日一贴、单词复习、单词测试、翻译、单词导入

5.4 测试用例

5.4.1 登录页面测试用例

表 5-1 登录页面测试用例表

用例编号	前置条件	操作	预期结果	测试结果
001	进入登录页面，选择登录	输入正确用户名和密码，点击登录	登录成功，跳转到主页	通过
002	进入登录页面，选择登录	输入用户名，点击登录	提示登录失败	通过
003	进入登录页面，选择登录	输入密码，点击登录	提示登录失败	通过
004	进入登录页面，选择登录	输入用户名，错误密码，点击登录	提示登录失败	通过

5.4.2 我的收藏测试用例

表 5-2 我的收藏测试用例表

用例编号	前置条件	操作	预期结果	测试结果
001	进入收藏页面，选择登录	输入想要查找的单词，点击搜索	搜索成功，页面出现收藏单词	通过
002	进入收藏页面，选择登录	输入错误单词，点击搜索	无搜索结果	通过

5.4.3 词库管理测试用例

表 5-3 词库管理测试用例表

用例编号	前置条件	操作	预期结果	测试结果
001	进入词库管理页面，选择单词	填写正确单词，选择搜索单词	查找出指定单词	通过
002	进入词库管理页面，选择单词	填写错误单词，选择搜索单词	搜索结果空白	通过
003	进入词库管理页面，选择新增单词	新增单词，填写单词信息，选择添加	提示添加成功	通过
004	进入词库管理页面，选择新增单词	新增单词，填写单词信息，选择关闭	退出新增单词界面	通过

5.4.4 翻译功能测试用例

表 5-4 翻译功能测试用例表

用例编号	前置条件	操作	预期结果	测试结果
001	进入翻译页面，选择单词	填写正确单词，选择英译中	查找出单词汉语意思	通过
002	进入翻译页面，选择单词	填写错误单词，选择英译中	无翻译结果	通过
003	进入翻译页面，选择所翻译的汉语	填写正确汉语，选择中译英	查找该汉语的英语意思	通过
004	进入翻译页面，选择所翻译的汉语	填写正确汉语，选择中译英	无翻译结果	通过

5.5 测试结果

通过系统测试及优化，该系统现阶段已经能够正常运行与使用。

6. 结 论

6.1 系统优点与不足

在对该系统开发期间，前台开发工具选择的是 VS，主体编写运用的是 C#语言，数据采集运用的是 Python 语言。首先对系统的需求进行了研究与分析，然后根据实际需求开展了总体设计，最后进行了详细设计。本系统实现了英语单词速查，集单词的查找储存、复习记忆、新单词的录入，可实时录入新的单词，提高了学生的学习效率。但是系统也存在着不足，爬虫数据单一，功能不齐全等。

6.2 用户体验调查分析

本系统使用人群主要为自学英语者提供便利。通过对使用该系统的学生进行调查与统计后，了解到绝大多数同学都认为该系统对于英语的学习具有积极作用。但是需注意的是，由于该系统在界面设计太过于单调；还有部分体验者反馈希望可以上线或者手机版，方便学习者可以随地随时的使用。

7. 结论与系统扩展

本次毕设在总体上基本完成了期初定下的设想，在系统设计与开发过程中增加了一些全新的设计理念，使得系统功能运行过程中更为顺畅。伴随着计算机技术的提升，计算机的普及应用，各项技术的不断更新，在开发过程中，遇到了很多没有接触过的问题，在系统设计与开发之前，我并未学习和接触过 Python，尽管 Python 的入门基础要求不高，并且比较好学习，但学习过程中会遇到很多新问题，为解决这些问题，需要去百度搜索相关问题解答答案。如果时间充足，我会更好的完善界面的一些设计，让用户有更好的体验，另外希望可以把整个系统上线，做成一个完完整整的 APP 来推向市场，让自己做的东西可以让更多人受益。

通过此次毕业设计的撰写，让我从中学习到了很多知识，同时也对很多新知识有了一定的了解，例如，学习了 Python 语言，还通过论文写作，巩固了学习过的知识，对大学学习期间接触到的知识都重新温习了一遍，做到了学以致用。通过本次毕业设计，我更加了解了自身存在的不足之处，在以后的学习和工作中，我会继续努力提高自身能力。

致谢

在此次毕业设计撰写的过程中，得到了很多人的支持与帮助，在此我要感谢我的指导老师，感谢我的同学和朋友，感谢所有人给予我的鼓励。首先，我要感谢指导教师，感谢您对论文写作的批评指正，从您的身上我学习到了很多，也懂得了很多的道理，感谢您给予论文写作提出的建议。如果没有您的帮助，毕业设计就不会如此顺利的截稿。

其次，感谢所有的任课老师，感谢你们传授我们专业知识，让我们在校期间不断充实自己，提高自身的素质和专业能力。

最后，再次感谢所有老师和领导，感谢你们的教导，让我们有了学习的动力，一分耕耘一分收获，从你们的身上，学习到的不仅仅是知识，还有做人的道理，对日后生活和工作都将具有重要意义。

参考文献

- [1] 刘卫国. Python 语言程序设计[M]. 北京: 电子工业出版社 ‘2016’ 1-2.
- [2] 嵩天, 黄天羽, 礼欣. Python 语言程序设计课程教学改革理想选择的理想选择[J]. 中国大学教学, 2016 (2):42-46.
- [3] 刘静. 关于面向非计算机专业开设 Python 语言的几点思考[J]. 考试周刊, 2018(23):130.
- [4] 史梦楚. Python 语言的探讨[J]. 中国新通信, 2017, 19(07):98.
- [5] 肖旻, 陈行. 基于 Python 语言编程特点及应用之探讨[J]. 电脑知识与技术, 2014, 10(34):8177-8178.
- [6] 飞思科技产品研发中心. 《C#应用开发详解》. 电子工业出版社, 2003 年 9 月
- [7] 耿祥义, 张跃平. 《C#实用教程》. 清华大学出版社, 2003 年 5 月
- [8] 孙涌. 《现代软件工程》. 北京希望电子出版社, 2003 年 8 月
- [9] 萨师煊, 王珊. 《数据库系统概论》. 高等教育出版社, 2002 年 2 月
- [10] Brown 等. 《C#编程指南 (第二版)》. 电子工业出版社, 2003 年 3 月
- [11] 清宏计算机工作室. 《C#编程技巧》. 机械工业出版社, 2004 年 5 月
- [12] 朱红, 司光亚. 《C# Web 编程指南》. 电子工业出版社, 2001 年 9 月
- [13] 赛奎春. 《C#工程应用与项目实践》. 机械工业出版社, 2002 年 8 月
- [14] 美. 霍尔著钟鸣等译. Servlet 与 C#权威指南. 机械工业出版社. 2002-10-1
- [15] 刘彬主编. C#数据库高级教程. 清华大学出版社. 2006-3-1

Design and Implementation English Abbreviation Quick System

WANG Jin-mu 105032016027 Advisor: LIN Ling

**Major in Computer Science and Technology College of Mathematics and
Informatics**

【Abstract】 With the continuous development of computer technology, people use computers not only to browse news, send email, chat and games, but also to achieve the purpose of learning. The design of an English word quick search system makes it more convenient for people to learn English in time, while enriching and improving people's learning needs. This system is developed in C / S mode. Based on the functional characteristics of Python programming language and crawler technology, this paper designs and implements an English word quick reference system, and stores the data in SQL server. The system is developed based on C language, which has the functions of user management, vocabulary maintenance, vocabulary learning, vocabulary review, vocabulary test, etc. In the design, considering the practicability, effectiveness and maintainability of the system, the three-layer architecture programming method is used to realize the function design in the programming, which reduces the code repetition, makes the system easy to maintain and manage, and provides the interface for further expansion.

【Keywords】 C / S; English learning; three-tier architecture