

北京郵電大學

本科 毕业 设计（论文）



题目：基于 Android 平台的专业术语辅助学习系统的设计与实现

姓 名 黄玺祯
学 院 计算机学院
专 业 网络工程
班 级 2014211312
学 号 2014211479
班内序号 26
指导教师 张海旸

2018 年 6 月

本科毕业设计（论文）诚信声明

本人声明所呈交的毕业设计（论文），题目《基于 Android 平台的专业术语辅助学习系统的设计与实现》是本人在指导教师的指导下，独立进行研究工作所取得的成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：_____ 日期：_____

基于 Android 平台的专业术语辅助学习系统的设计与实现

摘要

当今，移动设备发展势头愈发迅猛，五花八门的手机操作系统中，Android 系统具有最高的市场占有率。在国际化交流不断加强的今天，对于语言学习的需求也变得越来越普遍，在线翻译软件成为刚性需求。如果将翻译软件和搭载 Android 系统的移动设备结合起来，无疑会给语言学习者带来极大的便利。

但是目前为止市面上大多数翻译软件都是通用翻译软件，如金山词霸、有道词典、百度翻译等。这些通用翻译软件在进行专业词汇的翻译时，还有很多不足之处。本毕业设计研究的是计算机专业术语领域的翻译，经过调研发现，很多同学们在使用通用翻译软件时都遇到过各式各样的问题，其中查找不到某些词汇、不知道哪个释义比较准确这两个问题最为常见。因此，设计并实现本系统的主要目的是为了弥补通用翻译软件在翻译专业词汇领域上的缺陷，帮助学习者们能够更好的学习。

本文首先介绍了该项目的研究背景以及国内外翻译软件以及翻译平台的研究现状，阐述了该系统存在的社会意义。接着第二章详细介绍了设计本系统所需的一些技术，包括金山词霸平台 API 的使用、系统开发平台的简要介绍、C/S 架构的概述、网络传输 Socket 相关知识、MySQL 数据库的组成和性能等等。第三章从功能性需求和非功能性需求两个方面对该系统进行了分析，其中功能性需求主要包括在线翻译查询、释义投票、用户释义提交三大部分，非功能性需求包括系统性能要求和用户体验要求。接着根据需求对系统使用 MVC 模式进行层次架构设计、划分具体的功能模块，然后对系统涉及到的一些关键思想和技术进行阐述、设计服务器的数据库结构、设计客户端和服务器的交互信息等等。第四章对系统的核心功能模块进行了详细设计，通过使用时序图、流程图、类图等对系统的工作流程进行阐述，并加以伪代码和文字来进行辅助说明。第五章则对该系统进行了全面测试，涵盖了用户使用过程中的各种情形。除此之外，还在论文的最后一章提出了该系统设计过程中存在的不足之处以及优化的方向。

本人在老师的指导下独立完成了整个项目，包括需求分析、总体设计、详细设计以及系统测试。软件在测试后，每个模块均能够成功运行，满足了用户的需求，达到了最初设计该专业术语辅助学习系统的目的。

关键词 专业术语 Android 系统 在线翻译

The design and implementation of terminology assisted learning system based on android platform

ABSTRACT

Today, the number of mobile devices is growing rapidly. Among the various mobile operating systems, the Android system has the highest market share. With the frequent international communication, language learning has become more and more common in our daily life, on-line translation software has become a rigid demand. If you combine translation software with Android mobile phone, it will undoubtedly bring great convenience to language learners.

But so far, the most popular translation software on the application market is universal translation software, such as Kingsoft, Youdao Dictionary, Google Translation and so on. These common translation applications still have many deficiencies in the translation of professional vocabulary. So, this graduation project focuses on the translation of the computer science terms. After investigation, I was found that many students encountered various problems when using the universal translation software. For instance, the translation of some terms could not be found in the application, or there are serval translations in the target language so that learners cannot figure out which one is more accurate. Therefore, the main purpose of this project is designing and implementing an application on Android platform, in order to make up the deficiencies of universal translation applications and help more learners.

In the first chapter of this thesis, it introduces the research background of this project and the status of translation applications in China and abroad, also expounds the social significance of the system. Then in the second chapter, it detailly describes the techniques which is needed to design the system, including Kingsoft platform API, C/S architecture, socket technique, MySQL database. In the third chapter, this thesis analyzes the system from two aspects: functional requirements and non-functional requirements. Functional requirements mainly include querying online, voting for translation results and submitting users' own interpretations. Non-functional requirements include system performance and user experience. Then in this chapter, introducing how to use MVC pattern to design the hierarchical architecture and divides specific functional modules, explaining the design of database structure and the interaction method between client and server in this chapter. In the fourth chapter, demonstrating the workflow of core functional modules in this system. The workflow of the system is elaborated by using sequence diagrams, flowcharts, class diagrams, etc., and pseudo-codes and texts are used to help explain. The fifth chapter includes serval comprehensive tests of the system, I simulated user usage scenarios systematically in order to test robustness of this system. In the last chapter of the paper, listing the deficiencies and optimization directions of this system.

I independently completed the entire project under the guidance of professor, including requirements analysis, overall design, detailed design, and system testing. After this software was tested, each module was able to run successfully, satisfying the needs of the user and achieving the purpose of initially designing the terminology assisted learning system.

KEY WORDS terminology android system online translation

目 录

第一章 绪论	1
1.1 选题背景及研究意义	1
1.1.1 专业术语辅助学习系统的背景	1
1.1.2 论文的研究意义	1
1.2 国内外研究现状	2
1.3 论文主要研究工作	2
1.4 论文结构及安排	2
1.5 本章小结	3
第二章 相关技术介绍	4
2.1 移动学习理论	4
2.2 通用翻译软件	4
2.3 Android Studio 和 Eclipse 开发平台	5
2.4 C/S 架构	7
2.5 Socket	8
2.6 MySQL 数据库	9
2.6 本章小结	10
第三章 系统的总体设计	11
3.1 系统需求分析	11
3.1.1 功能性需求分析	11
3.1.2 非功能性需求分析	13
3.2 系统软件层次架构设计	14
3.3 系统功能模块设计	15
3.4 关键技术研究	15
3.5 系统的数据库设计	17
3.6 客户端和服务器交互信息设计	19
3.7 系统的界面设计	22
3.8 本章小结	23
第四章 系统详细设计与实现	24
4.1 核心功能模块设计	24
4.1.1 在线查询模块	24
4.1.2 用户投票模块	30
4.1.3 用户释义提交模块	34
4.1.4 释义排序处理模块	38
4.2 本章小结	42

第五章 系统测试	43
5.1 系统测试目标	43
5.2 系统功能测试以及结果分析	43
5.3 系统性能测试	54
5.4 本章小结	56
第六章 结束语	57
6.1 论文工作总结	57
6.2 问题与展望	57
参考文献	58
致 谢	59
外文译文	

第一章 绪论

1.1 选题背景及研究意义

1.1.1 专业术语辅助学习系统的背景

本课题来自于实际的应用需求。随着全球一体化的进程不断推进，中国对外政策不断开放，越来越多的行业需要进行跨地区跨国界的交流，从事此类行业的人很多时候都要阅读和参考外文文献，而其中绝大多数是英文文献。例如学生们阅读专业相关的外文书籍，或者一些技术人员察看专业资料等等。他们在进行工作或学习时，会经常接触到大量的专业领域术语，但是由于不同语言的人的生长环境和思维方式的不同，很多专业词汇很难以找到准确的所对应的中文意义，给人们阅读和理解书本内容造成了一定的困难。

另一方面，当前时代是一个互联网高速发展的时代，移动智能设备越来越普及。根据 CNNIC 2017 年 1 月 22 日发布的《中国互联网络发展状况统计报告》显示，截至 2016 年 12 月，“.CN”注册保有量超过 2000 万，居全球国家域名第一^[1]。中国网民规模接近 7.31 亿，互联网普及率高达 53.2%。其中，手机网民规模达 6.95 亿，占比达 95.1%，增速连续三年超过 10%^[1]。因此，将专业术语的学习放置在智能手机上，对于使用者而言是非常便捷且容易接受的。综上所述，研究移动平台上的专业术语辅助学习系统有着重要的社会意义。

1.1.2 论文的研究意义

专业术语的中英文翻译一直以来都是一个难点，为了解决这个问题，使得人们能够查找到某一专业词汇所对应的中文释义，现在市面上有着不少的辅助工具。比较传统辅助工具的如专门的词汇书，纸质辞典；基于互联网的如在线翻译平台、翻译软件等等。但是这些工具或多或少都有着缺陷：纸质书体积庞大携带不便，查找单词耗时耗力；在线翻译软件如 Google 翻译，有道翻译等等，在翻译专业术语时要么缺乏相关解释，要么给出很多不同的解释以至于使用者难以判断哪一个解释更为准确，给阅读外文论文书籍带来了困难。

为了弥补当前市面上存在的翻译工具的不足，本课题开发一套基于 Android 的专业术语辅助学习系统，用户可以通过本系统查询英文专业术语，界面上会显示出多条中文互译结果，其中每一条翻译结果经过用户投票，通过算法计算后按得分从高到低降序排列，从而确保用户们看到的推荐结果是最接近准确释义的结果。除此之外用户也可以将自己理解的专业术语的含义提交到系统中，通过对专业术语的翻译结果进行反馈，用以不断完善该辅助系统。

1.2 国内外研究现状

Google 公司在 2005 年收购 Android 系统，在国内的智能手机市场中，Android 市场占有率最高^{[2][3]}，公众接受度也是最高的。

在电子词典方面，从二十世纪以来，微型计算机逐步进入大众们的视野，电子词典也跟随着微型计算机的发展逐步受到重视。电子词典又可分为非智能电子词典和智能电子词典^[4]。非智能电子词典的功能较为单一，而且所有的词库都需要事先预置在学习机中，很占用学习机的存储空间；而智能电子词典主要基于移动操作平台，功能强大且种类繁多，在性能上与前者相比有着质的飞跃。

现如今随着嵌入式技术的快速发展，它在我们生活中的方方面面都起到了很大的推动作用。对于智能词典的快速普及，嵌入式操作系统也是一个强有力的手推手。

但是无论是基于移动设备的翻译软件还是基于网站的翻译平台，现目前市面上存在的绝大多数都是通用翻译。通用翻译有一个明显的不足之处是对于某些跨领域名词，会给出很多释义，这样会使得那些基础不是特别扎实的使用者们无法进行准确的判断，从而减慢学习的进度。这是目前翻译软件使用者的痛点之一，也是本系统主要的研究方向。

1.3 论文主要研究工作

本课程设计基于 Android 平台，分析和设计了一个在线的专业术语辅助学习工具，该工具由客户端和服务器端两大部分构成，其主要功能就是满足用户查询英文专业术语单词的需求。为了分析和设计该应用软件，主要做了以下工作：

- (1) 使用和研究了市面上通用翻译软件和翻译平台，分析当前市面上的翻译类辅助工具所存在的缺点，得出当前翻译软件的总体需求以及完成软件定位，划分出系统的大概功能模块；
- (2) 研究本课程设计所需要的技术，如开发环境 Android Studio 的使用、学习 Java 语言、数据库 MySQL、C/S 架构、Socket 连接以及翻译技术金山 API；
- (3) 最后对客户端各个功能模块、界面以及服务器所需的数据库进行了详细设计。功能模块包括用户注册登陆、用户信息管理、专业术语在线翻译模块（英语）、专业术语解释评价和反馈模块、专业术语数据库管理模块等。

1.4 论文结构及安排

第一章 绪论：本章作为前言部分，介绍了目前市面上存在的翻译软件种类，目前国内外翻译软件以及平台的发展状况，通用翻译平台在专业术语的翻译上存在的问题，引出本文的研究意义和目的，并简要说明了本文的研究工作。

第二章 相关理论技术简介：介绍了该课题所使用的技术，包括移动学习理论、金山词霸 API 的使用方法、Android 平台架构、系统结构用到的 C/S 架构以及 MySQL 数据库。

第三章 系统的总体设计：通过调研现有的专业术语词典以及计算机专业学生的翻译软件使用情况，得出系统的总体需求。对于系统的功能性需求，采用了用例表的方式对各个功能进行了详细的分析；对于系统的非功能性需求，从系统性能以及用户体验两个方面进行了分析。接着根据需求进行层次架构设计，本系统采用了 MVC 的框架。然后划分具体的功能模块，包括客户端和服务器。然后设计服务器的数据库结构，根据 E-R 图来确定数据库模型该如何建立。最后根据界面设计原则来进行客户端的用户界面设计。

第四章 功能设计：根据系统的总体设计目标，该系统采用了 C/S 架构，具有客户端和服务器两部分，而且对系统的功能模块进行了详细的划分。然后以表格的形式详细展示了数据库表的详细情况，以及描述了客户端和服务器交互信息的具体格式以及字段解释。

第五章 系统测试：对该系统的每一个功能模块都进行详细的测试，涵盖了用户在使用过程中可能会出现的各种情况，并且阐述了出现意外情况时系统会出现的提示信息。然后对测试结果进行了分析。

第六章 结束语：总结论文，指出目前该系统存在的不足之处有哪些方面，以及未来优化改进的方向。

1.5 本章小结

本章首先介绍了专业术语辅助学习系统的背景。现在市面上的翻译辅助工具种类繁多，但是大多数都是通用翻译工具，对于特定的专业词汇来说它们给出的结果并不具有很高的准确性，有时候甚至会误导使用者。然后阐述了该系统的研究目的和意义，也对目前电子词典的情况进行了简要介绍。在本章的最后还说明了论文的主要工作和本文的论文结构安排。

第二章 相关技术介绍

2.1 移动学习理论

移动学习英文为 M-Learning 是指利用无线移动通信网络技术，以无线移动通信设备来获取教育信息、教育资源的一种新型数字化学习形式^[5]。与其它形式的学习相比较，移动学习具有学习便捷性、教学个性化、交互丰富性、情境相关性等特点，是移动计算技术与数字化学习技术嫁接的产物^[5]。在它出现之前，人们在学习方面主要依赖的是纸质媒介，具有较大的局限性。在移动学习诞生后，人们的学习方式出现了较大的改变，它给学习者们带来了巨大的便利主要体现在以下方面：首先，移动设备小巧易于携带，只要使用者有使用意愿，就可以随时随地开始进行学习，几乎不受地点和时间的约束。其次，移动设备人机交互度高，操作性强，学习者可以根据自己的需求来管理自己的学习应用。但是移动学习并不能取代传统的学习方式，传统的学习方式适用于正式场合，而移动学习方式的场景和时间的灵活性，因此适用于非正式的学习^[6]，比如学习者在等待地铁的时候，或者在乘坐公交的时候。因此基于移动学习理论的项目一般都较为灵活，学习者可以随时随地开始和结束学习。比如电子词典、在线英语听力软件、驾考宝典等。

虽然移动学习无论是在理论还是应用方面都取得了不错的成绩，但是依旧存在一些问题需要解决：

(1) 无线通信问题

基于移动学习的项目和系统，都需要在网络稳定的条件下才能够完美展现其功能和作用，但是无线网络本身稳定性较差并且维护成本较高，因此这些限制在某些程度上并不利于移动学习在实践中取得较大的进步。

(2) 产品设计问题

基于移动学习的项目和系统基本上针对的都是零散的学习，而且在市面上移动学习软件虽然数不胜数，但是很多应用的界面设计和产品功能不够简洁，用户使用和操作的成本太高导致使用频率下降。因此一款移动的辅助学习软件能否普及开来，产品设计起到了至关重要的作用。

2.2 通用翻译软件

当前市面上常用的翻译软件有金山词霸，翻译平台有 Google translate、百度翻译等。接下来对这些翻译工具的专业术语翻译能力进行分析：

(1) 金山词霸是一个通用翻译软件，因此在查询计算机相关的专业术语时会出现常见问题那就是：

- 对于一词多义的单词，没有将计算机相关的释义单独提取出来。例如单词“TCP”，金山词霸给的意思是“行车调度台、训练调度台、传输控制协议”等等。而且“传输控制协议”这个解释被放在最后，并用灰色小一号字体弱化了显示效果。
 - 对于一词多义的单词，没有将释义进行排序，使得用户能够快速找出最准确的答案。例如单词“database”，金山词霸给的意思是“资料库、信息库、数据库”。虽然三个解释都可以用来翻译 database，但是在计算机领域无疑“数据库”这个解释是最为准确的。通用词典里将所有的释义都混在了一起，有些情况下会给学习者带来困扰。
- (2) Google translate 是一个通用翻译平台。在 2016 年进行升级优化后和原来相比已经有了很大的提升，但依旧存在不足之处。其中最主要的一个缺陷就是无法翻译缩略词。然而计算机专业术语里面有非常多的缩略词，比如 UDP、SQL 等，Google translate 并不支持此类缩略词的翻译，这对于学习者来说并不友好。
- (3) 百度翻译也是一个通用翻译平台。虽然加入了牛津词典和柯林斯词典，具有更为全面的单词释义库，但是也将所有的释义都混合在一起进行展示，和金山词霸有着相同的不足之处。

为了解决通用翻译软件在翻译计算机专业术语时存在的不足之处，本系统在通用翻译的基础上进行了优化。其中通用翻译使用金山词霸的 API 进行接入，主要使用了金山词霸的英译汉功能，key 和 w 是金山 API 的两个参数，其中 key 是开发者身份验证码，w 是要查询的单词。为了能够让第三方应用程序使用金山 API，金山提供了一个查词的接口，单词查询地址为 "<http://dict-co.iciba.com/api/dictionary.php?w=?&key=XX>"，此处的 XX 就是开发者身份验证码。查询结果可以用 JSON 和 XML 两种格式来返回，默认格式为 XML。在本系统里使用的返回格式为 XML，接收到返回信息后，会对原始返回结果系统做出一些处理。首先是将所有不是名词的解释给过滤掉，因为专业词汇的翻译均为名词；其次对于有着多个解释的单词，会把每个解释都单独提取出来，成为一条独立的结果显示在界面上，便于用户进行选择以及投票。

2.3 Android Studio 和 Eclipse 开发平台

本系统采用 Android Studio 平台和 Eclipse 进行开发，下面对这两个平台进行简要介绍。

系统客户端使用 Android Studio 3 进行开发。Android Studio 是 Google Android 操作系统的官方集成开发环境，基于 JetBrains 的 IntelliJ IDEA 软件，专门为 Android 开发而设计^[7]。它是 Eclipse Android 开发工具（ADT）的替代品，可作为原生 Android 应用程序开发的主要 IDE。

Android Studio 开发界面如图 2-1：

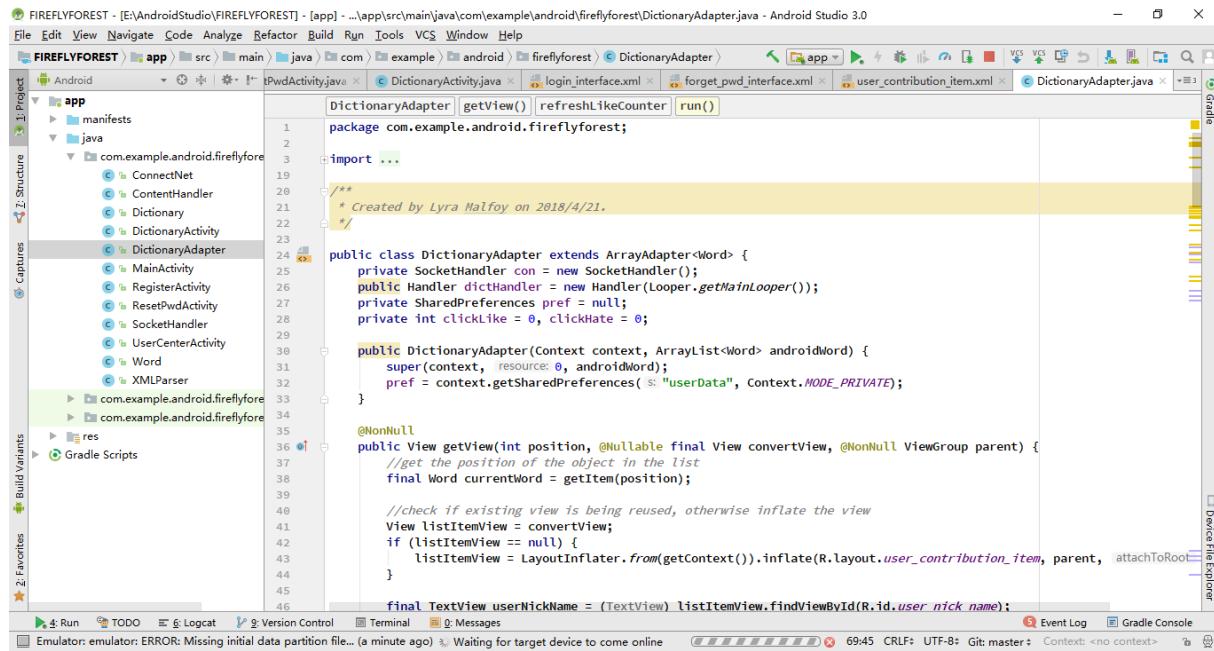


图 2-1 Android Studio 开发界面

Android Studio 目前取代 Eclipse 成为市面上主流的 Android 开发工具，这是因为 Android Studio 具有如下优势：

- (1) 基于 Gradle 的构建支持
- (2) Android 特定的重构和快速修复，代码自动补全以及提示
- (3) ProGuard 集成和应用程序签名功能
- (4) 丰富的布局编辑器，允许用户拖放 UI 组件，可在多个屏幕配置中预览布局
- (5) 内置 svn、git 工具支持
- (6) Android 虚拟设备（模拟器）在 Android Studio 中运行和调试应用程序

系统服务器使用 Eclipse Java Oxygen 进行开发。Eclipse 是用于计算机编程的集成开发环境（IDE），也是使用最广泛的 Java IDE，它包含一个基本工作区和一个用于定制环境的可扩展插件系统^[8]。Eclipse 主要以 Java 编写，主要用于开发 Java 应用程序。Eclipse 受到基于 Smalltalk 的 VisualAge 系列集成开发环境（IDE）产品的启发，最初的代码库来自 IBM VisualAge^[8]。包含 Java 开发工具的 Eclipse 软件开发工具包适用于 Java 开发人员。

Eclipse 开发界面如图 2-2。

使用 Eclipse 开发具有如下优势：

- (1) 强大的 Java 语言支撑功能，从编译、调试、查询方面来看，与 Java 语言完全契合；
 - (2) 除了内核外其他的功能几乎全部由插件来实现，可以实现大量的功能扩展
- Eclipse 是开源软件，自由、免费；

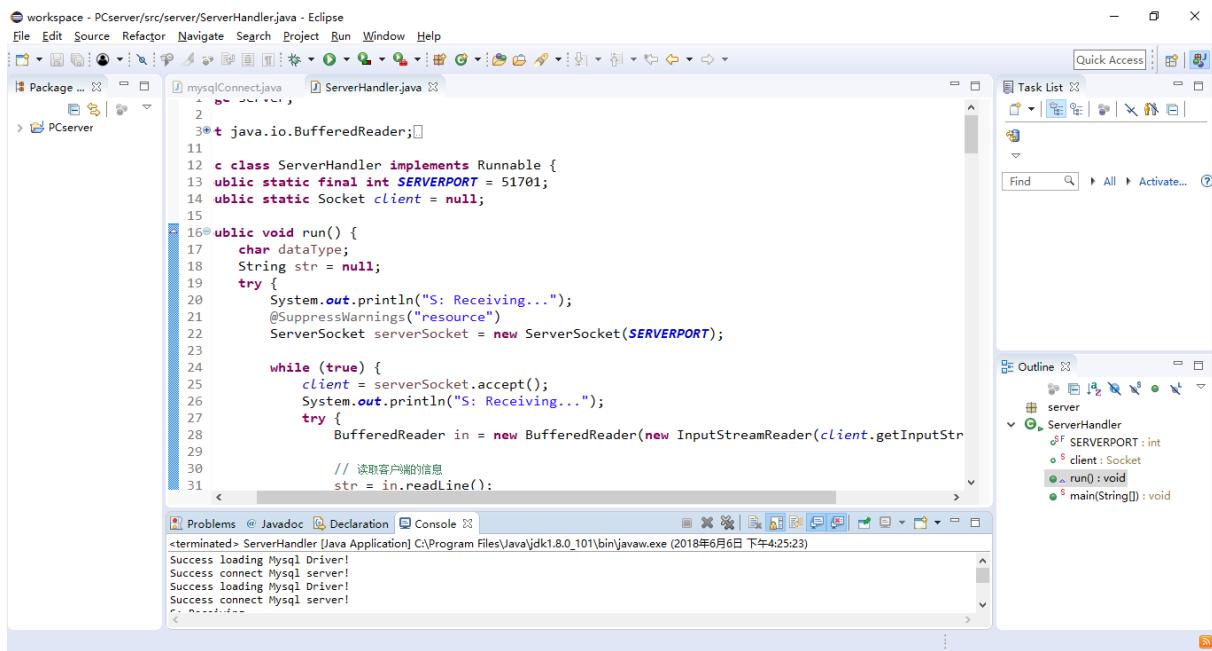


图 2-2 Eclipse 开发界面

2.4 C/S 架构

C/S 架构全称为 Client/Server，架构如图 2-3 所示。

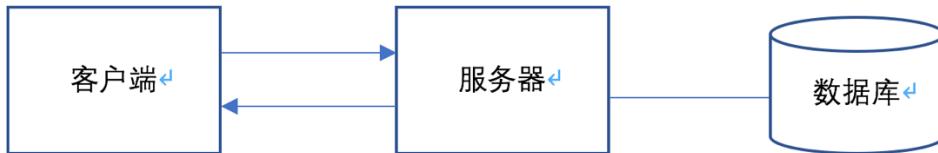


图 2-3 Client/Server 架构

C/S 架构分为客户机和服务器两层：第一层是在客户机系统上结合表示和业务逻辑两部分，第二层是通过网络连接了数据库服务器^[9]。客户端和服务器直接相连，这两个组成部分都承担着重要的角色，第一层的客户机并不是只有输入输出，除此之外它还可以处理一些计算、数据存储等方面的业务逻辑事务^[9]。第二层的服务器主要承担事务逻辑的处理，因为客户机可以分担一些逻辑事务，所以减轻了服务器的负担，使得网络流量增多^[9]。

C/S 架构的优点主要有如下几个方面：

(1) 客户端和服务器端直接相连

因为客户端和服务器端直接相连，点对点的模式更加安全可靠；可以直接操作本地文本，从而减少通信流量；而且当通信量较大的时候不容易出现拥塞现象，响应速度快。

(2) 客户端可以进行部分逻辑事务的处理

客户端可以帮助服务器端分担一些逻辑事务的处理，充分运用两者的硬件设施，避免资源的浪费^[9]；客户端有一套完整的运用程，出现错误时可以及时进行提示。

(3) 操作界面体验良好

操作界面可以按照用户的需求进行个性化定制，提高客户视觉体验。

但是经过长期的实践检验，C/S 架构的一些弊端也开始展现，主要有以下几个方面：

- (1) 客户端只能处理功能单一的多系统
- (2) C/S 架构适用于局域网，对网速要求比较高。而且若客户端所使用的系统不同，就要针对这些系统分别编写程序
- (3) 只有安装了客户端才哦能访问服务器，所以用户不能随时随地访问
- (4) 当用户数目增多时会出现通信拥堵，服务器响应变慢
- (5) 系统升级维护麻烦，初次投入成本后不能一劳永逸，后期需要很多成本

2.5 Socket

Socket 起源于 Unix，是一种“打开一读/写一关闭”模式的实现^[10]。具体来说，它是网络软件（协议栈）中该端点的表示，利用 IP 地址+协议+端口号唯一标示网络中的一个进程。Socket 在网络层次架构中的位置如图 2-4 所示。

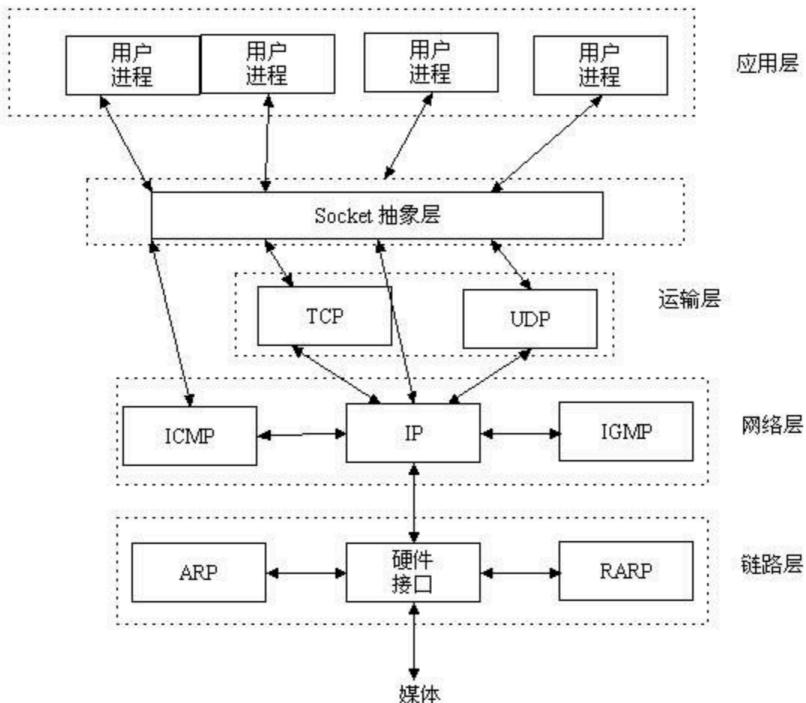


图 2-4 Socket 在网络层次架构中的位置

Socket 层的特征有如下几点：

- (1) Socket 层是一个抽象层，位于在应用层和传输层之间^[11]

- (2) Socket 层是一个协议无关层，在工作过程中 socket 会使用到下层协议，所有系统调用从协议无关的 Socket 开始^[11]
- (3) Socket 层中的每一个 Socket 都具有一个输入队列和一个输出队列，用于进行不同层之间的数据传输

2.6 MySQL 数据库

MySQL 是一个开源的关系型数据库管理系统，原开发者为瑞典的 MySQL AB 公司^[12]。它是一个开源的数据库，到目前为止已经迭代了很多版本，功能也越来越强大，如它为多种编程语言提供了 API，可以通过 API 直接对数据库进行操作；使用优化后的查询算法，有效的提高查询速度；支持大型数据库，可以处理具有上千万条记录的大型数据库等等。也正因为如此，MySQL 逐渐应用于大规模的网站以及应用，如 Facebook, google 等^[12]。

MySQL 的系统结构如图 2-5 所示：

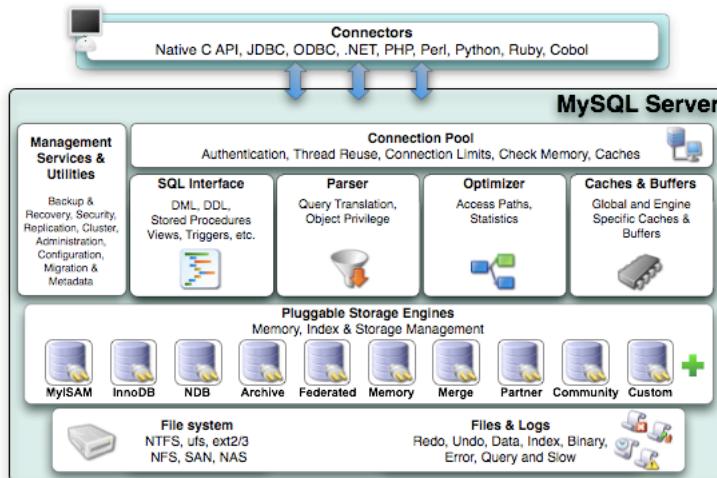


图 2-5 MySQL 系统结构

Connectors: 连接器，负责 MySQL 数据库和不同的编程语言之间的交互；

Connection Pool: 连接池，进行身份验证、线程重用，数据缓存等功能。还具有处理线程、管理用户的连接等需要进行缓存的需求；

SQL Interface: SQL 接口，用户通过 SQL 命令来查询所需结果；

Parser: 解析器，进行翻译对象查询，对象特权解析；SQL 命令传递到解析器的时候会被解析器验证和解析，如果在分析构成中出现错误，那么这个 SQL 语句就是不合理的^[12]；

Optimizer: 查询优化器，当使用 SQL 语句进行查询操作时，优化查询器会在查询语句执行之前对查询进行优化，优化查询策略为“选取-投影-联接”，这样可以有效地提升查询速度；

Cache 和 Buffer: 查询缓存，当进行查询时，首先会在查询缓存中进行结果查找。如果查询缓存有信息命中，就可以直接去查询缓存中取数据。这个缓存机制是由一系列小缓存组成的，比如表缓存，记录缓存，key 缓存，权限缓存等。查询缓存的存在也能大大提升查询的速度；

Management Services & Utilities: 系统管理和控制工具，负责备份和恢复的安全性保障，还有复制，集群，管理，配置，迁移和元数据的处理功能；

Engine: 存储引擎是 MySQL 中具体的与文件打交道的子系统，MySQL 的存储引擎是插件式的。它根据 MySQL AB 公司提供的文件访问层的一个抽象接口来定制一种文件访问机制，这种访问机制就叫存储引擎，常用引擎有 MyISAM、InnoDB、BDB，默认情况下 MySQL 是使用 MyISAM 引擎^[13]。

2.6 本章小结

本章介绍了进行专业词汇辅助学习 APP 开发时所要学习的技术与理论。首先介绍了移动学习理论，并将这一理论在本系统开发中得到运用和实践；其次介绍了金山 API 的词典查询技术；然后对开发平台进行了简要介绍，对 C/S 模型和 Socket 技术进行了系统的阐释和分析；最后对本系统要使用的持久化数据存储工具 MySQL 进行了介绍。

第三章 系统的总体设计

3.1 系统需求分析

该辅助学习系统是一个具有很强的用户需求针对性的 APP，主要针对的是计算机科学与技术方向的学习者们，因此在开发系统之前应当对系统的功能性需求和非功能性需求分析^[14]。由于设计的是专业术语辅助类软件，那么功能性需求就是使用特定算法给用户提供尽可能准确的翻译，非功能性指的就是用户相关体验。

3.1.1 功能性需求分析

为了对本系统功能性需求进行分析，我对计算机专业的同学进行了调研。他们都曾使用过有道翻译和金山词霸等通用翻译软件，以及 Google translate 和百度翻译等通用翻译平台。调研结果显示学生们在使用过程中遇到的困难主要集中在以下方面：

- 有些单词释义查询不到。目前市面上大多为通用翻译软件，对于计算机科学领域的专业名词收录不全面，所以有时候在阅读书籍时会遇到某些词汇看不懂也查不到的情况；
- 一词多义，无法辨别哪一个解释才是最准确的。很多专业课程使用的是国外编写的英文教材，专业名词颇多导致阅读非常吃力。其中有很多名词不仅在计算机领域有所使用，在其他学科领域中也有相应的概念，在使用通用翻译软件查询的时候会显示出很多结果，而学生往往会因为相应的专业知识不够而无法辨别哪一个才是最准确的解释，导致学生们在学习过程中出现误解从而减慢学习进度。

鉴于以上的调研结果，本系统的主要功能应从以下方面入手：

- 满足用户的查词需求，尽最大限度保证用户能够查询到计算机领域的专业词汇；
- 对具有多个解释的单词，将每个释义进行单独处理，使得在有多个释义的情况下，被大多数人认为是最准确的释义能够得到一个靠前的排名，从而方便后续用户的使用。

经分析可得，该专业术语辅助学习系统需要具备翻译查询、释义投票、用户释义提交三大功能，以满足用户使用需求。各个功能总体设计如下：

(1) 翻译查询

在联网的状态下，用户可使用该软件进行单词查询。翻译查询用例说明表如表 3-1 所示。

表 3-1 翻译查询用例

用例名称	翻译查询
------	------

行为角色	用户
简要说明	1) 用户输入要查询的单词 2) 点击查询图标，将查询单词提交给服务器 3) 服务器处无查询结果，将单词提交给金山词霸进行查找 4) 查看查询结果
前置条件	用户已经成功登录本系统
后置条件	屏幕上显示该单词的查询结果

(2) 释义投票

当用户点击查询后，相应的一个或者多个单词释义会逐条显示在界面上。在每一条释义的右上角都有一个赞同图标和一个反对图标。当用户觉得某条释义正确时，可以点击赞同图标，同时图标旁边的点赞数值也会发生相应的变化。反之如果用户觉得某条释义的结果不准确甚至是错误的，可以点击反对图标。用户对于同一个释义只能投一个赞成票或者反对票，重复点击赞成或者反对都会导致上一次的操作无效。释义投票用例说明表如表 3-2 所示。

表 3-2 释义投票用例

用例名称	释义投票
行为角色	用户
简要说明	1) 得到查询结果 2) 逐条查看屏幕上显示的释义 3) 对于某一条释义点击旁边的赞成或者反对图标，或者不做任何操作
前置条件	用户输入查询单词并且点击了搜索图标
后置条件	1) 根据用户的操作屏幕的显示结果也作出相应的变化 2) 数据库 dict 表内的该释义记录所对应的 like/hate 属性值被刷新 3) 数据库 operation 表内该操作用户的操作记录对应的 likeflag/hateflag 属性值被刷新

(3) 用户释义提交

当用户查看完所有的释义或者没有任何查询结果时，用户可以提交自己认为的准确的释义，帮助系统拓展数据库翻译条目。用户提交完释义后，进行下拉刷新就可以看到自己提交的释义。用户释义提交用例说明表如表 3-3 所示。

表 3-3 用户释义提交用例

用例名称	用户释义提交
行为角色	用户
简要说明	1) 用户没有得到满意的释义

	2) 在屏幕最下方的输入框中输入自己认为的或者从别处得到的准确释义 3) 点击提交按钮 4) 下拉刷新查看提交结果
前置条件	用户输入查询单词并且点击了搜索图标
后置条件	1) 用户提交的释义出现在屏幕上 2) 数据库 dict 表新增加一条记录

3.1.2 非功能性需求分析

对于一个软件而言，除了本身功能以外，系统的非功能性需求也是影响其易用性的重要因素之一。针对非功能性需求，本系统从一下两点进行了优化：

(1) 系统性能要求

由于本系统客户端所需要的数据几乎全部存放在服务器中，因此客户端和服务器端的交互非常频繁。当多个用户进行操作的时候会对服务器产生很高的负荷，因此在不影响正常功能的情况下，减少一部分客户端与服务器端的交互从而减轻服务器的压力。除此之外还要保证在信息传输过程中用户的信息不会泄漏。

为了实现上述要求，在本系统中使用了如下方式：

首先，当用户在客户端进行登录时，客户端会记录下来登陆用户的用户名以及用户等级，从而用户在进行释义提交和访问个人中心的时候，屏幕上显示的用户信息直接来源于本地文件，不用通过访问数据库取得；其次，用户在进行点赞或者反对时，屏幕上的统计数字在首次访问时存储在函数变量中，然后用户点击后函数变量值发生加减变化，最后将该值显示在屏幕上，而非每一次操作都要重新访问数据库取得最新信息；然后，要求用户进行密码输入的地方，输入的密码首先在客户端进行 SHA1 加密^[15]，因此无论是在传输过程中还是服务器数据库内存储的都是经过加密后的数据，从而保障用户信息的安全；最后，用户等级每 24 小时更新一次，从而避免了频繁的大量的数据更新导致服务器不堪重负而崩溃。

(2) 用户体验要求

在调研过程中，也有同学反馈曾下载过专业词典，但是交互界面粗糙且不友好，经短暂使用后就放弃了。因此在本系统的设计中，用户界面的设计不宜过于复杂，界面应该做到干净简单、配色合理。应当充分考虑到系统的易用性，功能设置一目了然。同时应当设置用户反馈意见的入口，以便于日后能够通过用户的反馈来进行软件的优化更新。

3.2 系统软件层次架构设计

本系统采用 MVC 模式，主要原因是 MVC 模式具有如下几个优点：

- (1) 视图层和业务层分离，使得系统代码具有低耦合性；
- (2) 允许不同的视图界面访问同一个服务器端的代码，具有高重用性和可适用性^[16]；
- (3) 可维护性，此模式将代码复杂度进行简化，使程序结构更加直观；

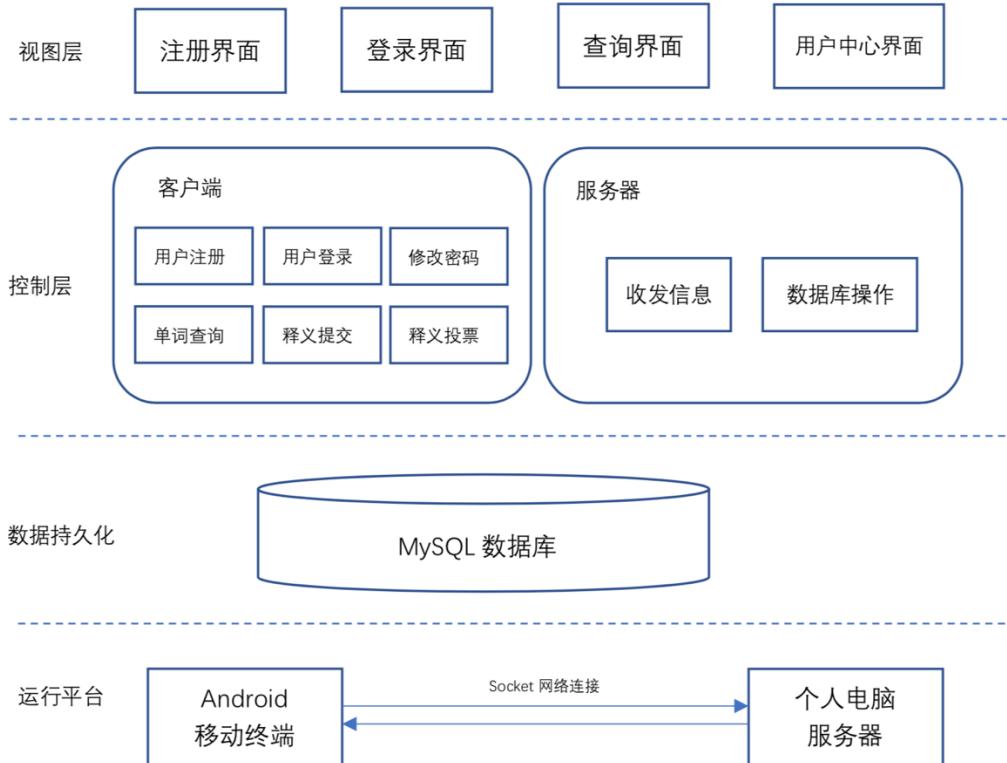


图 3-1 系统框架图

根据图 3-1，下面分别介绍本系统中视图层、控制层以及模型层的功能。

视图层：客户端各个界面显示；

控制层：它在本系统中对应为客户端和服务器的功能模块，客户端包括用户登录注册、修改密码、单词查询、释义提及和投票对应的 Activity。服务器对应和客户端交互的收发信息模块和数据库操作模块；

数据持久化层：数据库，用于存贮那些需要进行永久保存的数据；

运行平台：客户端在 Android 移动终端上运行，服务器在个人电脑上运行。客户端和服务器使用 Socket 进行通信。

3.3 系统功能模块设计

本系统根据用户需求，将整个软件系统分为两个大的子系统，分别为客户端和服务器。系统功能模块结构如图 3-2 所示。

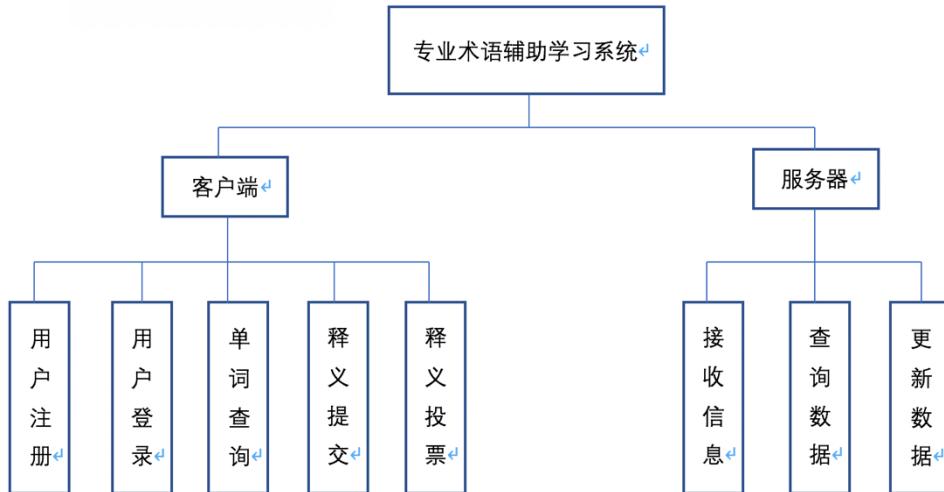


图 3-2 系统功能模块图

其中核心功能集中于客户端，客户端又可分为用户信息管理和翻译查询两大部分。其中用户信息管理包括用户注册、登录、退出三个模块；翻译查询包括在线查询、用户投票、用户释义提交三个模块。而服务器主要负责信息的存储和查询。

3.4 关键技术研究

相比于通用翻译软件，本系统特点在于对展示释义的处理。在本系统中，核心模块为单词查询模块、释义投票和用户释义提交模块。为了能够给用户展示最为精准的释义，核心模块采用了如下关键技术方法：

(1) 释义筛选以及拆分

通用词典的一大特征是对于具有多个释义的单词，不加区分的将所有释义都罗列出来，因此查询结果中会包含名词解释、动词解释、形容词解释等等。同时一个词性又可能会包含多个解释，如 closure 这个单词，名词词性有停业、关闭、结束、闭包等解释。显然这些解释并不是所有都适用于计算机领域，因此在释义处理时，第一步就是将释义进行筛选，然后拆分。这一操作主要用于单词释义来源为金山词霸的情况。

由于计算机专业术语释义均为名词，因此当有用户查询的单词具有多个释义时，如果该单词同时具有名词释义和非名词释义，系统会通过释义标签进行判断，只留下名词释义。

如果一个单词具有多个名词释义，金山词霸返回的结果中这些释义都被连接在一起成一个长字符串，因此本系统会将这些释义进行拆分，然后将释义来源、单词拼写、释义、点赞数、反对数以及总得分这些属性组合成一条信息，成为一个独立的元组存储在数据库中。

当用户进行单词查询时，如果该单词具有多个释义，那么查询结果会是一个按释义总得分从高到低进行排序后的集合，最后也是按照该顺序在界面上进行显示。

(2) 改进后的威尔逊算法

本系统将释义进行展示的时候并不是随机的，而是根据释义的总分数高低来进行排序，这也是本系统与其他的通用翻译软件的一大区别。释义的总分计算基于威尔逊算法，在常规算法的基础上根据系统需求加以改进。

威尔逊算法又叫做威尔逊分数区间，它是对正常近似区间的改进，因为实际覆盖概率更接近于标称值，由 Edwin Bidwell Wilson (1927) 开发^[17]。主要用于用于质量排序，数据含有赞成和反对，投票来源于所有时段的用户。综合考虑投票数与点赞率，得分越高，质量越高。

威尔逊算法公式如下：

$$n = like + dislike \quad \text{式 (3-1)}$$

$$percent = \frac{like}{n} \quad \text{式 (3-2)}$$

$$S = \frac{percent + \frac{z^2}{2n} - \frac{z}{2n}\sqrt{4n(1-percent)percent + z^2}}{1 + \frac{z^2}{n}} \quad \text{式 (3-3)}$$

like 表示赞成数，*dislike* 表示反对数，*n* 表示该评论总数，*percent* 表示好评率，*z* 是正态分布的分位数，在本系统中 *z* 取值为 1.96，置信区间为 98%，*s* 表示最终的威尔逊得分。

传统的（赞成票-反对票）排名方式 和（赞成票/反对票）的排名方式都是具有重大缺陷的。第一种方式没有考虑好评率，第二种方式当总票数很少的时候就会出现严重偏差。相比之下威尔逊算法具有很多的优良性质：

- 1) 投票总数增加，得分会趋向于正向反馈占总反馈的比例，因此得分可以很好的解释内容的质量；
- 2) 在总票数只有个位数的情况下，结果也能具有较高的准确性；
- 3) 置信区间可以通过调整 *z* 的值来设置；

基于本系统的情况，威尔逊算法做了一点改动。当用户进行投票或者反对的时候，提交给服务器的点赞或者反对的增长值与用户自身等级有关。因此算法里的赞成数和反对数不是该释义的真正票数，而是票数的权值。等级为 0 的用户点击一次票数权值增加 1，等级为 1 的用户点击一次票数权值增加 2，以此类推。但是该权值仅仅只是用来作为分数计算，不用做界面显示。同时，用户等级的增长也与该用户

提交的释义得分相关。如果一个用户提交的释义获得了广泛的认可，那么他的用户等级也会随之提升。

这样设计是因为考虑到用户等级跟他们的贡献率以及贡献质量有关，等级越高的用户显然对于计算机领域的认知更为深刻，因此他们的投票更加具有可信性。相应的被这些更具有权威性的用户所广泛认可的释义，应该有更快的总分增长速度，最后的释义排名也应该更加靠前。

(3) 冷启动处理

本系统的单词释义具有三个来源：系统预置、金山词霸平台、用户提交。

每个释义都具有一个初始分数，这个初始分数根据该释义的来源而有所差别。对于系统预置专业名词释义初始总得分均为 0.25，来源为金山词霸平台的释义初始总得分均为 0.2，来源为用户提交的释义初始总得分均为 0。

设计该方案主要出于以下几点考虑：

- 1) 当系统投入使用的初期阶段时，由于用户量不足会导致投票量过少甚至无人投票，因此很多释义的得分都是初始得分。如果不同来源的释义不加区分都具有相同的初始总得分，此时的释义展示顺序就不具备参考价值。因为系统预置的专业名词释义是经过筛选的，相比较来源于金山词霸和用户提交的释义具有更高的权威性和可靠性。
- 2) 在一定程度上避免出现用户恶意提交然后自己给自己点赞。如果某个单词的所有释义当前分数均为 0，此时有用户随便提交了一个释义，初始得分为 0，然后自己给自己点赞，这时候他的释义分数就会有所增加。以后其他用户在查询该单词时，那个恶意用户的释义反而会被置顶，这样就回导致一些用户被恶意释义所误导。

3.5 系统的数据库设计

本系统涉及三类实体，分别为用户信息表、词库表和用户操作记录表。

这些实体的属性分别为：

- (1) 用户信息表：用户名、密码、用户等级、当前状态；
- (2) 词库表：释义贡献者、单词拼写、释义、点赞权值、反对权值、总得分、点赞数、反对数；
- (3) 用户操作记录表：操作者、单词拼写、释义、点赞标识位、反对标识位；

E-R 图能够生动形象的反映出各个实体之间的关系^[18]。该系统涉及的三个实体之间的关系如图 3-3。

由图 3-3 可得，用户和用户操作是一对多的关系，一个用户可以具有多个用户操作，其中用户操作是弱实体集，依赖于用户这个强实体集。用户和用户操作之间由用户投票这一动作相联系。

用户和词库是多对多的关系，一个用户可以贡献多个单词释义，一个单词可以有来源于多个用户的释义，其中单词的释义属性是多值属性。用户和词库通过用户查询和提交两个动作相联系。

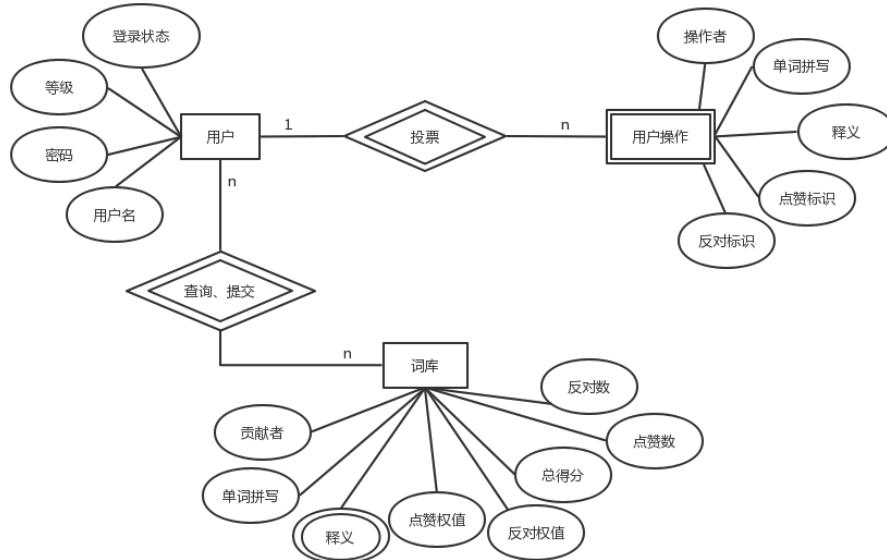


图 3-3 数据库 E-R 图

本系统的数据库表有三个，分别为 users、dict、operation。下面分别对各个表的设计进行阐述：

(1) 用户信息表

用户信息表里面存放的是用户的基本信息，如表 3-4 所示：

表 3-4 用户信息表

名称	类型	是否可空	说明
user_name	String	否	用户名
password	String	否	用户密码
level	int	否	用户等级
status	int	否	用户是否在线 在线 status=1 离线 status=0

(2) 词库表

词库表里面存放的是各个单词对应的释义、贡献者名称、点赞权值、反对权值、释义得分以及点赞数和反对数，如表 3-5 所示：

表 3-5 词库表

名称	类型	是否为空	说明
----	----	------	----

user_name	String	否	释义贡献者用户名
word	String	否	单词
interpret	String	否	单词释义
likes	int	否	点赞权值
hates	int	否	反对权值
scores	double	否	总得分
like_num	int	否	点赞数
hate_num	int	否	反对数

(3) 用户操作表

用户操作表里面存放的是用户对某一单词释义的操作情况，如表 3-6 所示：

表 3-6 用户操作表

名称	类型	是否为空	说明
user_name	String	否	释义操作者用户名
word	String	否	单词
interpret	String	否	单词释义
likeflag	int	否	是否进行过点赞操作
hateflag	int	否	是否进行过反对操作

3.6 客户端和服务器交互信息设计

本系统客户端和服务器采用 Socket 进行连接，客户端发送的不同请求具有不同的信息格式，下面将交互信息的格式以及各个字段的意义进行简要阐述：

(1) 用户登陆交互信息

客户端将用户登陆信息发送给服务器：

0	Username	Password
---	----------	----------

- 0:登录请求标识
- Username:用户名
- Password:密码

服务器返回信息：用户名以及密码（用户登陆成功）或者 null（该用户不存在）
登陆成功服务器返回信息格式：

Username	Level
----------	-------

(2) 用户注册交互信息

客户端将新用户信息发送给服务器：

1	Username	Password	Level	Status
---	----------	----------	-------	--------

- 1:注册请求标识
- Username:用户名
- Password:密码
- Level:用户等级，初始默认为 0
- Status:用户在线状态，初始默认为不在线

服务器返回信息：300（注册失败，该用户已经存在）或者 201（注册成功）

(3) 单词查询交互信息

客户端将要查询的单词发送给服务器：

2	Word
---	------

- 2:单词查询请求标识
- Word:要查询的单词

如果 dict 表中没有该单词，返回 null；

如果 dict 表中有该单词，服务器返回该单词的所有释义：

Username	Word	Interpret	Like_num	Hate_num	Scores	；
----------	------	-----------	----------	----------	--------	---

- Username:该单词释义贡献者
- Word:要查询的单词
- Interpret:单词释义之一
- Like_num:该释义点赞数
- Hate_num:该释义反对数
- Scores:总得分

如果在服务器上查询不到任何结果，调用金山 AIP。

客户端发送信息：<http://dict-co.iciba.com/api/dictionary.php?w=?&key=XX>

- W:要查询的单词
- Key:开发者身份验证码

金山返回结果如下，本系统只使用名词解释。

```
<dict num="219" id="219" name="219">
<key>computer science</key>
<ps>kəm'pjゅ:tə 'saɪəns</ps>
<pron>http://res.iciba.com/resource/amp3/0/0/80/c1/80c18801f6ce7e664f788b70032cdcd1.mp3</pron>
```

```

<ps>kəm'pjutər 'saɪəns</ps>
<pron>http://res-tts.iciba.com/8/0/c/80c18801f6ce7e664f788b70032cdcd1.mp3</pron>
<pos></pos>
<acceptation>计算机科学;
</acceptation>
<sent><orig>
My favorite subjects are Computer Science and Economics, especially Management of Computer Information Science.
</orig>
<trans>
本科学习经济管理专业,具备了初步的经济学修养.
</trans></sent>
</dict>

```

(4) 释义提交交互信息

客户端将用户提交的释义发给服务器:

3	Username	Word	Interpret	Like	Hates
---	----------	------	-----------	------	-------

- 3:用户释义提交请求标识
- Username:提交释义的用户
- Word:补充释义的单词
- Interpret:用户提交的释义
- Likes:该释义点赞权值, 默认为 0
- Hates:该释义反对权值, 默认为 0

服务器返回信息: 301 (已经存在该释义) 或者 202 (用户释义成功提交)

(5) 用户投票交互信息

客户端发送给服务器, 查询用户是否已经对该释义投过赞成票:

4	Username	Word	Interpret
---	----------	------	-----------

- 4:用户赞成历史操作查询标识
- Username:进行点击的用户
- Word:单词
- Interpret:进行打分的释义

客户端发送给服务器, 查询用户是否已经对该释义投过反对票:

5	Username	Word	Interpret
---	----------	------	-----------

- 5:用户反对历史操作查询标识
- Username:进行点击的用户
- Word:单词
- Interpret:进行打分的释义

服务器返回信息:204 (用户以前没有点过赞) 或者 304 (用户已经点过赞)

205 (用户以前没有点过反对) 或者 305 (用户已经点过反对)

客户端将用户投票信息发送给服务器:

6	Username	Word	Interpret	Likes
---	----------	------	-----------	-------

- 6:更新赞成票请求标识
- Username:进行点击的用户
- Word:单词
- Interpret:进行打分的释义
- Likes:用户本次操作增加的点赞权值, 计算方式为 (用户等级+1)。如果用户本次操作是点赞, 发送的 Hates 为 0, 表示没有进行反对操作

客户端将用户投票信息发送给服务器:

7	Username	Word	Interpret	Hates
---	----------	------	-----------	-------

- 7:更新反对票请求标识
- Username:进行点击的用户
- Word:单词
- Interpret:进行打分的释义
- Hates:用户本次操作增加的反对权值, 计算方式为 (用户等级+1)。如果用户本次操作是反对, 发送的 Likes 为 0, 表示没有进行点赞操作

(6) 用户密码修改交互信息

客户端将修改信息发给服务器:

8	Username	Password
---	----------	----------

- 8:修改密码请求标识
- Username:用户名
- Password:新密码

服务器返回信息: 204 (密码修改成功) 或着 404 (未找到该用户)

3.7 系统的界面设计

系统用户界面各个单元不是随意叠加的, 在设计过程中, 不仅要考虑到后台功能的实现, 还要考虑到用户界面上各个单元和底层逻辑的对应关系。如果不对应关系和整体结构进行考虑, 容易使整个系统出现混乱, 甚至系统崩溃。因此, 为了避免上述情况发生, 本系统在系统界面设计上采用了以下几个原则:

- (1) 为了方便用户操作, 该 APP 所有的功能均采用图像化显示界面, 直截了当的显示在用户眼前^[19], 无需考虑其他方法来使用本 APP;

-
- (2) 用户界面设计简洁明了、配色合理，操作入口清晰明显，用户对于所有功能都能一目了然^[20]；
 - (3) 由于用户在使用本 APP 时要进行大量的输入操作，因此界面的输入设计应该方便快捷，并且对于用户的输入应该首先检查其有效性，对于无效输入应当及时使用 Toast 给用户反馈；
 - (4) 对于没有查询到任何释义的情况应当使用 Toast 给出提示语或者背景显示提示图片；
 - (5) 用户的操作应当进行持久化存储，并且各个界面应与用户的行为保持一致^[21]，使得用户尽快适应软件的使用。

3.8 本章小结

本章从系统需求入手，首先对系统的功能性需求和非功能性需求进行了分析。接着描述了该系统的软件层次架构和功能模块设计，然后对系统中使用的关键技术和思想进行了阐述，接着采用 E-R 图和文字说明了该系统数据库的设计方案，最后简要说明了本系统的信息交互格式和采用的界面设计原则。

第四章 系统详细设计与实现

前几章对该选题的背景、相关理论技术、以及对系统需求进行了分析，本章将对前几章提及的系统核心模块进行详细的设计描述。

4.1 核心功能模块设计

4.1.1 在线查询模块

用户成功登陆后，在联网状态下输入要进行查询的单词，点击确认后发送给服务器进行查询，最终查词得到的结果会逐条显示在界面上。

(1) 在线查询模块时序图如图 4-1 所示：

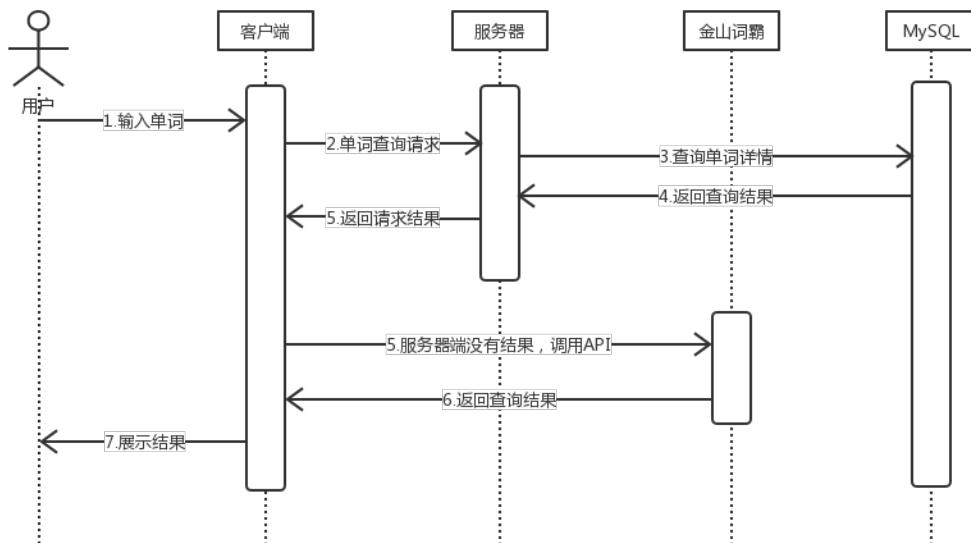


图 4-1 在线查询模块时序图

(2) 在线查询的流程图如图 4-2 所示。

根据图 4-2，在线查询详细流程如下：

- 1) 用户输入要查询的单词；
- 2) 客户端判断用户输入是否为空，如果为空返回第 1 步，不为空走到第 3 步；
- 3) 客户端将用户输入的单词转化为字符串，发送给服务器；
- 4) 服务器接收到字符串后在数据库 dict 表中查询该单词，如果表中存在该单词的释义，则将所有释义以字符串形式返回给客户端；如果表中不存在该单词，则返回码为 405；
- 5) 如果客户端收到返回码是 405，则到第 5 步；如果收到返回信息不是 405，则到第 6 步；

- 6) 客户端和通过 API 和金山词霸服务器建立连接，从金山词霸获得该单词的释义，如果有释义返回，走到第 6 步；如果没有信息返回，则在界面上显示提示信息，通知用户该单词目前没有释义；
 7) 客户端收到后对释义进行处理并显示在界面上。

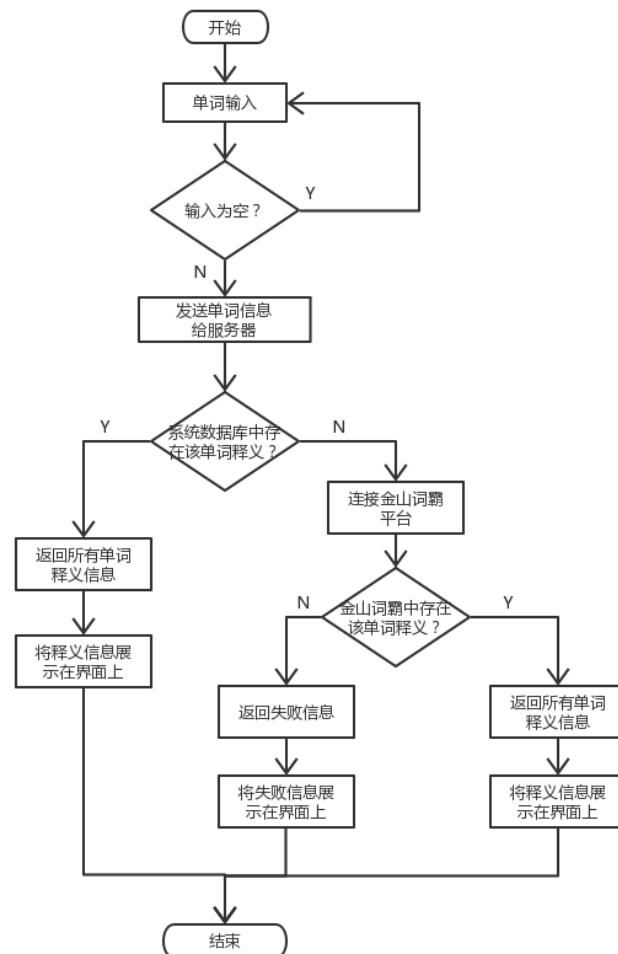


图 4-2 在线查询模块流程图

(3) 在线查询模块类图如图 4-3 所示。

下面对涉及到的各个类以及主要函数进行详细解释，其中 DictionaryActivity、DictionaryAdapter、Dictionary、ConnectNet、ContentHandler 和 SocketHsndler 这六个为主要类，其他为内联类。类与类之间的详细关联见下图 4-3。

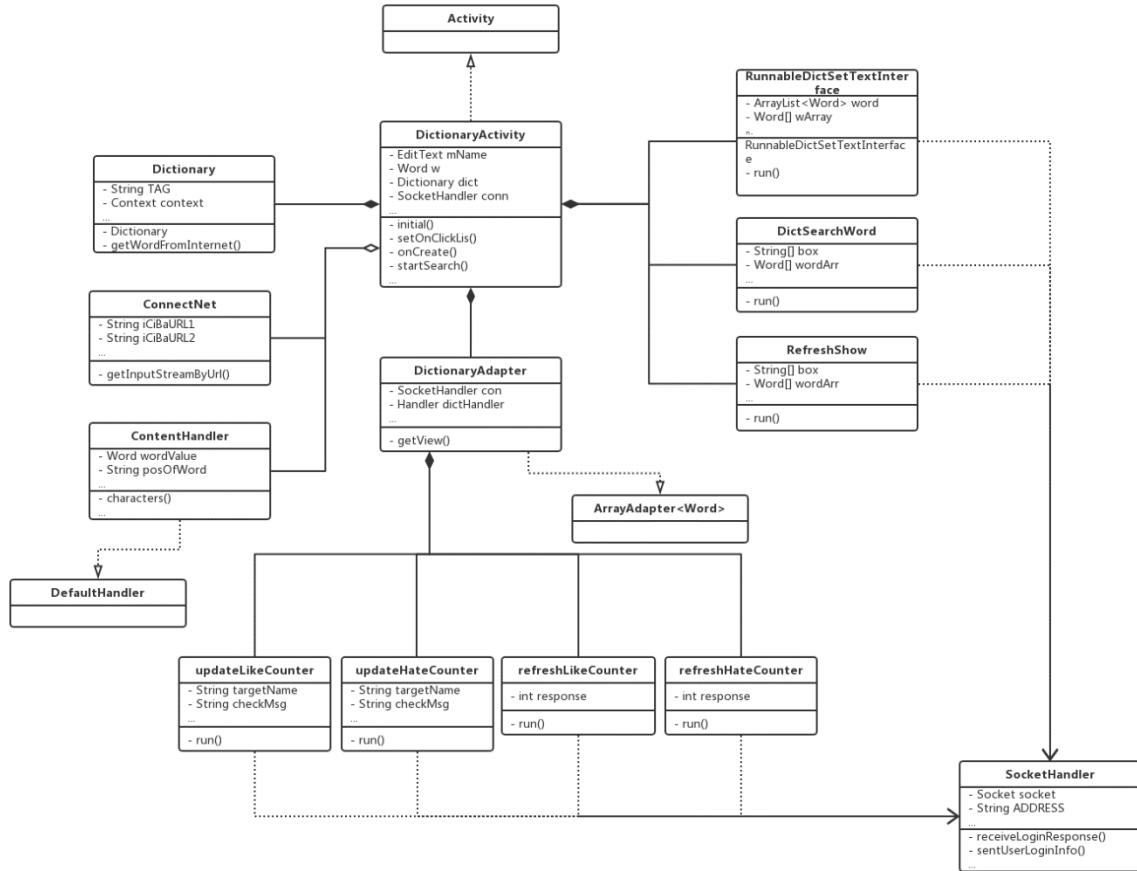


图 4-3 在线查询模块类图

涉及在线查询功能的主要函数伪代码如下：

- 1) DictionaryActivity: 将用户输入单词进行保存并处理接收到的数据。
提取用户输入单词函数，将用户输入数据进行存储并调用单词查询线程。伪代码如图 4-4:

```

1   public void startSearch() {
2       /*str 和 searchWord 均为字符串类型, imm 为 InputMethodManager 类型*/
3       将用户输入转化为字符串, 赋值给 str;
4       If str = null or str =""
5           then return
6       str 的值赋给 searchedWord;
7       开启新线程 DictSearchWord();
8       打开输入法窗口 imm;
9       If imm != null
10          then 关闭输入法窗口
11      }
  
```

图 4-4 开始查询函数伪代码

单词释义数据处理函数，主要功能为将单词的多个释义分割为多条记录，然后用于展示，伪代码如图 4-5：

```

1  public class DictSearchWord implements Runnable {
2      /*temp 为字符串数组类型
3      *wordArr 为自定义 Word 数组类型
4      * words 为 ArrayList<Word>类型，其余均为字符串类型*/
5      连接服务器;
6      从服务器获取的数据赋值给 temp;
7      If temp = null
8          then 连接金山词霸;
9          if 金山词霸返回为空
10             then 提示用户该单词目前没有任何释义;
11             else if 返回不为空
12                 then 将返回信息进行分割，放入字符串数组 ingerInterpret;
13                 for i=0 to ingerInterpret.length
14                     取出释义信息放入 wordArr 中;
15                 end
16                 将释义展示在界面上;
17             else if temp != null
18                 then 清空 words
19                 for i=0 to temp.length
20                     取出释义信息存放在 wordArr 中;
21                 end
22                 将释义展示在界面上;
23 }
```

图 4-5 单词释义处理函数伪代码

2) DictionaryAdapter：将释义展示在界面上

列表显示函数，伪代码如图 4-6：

```

1  public View getView(int position, final View convertView, ViewGroup parent){
2      /*currentWord 为 Word 类型
3      *listItemView 为 View 类型
4      *页面文本为 TextView 类型，图片为 ImageView 类型*/
5
6      当前释义位置赋值给 currentWord;
7      当前视图信息赋值给 listView;
8      .....
9      获取当前页面组件 ID 并存入相应的变量中;
10     userNickName.setText(currentWord.getUserName()); //设置页面文本
11     .....
12     imageView.setImageResource(currentWord.getLikeImageResourceID()); //设置图片
13     imageView.setVisibility(View.VISIBLE);
14     imageView.setOnClickListener(mListener); //设置监听器
15     .....
16 }
```

图 4-6 列表显示函数伪代码

3) Dictionary：向金山词霸发起连接并对解析模块进行初始化

获取网络释义函数，生成单词查询的 URL，初始化 XML 解析模块。伪代码如图 4-7：

```

1  public Word getWordFromInternet(String searchedWord) {
2      /*word 为 Word 类型, tempUrl 为 String 类型, in 为 InputStream 类型*/
3      金山词霸 URL 赋值给 tempUrl;
4      输入信息流赋值给 in;
5      if in != null
6          then 初始化信息解析模块
7          return word
8  }

```

图 4-7 网络获取释义函数伪代码

- 4) ConnectNet: 使用 HTTP 连接金山词霸平台, 根据给出的 URL, 从网络获得输入流

获取输入流函数, 发起数据请求并设置超时时间。伪代码如图 4-8:

```

1  public InputStream getInputStreamByUrl(String urlStr) {
2      /* tempiInput 为 InputStream 类型
3      connection 为 HttpURLConnection 类型*/
4      HttpURLConnection 的连接赋值给 connection;
5      发起 GET 类型连请求;
6      设置连接超时时间 8000ms;
7      设置信息接收超时时间 10000ms;
8      if 返回码=200
9          then 返回数据赋给 tempiInput
10     else
11         打印错误日志
12 }

```

图 4-8 输入流获取函数伪代码

- 5) ContentHandler: 使用 SAX 解析器对 XML 进行解析

字符处理函数, 获取节点中内容并对内容进行筛选。伪代码如图 4-9:

```

1  public void characters(char[] ch, int start, int length){
2      /*str 为字符串对象类型
3      *flag 为整数类型
4      *wordValue 为 Word 类型, 其余均为字符串类型*/
5      for i=0 to start+length
6          if ch[i] = '\n'
7              return;
8      end
9
10    将 XML 信息存储在 str 中等待解析;
11    if tagName = "key"
12        then 该信息为单词拼写, 将它存储在 wordValue 中;
13        else if tagName = "pos"
14            then if str = n. or str = abbr. or str = ""
15                then 该释义为名词, flag<-1
16                else if tagName = "acceptation" and flag=1
17                    then 取出所有名词释义, 每个名词释义之间以分号或逗号隔开
18                    将释义存储在 wordValue 中
19 }

```

图 4-9 XML 解析函数伪代码

6) SocketHandler: 与服务器进行数据交互

消息发送函数。伪代码如图 4-10:

```

1  public void searchWord(String wordInfo) {
2      /*ADDRESS 为字符串类型
3      *socket 为 java.net.Socket 类型
4      *serverAdd 为 InetAddress 类型*/
5      if wordInfo == null
6          then return
7      服务器的 IP 地址赋值给 serverAddr;
8      应用 Server 的 IP 和端口建立 Socket 对象, Socket(地址, 端口号);
9      初始化 PrintWriter out;
10     out.flush();
11 }
```

图 4-10 查询消息发送函数伪代码

接收服务器返回数据函数。伪代码如图 4-11:

```

1  public String[] searchWordResponse() {
2      /*br 为 BufferedReader 类型, response 为 String 类型
3      *res 为字符串数组类型*/
4      response<-br.readLine(); //通过输入流读取器对象 接收服务器发送过来的数据
5      if response == null
6          then return null
7      else
8          res <- response.split(";") //把接收到的字符串以分号分割, 存入数组
9          return res
10 }
```

图 4-11 查询结果接收函数伪代码

(4) 在线查询模块界面如图 4-12 所示:

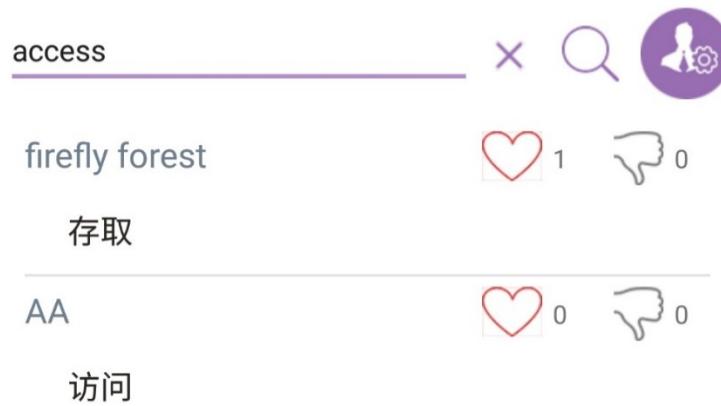


图 4-12 在线查询模块界面

单词输入: 位于屏幕最上方, 由输入栏、清空图标和查询图标三部分组成。当用户点击查询图标时, 客户端就会记录下用户输入内容, 然后发送给服务器或者金山词霸平台进行查询。如果用户输入失误, 则可以点击清空图标将输入栏清空;

查询结果展示：每一条信息都是单词的一个释义，其中包括提交该释义的用户名，名称下面是释义。右侧是点赞按钮/点赞数、反对按钮/反对数。

4.1.2 用户投票模块

对于查询到的每一个释义，用户都可以进行点赞或者反对，对于同一个释义不能够进行多次点赞或者反对，重复操作会导致上一次的操作无效。同时用户的每一个点赞或者反对都会对释义的得分有影响，最终经过大量用户的投票情况和威尔逊算法的计算后，那些得到广泛认可的释义会排到最前面。

(1) 用户投票模块时序图如图 4-13 所示：

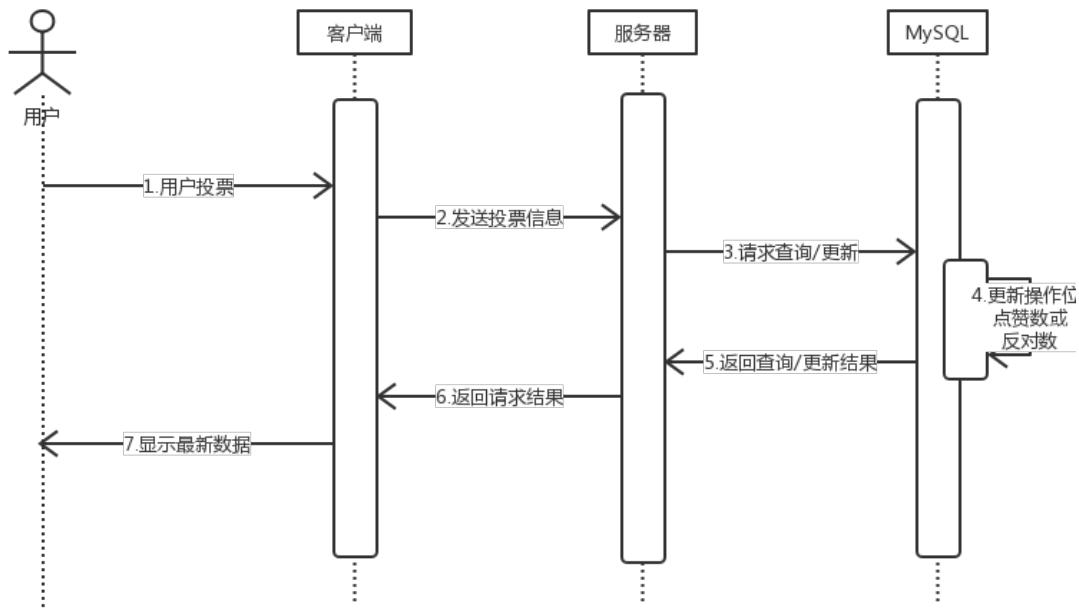


图 4-13 投票模块时序图

(2) 用户投票的流程图如图 4-14 所示。

根据图 4-14，用户投票详细流程如下：

- 1) 用户对某个释义点击赞成或者反对；
- 2) 客户端将操作用户的用户名、被操作的单词以及释义、用户具体操作信息组成一个长字符串发送给服务器；
- 3) 服务器接收到客户端发送的信息后，根据信息内容重新计算该释义的总分数，并且更新 operation 表中的用户操作标识位；
- 4) 客户端更新界面上用户操作后的赞成和反对的统计数字。



图 4-14 用户投票模块流程图

(3) 用户投票模块类图如图 4-15 所示：

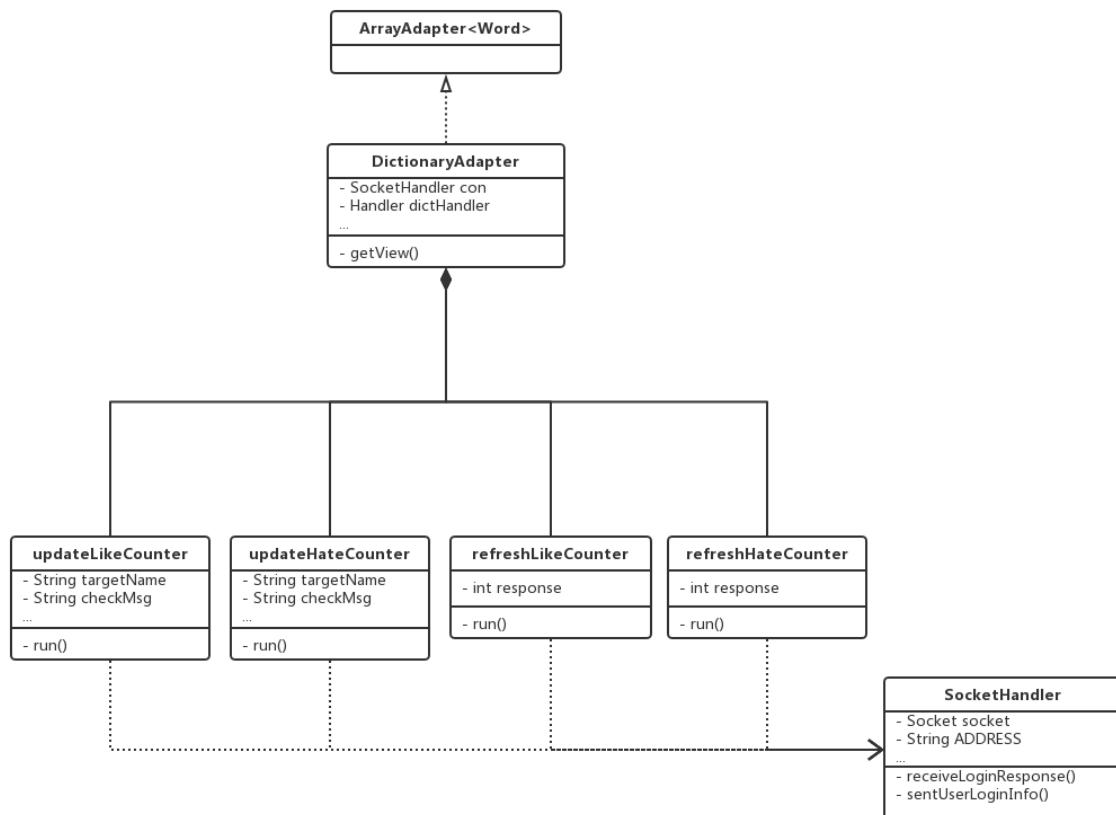


图 4-15 用户投票模块类图

DictionaryAdapter 继承 ArrayAdapter 类，主要作用为将每一条单词释义信息展示在界面上，以及对计数值进行更新操作。SocketHandler 主要用于连接服务器、发送和接收信息。涉及用户投票功能的主要函数伪代码如下：

- 1) DictionaryAdapter：连接后端数据和前端显示的适配器接口类
列表显示函数。伪代码如图 4-16：

```

1  public View getView(int position, final View convertView, ViewGroup parent){
2      /*currentWord 为 Word 类型
3      *listItemView 为 View 类型
4      *页面文本为 TextView 类型，图片为 ImageView 类型*/
5
6      当前释义位置赋值给 currentWord;
7      当前视图信息赋值给 listView;
8      .....
9      获取当前页面组件 ID 并存入相应的变量中;
10     userNickName.setText(currentWord.getUserName()); //设置页面文本
11     .....
12     imageView.setImageResource(currentWord.getLikeImageResourceID()); //设置图片
13     imageView.setVisibility(View.VISIBLE);
14     imageView.setOnClickListener(mListener); //设置监听器
15     .....
16 }
```

图 4-16 列表显示函数伪代码

更新界面赞同统计数函数，更新反对统计数函数与之逻辑完全相同。伪代码如图 4-17 和 4-18：

```

1  class updateLikeCounter implements Runnable {
2      /*likeRes 为整型变量
3      *其余全为字符串型变量*/
4
5      获取当前操作用户的姓名;
6      获取当前操作用户的等级;
7      获取当前单词拼写;
8      获取当前单词释义;
9      .....
10     发送用户历史操作查询请求;
11     服务器返回信息赋值给 likeRes;
12     if likeRes = 204 //用户没有进行过操作
13         then
14             该释义增加点赞数为(用户等级+1), 发送给服务器
15             启动界面统计数字刷新线程
16         else if likeRes = 304 //用户曾经进行过操作, 再次点击取消点赞
17             then
18                 该释义增加点赞数为-(用户等级+1), 发送给服务器
19                 启动界面统计数字刷新线程
20 }
```

图 4-17 更新统计数字函数伪代码

```

1  class refreshLikeCounter implements Runnable {
2      /*response 是整形变量， clickLike 是整形变量*/
3
4      public refreshLikeCounter(int response) {
5          this.response<-response;
6      }
7
8      public void run() {
9          if response = 204
10             then 点赞数据加一
11         else if response = 304 and clickLike=1
12             then 点赞数据减一
13         else if response = 304 and clickLike>1
14             then 用户上次操作失效， 点赞数据不变
15     }

```

图 4-18 统计数字界面刷新函数伪代码

2) SocketHandler:

发送检查用户操作历史请求函数。伪代码如图 4-19:

```

1  public void checkLikeOperation(String checkInfo) {
2      /*ADDRESS 为字符串类型
3      *socket 为 java.net.Socket 类型
4      *serverAdd 为 InetAddress 类型*/
5
6      服务器的 IP 地址赋值给 serverAddr;
7      应用 Server 的 IP 和端口建立 Socket 对象， Socket(地址, 端口号);
8      创建 PrintWriter 实例 out;
9      out.flush();
10 }

```

图 4-19 用户操作历史请求函数伪代码

接收检查结果函数。伪代码如图 4-20:

```

1  public int receiveLikeOperation() throws IOException {
2      /*br 为 BufferedReader 类型, response 为 String 类型
3      * res 为字符串数组类型*/
4      response<-br.readLine();    //通过输入流读取器对象 接收服务器发送过来的数据
5      res <- Integer.parseInt(response); //将字符串转化为整型
6      return res;
7  }

```

图 4-20 接收操作结果函数伪代码

(4) 用户投票模块界面如图 4-21 所示:

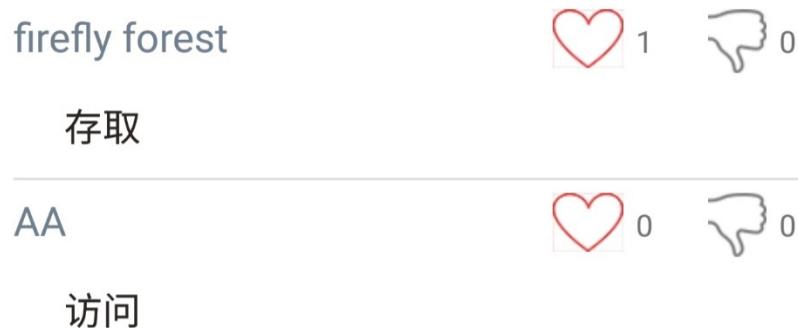


图 4-21 用户投票界面

赞成票：爱心为投票按钮，右边为该释义当前总赞成数。当用户进行点击时，统计数字会根据用户之前有无进行过点赞发生相应变化。

反对票：向下的大拇指为反对按钮，右边为该释义当前总反对数。当用户进行点击时，统计数字会根据用户之前有无进行过反对发生相应变化。

4.1.3 用户释义提交模块

用户进行查询后，如果对现有结果不满意或者没有查询到任何结果，可以从其他途径得到解释后给本系统。

(1) 用户释义提交模块时序图如图 4-22 所示：

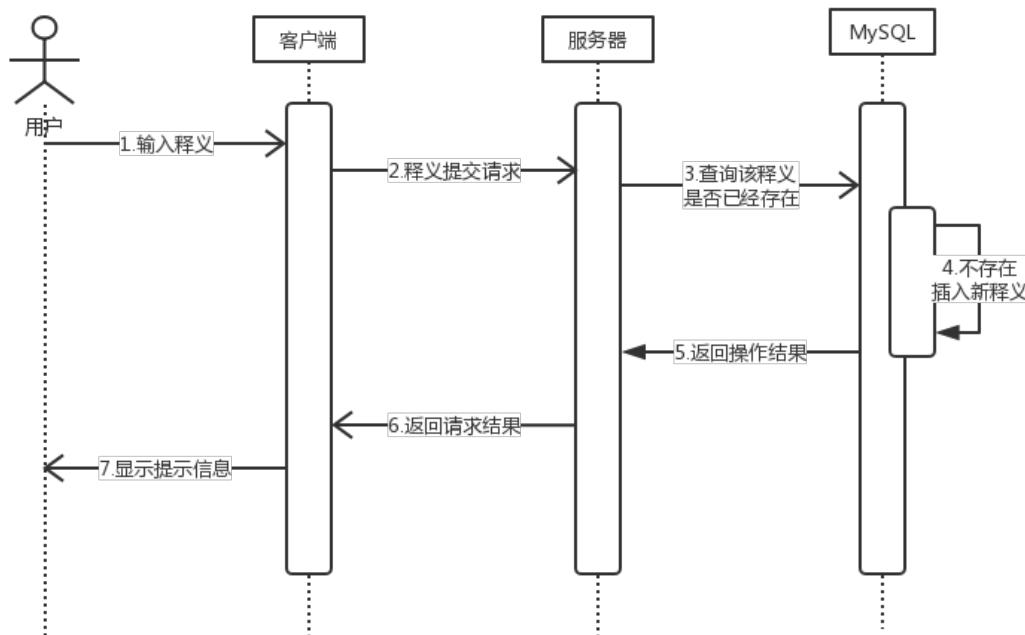


图 4-22 用户释义提交模块时序图

(2) 用户释义提交流程图如图 4-23 所示：

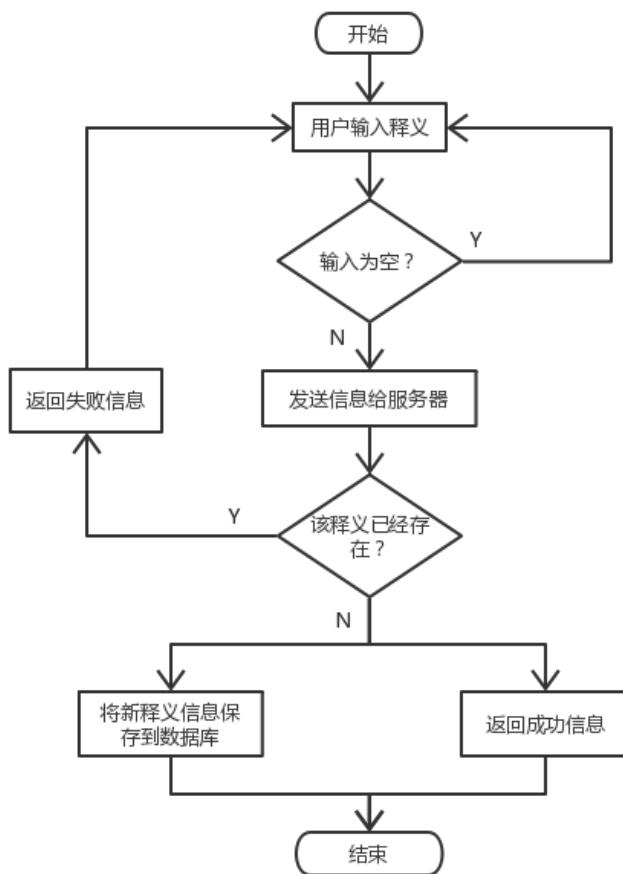


图 4-23 用户释义提交模块流程图

根据图 4-23，用户释义提交详细流程如下：

- 1) 用户在输入框中输入释义，点击提交；
- 2) 客户端判断用户输入是否为空，如果为空返回第 1 步，不为空走到第 3 步；
- 3) 客户端将用户输入的释义转化为字符串，连同该单词一起发送给服务器；
- 4) 服务器接收到信息后在数据库 dict 表中查询该单词的所有释义，看用户提交的释义是否已经存在，如果不存在则插入 dict 表中，然后给客户端发送返回码 202；如果该释义已经存在给客户端发送返回码 301；
- 5) 客户端接收到服务器的返回码，如果为 202，则提醒用户刷新后查看提交的内容；如果为 301，弹出提示框告诉用户该释义已经存在。

(3) 用户释义提交模块类图如图 4-24 所示。在释义提交模块中，有 3 个类是主要类，分别为 DictionaryActivity、DictionaryAdapter、和 SocketHsndler，其余全为内联类。类与类之间的详细关联如下图 4-24。

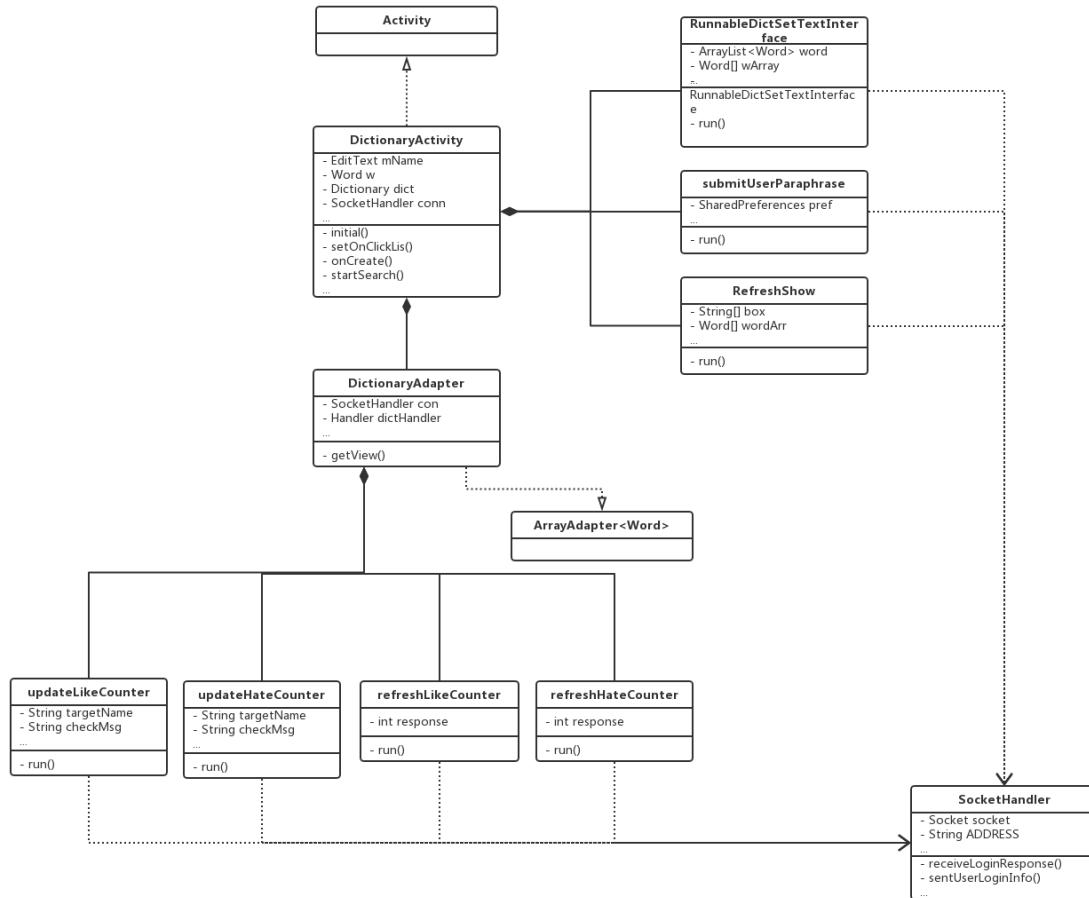


图 4-24 用户释义提交模块类图

1) DictionaryActivity: 主要功能为发起释义提交请求以及下拉刷新

提取用户输入释义函数，调用用户释义处理线程。伪代码如图 4-25 所示：

```

1  class submitUserParaphrase implements Runnable {
2      /* pref 为 SharedPreferences 型变量， response 为整型变量
3      * 其余全为字符串型变量*/
4
5      将用户输入转化为字符串， 赋值给 userParaphrase;
6      if userParaphrase = null or userParaphrase = ""
7          then 使用 Toast 提示用户输入为空
8          else
9              获取释义提交者名称;
10             单词拼写+用户名称+释义赋值给 sendInfo;
11             将 sendInfo 发送给服务器;
12
13             服务器返回信息赋给 response;
14             if response = 301
15                 then 使用 Toast 提示用户该释义已经存在
16                 else if response = 202 使用 Toast 提示用户刷新后查看提交的释义
17     }
  
```

图 4-25 用户释义提交函数伪代码

下拉刷新函数。伪代码如图 4-26 所示：

```

1  pclass RefreshShow implements Runnable{
2      /*box、temp 为字符串数组类型
3      *wordArr 为自定义 Word 数组类型
4      * words 为 ArrayList<Word>类型，其余均为字符串类型*/
5      连接服务器进行单词查询；
6      从服务器获取的数据赋值给 temp;
7      清空 words;
8      for i=0 to temp.length
9          box <- temp[i].split("\\|");
10         取出释义信息存放在 wordArr 中;
11     end
12     开启 RunnableDictSetTextInterface 线程;
13 }
```

图 4-26 下拉刷新函数伪代码

- 2) **DictionaryAdapter:** 为适配器类，主要功能为连接后端数据和前端显示的适配器接口

列表显示函数。伪代码如图 4-27 所示：

```

1  public View getView(int position, final View convertView, ViewGroup parent){
2      /*currentWord 为 Word 类型
3      *listItemView 为 View 类型
4      *页面文本为 TextView 类型，图片为 ImageView 类型*/
5
6      当前释义位置赋值给 currentWord;
7      当前视图信息赋值给 listView;
8      .....
9      获取当前页面组件 ID 并存入相应的变量中;
10     userNickName.setText(currentWord.getUserName()); //设置页面文本
11     .....
12     imageView.setImageResource(currentWord.getLikeImageResourceID()); //设置图片
13     imageView.setVisibility(View.VISIBLE);
14     imageView.setOnClickListener(mListener); //设置监听器
15     .....
16 }
```

图 4-27 列表显示函数伪代码

- 3) **SocketHandler:** 主要用于连接服务器、发送和接收信息。

发送给服务器用户释义函数。伪代码如图 4-28 所示：

```

1  public void sendInterpretMsg(String wordInfo){
2      /*ADDRESS 为字符串类型
3      *socket 为 java.net.Socket 类型
4      *serverAdd 为 InetAddress 类型*/
5
6      服务器的 IP 地址赋值给 serverAddr;
7      应用 Server 的 IP 和端口建立 Socket 对象, Socket(地址, 端口号);
8      创建 PrintWriter 实例 out;
9      out.flush();
10 }

```

图 4-28 释义提交请求函数伪代码

服务器反馈接收函数。伪代码如图 4-29 所示：

```

1  public int receiveInterpretResponse() throws IOException{
2      /*br 为 BufferedReader 类型, response 为 String 类型
3      * res 为字符串数组类型*/
4      response<-br.readLine(); //通过输入流读取器对象 接收服务器发送过来的数据
5      res<- Integer.parseInt(response); //将字符串转化为整型
6      return res;
7
8
9 }

```

图 4-29 提交反馈函数伪代码

(4) 用户释义提交界面如图 4-30 所示：



图 4-30 用户投票界面

释义输入：位于屏幕最下方，由输入栏和提交按钮两部分组成。输入栏内有提示信息，用户开始输入后消失。当用户点击提交按钮后，客户端就会记录下用户输入内容，然后发送给服务器进行处理。

4.1.4 释义排序处理模块

该处理模块与查询后的释义显示顺序、用户点赞以及用户释义提交三个模块均有联系。是改进后的威尔逊算法的具体实现模块，也是本系统核心部分之一。

(1) 处理流程图如图 4-31 所示：

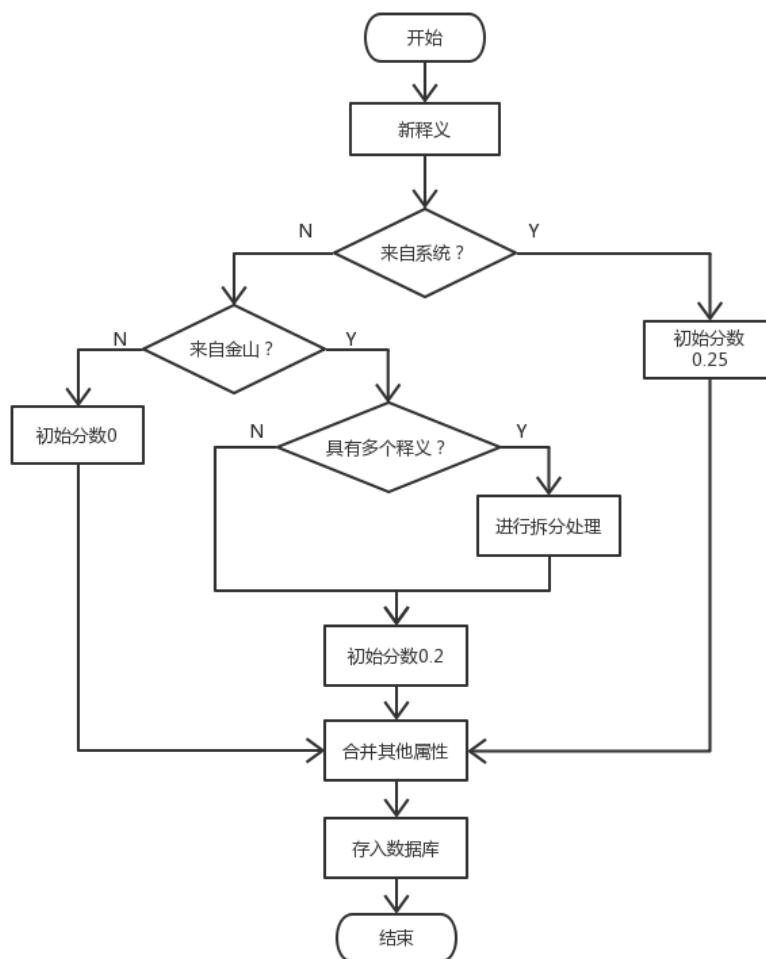


图 4-31 新释义处理流程图

根据图 4-31，新释义处理流程如下：

- 1) 有新释义要导入数据库；
- 2) 释义是否来自于系统导入。如果是，初始分数设置为 0.25，然后走到第 6 步。如果不是，走到第 3 步；
- 3) 释义是否来自于金山词霸。如果是，走到第 4 步。如果不是，初始分数设置为 0，走到第 6 步；
- 4) 金山词霸提供的释义是否包含多个结果。如果包含，进行拆分处理，否则不进行任何处理，然后走到第 5 步；
- 5) 初始分数设置为 0.2；
- 6) 单词释义合并其他属性，如贡献者、单词拼写等；
- 7) 将合并后的完整信息放入数据库中；

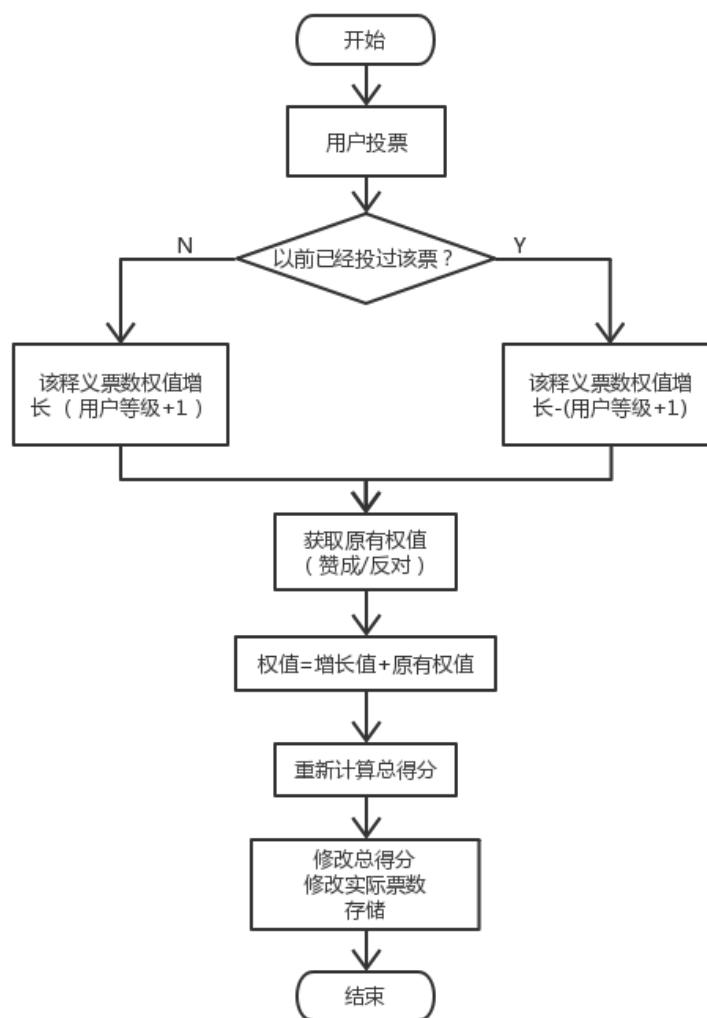


图 4-32 新释义处理流程图

根据图 4-32，新释义处理流程如下：

- 1) 用户进行投票；
- 2) 如果用户之前已经对该释义进行过投票，那么在此进行投票会使得上一次投票失效，如果没有进行过投票，那么此次投票有效；
- 3) 从服务器端获取该释义原来的权值（赞成/反对，视用户具体操作而定）；
- 4) 重新计算新的总票数；
- 5) 重新计算该释义总得分；
- 6) 更新数据库中总得分属性值；
- 7) 更新实际的票数；

(2) 具体实现部分工程代码如下：

- 释义筛选以及拆分处理的具体实现函数如下，代码截取了该功能实现的关键部分

```
public void characters(char[] ch, int start, int length)
    throws SAXException {
    .....
    // 一行一行的开始设置文本
    String str = new String(ch, start, length);
    if (tagName == "key") {
        wordValue.setWord(str);
    } else if (tagName == "pos") {
        if (str.equals("n.")) {
            posOfWord = posOfWord + str + " ";
            flag = 1;
        }
    } else if ((tagName == "acceptation") && (flag == 1)) {
        interpret = interpret + str;
        interpret = wordValue.getInterpret() + interpret;
        wordValue.setInterpret(interpret);
        interpret = ""; // 初始化操作，预防有多个释义
        flag = 0;
    }
}
```

- 威尔逊算法

```
public static double calculateScore(int like, int hate) {
    double score = 0.0;
    int n = 0;
    double p = 0.0, numerator = 0.0, denominator = 0.0;

    n = like + hate;
    if (n == 0) {
        System.out.println("score is:" + score);
        return score;
    } else {
        p = like / n;
        numerator = (p + (Za * Za) / (2 * n) - Za * Math.sqrt((p * (1 - p)) / n + (Za * Za) / (4 * n * n)));
        denominator = 1 + Za * Za / n;
        score = numerator / denominator;
        System.out.println("score is:" + score);

        return score;
    }
}
```

4.2 本章小结

本章主要描述的是系统详细设计过程，使用图表与代码、文字相结合的方式对系统的核心功能模块的工作流程进行介绍，并且详细描述了核心功能模块的具体实现过程。

第五章 系统测试

本章对该专业术语辅助学习系统进行测试。首先明确总体测试目标，然后对各个模块的功能进行测试，并对测试结果进行分析。

5.1 系统测试目标

测试目标是为了检验该系统能否正常运行，提高软件系统的可靠性^[22]。为了找出可能在软件运行过程中出现的问题，要设计详细全面的测试用例，从而能够及时进行系统修复^[23]。

5.2 系统功能测试以及结果分析

该系统客户端是在 Android Studio 3 的环境下开发的，服务器是在 Eclipse Java Oxygen 环境下开发的。测试机为 Samsung S9，Nexus 6。下面是对系统进行功能测试的结果，每个测试至少 5 轮。

(1) 用户注册测试以及结果

用户注册子模块主要包括用户名输入、密码输入以及确认和取消按钮的点击事件，测试用例如表 5-1 所示：

表 5-1 用户注册测试用例

子模块	前置条件	操作步骤	预期结果	实际结果
用户名输入	用户已经联网，密码和确认密码均正确输入	输入为空，点击确认注册按钮	弹出提示框，提示用户名输入为空	与预期一致
		输入一个已经存在的账户，点击确认注册按钮	弹出提示框，提示该用户名已经被注册	与预期一致
		正确输入用户名，点击确认注册按钮	用户成功注册，跳转至登录界面	与预期一致
密码输入	用户已经联网，用户名正确输入	输入为空，点击确认注册按钮	弹出提示框，提示密码输入为空	与预期一致

	密码长度小于 6 位, 点击确认注册按钮	弹出提示框, 提示密码长度小于 6 位	与预期一致
	密码和确认密码不一致, 点击确认注册按钮	弹出提示框, 提示两次密码输入不一致	与预期一致
	正确输入密码以及确认密码, 点击确认注册按钮	用户成功注册, 跳转至登录界面	与预期一致
取消按钮	当前界面为用户注册界面	用户点击取消按钮	跳转至登录界面
确认按钮	用户信息均已正确输入	用户点击确认按钮	跳转至登录界面

根据测试用例来进行测试的一部分结果如下图：

图 5-1 为用户名输入正确，但是两次输入密码不一致时的情况：



图 5-1 用户注册密码测试

图 5-2 为密码输入正确，但是用户名已经被注册时的情况：



图 5-2 用户注册用户名测试

(2) 用户登录测试以及结果

用户登录子模块主要包括用户名输入、密码输入以及确认按钮的点击事件，测试用例如表 5-2 所示：

表 5-2 用户登录测试用例

子模块	前置条件	操作步骤	预期结果	实际结果
用户名输入	用户已经联网，密码正确输入	输入为空，点击登录按钮	弹出提示框，提示用户名输入为空	与预期一致
		输入错误的用户名，点击登录按钮	弹出提示框，提示该用户名不存在或者提示用户检查账号与密码	与预期一致
		正确输入用户名，点击登录按钮	用户成功注册，跳转至登录界面	与预期一致

密码输入	用户已经联网，用户名正确输入	输入为空，点击登录按钮	弹出提示框，提示密码输入为空	与预期一致
		密码长度小于6位，点击登录按钮	弹出提示框，提示密码长度小于6位	与预期一致
		输入错误密码，点击登录按钮	弹出提示框，提示用户检查账户和密码	与预期一致
		正确输入密码，点击登录按钮	用户成功注册，跳转至登录界面	与预期一致
确认按钮	用户信息均已正确输入	用户点击确认按钮	跳转至查询界面	与预期一致

根据测试用例来进行测试的一部分结果如下图：

图 5-3 为用户名输入正确，但是密码长度小于 6 位时的情况：



图 5-3 用户登录密码测试

图 5-4 为密码输入正确，但是用户名为空时的情况：



图 5-4 用户登录用户名测试

(3) 查询测试以及结果

单词查询子模块主要包括输入框以及删除图标、搜索图标的点击事件，测试用例如表 5-3 所示：

表 5-3 单词查询测试用例

子模块	前置条件	操作步骤	预期结果	实际结果
输入框	无文本输入	查看样式显示是否正确	显示灰色提示信息“请在这里输入单词”	与预期一致
	有文本输入		光标闪烁，软键盘弹出	与预期一致
删除图标	输入框内有信息	点击删除图标	输入框内信息全部清除	与预期一致
搜索图标	用户已经联网，输入框为空	点击搜索图标	弹出提示框，提示用户输入框内不能为空	与预期一致
	用户已经联网，输入框不为空		屏幕上显示该单词的所有释	与预期一致

			义，按照分数 降序排列	
--	--	--	----------------	--

根据测试用例来进行测试的一部分结果如下图：

图 5-5 为正确输入并且联网时的情况：

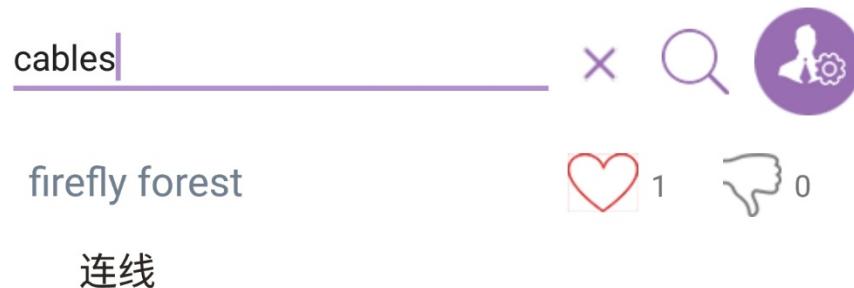


图 5-5 单词查询测试

(4) 用户投票测试以及结果

用户投票子模块主要包括赞同图标、反对图标点击事件以及下拉刷新事件，测试用例如表 5-4 所示：

表 5-4 用户投票测试用例

子模块	前置条件	操作步骤	预期结果	实际结果
赞同图标	用户之前从来没有对该释义点过赞	用户第一次点击赞同	赞同数在原来的基础上加 1	与预期一致
		用户又一次点击赞同	赞同数减 1，变为原来的值	与预期一致
	用户之前对该释义点过赞	用户第一次点击赞同	赞同数在原来的基础上减 1	与预期一致
		用户又一次点击赞同	赞同数加 1，变为原来的值	与预期一致
反对图标	用户之前从来没有对该释义点过反对	用户第一次点击反对	反对数在原来的基础上加 1	与预期一致
		用户又一次点击反对	反对数减 1，变为原来的值	与预期一致
	用户之前对该释义点过反对	用户第一次点击反对	反对数在原来的基础上减 1	与预期一致
		用户又一次点击反对	反对数加 1，变为原来的值	与预期一致

下拉刷新	目前为单词查询界面	手指下拉屏幕	出现旋转图标，显示该单词的最新释义以及最新排名情况	与预期一致
------	-----------	--------	---------------------------	-------

根据测试用例来进行测试的一部分结果如下图：

图 5-6 为用户对释义进行一次点赞后的情况：



图 5-6 释义一次投票测试

图 5-7 为用户对同一个释义进行又一次点赞后的情况，可见之前点的赞会取消：



图 5-7 释义两次投票测试

(5) 用户释义提交测试以及结果

用户释义提交子模块主要包括输入框、提交按钮的点击事件、下拉刷新事件，测试用例如表 5-5 所示：

表 5-5 用户释义提交测试用例

子模块	前置条件	操作步骤	预期结果	实际结果
输入框	无文本输入	查看样式显示是否正确	显示灰色提示信息“请在这里输入释义”	与预期一致
	有文本输入		光标闪烁，软键盘弹出	与预期一致
提交按钮	用户已经联网，输入框为空	点击提交按钮	弹出提示框，提示用户提交释义不能为空	与预期一致
	用户已经联网，输入框不为空		弹出提示框，提示用户刷新后可以看到提交的释义	与预期一致
下拉刷新	目前为单词查询界面	手指下拉屏幕	出现旋转图标，显示单词的最新释义以及最新排名情况	与预期一致

根据测试用例来进行测试的一部分结果如下图：

图 5-8 为用户输入释义为空时的情况：

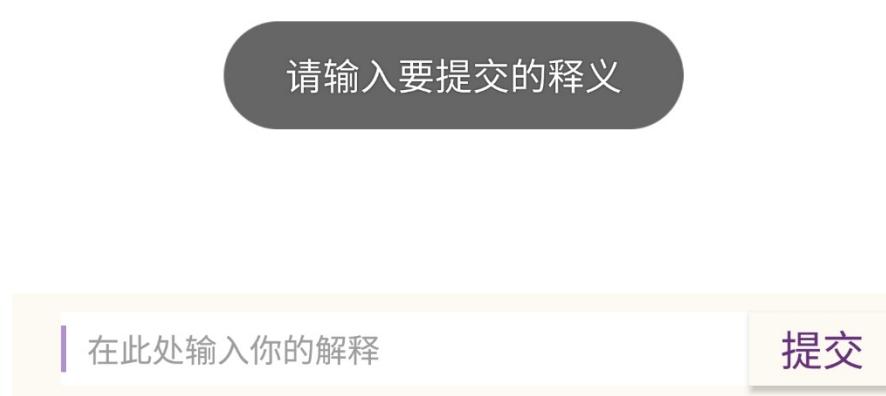


图 5-8 释义提交框测试

图 5-9 为用户输入释义后进行下拉刷新：

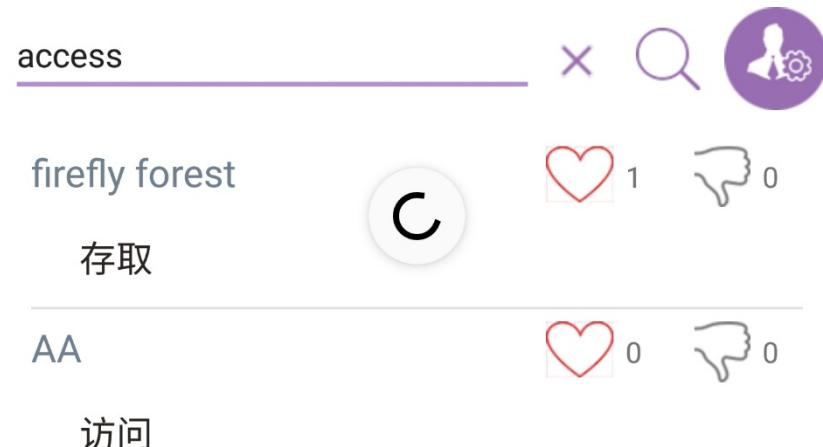


图 5-9 下拉刷新测试

(6) 修改密码测试以及结果

修改密码子模块主要包括输入框以及确认按钮、取消按钮的点击事件，测试用例如表 5-6 所示：

表 5-6 修改密码测试用例

子模块	前置条件	操作步骤	预期结果	实际结果
用户名输入	用户已经联网，密码正确输入	输入为空，点击确认按钮	弹出提示框，提示用户名输入为空	与预期一致
		输入不存在的用户名	弹出提示框，提示该用户名不存在	与预期一致
		正确输入用户名，点击确认按钮	用户成功修改密码，跳转至登录界面	与预期一致
密码输入	用户已经联网，用户名正确输入	输入为空，点击确认按钮	弹出提示框，提示密码输入为空	与预期一致
		密码长度小于 6 位，点击确认按钮	弹出提示框，提示密码长度小于 6 位	与预期一致
		密码和确认密码不一致，点击确认按钮	弹出提示框，提示两次密码输入不一致	与预期一致

		正 确 输入 密 码， 点击确认 按 钮	用 户 成 功 改 密 码， 跳 转 至 登 录 界 面	与 预 期 一 致
取消按钮	当 前 界 面 为 修 改 密 码 界 面	用 户 点 击 取 消 按 钮	跳 转 至 登 录 界 面	与 预 期 一 致
确认按钮	用 户 信 息 均 已 正 确 输 入	用 户 点 击 确 认 按 钮	提 示 用 户 成 功 改 密 码， 跳 转 至 登 录 界 面	与 预 期 一 致

根据测试用例来进行测试的一部分结果如下图：

图 5-10 为用户名正确输入，两次密码长度不一时的情况：

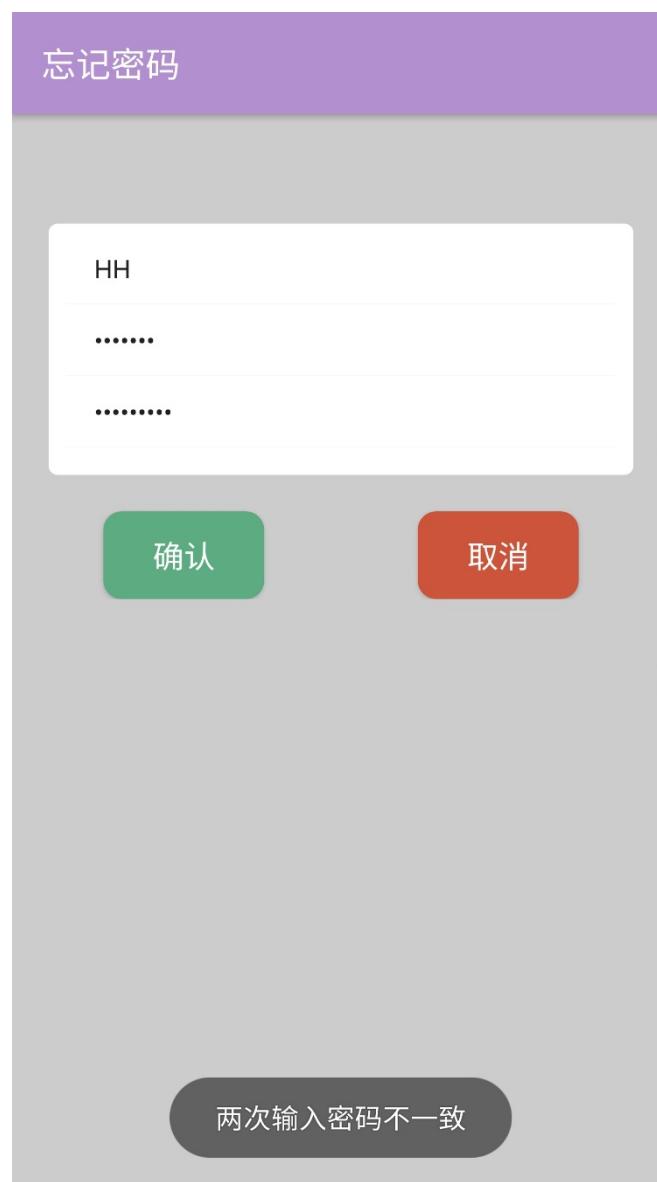


图 5-10 修改密码界面密码输入测试

图 5-11 为密码正确输入但是不存在该用户时的情况：

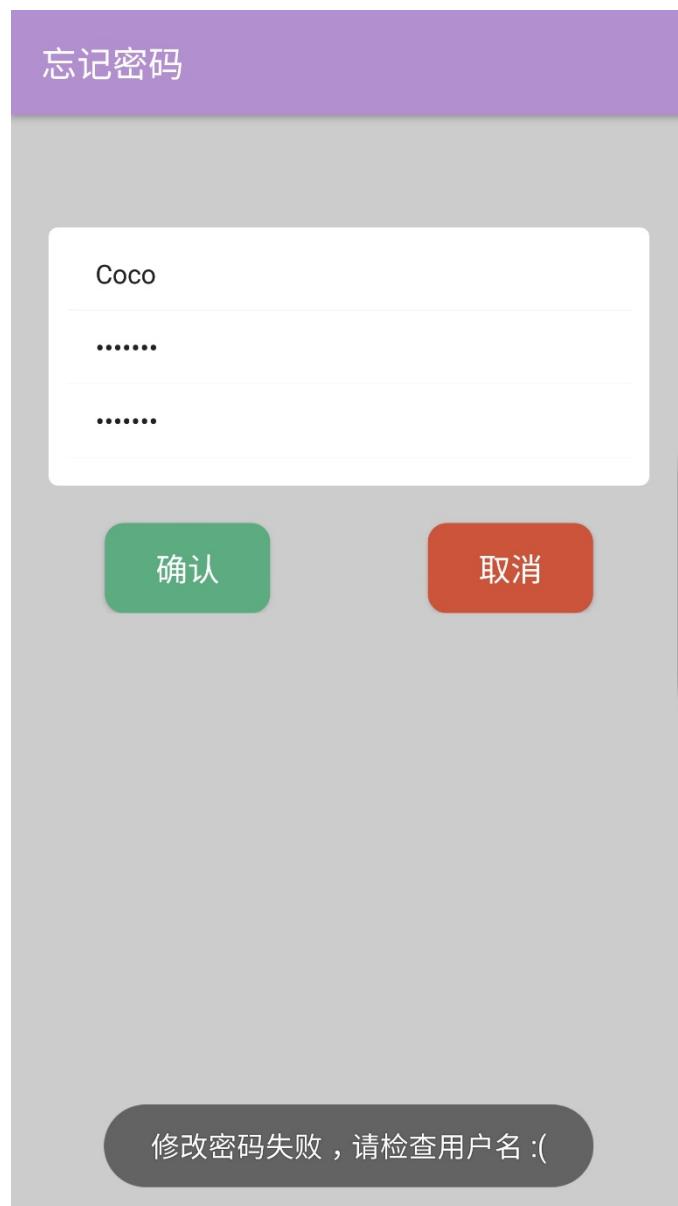


图 5-11 用户名检测测试

(7) 用户登出测试以及结果

用户登出子模块主要包括用户信息显示以及返回按钮、退出登录按钮的点击事件，测试用例如表 5-7 所示：

表 5-7 用户登出测试用例

子模块	前置条件	操作步骤	预期结果	实际结果
用户信息显示	当前为用户中心界面	查看样式	显示本机登录的用户名和用户等级	与预期一致
返回按钮	当前为用户中心界面	点击返回	回到查询界面	与预期一致

退出登录按钮	当前为用户中心界面	点击退出登录	跳转到登录界面	与预期一致
--------	-----------	--------	---------	-------

根据测试用例来进行测试的一部分结果如下图：

图 5-12 为检测用户信息是否能正确显示：



图 5-12 用户信息显示测试

5.3 系统性能测试

除了功能测试之外，为了体现出本系统与其他通用翻译软件相比的区别，还要对系统性能进行测试。在此测试过程中选取了 5 个单词作为参照，模拟已经有用户使用了一段时间之后的情况。与金山词霸的对比如图 5-13 至图 5-17，其中左图为本系统查询情况，右图为金山词霸查询情况：

The image shows a comparison of search results for the word 'browser' between two platforms. On the left, the 'kingsoft' search results are displayed, while on the right, the 'browser' search results from the Kingsoft Dictionary app are shown. Both results include the word's definition, pronunciation, and example sentences.

kingsoft	browser
浏览器（用于在互联网上查阅信息）	英 ['braʊzə(r)] ↗ 美 ['braʊzə] ↗
浏览程序	常用词典 牛津词典 NEW 柯林斯高阶
浏览图书报刊者	n. 浏览程序；浏览器（用于在互联网上查阅信息）；浏览图书报刊者；逛商店的人；
逛商店的人	

图 5-13 单词 browser 测试

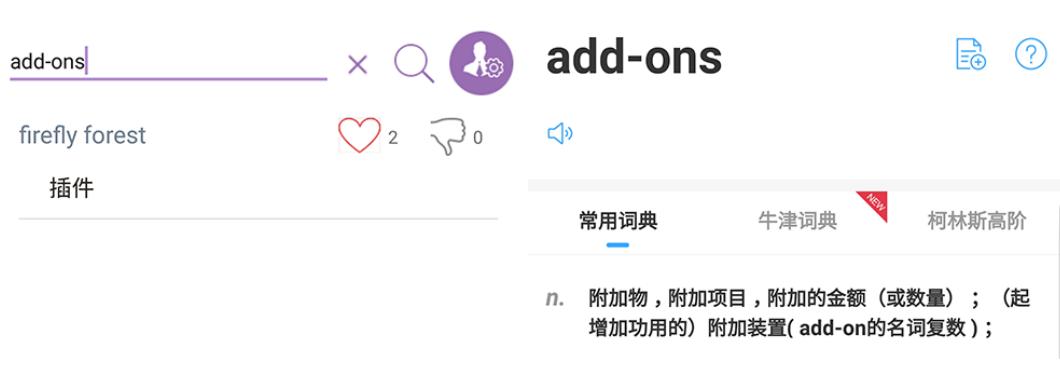


图 5-14 单词 add-ons 测试



图 5-15 单词 word 测试



图 5-16 单词 cables 测试



图 5-17 单词 clarity 测试

通过测试可知，和金山词霸相比，本系统对于专业领域词汇的释义更加具有针对性。通过用户投票，那些更加符合计算机专业领域的释义会具有更好的排名，这样可以使得用户们能够快捷、准确的得到精准的释义，减弱了用户们自己去进行筛选的环节。

5.4 本章小结

本章对该专业术语辅助学习系统的各个功能都严格按照测试用例进行了测试，测试包括功能测试和性能测试，结果良好，各功能正常运转。本系统总体上来说达到了最初的设计要求。

第六章 结束语

6.1 论文工作总结

针对现在国内以及国外各个翻译软件以及平台对于专业术语的翻译支持情况，本设计从用户需求、功能性需求和非功能性需求三个方面进行了详细阐述，然后对该系统的总体框架设计进行了详细讨论，最后对本系统的各个功能模块、系统数据库以及用户界面进行了详细设计。在进行分析和设计的过程中，我积累了很多经验，也发现了关于该系统的很多不足之处，还有待在后期逐步进行完善。

主要成果有：

- (1) 通过对计算机专业同学的访谈以及同类型软件的调研，明确了当前翻译类软件在专业术语的翻译上存在的问题，并从功能性需求和非功能性需求两个方面进行了详细的需求分析；
- (2) 根据需求分析，确定了系统的层次架构以及功能模块，并对功能模块进行了详细设计与实现；
- (3) 通过 E-R 图，先对本系统的数据库进行了概要设计，然后对数据库表进行了详细设计；
- (4) 系统功能一定程度上弥补了当前通用翻译软件存在的缺陷，系统测试结果良好。

6.2 问题与展望

尽管本系统完成了用户所需的专业术语查询要求，弥补了市面上通用翻译软件存在的问题，但是由于时间紧张，因此还有很多功能是需要后期不断完善的。目前该版本需要优化的功能有：

- (1) 当前只能够进行英译汉，后期可添加中译英；
- (2) 现在系统支持计算机专业术语的翻译，后期可以加入其余领域的专业术语；
- (3) 用户界面只显示用户名和用户等级信息，用户不能够查看自己在哪些单词下提交过释义，后期可以增加该查看功能；

以上三点是我觉得在下一个版本需要进行重点优化的功能，为了给用户提供更好的翻译体验。

参考文献

- [1] CNNIC 发布第 39 次《中国互联网络发展状况统计报告》[J].中国信息安全,2017(02):24.
- [2] 卿斯汉.Android 安全研究进展[J].软件学报,2016,27(01):45-71.
- [3] Mawston N. Strategy Analytics. Android shipped 1 billion smartphones worldwide in 2014. 2014. <http://www.strategyanalytics.com/%20default.aspx?mod=reportabstractviewer&a0=10539>
- [4] 张相明.21 世纪词典学研究的电子化发展新趋势——兼评《电子词典学》[J].辞书研究,2014(04):10-19+93.
- [5] 叶成林,徐福荫.移动学习及其理论基础[J].开放教育研究,2004(03):23-26.
- [6] 王郁瑶. 基于 Android 的翻译类 App 客户端软件的分析与设计[D].北京邮电大学,2017.
- [7] Ducrohet, Xavier; Norbye, Tor; Chou, Katherine (May 15, 2013). "Android Studio: An IDE built for Android". Android Developers Blog. Google. Retrieved May 16, 2013.
- [8] "Where did Eclipse come from?". Eclipse Wiki. Retrieved 16 March 2008.
- [9] 浅谈 C/S 和 B/S 架构. (2015, April 26). In CSDN. Retrieved May 30, 2018, from <https://blog.csdn.net/tennysonsksky/article/details/45062079>
- [10] Socket 详解. (2016, September 30). In CSDN. Retrieved June 7, 2018, from https://blog.csdn.net/qq_26399665/article/details/52421723
- [11] Socket 实现. (2017, June 27). In cnblogs. Retrieved June 7, 2018, from <https://www.cnblogs.com/kakawater/p/7085122.html>
- [12] MySQL. (2018, May 18). In Wikipedia, the free encyclopedia. Retrieved May 30, 2018, from <https://en.wikipedia.org/wiki/MySQL>
- [13] MySQL 体 系 结 构 . (2012, February 6). In cnblog. Retrieved May 30, 2018, from <http://www.cnblogs.com/yjfs12/archive/2012/02/06/2339496.html>
- [14] 王继成,高珍.软件需求分析的研究[J].计算机工程与设计,2002(08):18-21.
- [15] 赵硕.基于 SHA-1 的加密算法[J].齐齐哈尔大学学报(自然科学版),2014,30(03):53-56.
- [16] Dragos-Paul Pop,Adam Altar. Designing an MVC Model for Rapid Web Application Development[J]. Procedia Engineering,2014,69.
- [17] Wilson, E. B. (1927). "Probable inference, the law of succession, and statistical inference". *Journal of the American Statistical Association*. 22: 209–212. doi:10.1080/01621459.1927.10502953. JSTOR 2276774
- [18] 杜蓬,吕海鸥.从 ER 图到关系模型的数据依赖图法[J].天津师大学报(自然科学版),1990(02):54-58.
- [19] 用户界面设计 20 大设计原则与规范总结. (2015, January 7). In Mockplus. Retrieved May 30, 2018, from <http://www.mockplus.cn/blog/post/31>
- [20] 【 UI 设计 】 ——界面设计原则 . (2018, March 4). In CSDN. Retrieved May 30, 2018, from <https://blog.csdn.net/zt15732625878/article/details/79440469>
- [21] 软 件 界 面 设 计 原 则 . (2008, April 30). In CSDN. Retrieved May 30, 2018, from <https://blog.csdn.net/guohaif/article/details/2346249>
- [22] 淡海英.软件项目中系统测试概述[J].工业仪表与自动化装置,2014(06):15-16+42.
- [23] 聂长海.关于软件测试的几点思考[J].计算机科学,2011,38(02):1-3+27.

致 谢

经过两个月的工作，我的毕业设计也算是告一段落了。首先，我要感谢我的毕业设计导师张海旸老师。因为我之前从来没有接触过 Android 开发，刚拿到这个毕设题目时候可以说是一无所知，在此期间是张老师一直在引导我，为我指明学习的方向，在我遇到困难时给了我许多宝贵的建议和帮助。在两个月的开发过程中，我学习到了很多宝贵的经验，在此向张老师致以最崇高的敬意。

感谢我的朋友家人们，在我最为迷茫的时候，一如既往的支持我、理解我。这两个月来经历了太多的意料之外的事，是家人和朋友们的鼓励让我重新站起，重新规划未来。你们是我最强大的后盾。

感谢小美同学对我的帮助，陪我度过那段阴霾的日子。

感谢各位专家和老师们在百忙之中抽出空来评审本论文。