# 摘要：近年来，随着手机的日趋流行，手机软件也日益发展壮大。而在全球化的背景下，对于英语的学习也变得越来越重要。手机电子词典软件作为学习英语的工具，凭借其便捷、简单、高效的优势，日益受到年轻人的青睐。它通过使用联网数据库或者本地数据库来实现对单词、句子的查询，并且显示在图形用户界面上。本项目拟开发一款跨平台的电子词典软件，包括三部分:第一部分是词典，第二部分是翻译，第三部分是个人中心。词典部分包括四六级等关于英文的题库。该部分主要是用户根据自己的实际需要选取相应词典进行单词学习。翻译部分主要是调用百度API来实现中英文互译，实现生词加入收藏。个人中心部分则是存储用户的个人信息，以及用户生词本的维护。

# 目录

# 引言

## 项目介绍

## 课程设计选题《单词记忆APP》

## 项目的目的和意义

全球化渗入到我们生活的方方面面，英语在我们学习、生活和工作中扮演着重要的角色。手机词典软件的出现给人们对于英语的学习与使用带来了极大的便利，通过使用手机词典软件，我们可以轻松解决所遇到生疏的英语单词。随着手机的使用日益普遍，支持各种系统系统的手机应用软件也日益增多，电子词典软件就是其中很好的学习英语的工具。

而专注于某一种手机系统开发的APP普及型不强，且若分别针对每种手机系统进行开发，会浪费大量人力物力。因此有了跨平台APP开发框架的出现。[跨平台APP开发框架](http://www.kingwins.com.cn/content-6846.html" \t "https://www.kingwins.com.cn/_self)使开发人员更容易为多个平台构建APP，而无需花费大量精力在每个平台上编写相同功能的代码。这些框架的作用是，它们允许你写下一段代码，然后在许多其他平台上重用该代码。由于各个平台的某些部分之间存在很大差异，因此开发人员必须对它们进行编码。但是，使这些框架成为一个有利可图的选择的事实是它们允许用户重用大约70％的代码。而且，它可以极大地降低在金钱和时间方面开发成本。

Ionic是一个开源的移动应用程序开发框架，它可以轻松地使用web技术构建高质量的跨平台的移动应用。可以让我们快速开发移动App、移动端WEB页面、微信公众平台应用，混合app web页面。Ionic4目前已经发布了基于Angular的正式版本，对 React 和 Vue 的支持尚处于 alpha 状态，未来我们可以用自己喜欢的语言结合Ionic来开发移动端应用。Ionic4是Ionic有史以来最快、最小和最可扩展的版本。它的优异的性能则让 Ionic 成为了构建高性能 PWA 的最佳 UI 框架。 Ionic 团队花了一年的时间使用 Stencil 构建一个 Web 组件管道来生成 Ionic 组件,以达到 Google 规定的性能标准，使用新的异步组件加载和交互的方法。确保Ionic4有更快的运行速度。

本手机词典软件是就是基于ionic框架进行开发的，功能明确，内容准确，操作简单，适合大部分人群使用。在设计本软件之前，我们首先对手机软件市场做了充分的调研，明确了人们对于手机电子词典软件的功能需求，然后进行了详细的软件需求分析以及软件的整体结构设计，确保本软件的每一个功能对于用户来说都是有意义的。接着我们对软件进行了详细设计，将项目分成多个设计模块，完成各个设计模块的功能，最后将各个模块通过设计好的框架整合在一起。最后当整个项目完成后，对软件做性能测试，确保软件按照正确方式运行。

## 相关技术介绍

### ionic + angular + cordova混合框架

混合开发，是介于H5和原生之间的一种开发模式，既有原生的部分，也有H5的部分，结合了各自有优点，这种开发模式，既可以做到跨平台，也能上各大应用市场。

**1、cordova —— 应用的壳**

cordova 是Apache的开源框架，是由PhoneGap演化而来的一套核心代码。可以理解为Webkit和Google Chrome的关系。cordova 提供了一系列移动平台的壳，例如 iOS、Android、ubuntu phone os、Blackberry、Windows Phone、Palm WebOS、Bada和Symbian。主要用于嵌套H5，将H5打成应用市场认可的包，并且提供一套基于js的H5和各个移动平台通讯的API。比如获取手机文件、相机、手机信息、拨打电话等等基础功能。cordova 把这些和手机系统互相通讯的功能，封装成一个个插件，需要用到某一个功能，添加即可调用。如果cordova提供的插件不能满足你和手机的通讯， cordova也支持自定义插件，可以自己开发一个插件。

**2、ionic —— 应用的界面**

ionic 是一个强大的 HTML5 应用程序开发框架(HTML5 Hybrid Mobile App Framework )。 可以帮助您使用 Web 技术，比如 HTML、CSS 和 Javascript 构建接近原生体验的移动应用程序。ionic 主要关注外观和体验，以及和你的应用程序的 UI 交互，特别适合用于基于 Hybird 模式的 HTML5 移动应用程序开发。ionic是一个轻量的手机UI库，具有速度快，界面现代化、美观等特点。为了解决其他一些UI库在手机上运行缓慢的问题，它直接放弃了IOS6和Android4.1以下的版本支持，来获取更好的使用体验。

Ionic有如下特点：

1.ionic 基于Angular语法，简单易学。

2.ionic 是一个轻量级框架。

3.ionic 完美的融合下一代移动框架，支持 Angularjs 的特性， MVC ，代码易维护。

4.ionic 提供了漂亮的设计，通过 SASS 构建应用程序，它提供了很多 UI 组件来帮助开发者开发强大的应用。

5.ionic 专注原生，让你看不出混合应用和原生的区别

6.ionic 提供了强大的命令行工具。

7.ionic 性能优越，运行速度快。

**3、angular —— 应用核心**

angular一个套js库，类似jQuery 。主要用于，界面的数据绑定，和界面逻辑的处理。

**4、****NG-ZORRO-MOBILE移动端UI组件库**

Ant Design是蚂蚁金服体验技术部开发出一个服务于企业级产品的设计组件库，来帮助业务快速设计出高质量的产品原型。基于『确定』和『自然』的设计价值观，通过模块化的解决方案，降低冗余的生产成本，让设计者专注于更好的用户体验。而NG-ZORRO-MOBILE是Ant Design基于angular的兼容ionic的移动端版本。

## 后端SSH框架

SSH框架是Struts、Spring、Hibernate的一个合成框架，使用Struts作为系统的整体基础架构，负责MVC的分离，在Struts框架的模型部分，控制业务跳转，利用Hibernate框架对持久层提供支持，Spring做支持，支持struts和hibernate。具体做法是：用面向对象的分析方法根据需求提出一些模型，将这些模型实现为基本的Java对象，然后编写基本的DAO(Data Access Objects)接口，并给出Hibernate的DAO实现，采用Hibernate架构实现的DAO类来实现Java类与数据库之间的转换和访问，最后由Spring做支持，支持struts和hibernate。其实ssh框架最主要的本质是：“高内聚、低耦合”。

**1、Struts2**

JSP、Servlet、JavaBean技术的出现给我们构建强大的企业应用系统提供了可能。但用这些技术构建的系统非常的繁乱，所以在此之上，我们需要一个规则、一个把这些技术组织起来的规则，这就是框架，Struts便应运而生。

Struts2工作流程

第一步：客户端初始化一个指向Servlet容器（例如Tomcat）的请求

第二步：这个请求经过一系列的过滤器（Filter）（这些过滤器中有一个叫做ActionContextCleanUp的可选过滤器，这个过滤器对于Struts2和其他框架的集成很有帮助，例如：SiteMesh Plugin）

第三步：接着FilterDispatcher被调用，FilterDispatcher询问ActionMapper来决定这个请是否需要调用某个Action

第四步：如果ActionMapper决定需要调用某个Action，FilterDispatcher把请求的处理交给ActionProxy

第五步：ActionProxy通过Configuration Manager询问框架的配置文件，找到需要调用的Action类

第六步：ActionProxy创建一个ActionInvocation的实例。

第七步：ActionInvocation实例使用命名模式来调用，在调用Action的过程前后，涉及到相关拦截器（Intercepter）的调用。

第八步：一旦Action执行完毕，ActionInvocation负责根据struts.xml中的配置找到对应的返回结果。

**2、Hibernate**

Hibernate是一个开源的对象关系映射框架，对JDBC进行了轻量级的对象封装；

hibernate的工作原理

第一步：Configuration.config() 来读取xml配置文件

第二步：Configuration.config()来读取配置文件里面的映射信息

第三步：创建会话工厂

第四步：打开会话

第五步：开启事务

第六步：持久化到数据库

第七步：关闭会话

第八步：关闭会话工厂

**3、spring**

spring提供了两个核心功能，一个是IoC(控制反转)，另外一个便是Aop(面向切面编程)，

AOP 让开发人员可以创建非行为性的关注点，称为横切关注点，并将它们插入到应用程序代码中。使用 AOP 后，公共服务 （比如日志、持久性、事务等）就可以分解成方面并应用到域对象上，同时不会增加域对象的对象模型的复杂性。

IOC 允许创建一个可以构造对象的应用环境，然后向这些对象传递它们的协作对象。正如单词 倒置 所表明的，IOC 就像反 过来的 JNDI。没有使用一堆抽象工厂、服务定位器、单元素（singleton）和直接构造（straight construction），每一个对象都是用其协作对象构造的。因此是由容器管理协作对象（collaborator）。

# 系统需求分析

## 软件功能

### 需求分析

随着科学技术的发展，人类社会日新月异，智能手机自出现以来就在人类的生产生活中饰演着重要的角色，越来越成为大家日常生活中必不可少的生活工具。在学校、在家中、在交通工具上甚至在工作中随处可见智能手机的身影。与此同时手机软件也日益发展壮大。而在全球化的背景下，对于英语的学习也变得越来越重要。手机电子词典或生词记忆软件作为学习英语的工具，凭借其便捷、简单、高效的优势，日益受到年轻人的青睐。

本项目APP主要针对用户有需要英语四六级考试、考研、出国留学考雅思、托福等在校大学生。还有就是初高中需要背英语单词的学生，以及想要提升自己词汇量的上班白领。这些用户的特征有（1）学生群体，受教育程度高，主动学习能力强；（2）好奇心强，对好玩有趣易用的新鲜事物感兴趣；（3）备考需要，想要提升自己的考试能力。经分析，这些用户的需求有（1）提高词汇量，满足考试、生活及工作需要；（2）充分利用碎片化时间，更有效率地学习；（3）需要有一套合理、科学的背单词学习计划，日积月累，成长看得见；（4）需要更有趣味地记单词，拒绝传统式枯燥、而且容易忘的记单词方式；（5）在记单词的时候，也想要提高自己的听力。

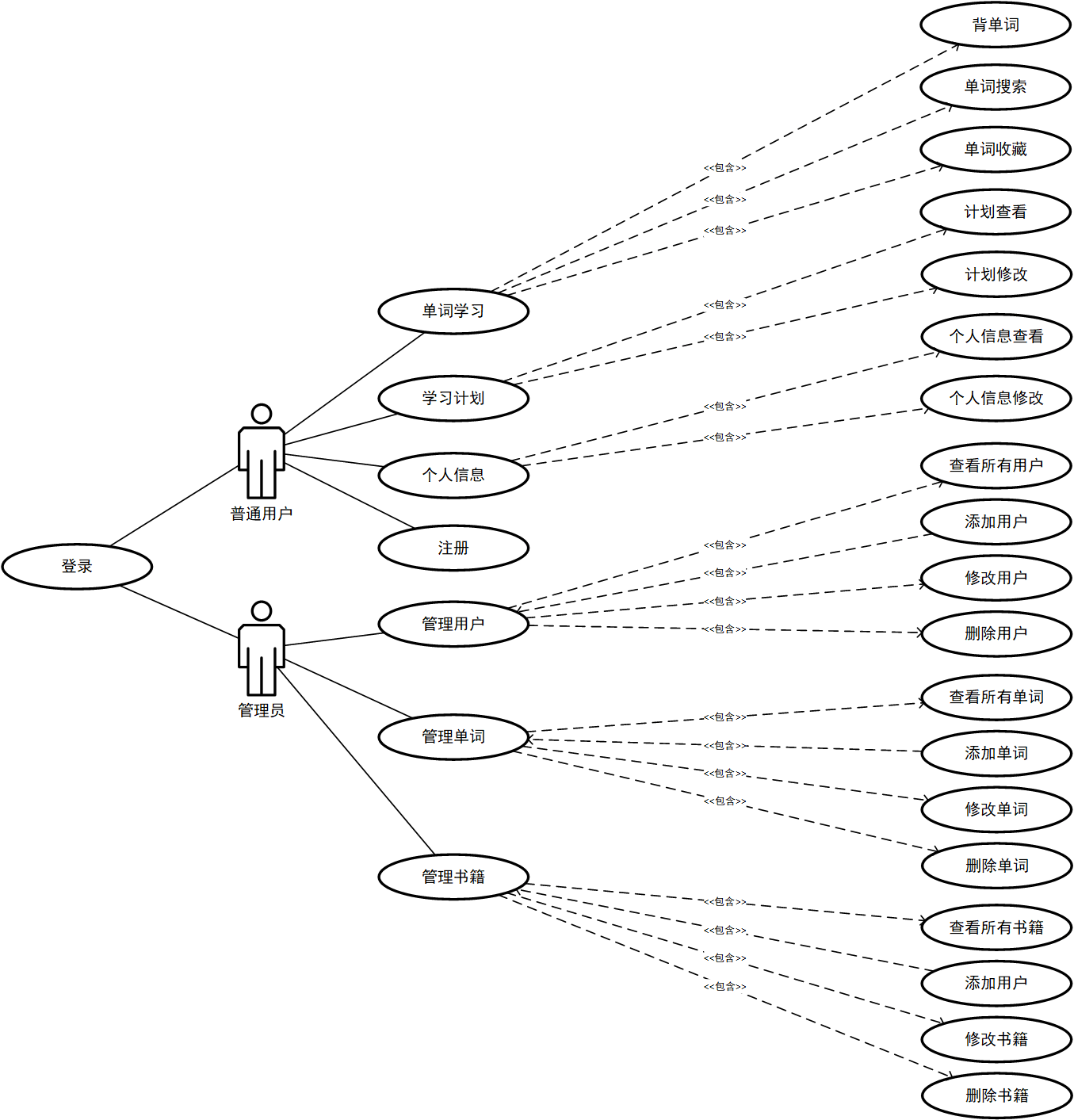
为满足目标用户的需求，基本可以去确定本电子词典软件设计所要完成的功能目标。以下是本项目中电子词典具备的功能。

1. 登录界面，用户进行登录，保障用户信息安全；
2. 注册界面，供新用户进行注册；
3. 主菜单界面，供用户选择要执行的操作界面；
4. 单词列表界面，供用户查看当前词典所有单词以及学习中收藏的单词
5. 计划更改界面，供用户选择词典以及制定每日所学习的单词数
6. 单词学习界面，供用户进行单词记忆主要功能模块；
7. 翻译界面，供用户进行生词的英汉互译；
8. 翻译收藏界面，供用户收藏自己查询的易混单词；
9. 单词本界面，供用户对学习问题及生词进行标注；
10. 个人中心页面，供用户修改个人信息及头像；

在设计软件之前，我们首先要对要解决的问题进行详细的分析，弄清楚问题的要求，包括需要输入什么，要得到什么结果，最后应输出什么。使用电子词典软件的最大群体是学生，他们需要准确性很强的单词解释，而对操作界面没有太高的要求。因而，我们最重要的就是增强电子词典软件的准确性。我们使用有道API、百度API这样准确性很强的电子词典接口，在词典中加入四六级、雅思托福等实用性强的词库。软件设计的布局要清晰，功能不宜太复杂，只有这样才能使不参与本软件编写的人熟练的使用。

## 功能性需求

根据项目所要实现的目标，我们以用例图的方式来描述系统的功能需求。如图xxx



# 项目介绍

## 系统的开发环境

操作系统：Windows10

开发工具：webstorm、IntelliJ IDEA

前端框架：ionic框架

后端框架：structs2+spring+hibernate（SSH框架）

后端服务器：MySql 5.7

后台服务器：tomcat 9.x

## 系统的技术路线

本系统前端使用ionic开发框架，完成界面的设计和方法的实现。后端使用SSH框架，用面向对象的分析方法根据需求提出一些模型，将这些模型实现为基本的Java对象，然后编写基本的DAO接口，采用Hibernate架构实现的DAO类来实现Java类与数据库之间的转换和访问，最后由Spring做支持，支持struts和hibernate。

## 系统的具体功能

### 登录注册模块

该模块主要解决系统中的用户登录及个人信息问题。在本系统中，用户在进行注册时，需要填写邮箱、密码、昵称和手机号码等几个重要信息。用户注册成功之后，在数据库中存入用户的信息。需要注意的是用户注册时填写的手机号码需要进行验证，填写验证码。为保证用户在注册时输入密码的正确性，要设置密码的重复输入对比。用户登录之后，就可以正常使用系统中的所有功能。

### 单词学习模块

本模块为系统的主要功能模块，用于解决用户的单词记忆需求。在该模块中，用户可选择自己需要的词典，例如四级词汇、六级词汇、雅思词汇等。用户选择词典后，可根据实际情况，给自己制定每天的学习计划。若初次登陆未进行计划即为默认计划（学习四级词汇，每天学习25个单词）。计划制定完成后即可开始对本词典内容的学习。在开始学习之前，首先会询问用户是否认识该单词，对于用户熟知的单词，会适当减少复习次数，反之增加复习次数。在询问完所有今日计划的单词后，便会开始复习。对于每个复习的单词，会提供三个错误选项以及一个正确选项，正确选项即为复习单词正确的中文释义。若用户在复习中选对了单词，则该单词的复习次数会减1，当减到0时该单词复习结束；反之若用户选错了释义，该单词的复习次数会加2，并且显示正确的中文释义。在学习过程中，若遇到反复错误的单词，可以点击右上角的星星按钮对单词进行收藏，收藏后的单词可以在单词列表中快速查看。

### 单词列表模块

本模块用于显示当前用户所学习的词典的所有单词，供用户快速查看所有的单词以及在学习中收藏的单词。

### 单词翻译模块

本模块主要解决用户的生词查询需求。当用户需要对不熟悉的英语单词进行查询或是想要查询汉语词语对应的英文单词时，可在搜索框中进行查询。此时，会返回查询单词或句子对应的中英文翻译。用户还可对翻译进行加入收藏操作。

### 翻译收藏模块

本模块主要为单词翻译模块进行服务。用户可把查询到的生词进行加入收藏的操作，以供用户进行后续的复习查看操作，也避免出现用户对同一词语进行反复查询的情况。

### 阅读模块

本模块显示在翻译界面当中，主要为用户提供一些值得学习的英文文章，目的是为了让用户通过阅读扩大视野，提高英语阅读能力。

### 个人中心模块

本模块主要是让用户可以对在本系统中存储的的个人信息进行管理。用户可在本模块中进行个人信息修改，修改的内容主要包括用户昵称、密码、头像。如果用户需要修改密码，那么修改之后，用户需要使用修改后的新密码重新进行登录。如果修改的是用户昵称或头像，保存修改返回个人中心之后，在个人中心显示的昵称或头像已经修改。

### 今日计划版块

今日计划部分设计为拥有一个记事本的基本功能，意在让用户可以在系统内进行学习规划、学习状态记录以及学习备忘。在本计划模块，用户可以实现对计划或备忘的增加、修改、删除和查看等基本功能。通过本版块，用户可以有针对性的实现系统中学习情况记录，方便用户进行学习。

### 单词进度模块

用户可在个人中心页面查看自己的学习进度。查看整体学习进度后，用户可衡量自己是否需要进行学习计划的调整，并可选择继续学习。

# 系统设计

## 总体设计

### 处理流程

根据需求分析设计子系统如下：

（1） 用户接口子系统

（2） 背单词子系统

（3） 搜索单词子系统

（4） 修改信息子系统

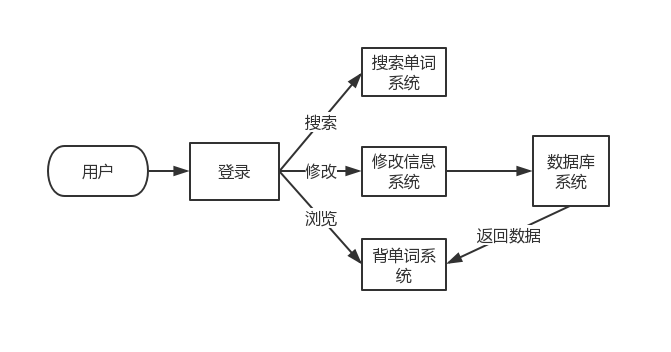


图2.1 子系统之间的关系图

### 总体结构和模块外部设计

与外部模块主要体现在，通过有道词典的接口对单词检索上。

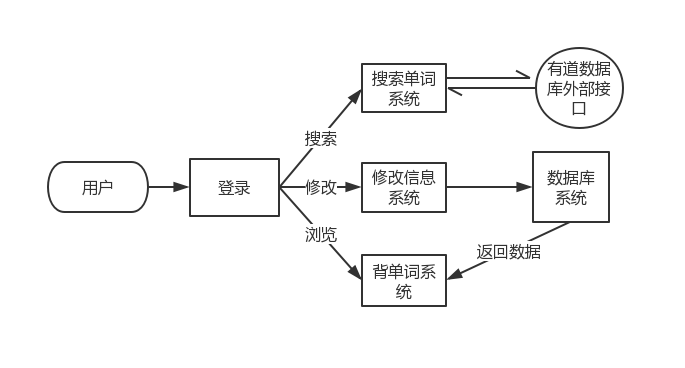


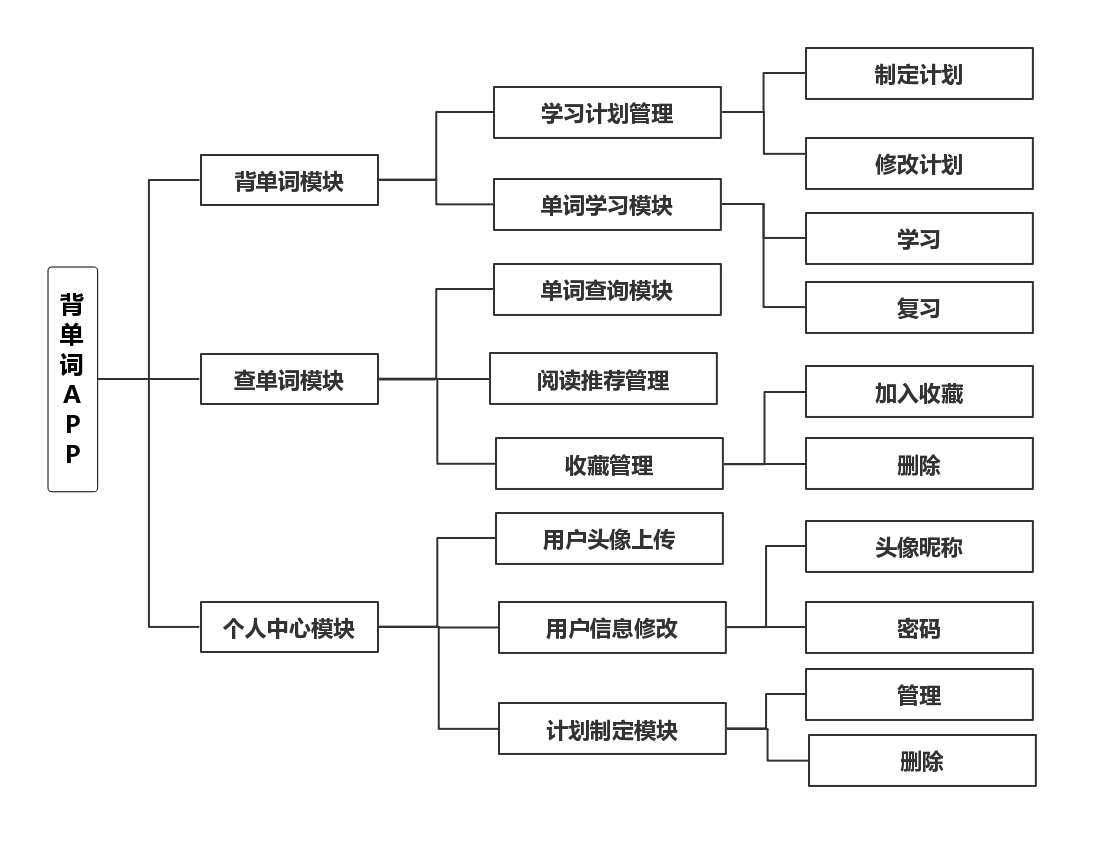
图2.2 总体结构图

### 系统类关系设计



## 系统功能模块设计

此系统为基于ionic、cordova和Angular开发的一款手机APP，主要针对有英语学习需求的年轻用户，使系统用户可以为自己定制学习计划，有目的的利用碎片化时间学习。本系统中的功能主要包括用户登录、用户注册、英汉互译、推荐阅读、个人中心、计划制定、单词背诵、单词复习等功能。如图xxx所示，下面将对每个功能进行详细说明介绍。



### 用户登陆修改信息模块

用户注册登录模块是用户完整使用系统中所有功能的前提。只有在用户注册并且登录之后，系统才可以正常记录用户的学习情况。因此，只有用户登录之后，才可以正常使用词典与个人中心模块。否则，用户只能使用翻译模块。用户登陆系统后，在个人信息中心中对个人信息修改并提交给后端服务器，从而实现对用户信息的修改。

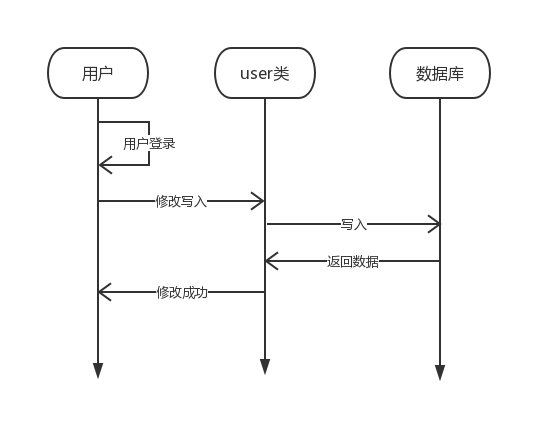


图2.3 用户登陆与信息修改时序图

### 单词学习模块

单词模块，是用户进行单词背诵的主要模块。用户可根据目前自己的需要或学习进度选择自己需要的词典，然后给自己制定学学习计划。例如每天学习30个单词等。接下来用户就可以进行学习，结束后还可以进行复习。用户登陆系统后，调用Word类，可以提取数据库中的单词数据，从而将获取的数据展示到前端页面上。

单词模块包括单词学习与单词复习功能。用户在第一次登录后，可以对自己所需要学习的单词书（如四级词汇等）进行选择、以及每天所学习的单词数量进行计划（如每天学习25个），否则将会按照默认计划（学习四级词汇，每天25个单词）进行学习。若用户在学习过程中遇到某些需要特殊记忆的单词，可以将其加入收藏中，以便多次查看复习。

单词学习的过程包括单词初步学习与重复复习过程，如图与图所示。开始时，会创建一个单词复习列表，在单词初步学习过程中，首先会询问用户是否认识该单词，对于用户熟知的单词，会适当减少复习次数，反之增加两次复习次数。在询问完所有今日计划的单词后，便会开始重复复习。对于每个复习的单词，会提供三个错误选项以及一个正确选项，正确选项即为复习单词正确的中文释义。若用户在复习中选对了单词，则该单词的复习次数会减1，当减到0时该单词复习结束，该单词便从需要复习的单词列表中删去；反之若用户选错了释义，该单词的复习次数会加2，并且显示正确的中文释义，然后从单词复习列表中随机选取下一个复习的单词，直到单词复习列表为空。



图 单词初步学习流程图



图 单词重复复习过程

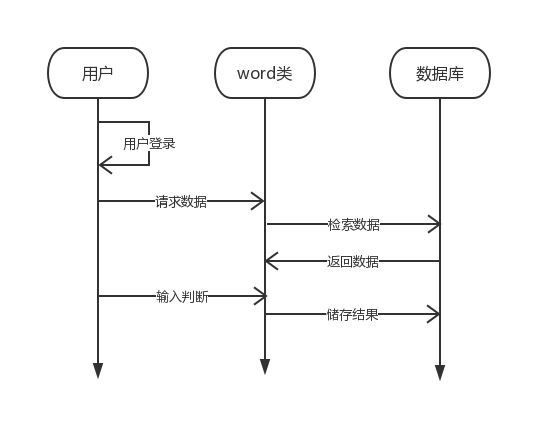


图2.4 背单词时序图

### 搜索模块

搜索模块能够实现短语或句子的英汉互译。本模块为用户进行单词记忆中遇到的困难提供帮助，返回用户想要查询的结果。用户登陆系统后，进入搜索功能，系统会调用“百度词典”的接口，从而实现对单词的搜索。

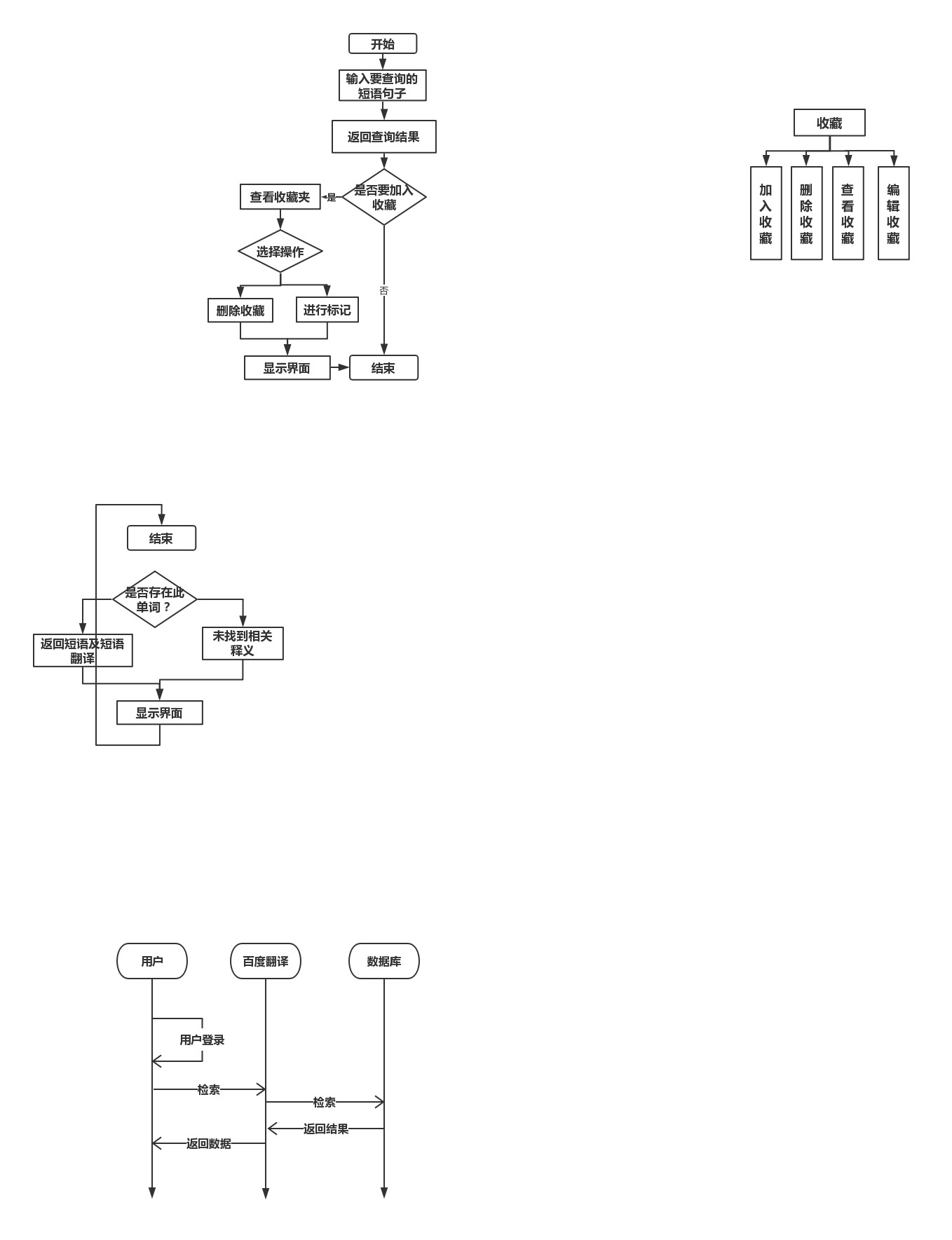


图2.5 搜索模块时序图

### 查看词典

用户登陆系统后，前端将向后端请求词典数据，并将后端获取的词典数据，显示在前端页面上。

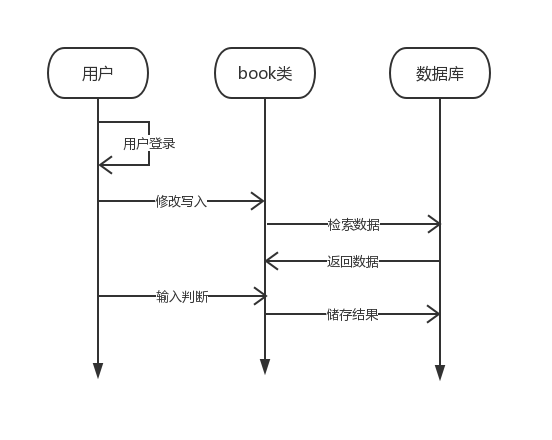
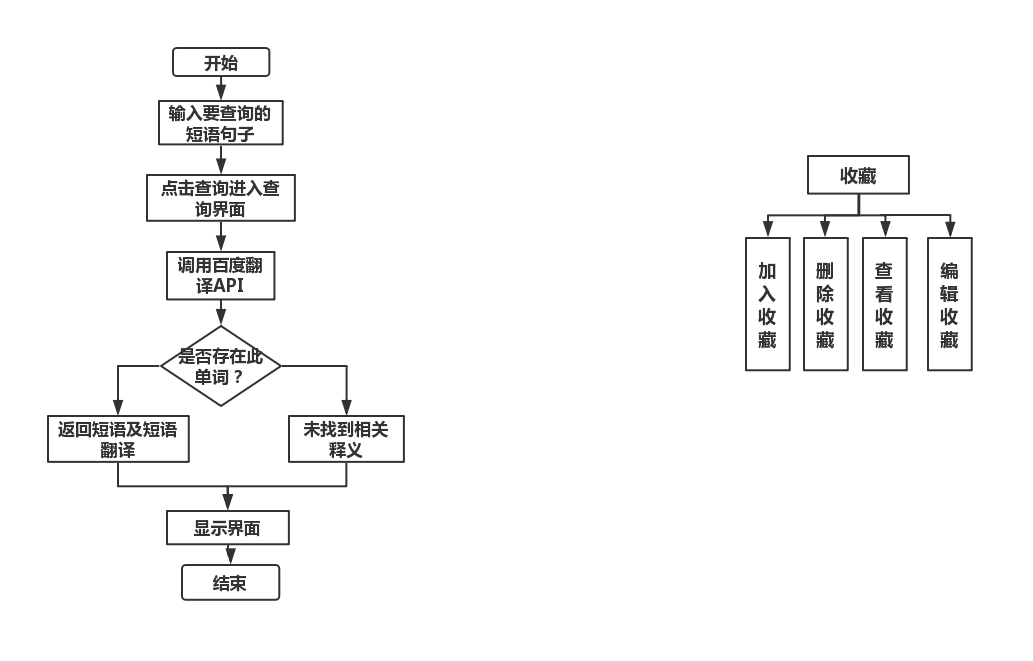


图2.6 查看单词书时序图

### 收藏模块

收藏模块，本模块主要为单词搜索模块进行服务。用户在搜索易错词或者容易错的单词之后，为避免一直重复查询，用户可将其加入到收藏里面，方便下次查看。该模块的功能结构如图：



下面对其功能进行说明：

1. 加入收藏

用户在进行中英互译查询之后，点击翻译结果右边的星号，可选择将此单词加入收藏。加入收藏之后的单词或短语可在用户收藏夹中查看。

1. 删除收藏

删除收藏操作是在用户收藏夹中进行操作的。用户可对收藏夹中自己已经熟悉的单词短语进行移出收藏夹操作。

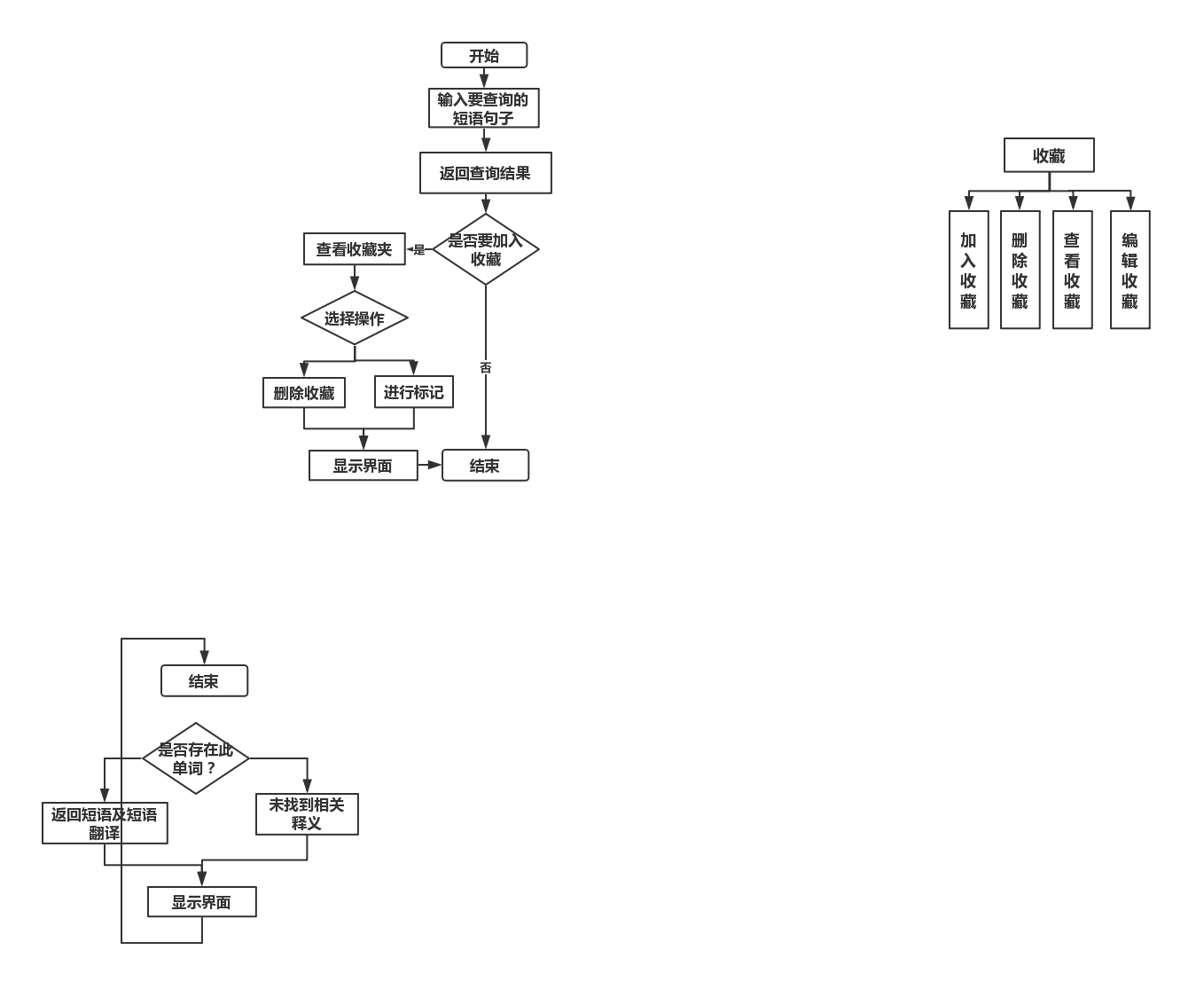
1. 查看收藏

用户所有加入收藏的单词短语都会保存在用户收藏夹中，用户可在收藏夹中查看自己已经收藏过的单词短语。

1. 编辑收藏

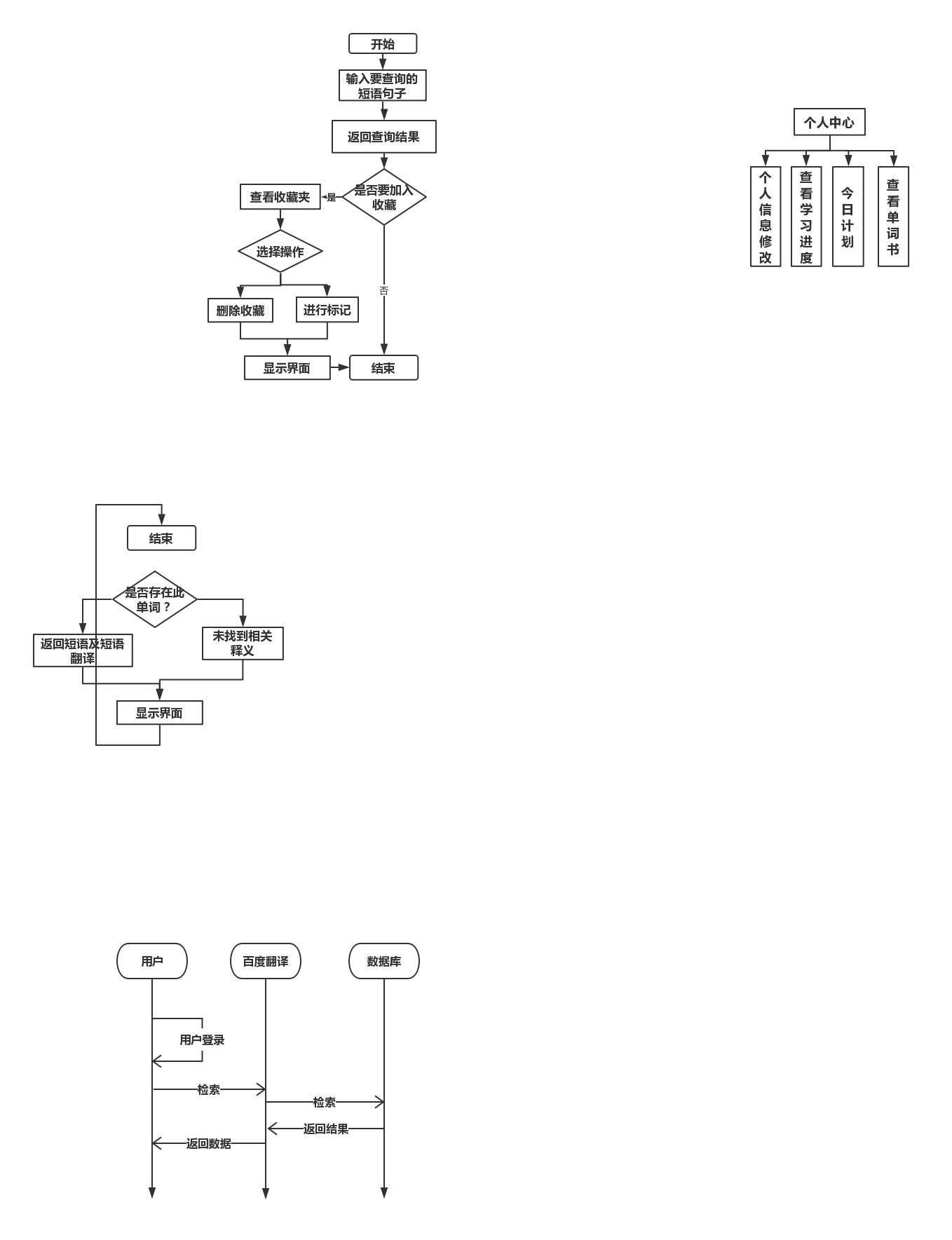
用户可对收藏夹中的单词短语进行编辑，为其加上标记以区分重要程度或用户的熟悉程度。

其处理流程图，如图xxx所示：



### 个人中心模块

个人中心模块，用来记录或设置用户个人在系统中的信息。在该模块中，可以实现用户头像上传，用户个人信息修改，查看正在学习的词典，查看自己的学习进度，以及制定自己的学习计划，记录自己学习中的问题。该模块的功能结构如图xxx：



图xxx

下面对其功能进行说明：

1. 个人信息修改

用户在个人中心模块可以对个人信息进行修改，可以修改的内容包括用户在系统中的昵称、密码以及用户的头像。用户头像修改部分，可以上传本地头像，也可以拍照上传。密码修改后，用户需要使用新密码重新登录系统。

1. 查看学习进度

用户的学习完成情况会在系统中进行记录，用户可在本模块中查看用户关于某本词典的学习进度。

1. 今日计划模块

用户可添加、修改、删除学习计划，时刻记录自己的学习问题。

1. 查看单词书

用户可在本模块查看自己开始学习或已经学完的单词书，查看自己的学习进行情况。

其处理流程图如图xxx：

### 今日计划模块

今日计划模块，本模块提供给用户一个制定本系统内学习计划、进行备忘记录、易错易混单词记录的空间。用户可将在本系统内学习时的想法及遇到的问题进行单独记录，方便用户下次打开系统进行学习时查看。该模块的功能结构如下：

### 管理员管理模块

管理员通过另外的页面进行登录，登录后管理员便可以通过页面对用户、单词、词典进行管理（包括增删改查操作），下图是管理员调用book类，对用户信息进行管理。

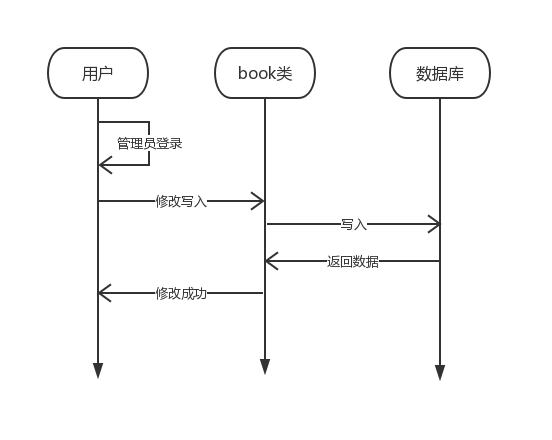


图2.7 管理员对单词本进行管理时序图

## 数据库设计

### 逻辑结构设计

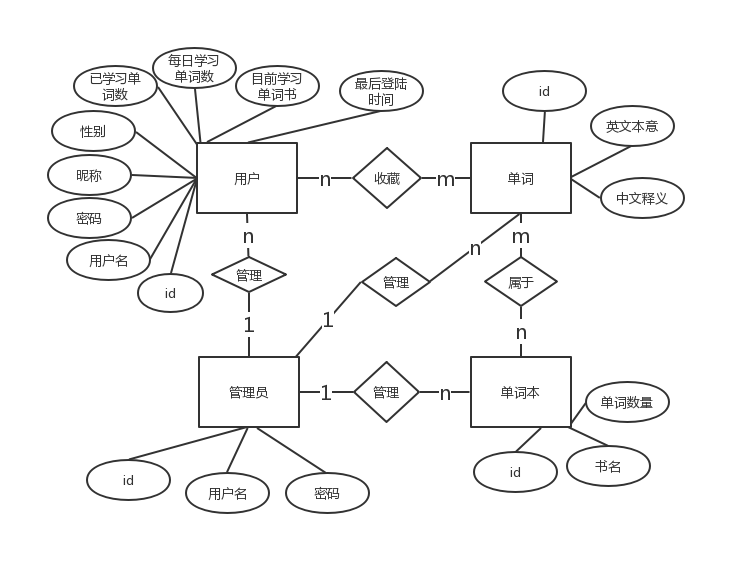


图2.8 数据库E-R图

### 物理结构设计

User表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 是否可空 | 说明 |
| U-id | integer | 否 | 用户序号 |
| 用户名 | string | 否 | 用于登录的用户名 |
| 密码 | string | 否 | 用于登录的密码 |
| 昵称 | string | 是 | 昵称 |
| 性别 | string | 是 | 性别 |
| 已学习单词数 | int | 是 | 该用户已学习单词数 |
| 目前学习单词书 | string | 是 | 该用户目前学习单词书 |
| 每天学习单词数 | int | 是 | 该用户每天学习单词数 |
| 最后登录时间 | timestamp | 是 | 最后登录时间 |

book表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 是否可空 | 说明 |
| B-id | integer | 否 | 单词本序号 |
| 书名 | string | 否 | 单词本名称 |
| 单词数量 | intg | 否 | 单词本含有的单词数 |

Word表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 是否可空 | 说明 |
| W-id | integer | 否 | 单词序号 |
| 英文本意 | string | 否 | 单词本身 |
| 中文释义 | string | 否 | 单词的中文释义 |

administrator表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 是否可空 | 说明 |
| A-id | integer | 否 | 管理员序号 |
| 用户名 | string | 否 | 用于登录的用户名 |
| 密码 | string | 否 | 用于登录的密码 |

用户收藏单词信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 是否可空 | 说明 |
| id | integer | 否 | 该记录序号 |
| U-id | integer | 否 | 用户序号 |
| W-id | integer | 否 | 单词序号 |
| 收藏时间 | timestamp | 是 | 用户收藏单词的时间 |

单词本具体信息表（记录单词本包含的单词）

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 是否可空 | 说明 |
| id | integer | 否 | 该记录序号 |
| B-id | integer | 否 | 单词本序号 |
| W-id | integer | 否 | 单词序号 |
| 备注 | string | 是 | 其他信息 |

管理员对用户管理记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 是否可空 | 说明 |
| id | integer | 否 | 该记录序号 |
| A-id | integer | 否 | 管理员序号 |
| U-id | integer | 否 | 用户序号 |
| 管理信息 | string | 否 | 记录管理员对用户信息的具体操作 |
| 操作时间 | timestamp | 是 | 记录该操作发生的时间 |

管理员对单词管理记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 是否可空 | 说明 |
| id | integer | 否 | 该记录序号 |
| A-id | integer | 否 | 管理员序号 |
| W-id | integer | 否 | 单词序号 |
| 管理信息 | string | 否 | 记录管理员对单词信息的具体操作 |
| 操作时间 | timestamp | 是 | 记录该操作发生的时间 |

管理员对用户管理记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 是否可空 | 说明 |
| id | integer | 否 | 该记录序号 |
| A-id | integer | 否 | 管理员序号 |
| B-id | integer | 否 | 单词本序号 |
| 管理信息 | string | 否 | 记录管理员对单词本信息的具体操作 |
| 操作时间 | timestamp | 是 | 记录该操作发生的时间 |

## 后端服务器API规约

后端服务器采用基于Java的SSH框架构建，并使用RESTful接口风格，使得接口风格统一方便调用，且RESTful风格的接口是一种无状态的接口，在调用接口时不需要考虑上下文、不用考虑当前状态，极大的降低了复杂度。对于每个相关的类，都提供了一套接口供前端编程者使用。如下接口中的数据交互均为整理好了的 JSON数据，方便直接调用。

### 用户相关API



### 单词相关API



### 词典相关API



### 收藏相关API



### 管理员相关API



# 各模块具体功能实现

对于前后端的交互，交由一个data.service的服务类来管理。Service是

angular中一个特殊的类，它会在应用的整个生命周期中始终在内存里，所以用service来存储类可以方便不同的类之间获取从后端获取的各种数据，如用户数据、单词数据、词典数据等。

## 用户登陆实现

用户在登录页面输入用户名和密码提交后，前端将用户输入的用户名与密码以JSON文件格式发送给后端，后端服务器对接收到的账号密码进行处理，若账号密码正确后端返回状态值为200的正确信息以及登录者的ID，前端再通过ID向后端服务器请求登陆者的完整信息；若账号密码错误，则返回状态值为500的错误信息，拒绝访问，前端提示用户账号密码错误，重新输入账号密码。

### 向后端服务器提交用户登录信息算法

loginCheck(form: NgForm) {

if (this.valid) {

let request = {

"username": form.value.username,

"password": form.value.password

};

console.log(request);

this.http.post("http://127.0.0.1:8088/untitled2\_war\_exploded/user", request,{ observe: 'response' })

.subscribe(

(data) =>{

console.log(data);

this.userService.userLogin(data.status - 200);

},

(error: any) =>{

console.log(error);

this.errortext = error.error.description;

this.presentToast();

}

);

}

}

### 用户登录成功向后端服务器获取相关信息算法

/\*\*

\* 用于用户在登录时获取用户的信息

\*

\* @param id

\*/

userLogin(id: number){

this.http.get(this.serviceUrl + 'user/'+ id + '.json')

.subscribe((data: any)=> {

console.log(data);

this.currUser = new User(data.id, data.username, data.password, data.nickname, data.lastlogin, data.dailyWords, data.gender, data.studied, data.curbook);

console.log(this.currUser);

this.getAllWords();

this.getBooks();

this.router.navigateByUrl("user");

}

)

};

## 用户注册实现

### 向后端服务器提交用户注册信息算法

register(){

console.log("submit");

let gender;

if (this.isValid()) {

let request = {

"username": this.username,

"password": this.password,

"nickname": this.nickname,

"gender": gender = this.gender[0]

};

console.log(request);

this.http.post("http://127.0.0.1:8088/untitled2\_war\_exploded/user", request)

.subscribe(

(data : any) =>{

console.log(data);

// this.router.navigateByUrl("user");

},

(error: any) =>{

console.log(error);

this.errortext = error.error.description;

this.presentToast();

}

);

}else {

const toast = ToastService.fail('请填写完毕再提交', 1000);

return

}

}

## 用户信息修改实现

### 向后端服务器提交用户更改的信息算法

submit() {  
 if (this.oldPassword == this.dataService.currUser.password) {  
 let gender;  
 let request = {  
 "id": this.dataService.currUser.id,  
 "studied": this.dataService.currUser.studied,  
 "dailyWords": this.dataService.currUser.wNumPerD,  
 "curbook": this.dataService.currUser.curbook,  
 "username": this.dataService.currUser.username,  
 "password": this.newPassword,  
 "nickname": this.nickname,  
 "gender": gender = this.gender[0]  
 };  
 console.log(request);  
 this.http.post('http://127.0.0.1:8088/untitled2\_war\_exploded/user', request, {observe: 'response'}).subscribe(  
 (data) => {  
 console.log(data);  
 }  
 )  
 }else {  
 this.failToast();  
 }  
}

## 单词学习流程实现

单词学习算法共有两个步骤：初步学习与单词复习。在初步学习页面，会将所学的单词生成一个单词列表，然后依次显示所学单词的英文，并询问用户是否认识这个单词，如果不认识，将会在复习时增加两次复习的次数；如果认识，则为默认次数（两次），直到询问完所有的应学单词为止。

进入复习阶段，所有的应学单词将会依次显示，并且同时显示藏在三个错误中文释义中的正确中文释义，如果用户选对了该单词的中文释义，将会减少一次复习次数，当复习次数减为0时，该单词将会从单词列表中删去。该单词复习完后将选择下一个单词为复习单词，并循环直到单词列表为空则整个学习过程结束。

### 初步学习页面

<Navbar [icon]="icon"

[mode]="'light'"

(onLeftClick)="onLeftClick()"

>

{{preWord.en}} {{preWord.zh}}

</Navbar>

<ng-template #icon>

<Icon [type]="'left'"></Icon>

</ng-template>

<ion-content>

<WhiteSpace [size]="'xl'"></WhiteSpace><WhiteSpace [size]="'xl'"></WhiteSpace><WhiteSpace [size]="'xl'"></WhiteSpace><WhiteSpace [size]="'xl'"></WhiteSpace>

<ion-grid style="height: 90%">

<ion-row style="height: 30%"><ion-col size="12" style="text-align: center; ">

<h1>{{curWord.en}}</h1></ion-col>

<ion-row [hidden]="!showZH"><ion-col size="12" style="text-align: center;"><h3 style="text-align: center; ">{{curWord.zh}}</h3></ion-col> </ion-row>

</ion-row>

<WhiteSpace [size]="'xl'"></WhiteSpace>

<ion-row [hidden]="showZH"><ion-col size="12"><div Button [type]="'ghost'" [inline]="true" (click)="isKnow(true)" style="width: 100%">认 识</div></ion-col></ion-row>

<WhiteSpace ></WhiteSpace>

<ion-row [hidden]="showZH"><ion-col size="12"><div Button [type]="'ghost'" [inline]="true" (click)="isKnow(false)" style="width: 100%">不认识</div></ion-col></ion-row>

<ion-row [hidden]="!showZH"><div Button [type]="'ghost'" [inline]="true" (click)="next()" style="width: 100%">下一个</div></ion-row>

</ion-grid>

</ion-content>

### 初步学习算法

/\*\*

\* 根据用户是否知道这个单词来确定复习的次数，并跳转到下一个单词

\*

\* @param isknow

\*/

isKnow(isknow: boolean){

if(this.studied == 24) //当等于用户每天需要学习的单词数量时时跳转到复习页面

this.router.navigateByUrl("user/tabs/tab1/review");

if(!isknow)

this.curWord.wrongTime+=2;

this.curWord.wrongTime+=2; //至少复习两次

this.showZH = true;

}

/\*\*

\* 跳转到下一个单词

\*/

next(){

this.preWord = this.curWord;

//如果太长了就把超过10的部分用...代替

if(this.preWord.zh.length > 10){

this.preWord.zh = this.preWord.zh.slice(0,9);

this.preWord.zh.concat("...");

}

this.studied++;

this.curWord = this.words[this.studied];

this.showZH = false;

}

### 单词复习算法

words: Word[] = []; //将要学习的单词数组

currentWords: Word[] = []; //当前学习的单词与三个错误单词

curWord: Word; //当前正在复习的单词

preWord: Word = new Word(0,"","",0); //上一个复习的单词

studied: number = 0; //已学习的单词数量

showZH: boolean = false;

constructor(private router: Router, private dataService: DataService) { }

/\*\*

\* 进入下一个单词的学习的方法

\*/

next(){

this.preWord = this.curWord;

this.studied++;

this.currentWords = this.getStudyWords(this.studied);

console.log(this.currentWords);

this.showZH = false;

if(this.words.length > 0 && this.studied >= this.words.length - 1) {

this.studied = 0;

console.log(this.words);

}

}

/\*\*

\* 返回一个四个单词对象的数组，一个是正确的，三个是错误的

\*

\* @param num

\*/

getStudyWords(num: number): Word[]{

let words = [];

let threRadom = this.getThreRadom(num);

this.curWord = this.words[num];

for(let i=0; i<threRadom.length; i++)

words.push(this.words[threRadom[i]]);

return words;

}

/\*\*

\* 返回一个包含四个数字的数组

\*

\* @param num 找出三个非num的数字与num一起返回

\*/

getThreRadom(num: number): number[]{

let threRadom = [];

let i = 0;

let flag = false; //判断num是否在threRadom中

//找出非num的三个数

while (i < 3){

let temp = ReviewComponent.randomNum(0, this.words.length-1);

if(threRadom.includes(temp)) continue;

if(temp == num){

threRadom.push(num);

flag = true;

continue;

}else if(temp > num){

if(flag) threRadom.push(temp);

else{

threRadom.push(temp);

threRadom.push(num);

flag = true;

}

}else if(temp < num){

if(flag) threRadom.push(temp);

else{

threRadom.push(num);

threRadom.push(temp);

flag = true;

}

}

i++;

}

// console.log(threRadom);

return threRadom;

}

/\*\*

\* 产生大于等于min，小于等于max的随机数

\*

\* @param min

\* @param max

\*/

static randomNum(min: number, max: number): number{

let range = max - min;

let rand = Math.random();

return min + Math.round(rand \* range);

}

isRight(word: Word){

console.log("current: " + this.curWord.en);

console.log("input: "+ word.en);

if(word.en != this.curWord.en) {

//如果错误那么错次+1

this.curWord.wrongTime++;

console.log("wrong");

this.showZH = true;

}

else {

console.log("correct");

this.curWord.wrongTime--;

//学习完成后从需要复习的单词列表中删除这个单词

if(this.curWord.wrongTime <= 0){

console.log(this.curWord.en + "学习完毕");

this.words.splice(this.words.indexOf(this.curWord), 1);

console.log(this.words);

}

this.next();

}

}

### 单词复习页面

<!--页头-->

<Navbar [icon]="icon"

[mode]="'light'"

(onLeftClick)="onLeftClick()"

>

{{preWord.en}} {{preWord.zh}}

</Navbar>

<ng-template #icon>

<Icon [type]="'left'"></Icon>

</ng-template>

<ion-content>

<WhiteSpace [size]="'xl'"></WhiteSpace><WhiteSpace [size]="'xl'"></WhiteSpace>

<ion-grid style="height: 90%">

<ion-row style="height: 30%"><ion-col size="12" style="text-align: center; ">

<h1>{{curWord.en}}</h1></ion-col>

<ion-row [hidden]="!showZH"><ion-col size="12" style="text-align: center;"><h3 style="text-align: center; ">{{curWord.zh}}</h3></ion-col> </ion-row>

</ion-row>

<WhiteSpace [size]="'xl'"></WhiteSpace>

<ion-row [hidden]="showZH"><ion-col size="12"><div Button [type]="'ghost'" [inline]="true" (click)="isRight(currentWords[0])" style="width: 100%">{{currentWords[0].zh}}</div></ion-col></ion-row>

<WhiteSpace ></WhiteSpace>

<ion-row [hidden]="showZH"><ion-col size="12"><div Button [type]="'ghost'" [inline]="true" (click)="isRight(currentWords[1])" style="width: 100%">{{currentWords[1].zh}}</div></ion-col></ion-row>

<WhiteSpace [size]="'xl'"></WhiteSpace>

<ion-row [hidden]="showZH"><ion-col size="12"><div Button [type]="'ghost'" [inline]="true" (click)="isRight(currentWords[2])" style="width: 100%">{{currentWords[2].zh}}</div></ion-col></ion-row>

<WhiteSpace ></WhiteSpace>

<ion-row [hidden]="showZH"><ion-col size="12"><div Button [type]="'ghost'" [inline]="true" (click)="isRight(currentWords[3])" style="width: 100%">{{currentWords[3].zh}}</div></ion-col></ion-row>

<ion-row [hidden]="!showZH"><div Button [type]="'ghost'" [inline]="true" (click)="next()" style="width: 100%">下一个</div></ion-row>

</ion-grid>

</ion-content>

## 单词计划实现

### 单词计划页面

<ion-content>

<Navbar [icon]="icon"

[mode]="'light'"

(onLeftClick)="onLeftClick()"

>

我的计划

</Navbar>

<ng-template #icon>

<Icon [type]="'left'"></Icon>

</ng-template>

<div><h1>{{getValue(value)}}</h1></div>

<WhiteSpace [size]="'xl'"></WhiteSpace>

<WhiteSpace [size]="'xl'"></WhiteSpace>

<WhiteSpace [size]="'xl'"></WhiteSpace>

<PickerView [data]="books"

[cascade]="false"

[(ngModel)]="value"

(ngModelChange)="onChange($event)"

></PickerView>

<WhiteSpace [size]="'xl'"></WhiteSpace>

<WhiteSpace [size]="'xl'"></WhiteSpace>

<WhiteSpace [size]="'xl'"></WhiteSpace>

<WhiteSpace [size]="'xl'"></WhiteSpace>

<WhiteSpace [size]="'xl'"></WhiteSpace>

<WhiteSpace [size]="'xl'"></WhiteSpace>

<div Button [type]="'primary'" (click)="submitChange()">确定修改</div>

</ion-content>

### 单词计划相关代码

books = [];

value = [];

constructor(private dataService: DataService, private router: Router) { }

ngOnInit() {

for(let i=0; i<this.dataService.books.length; i++){

this.books[i] = {

label: this.dataService.books[i].name,

children: [{label: '10',},

{label: '15',},

{label: '20',},

{label: '25',},

{label: '30',},

{label: '35',},

{label: '40',},

{label: '45',},

{label: '50',},

{label: '55',},

{label: '65',},

{label: '70',}

]

};

}

// if(this.dataService.currUser.curbook != 0){

// this.value = [this.dataService.books[this.dataService.currUser.curbook].name+'', this.dataService.currUser.wNumPerD+''];

// }else {

this.value = ['四级词汇', 25+''];

// }

console.log(this.books);

}

onLeftClick() {

this.router.navigateByUrl("../");

}

onChange(result) {

console.log(this.value, result);

}

getValue(result) {

let value = [];

let temp = '';

result.forEach(item => {

value.push(item.label || item);

temp += item.label || item;

});

return value;

}

submitChange() {

this.dataService.currUser.wNumPerD = parseInt(this.getValue(this.value)[1]);

for(let i=0; i<this.dataService.books.length; i++){

if(this.getValue(this.value)[0] == this.dataService.books[i].name){

this.dataService.currUser.curbook = i;

}

}

console.log(this.dataService.currUser);

}

## 单词列表实现

### 单词列表页面

<ion-content>

<Navbar [icon]="icon"

[mode]="'light'"

(onLeftClick)="onLeftClick()"

>

单词列表

</Navbar>

<ng-template #icon>

<Icon [type]="'left'"></Icon>

</ng-template>

<Tabs [useOnPan]="true"

[swipeable]="true"

[activeTab]="index"

[page]="3"

[tabBarActiveTextColor]="'#1890ff'"

(onChange)="onChange($event)"

(onTabClick)="onTabClick($event)"

>

<TabPane [title]="titleTemplate">

<ng-template #titleTemplate>

<div>应学单词</div>

</ng-template>

<div style=" align-items: center;background-color: #fff">

<List [className]="'my-list'">

<div \*ngFor="let word of words">

<ListItem [extra]="extra" style="width: 100%; height: 10px; align-content: center;"><ion-icon name="star" size="large"></ion-icon>{{word.en}}</ListItem>

<ng-template #extra>

{{word.zh}}

</ng-template>

</div>

</List>

</div>

</TabPane>

<TabPane [title]="titleTemplate2">

<ng-template #titleTemplate2>

<div>收藏单词</div>

</ng-template>

<div style="display: flex; align-items: center;justify-content: center;height: 150px;background-color: #fff">

<List [className]="'my-list1'">

<ListItem [extra]="extra" style="width: 100%; height: 10px; align-content: center;"><ion-icon name="star" size="large"></ion-icon>{{words[2].en}}</ListItem>

<ng-template #extra>

{{words[2].zh}}

</ng-template>

</List>

</div>

</TabPane>

</Tabs>

</ion-content>

## 单词搜索实现

百度翻译开放平台是百度面向广大开发者提供开放服务的平台。它的通用翻译API支持28种语言互译，覆盖中、英、日、韩、西、法、泰、阿、俄、葡、德、意、荷、芬、丹等；支持28种语言的语种检测。我们只需要通过调用通用翻译API，传入待翻译的内容，并指定要翻译的源语言（支持源语言语种自动检测）和目标语言种类，就可以得到相应的翻译结果。通用翻译API HTTP地址为http://api.fanyi.baidu.com/api/trans/vip/translate。在调用服务时，我们需要向该地址通过POST或GET方法发送下列字段来访问服务，如表xxx



签名是为了保证调用安全，使用MD5算法生成的一段字符串，生成的签名长度为 32位，签名中的英文字符均为小写格式。签名生成方法如下：1、将请求参数中的 APPID(appid), 翻译query(q, 注意为UTF-8编码), 随机数(salt), 以及平台分配的密钥按照 appid+q+salt+密钥 的顺序拼接得到字符串1。2、对字符串1做md5，得到32位小写的sign。

在翻译过程中有三个步骤：1、请先将需要翻译的文本转换为UTF-8编码；2、在发送HTTP请求之前需要对各字段做[URL encode](http://api.fanyi.baidu.com/api/trans/product/apidoc" \l "appendix)；3、在生成签名拼接 appid+q+salt+密钥字符串时，q不需要做URL encode，在生成签名之后，发送HTTP请求之前才需要对要发送的待翻译文本字段q做URL encode。最后返回的结果字段为json字段。具体实现如下：

**将需要翻译的文本转换为UTF-8编码**

encode(params) {  
 var str = '';  
 if (params) {  
 for (var key in params) {  
 if (params.hasOwnProperty(key)) {  
 var value = params[key];  
 str += encodeURIComponent(key) + '=' + encodeURIComponent(value) + '&';}}  
 str = '?' + str.substring(0, str.length - 1);}  
 if (str == '?') {  
 return "";}  
 return str;}

调用API接口实现翻译

translate2() {  
 this.tag = true;  
 const bd\_params = {  
 q: '', // 要翻译的文本  
 from: 'auto', // 源语言  
 to: 'auto', // 目标语言  
 appid: '20190531000303832', // 应用ID  
 salt: 0, // 实时计算，生成随机数  
 sign: '' // 实时计算  
 };  
 bd\_params.q = this.params.q;  
 let bd\_secret = 'p\_5QJjTYSc3QGIk8wmMm';  
 bd\_params.salt = new Date().getTime();  
 let str1 = bd\_params.appid + bd\_params.q + bd\_params.salt + bd\_secret;  
 bd\_params.sign = MD5(str1); //使用MD5加密  
 this.http.get("/api/trans/vip/translate" + this.encode(bd\_params)).subscribe(  
 (data: any) => {  
 console.log(data);  
 this.displayParams = data.trans\_result[0]; //选取翻译结果中的第一条  
 },  
 (error: any) => {  
 console.log(error);  
 }  
 );  
}

## 查询单词收藏功能实现

## 个人信息修改实现

## 今日计划部分实现

# 功能展示

## 单词学习系统主要界面

### 登陆界面

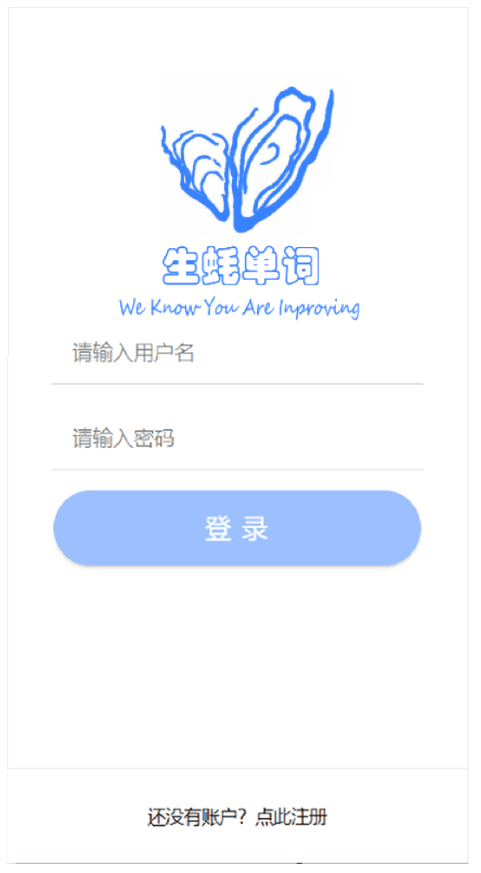


图 4.1 登录界面

### 用户注册界面

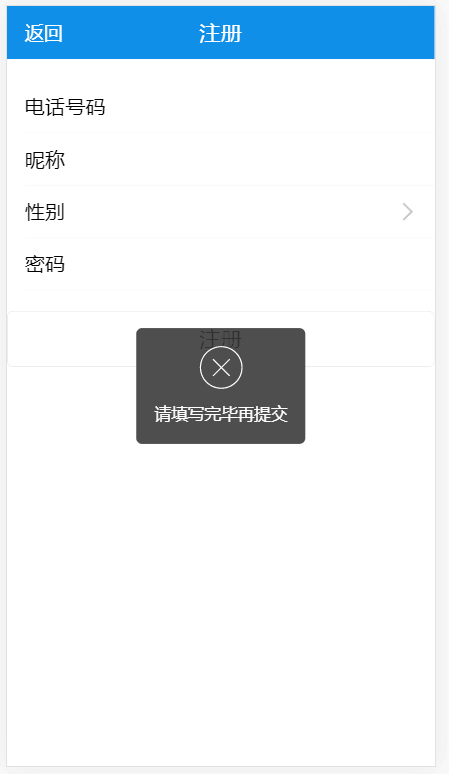
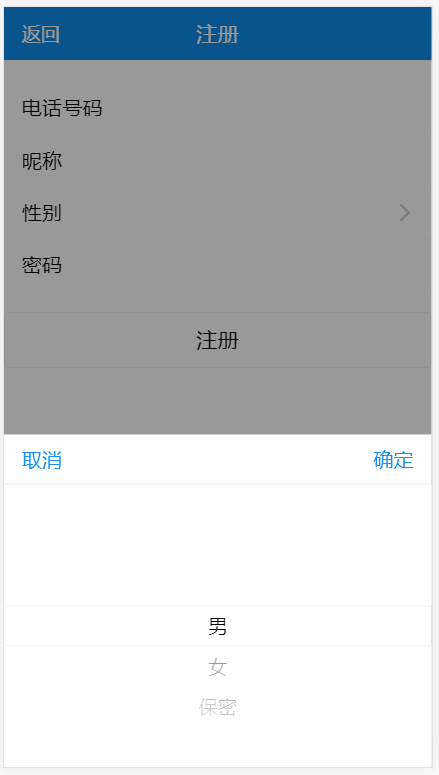
 

图 注册-未填写提交 图 注册-性别选择

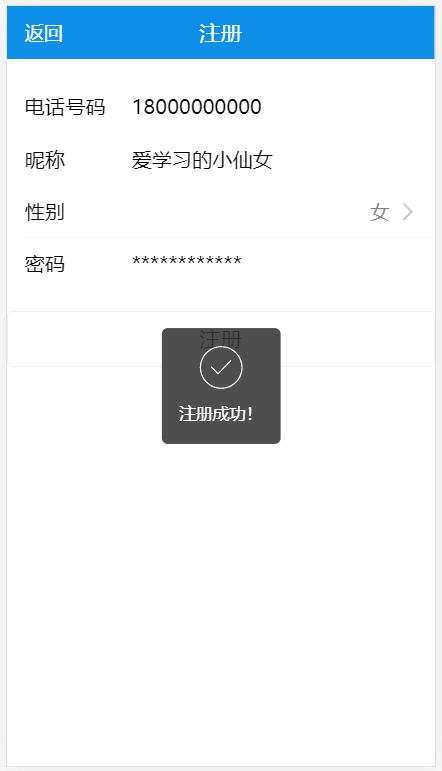
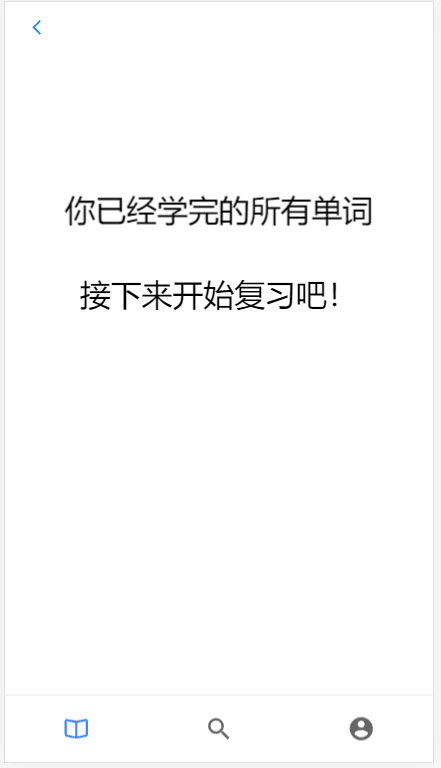


图 注册成功

### 单词学习界面

图 学习信息显示页——初次登陆



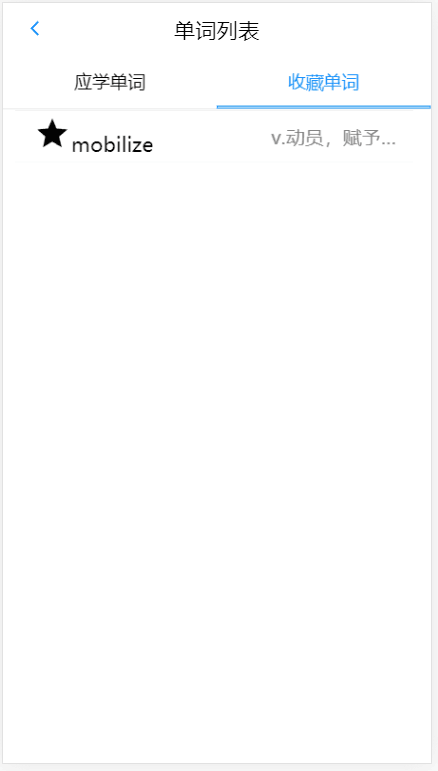




### 计划修改页面



### 单词列表页面



### 单词搜索界面

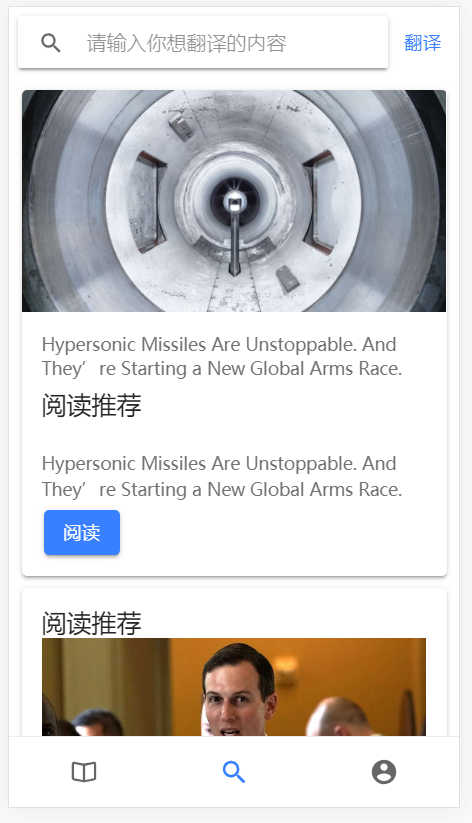


图4.6 单词查询页

### 个人信息查看与修改界面

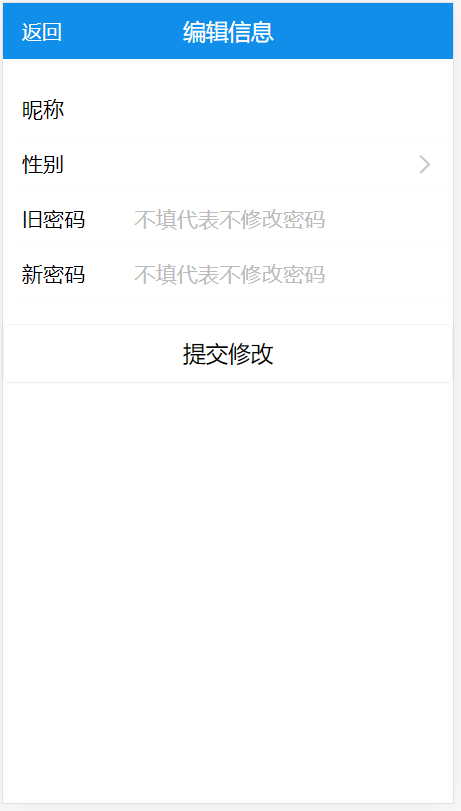
 

图4.7 个人信息查看页 图4.8 个人信息修改页面

### 背单词系统管理系统

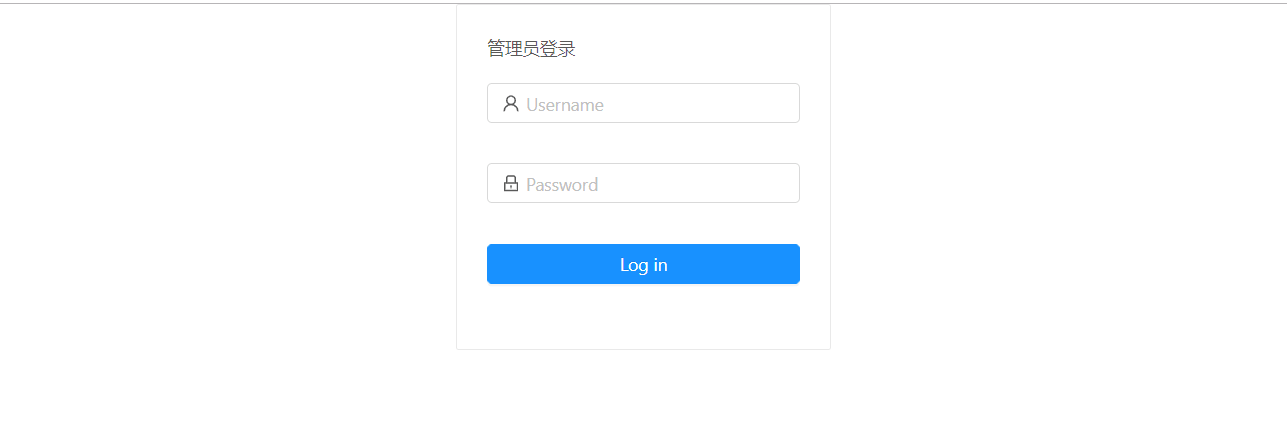


图4.9 管理员用户登陆页



图4.10 管理员管理用户信息页

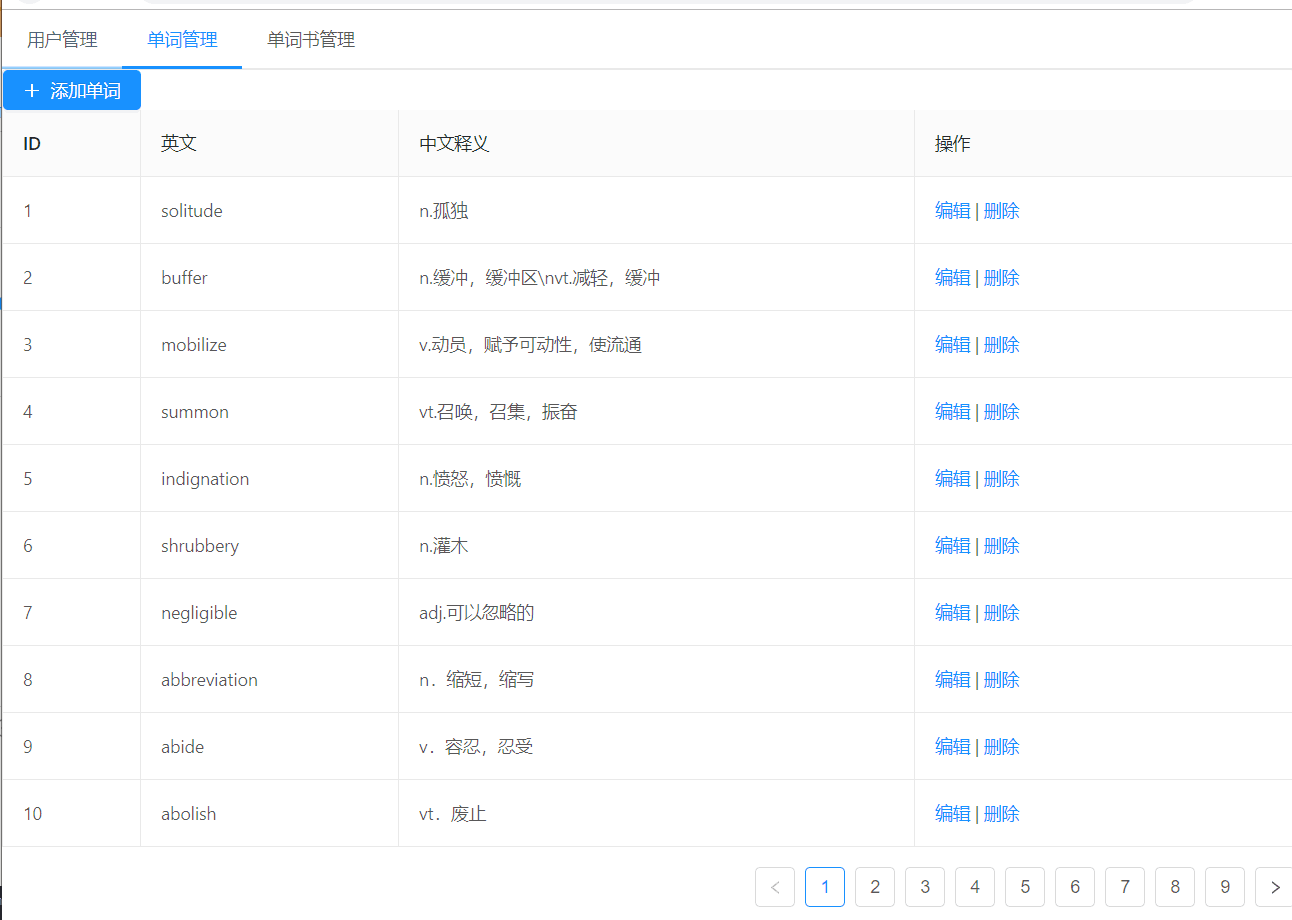


图4.11 管理员管理单词页

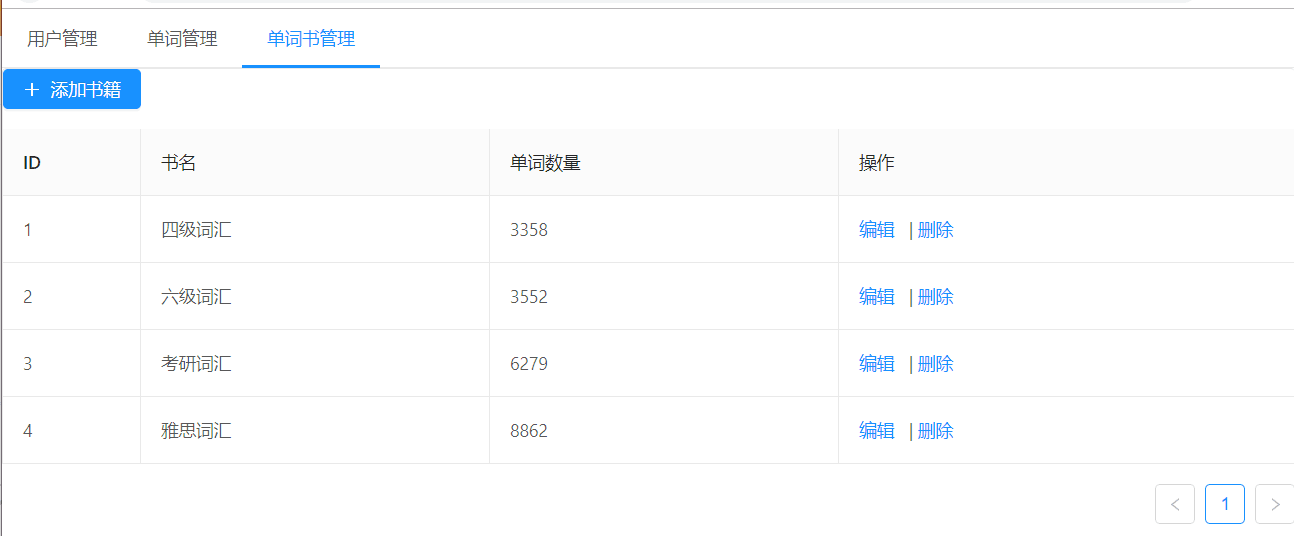


图4.12 管理员管理单词书页

7项目总结