# 基于局域网的移动设备与后台通讯机制

# 的跨平台的研究和实现报告

--------徐建海[南京中医药大学]

目录

[基于局域网的移动设备与后台通讯机制 1](#_Toc399868172)

[的跨平台的研究和实现报告 1](#_Toc399868173)

[基于局域网的移动设备与后台通讯机制 2](#_Toc399868174)

[的跨平台的研究和实现报告 2](#_Toc399868175)

[一、 背景: 2](#_Toc399868176)

[二、系统需求： 3](#_Toc399868177)

[三、需求分析： 3](#_Toc399868178)

[3.1信息的抓取： 3](#_Toc399868179)

[3.2数据的存储 3](#_Toc399868180)

[3.3数据的转发 3](#_Toc399868181)

[3.4详细内容 4](#_Toc399868182)

[四.数据库设计 7](#_Toc399868183)

[五、项目实施 8](#_Toc399868184)

[5.1功能模块的划分 8](#_Toc399868185)

[5.2日志系统功能 8](#_Toc399868186)

[5.3请求转发功能 9](#_Toc399868187)

[5.4后台处理模块 10](#_Toc399868188)

[6.维护 24](#_Toc399868189)

[6.1爬虫的优化 24](#_Toc399868190)

[6.2 管理员后台的建立 24](#_Toc399868191)

[6.3后台程序的优化 24](#_Toc399868192)

# 基于局域网的移动设备与后台通讯机制

# 的跨平台的研究和实现报告

## 背景:

### 1.简介

随着移动互联网的兴起，手机等移动设备成为了新时代下经济的刺激点，基于移动的开发工程如雨后春笋般涌现，但是不同移动设备之间兼容性导致了大量的重复开发：同样的后台内容，可能需要同时开发 ios 设备、ipad设备以及android设备和黑莓等移动终端，不仅如此，不同手持设备之间在开发过程中还存在不兼容的情况（如分辨率等问题），严重阻碍了开发的进程，增加了开发工程的人员，降低了资源利用率，加重了开发的费用。因此，如何实现跨平台的开发成为了当今移动工程领域的首要解决问题。

基于html5的移动开发在最近几年异常火热，作为移动设备不兼容的终结者备受关注。

相比传统的移动开发手段，通过开发工具如phonegap，可以大大减少移动开发过程中的低效的资源利用率，精简人员开发数量，提高工作效率。在本研究实现报告中，使用HBuild工具，快速开发一套简单的前后台交互式实例，实现前端的信息更新。

## 二、系统需求：

1. 该网站后台需要能提供常用网站如Voa standard English(（http://www.51voa.com/)的数据信息以及水滴英语作文网（<http://www.adreep.cn/>）的部分文章数据，并且能够及时更新相应网站的信息。
2. 用户需要具有用户名和密码从移动端登陆app,在提供正确的当前密码的情况下菜可以修改密码，修改密码后需要重新登陆。用户可以选择退出程序。
3. 用户登陆后，可以选择性更新到最新的文章、作文和听力，当然，点击相应的信息条目能够显示该信息的具体内容。
4. 用户可以收藏浏览到的听力、文章、作文，并能够查看自己的收藏内容。
5. 用户可以删除自己收藏的内容。
6. 该应用能够跨平台使用，在ios、android、windows phone 等手持设备上（平板和手机为主）能够运行良好。

## 三、需求分析：

### 3.1信息的抓取

对Voa standard English（<http://www.51voa.com/VOA_Standard_1.html>）和水滴英语作文网（<http://www.adreep.cn/>）的数据进行全部抓取，并且每天定时进行增量抓取。使用集合数据库进行数据的存储，提高数据的响应度。

### 3.2数据的存储

使用redis+mongodb两种数据库的搭配，实现爬虫的信息保存的功能，同时使用文件系统的目录存放mp3音频文件。

### 3.3数据的转发

数据主要分为音频数据和文本数据，对文本数据的请求转发到后台处理程序，由相关程序获取集合数据库中数据并将结果进行转发，对音频数据的请求将会通过ngnix进行静态资源的重定向，减少服务器和数据库的压力。整个转发过程使用局域网组网，通过无线路由器模拟真实网络请求。

### 3.4详细内容

说明:由于采用了mongodb文档类型的数据库，所以数据项的长度和取值范围多余，这里仅仅列出。。

#### 3.4.1数据项

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 含义说明 | 别名 | 类型 | 长度 | 取值范围 |
| 听力标题 | 抓取到的听力的标题 | Listening\_title | 字符串 |  |  |
| 听力发布时间 | 抓取到的听力的发布时间 | Listening\_publish | 字符串 |  |  |
| 听力收藏时间 | 用户收藏该听力的时间 | Listening\_collect | 字符串 | 50 |  |
| 听力内容 | 抓取到的听力的内容 | Content | 字符串 | 50 |  |
| 听力简介 | 听力内容的一定文字的提取 | Intro | 字符串 | 50 |  |
| 听力音频地址 | 听力音频在文件系统上的绝对路径 | MP3 | 字符串 |  |  |
| “阅读”标题 | 抓取到的“阅读”的标题 | Reading\_title | 字符串 |  |  |
| “阅读”发布时间 | 抓取到的“阅读”的发布时间 | Reading\_collect | 字符串 |  |  |
| “阅读”收藏时间 | 用户收藏“阅读”的时间 | Reading\_collect | 字符串 |  |  | |
| “阅读”内容 | 抓取到的“阅读”的内容 | Content | 字符串 |  |  |
| 作文标题 | 抓取到的作文的标题 | Article\_title | 字符串 |  |  |
| 作文发布时间 | 抓取到的作文的发布时间 | Article\_publish | 字符串 |  |  |
| 作文收藏时间 | 用户收藏作文的时间 | Article\_collect | 字符串 |  |  |
| 用户名 | 用户的邮箱地址 | Name | 字符串 |  |  |
| 密码 | 用户的密码 | Pass | 字符串 |  |  |

#### 3.4.2数据结构

##### (i)用户

含义说明: 定义了一个描述用户的信息结构

组成: 用户名、密码、收藏组（听力、阅读、作文收藏列表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 含义说明 | 别名 | 类型 | 长度 | 取值范围 |
| \_id | 唯一性标志 | 用户标志 | Byte | 12 | - |
| name | 用户名称 | 邮箱 | String |  |  |
| collections | 用户的收藏 |  | 数组 |  |  |
| Listenings | 收藏的听力 | 听力列表 | 列表 |  |  |
| Articles | 收藏的文章 | 文章列表 | 列表 |  |  |
| Readings | 收藏的阅读 | 阅读列表 | 列表 |  |  |
| Title | 收藏的标题 |  | String |  |  |
| Intro | 收藏内容的简介 | 简介 | String |  |  |
| Publish\_date | 收藏的内容被发布的时间 | 发布时间 | String |  |  |
| Collect\_date | 内容被发布的时间 | 收藏时间 | String |  |  |

##### (ii)听力

含义说明: 定义了一个描述听力的信息结构

组成: 听力标题、听力文本、听力音频、发布时间

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 含义说明 | 别名 | 类型 | 长度 | 取值范围 |
| \_id | 唯一性标志 | 用户标志 | Byte | 12 |  |
| Title | 听力的标题 | 标题 | String |  |  |
| Content | 听力的文本 | 内容 | String |  |  |
| Mp3 | 听力的音频绝对地址 | 音频源 | String |  |  |
| Time | 听力的发布时间 | 发布时间 | String |  |  |

##### (iii)阅读

含义说明: 定义了一个描述阅读的信息结构

组成: 阅读标题、阅读文本、发布时间

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 含义说明 | 别名 | 类型 | 长度 | 取值范围 |
| \_id | 唯一性标志 | 用户标志 | Byte | 12 |  |
| Title | 文章的标题 | 标题 | String |  |  |
| Content | 文章的文本 | 内容 | String |  |  |
| Time | 文章的发布时间 | 发布时间 | String |  |  |

##### (iv)作文

含义说明: 定义了一个描述作文的信息结构

组成: 作文标题、作文文本、发布时间

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 含义说明 | 别名 | 类型 | 长度 | 取值范围 |
| \_id | 唯一性标志 | 用户标志 | Byte | 12 |  |
| Title | 作文的标题 | 标题 | String |  |  |
| Content | 作文的文本 | 内容 | String |  |  |
| Time | 作文的发布时间 | 发布时间 | String |  |  |

## 四.数据库设计

由于采用mongodb+redis模型，属于非关系型数据库，因此这里不讨论相应的概念逻辑物理数据库设计。这里简单罗列一下数据库的实现:

Mongdb中的数据库模型:

数据库: Voas:

集合(听力）

Listening:

Title,content,time,mp3

集合(阅读和文章)

Reading ,article:

Title,content,time

集合(用户)

Users:

Name,pass,

listening:

[collect\_date:,publish\_dte:,intro:],

readings:

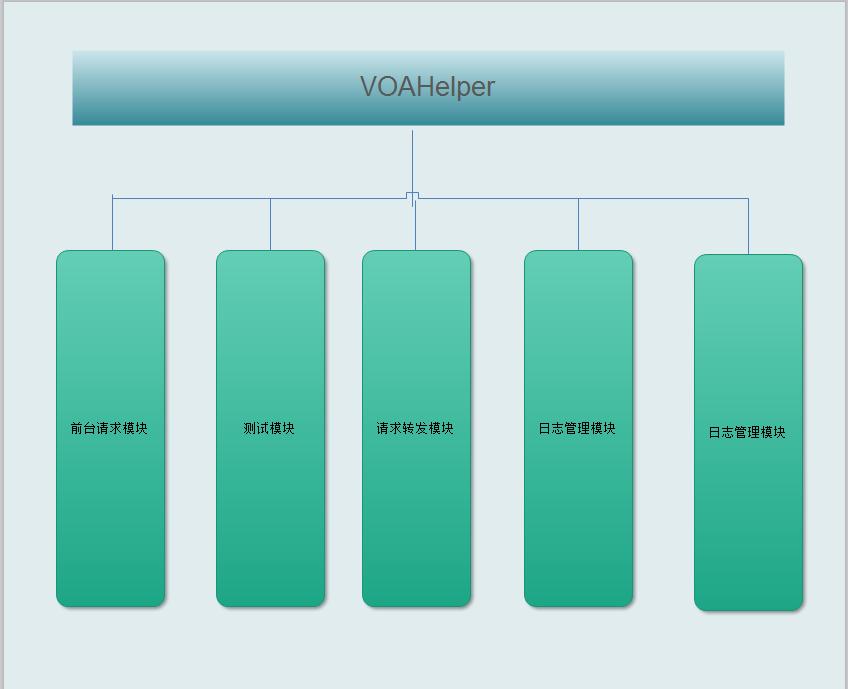
[collect\_date:,publish\_dte:,intro:],

Articles:

[collect\_date:,publish\_dte:,intro:],

## 五、项目实施

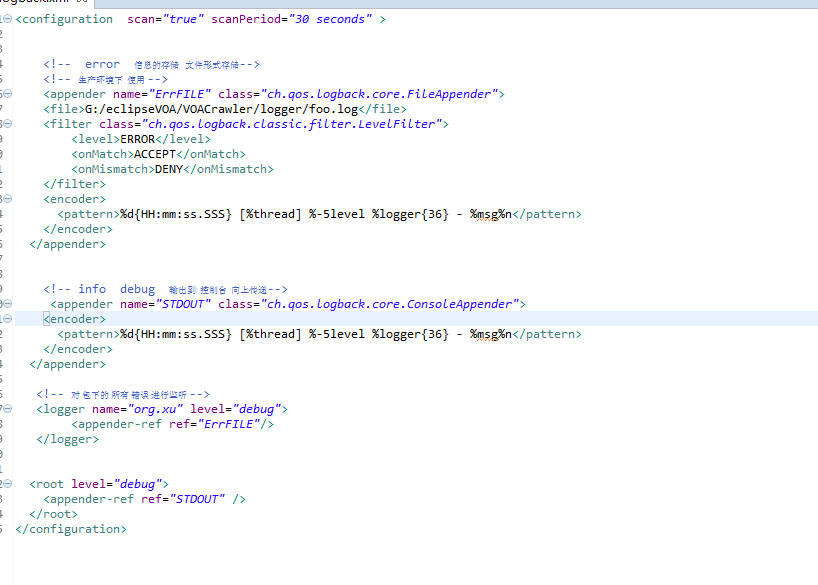
### 5.1功能模块的划分



### 5.2日志系统功能

**采用logback+slf4j搭建日志框架**

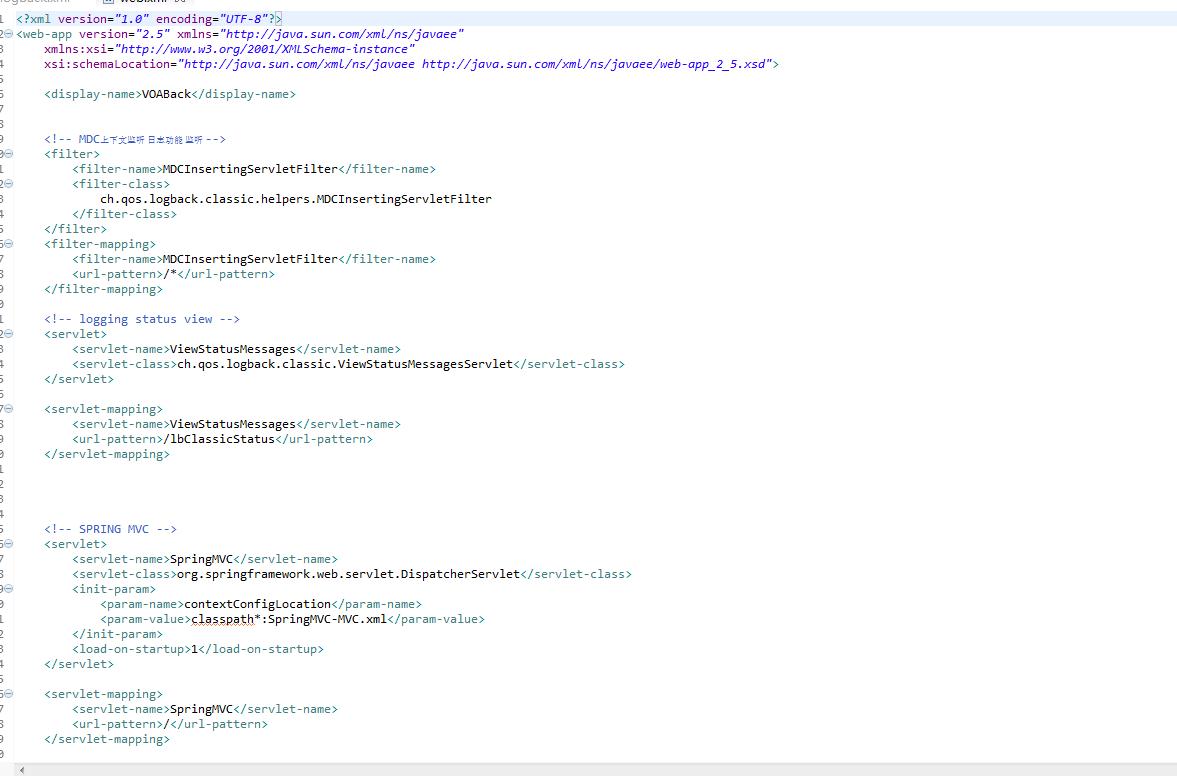
**相关的配置文件:**



### 5.3请求转发功能

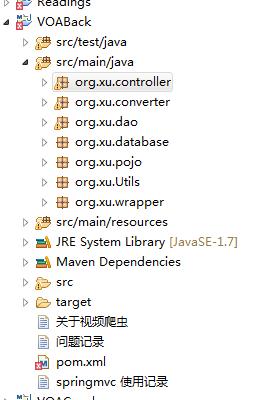
**通过java的web转发功能实现页面的请求转发:**

**这里是相关的配置：**

****

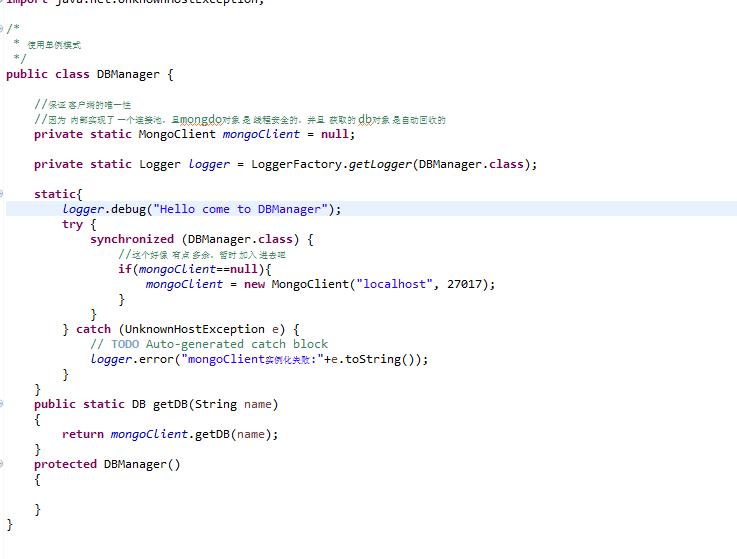
### 5.4后台处理模块

**5.4.1所有后台程序截图如下:**



##### 5.4.2数据库操作类

使用单例模式调用



##### 5.4.3 Dao层的类

**Dao层的类的总体结构图：**



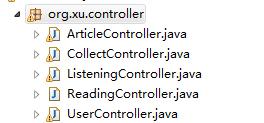
**以ListeningDao为例子**





##### 5.4.4 Controller前台交互类

**Controller层的类的总体结构图：**



**以ListeningController类为例:**

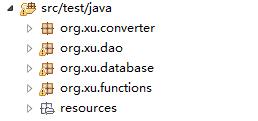


#### 5. 5基于Junit的测试功能模块

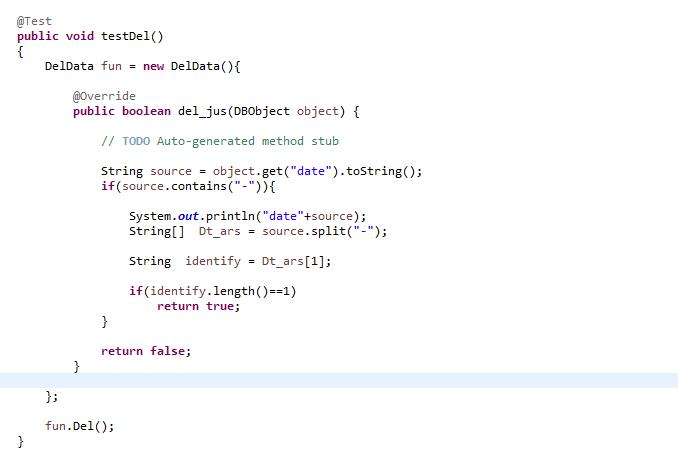
**5.5.1简介**

**使用junit测试dao服务处理层、数据库操作层、数据清洗。**

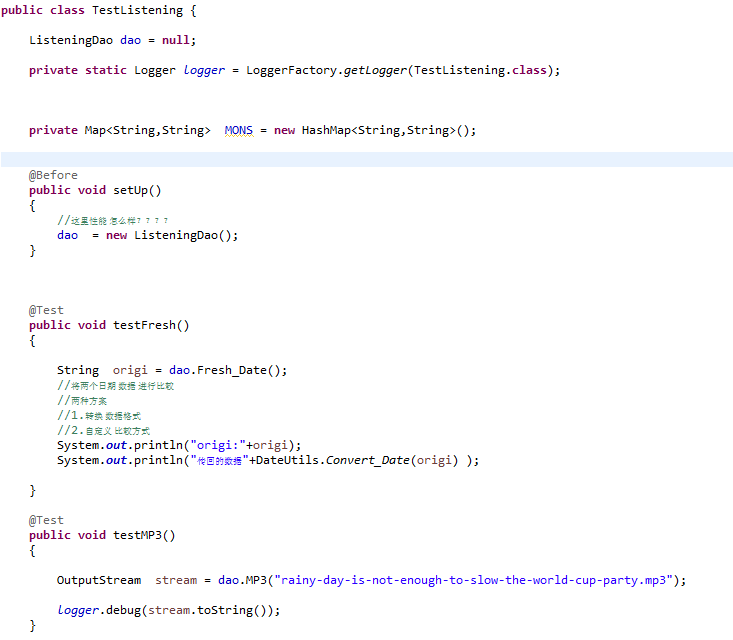
**5.5.2 所有结构如下:**

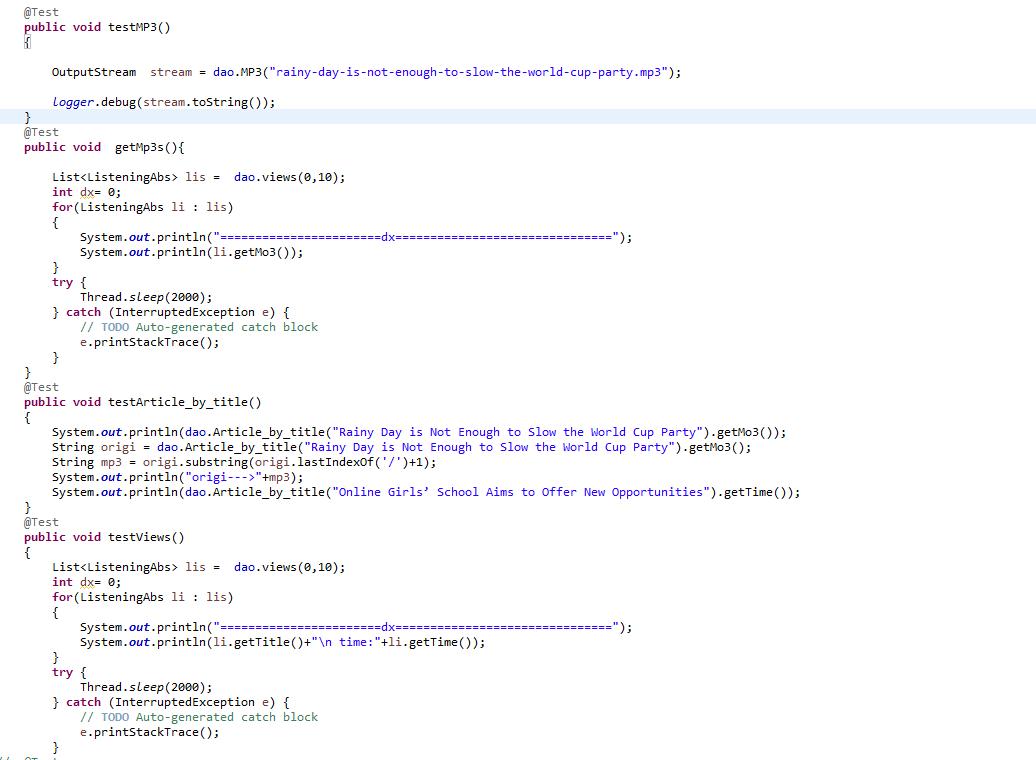


**5.5.3 针对converter数据清洗下日期操作的测试**

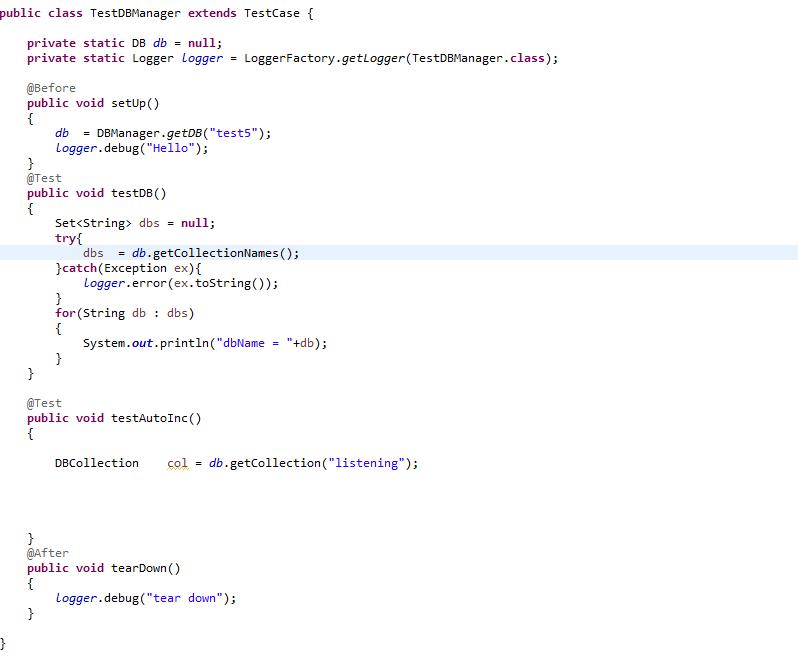


**5.5.4 针对dao后台处理下的listeningDao的部分截图**



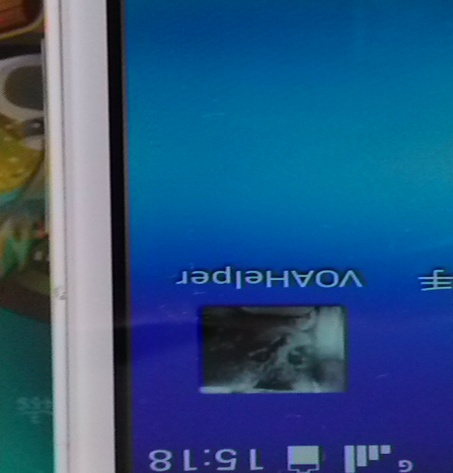


**5.5.5 针对DBManager数据库管理层下的的部分截图**

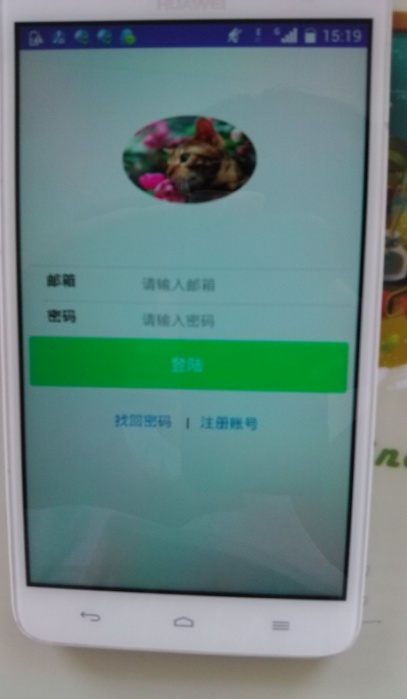
****

#### 5.6 前台

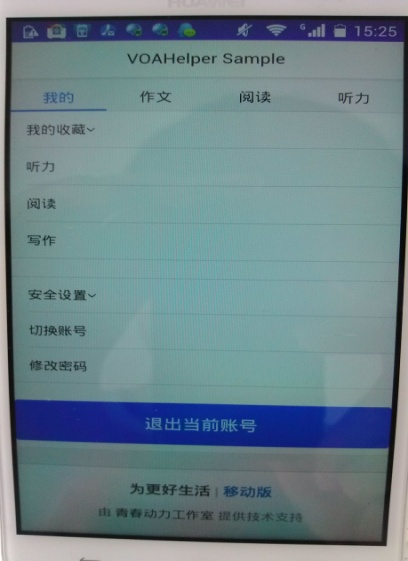
**App图标**

****

##### 登陆界面



##### 首页



#### 作文概览界面



#### 阅读概览界面



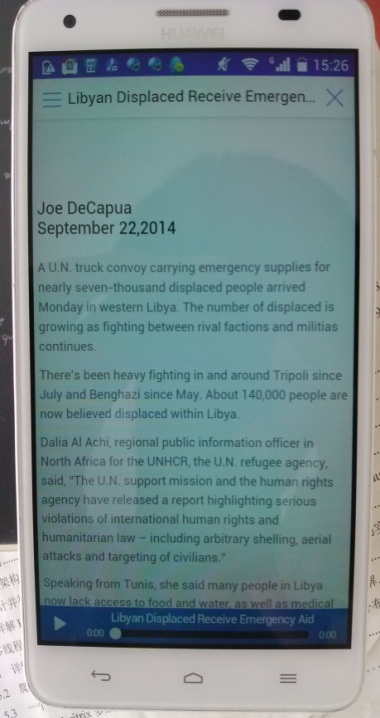
#### 听力概览界面



#### 浏览所有内容的界面



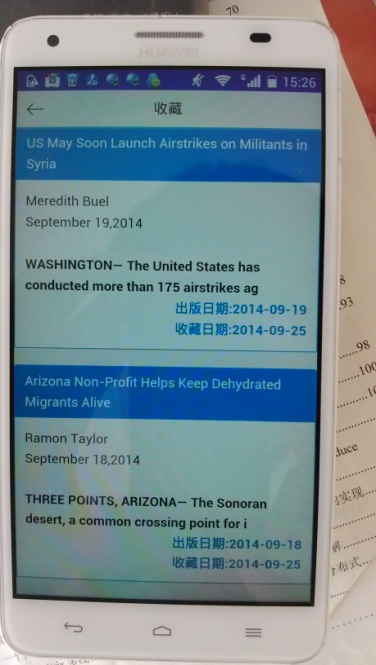
#### 听力详细内容界面



#### 侧边栏界面



#### 收藏界面

**

#### 修改密码界面



## 6.维护

### 6.1爬虫的优化

建议使用nutch分布式进行爬取，这样可以更快的获取大量英语资源用于管理员筛选。

### 6.2 管理员后台的建立

建立管理员后台，方便用户对爬虫获取的数据进行相应的增、删、改、查，实现对内容的优化，给于使用者更愉悦的体验。

### 6.3后台程序的优化

考虑到node的优良特性，建议使用nodejs实现与手机端的交互，但是管理员的后台建议使用java安全的语言机制，配合shiro或者springsecurity实现权限的管理，后天界面可以考虑使用 angularJs。

### 6.4请求转发的优化

建议使用ngnix转发静态资源，比如mp3音频、jquery资源、css资源等，减轻后台处理程序的压力。