## 垃圾分类小程序详细设计说明书

一、概述

1、系统简述

垃圾分类小程序使一款利用图像识别、语音识别技术和文字输入方式以便用户快速查询垃圾分类的微信小程序应用，同时具有游戏和科普知识模块，集趣味和实用性于一体，致力于让用户提高垃圾分类，循环利用的环保意识。

2、软件设计目标

1. 科普垃圾分类知识
2. 使查询垃圾分类更加快速便捷
3. 使垃圾分类更加具有趣味性，增强人们垃圾分类的观念

3、参考资料

（1）聚合数据垃圾分类接口文档：<https://www.juhe.cn/docs/api/id/490?f=375>

（2）天行数据垃圾分类接口文档：<https://www.tianapi.com/apiview/97>

（3）百度图像识别开发文档：<https://ai.baidu.com/aidoc/IMAGERECOGNITION/8k3e7f69o>

（4）微信官方小程序开放文档：<https://developers.weixin.qq.com/miniprogram/dev/framework/>

（5）《Information technology Project Management》8th Kathy Schwalbe 机械工业出版社

（6）《城市生活垃圾分类处理》冀海波 河北科学技术出版社

（7）《垃圾分类小百科》全国通用版 垃圾分类小百科编写组编写 北京联合出版公司

（8）《微信小程序：开发入门及案例分析详解》李俊、边思 机械工业出版社 2017

（9）百度语音识别开发文档：<https://ai.baidu.com/ai-doc/SPEECH/6k38lxp0r>

（10）Java后台与微信小程序交互：<https://blog.csdn.net/qq_42332821/article/details/89854353>

（11）微信小程序底层架构：<https://www.jianshu.com/p/ca720137818f>

（12）微信小游戏：<https://zhuanlan.zhihu.com/p/32826363>

4、修订版本记录

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 修改说明 | 修改人 | 版本 |
| 2020/03/11 | 图像识别查询垃圾分类功能及文字模糊查询功能模块详细设计说明 | 杨官梅 | V1.0 |
| 2020/03/11 | 类图、语音识别垃圾分类、小游戏模块、非功能详细设计 | 文珮全 | V2.0 |
| 2020/03/20 | 类图、图像识别查询、模糊查询、历史查询记录顺序图修改 | 杨官梅 | V3.0 |
| 2020/03/21 | 类图、语音识别垃圾分类、小游戏模块顺序图修改 | 文珮全 | V4.0 |
| 2020/04/05 | 类图 | 杨官梅 | V4.1 |
| 2020/04/22 | 类图、语音识别垃圾分类、小游戏模块顺序图修改 | 文珮全 | V5 |

二、术语表

三、设计概述

1、系统的复用计划：库、框架、模式、构件等方面的复用

2、系统接口设计(根据概要设计进一步细化设计)

• 各种提供给用户的界面

1. 首页：



1. 显示图像识别返回结果页面



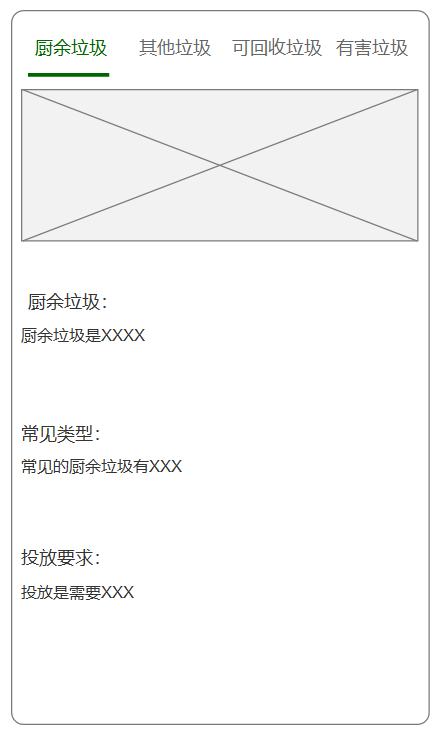
1. 显示垃圾分类结果详情页面



1. 文字模糊查询结果页面



1. 科普垃圾分类页面



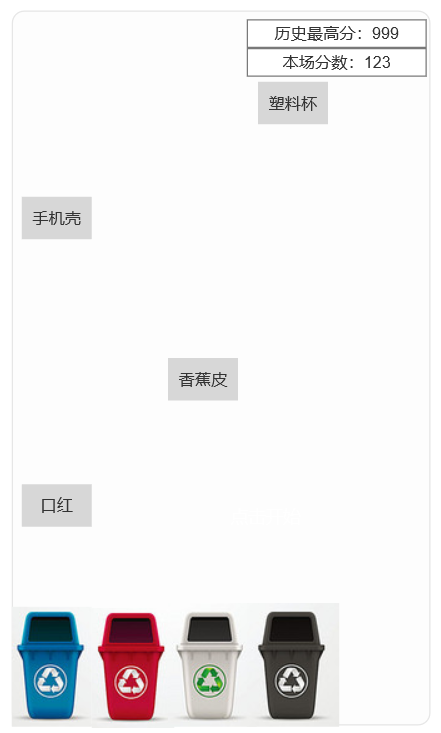
1. 语音识别界面



1. 语音识别垃圾分类结果界面

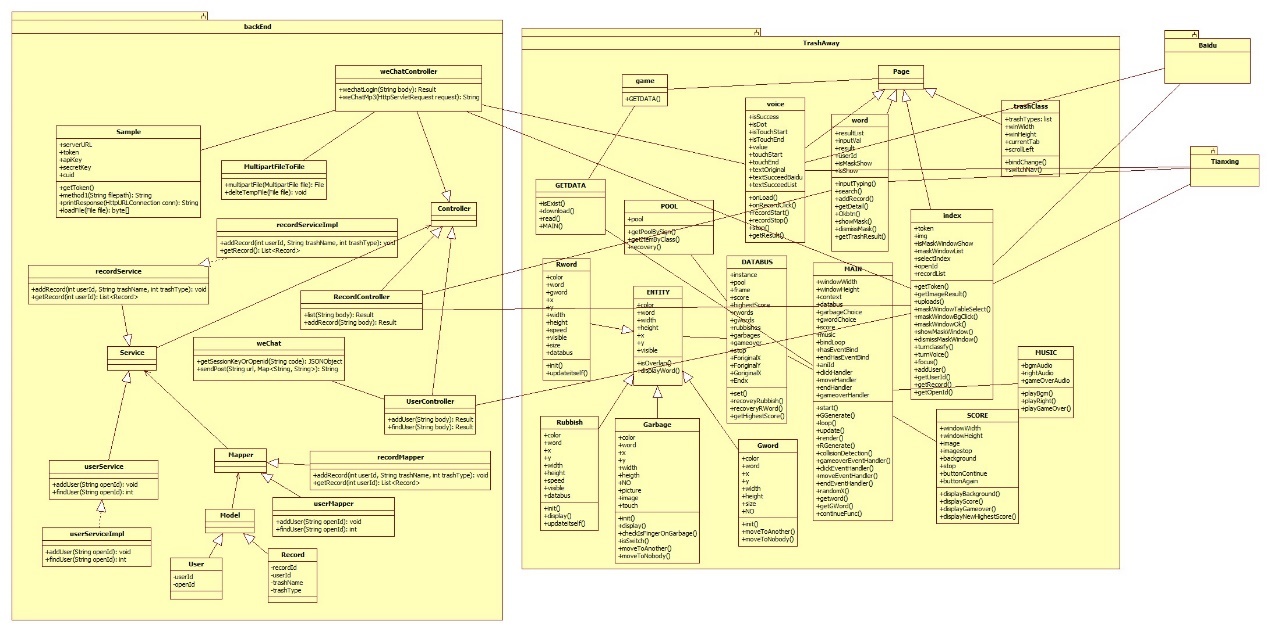


1. 游戏界面



• 系统外部接口设计：与外部系统的交互设计

1. 对象模型设计



backEnd：后台服务器

**userController:用户控制器**

**属性：**

**方法：**

1、addUser():增加用户方法

参数：请求体String body

返回值：Json数据格式

2、findUser():查询用户是否已经存在的方法

参数：请求体String body

返回值：Json数据格式

**userService:用户Service类**

**属性：**

**方法：**

1、addUser():增加用户方法

参数：String openId(微信用户唯一标识)

返回值：void

2、findUser()查询用户是否存在的方法

参数：String openId(微信用户唯一标识)

返回值：int userId(用户Id)

**userServiceImpl:userService实现类**

**属性：**

**方法：**

1、addUser():增加用户方法

参数：String openId(微信用户唯一标识)

返回值：void

2、findUser():查询用户是否存在的方法

参数：String openId(微信用户唯一标识)

返回值：int userId(用户Id)

**userMapper:映射数据库user表的类(与userMapper.xml配置文件对接)**

**属性：**

**方法：**

1、addUser():增加用户方法

参数：String openId(微信用户唯一标识)

返回值：void

2、findUser()查询用户是否存在的方法

参数：String openId(微信用户唯一标识)

返回值：int userId(用户Id)

**recordController:历史记录控制器**

**属性：**

**方法：**

1、addRecord():增加历史记录方法

参数：请求体String body

返回值：Json数据格式

2、list:获取历史查询记录方法

参数：请求体String body

返回值：Json数据格式

**recordService:历史记录Service类**

**属性：**

**方法：**

1、addRecord():增加历史方法

参数：int userId(用户Id)、String trashName(垃圾名称)、int trashType(垃圾类型)

返回值：void

2、getRecord():获取用户历史查询记录的方法

参数：int userId(用户Id)

返回值：List<Record>（record列表）

**recordServiceImpl:recordService实现类**

**属性：**

**方法：**

1、addRecord():增加历史方法

参数：int userId(用户Id)、String trashName(垃圾名称)、int trashType(垃圾类型)

返回值：void

2、getRecord():获取用户历史查询记录的方法

参数：int userId(用户Id)

返回值：List<Record>（record列表）

**recordMapper:映射数据库record表的类(与recordMapper.xml配置文件对接)**

**属性：**

**方法：**

1、addRecord():增加历史方法

参数：int userId(用户Id)、String trashName(垃圾名称)、int trashType(垃圾类型)

返回值：void

2、getRecord():获取用户历史查询记录的方法

参数：int userId(用户Id)

返回值：List<Record>（record列表）

**weChatController:获取微信用户登录openid语音识别相关的控制器**

**属性：**

**weChat w:****获取微信用户登录openid的方法所在的类实例w**

**方法：**

1. weChatLogin():获取微信用户登录openid的方法

参数：@RequestBody String body

返回值：Result

1. weChatMp3():语音识别使用到的方法

参数：HttpServletRequest request

返回值：String

**MultipartFileToFile:****将MultipartFile转换成File的方法类**

**属性：**

**方法：**

1. multipartFileToFile():将MultipartFile转换成File的方法

参数：MultipartFile file

返回值：File

**Sample:百度语音识别的方法类**

**属性**：

String serverURL:百度语音识别的地址

String token:百度语音身份识别token

String apiKey:百度语音识别apiKey

String secretKey:百度语音识别secretKey

String cuid:设备唯一标识，百度语音识别要求上传的参数之一

**方法：**

1、getToken():获取百度语音识别token的方法

2、method1():上传pcm文件进行语音识别的方法

参数：File file:音频文件

1. printResponse():连接服务端打印响应内容的方法

参数：HttpURLConnection conn

返回值：String

1. loadFile()”将文件转换成byte[]的方法

参数：File file

返回值：byte[]

**weChat:获取微信用户登录openid的方法**

**属性：**

**方法：**

1. getSessionKeyOropenid():获取微信登录openid的方法

参数：String code 微信端登录的code 值

返回值：JSONObject

1. sendPost()：Post方式请求服务端方法

参数：String url 服务端访问地址、Map<String,String>Params 请求参数

返回值：String

**TrashAway:微信小程序前端：**

**index：微信小程序首页面**

**属性：**

token:请求图像识别的token

img：图像识别的上传的图片

isMaskWindowShow:弹窗显示标记

maskWindowList:弹窗应该显示的图片识别的结果列表

selectIndex:选择弹窗的元素Id

openId:微信用户的唯一标识

recordList：用户历史记录列表

**方法：**

getToken():获取图像识别的token

getImageResult():上传图片调用百度api获取图片识别结果方法

uploads():上传图片进行图片识别的方法

maskWindowTableSelect():获取点击弹窗元素id的方法

maskWindowBgClick():点击弹窗背景的响应方法

maskWindowOk():点击弹窗“确定“按钮的响应方法

showMaskWindow():显示弹窗的方法

dimissMaskWindow():关闭弹窗的方法

turnclassfy():加载游戏的方法

turnVoice():跳转到语音识别页面的方法

focus():获取指针焦点后的响应方法

getRecord()：获取历史查询记录的方法

addUser():新增用户方法

getUserId()：获取用户Id的方法

getOpenId():获取微信用户唯一标识的方法

**word:微信小程序模糊查询word页面**

**属性：**

resultList：垃圾模糊查询的结果列表

inputVal：搜索框输入值

result：垃圾分类详情的结果

userId：用户Id

isMaskShow：弹窗弹出的标志

isShow:没有垃圾分类结果的图标显示标志

**方法：**

inputTyping():获取搜索框输入值的方法

search():模糊查询垃圾分类的方法

addRecord():增加历史查询记录的方法

getDetail():获取垃圾分类的详情的方法

Okbtn():点击弹窗“知道了“按钮后的响应方法

showMask():弹出弹窗的方法

dismissMask():关闭弹窗方法

getTrashResult():带参数模糊查询获取垃圾分类的方法

参数：String text(垃圾名)

**trashClass:微信小程序垃圾分类科普trashClass页面**

**属性：**

trashTypes:垃圾分类科普内容：厨余垃圾、其他垃圾、可回收垃圾、有害垃圾的科普内容

winWidth:手机屏幕宽

winHeight:手机屏幕高

currentTab:目前页面所处的tab

scrollLeft:屏幕向左滑动的距离

**方法：**

bindChange():tab点击的响应方法

switchNav()：tab华东的响应方法

Baidu：百度图象识别服务

Tianxing:t天性数据垃圾分类接口服务

**voice:微信小程序语音查询页面**

**属性：**

 isSuccess（boolean）

 isDot（String）

 isTouchStart（boolean）

 isTouchEnd（boolean）

 value（int）

 touchStart:（boolean）

 touchEnd（boolean）

 textOriginal（String）

 textSucceedBaidu（String）

 textSucceedList（String）

**方法：**

onLoad：页面加载函数

onRecordClick：点击函数

recordStart：开始录音

recordStop：结束录音

stop：结束录音的监听并上传录音记录并获取服务端返回的语音识别结果

getResult：获取垃圾分类结果

**小游戏类：**

**game类**

**属性：**

**方法**：

GETDATA（）：访问网络或本地存储，获取垃圾分类数据

**GETDATA类：**

**属性：**

**方法：**

isExist()：判断本地存储文件是否存在

download()：从网上下载垃圾分类数据文件并存储在本地

read()：读取存储在本地的垃圾分类文件

MAIN()：执行小游戏逻辑的主函数

**MAIN类**

**属性：**

windowWidth（int）

windowHeight（int）

context（any）

databus（DATABUS）

garbageChoice（GARBAGE）

gwordChoice（GWORD）

score（SCORE）

music（MUSIC）

bindLoop（callback）

hasEventBind（boolean）

endHasEventBind（boolean）

aniId（int）

**方法：**

start()：重新开始一局游戏

GGenerate()：初始化垃圾桶和垃圾桶文字，并将其放入对象池

loop()：循环更新画面各种元素以及打印到屏幕上的函数

update()：更新个种元素的位置等属性

render()：将元素打印到屏幕上

RGenerate()：新建或从对象池中取垃圾桶和垃圾桶文字对象并初始化

collisionDetection()：垃圾与垃圾桶碰撞检测

gameoverEventHandler()：游戏结束后的事件处理器

clickEventHandler()：游戏进行时的点击事件处理器

moveEventHandler()：游戏进行时的手指移动事件处理器

endEventHandler()：游戏进行时的手指离开屏幕的事件处理器

randomX()：随机生成垃圾的x坐标

getword()：获取垃圾的文字

getGWord()：获取垃圾对应分类的名字

continueFunc()：暂停后继续游戏

**DATABUS类**

**属性：**

instance（any）

pool（POOL）

frame（int）

score（int）

highestScore（int）

rwords（Array）

gwords（Array）

rubbishes（Array）

garbages（Array）

gameover（boolean）

stop（boolean）

ForiginalX（int）

ForiginalY（int）

GoriginalX（int）

Endx （int）

**方法：**

set()：在游戏重新开始时重置里面的数据

recoveyRubbish()：回收RUBBISH类

recoveryRWord()：回收RWORD类

getHighestScore(）：获取存储在手机缓存的最高分

**POOL类**

**属性：**

poolDic（symbol）

**方法：**

getPoolBySign()：通过对象标识获取对应的对象池

getItemByClass：根据传入的对象标识，查询后确定是新建亦或是从已有的对象池中取

recovery():将对象回收到对象池

**ENTITY类**

**属性：**

color（string）

word（string）

width（int）

height（int）

x（int）

y（int）

visible（boolean）

**方法：**

isOverlap()：判断垃圾文字与垃圾桶是否碰撞

displayWord()：将文字显示到屏幕上

**Rword类**

**属性：**

color（string）

word（string）

gword（string）

width（int）

height（int）

x（int）

y（int）

speed（int）

visible（boolean）

size（string）

databus（DATABUS）

**方法：**

init()：将从对象池中取出来的对象进行初始化

updateitself()：更新对象的位置信息

**Rubbish类**

**属性：**

color（string）

word（string）

width（int）

height（int）

x（int）

y（int）

speed（int）

visible（boolean）

databus（DATABUS）

**方法：**

init()：将从对象池中取出来的对象进行初始化

display()：将对象显示到屏幕上

updateitself()：更新对象的位置信息

**Garbage类**

**属性：**

color（string）

word（string）

width（int）

height（int）

x（int）

y（int）

NO（int）

picture（string）

image（Image）

touch（boolean）

**方法：**

init()：初始化对象

display()：显示垃圾桶

checkIsFingerOnGarbage()：检查点击事件的手指是否选中垃圾桶

isSwitch()：返回垃圾桶移动后的坐标

moveToAnother()：移动到有垃圾桶的位置

moveToNobody()移动到没有垃圾桶的位置

**Gword类**

**属性：**

color（string）

word（string）

width（int）

height（int）

x（int）

y（int）

size（string）

NO（int）

**方法：**

init()：初始化对象

moveToAnother()：移动到有垃圾桶的位置

moveToNobody()移动到没有垃圾桶的位置

**SCORE类**

**属性：**

windowWidth（int）

windowHeight（int）

image（Image）

imagestop（Image）

background（Image）

stop（stop）

buttonContinue（buttonContinue）

buttonAgain（buttonAgain）

**方法：**

 displayBackground（）：显示游戏背景

displayScore()：游戏时显示分数

displayGameover()：显示游戏失败/暂停的按钮画面

displayNewHighestScore（）：显示刷新纪录后的画面

**MUSIC类**

**属性：**

bgmAudio（Audio）

rightAudio（Audio）

gameOverAudio（Audio）

**方法：**

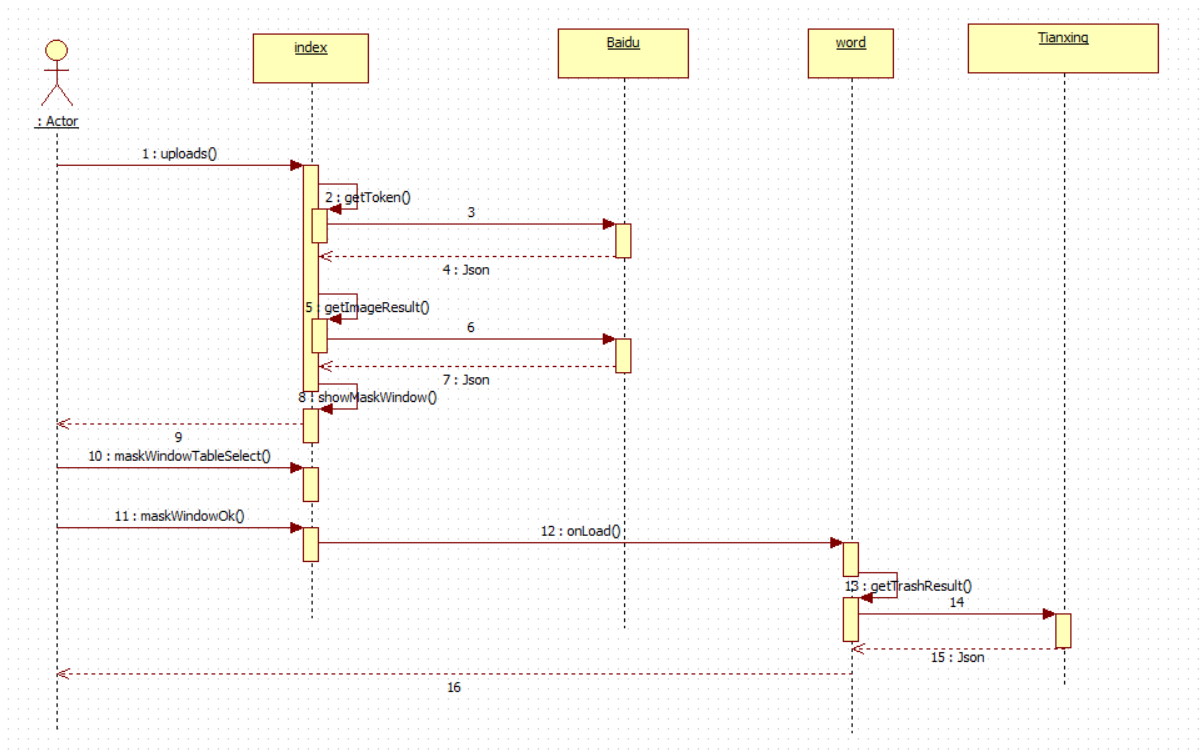
playBgm()：播放背景音乐

playRight()：播放得分时的音乐

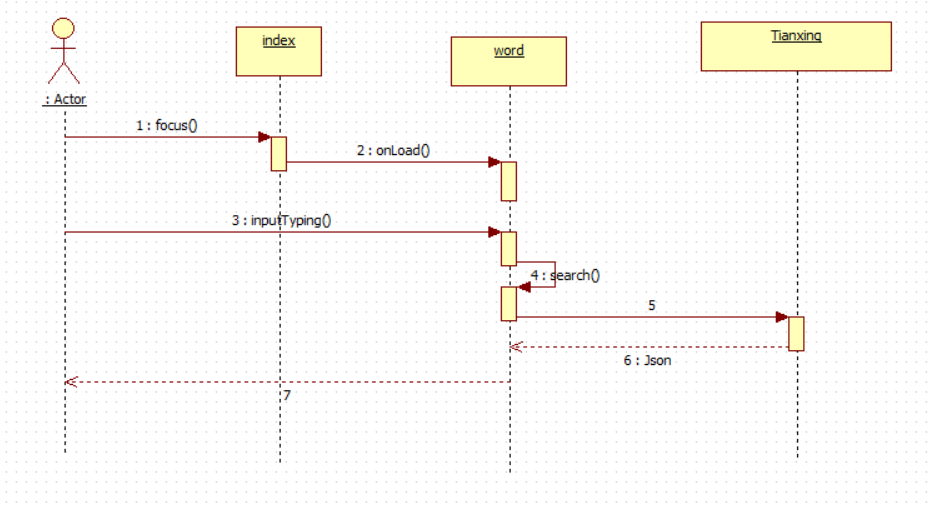
playGameOver()：播放游戏结束的音乐

4、系统用例实现详细设计

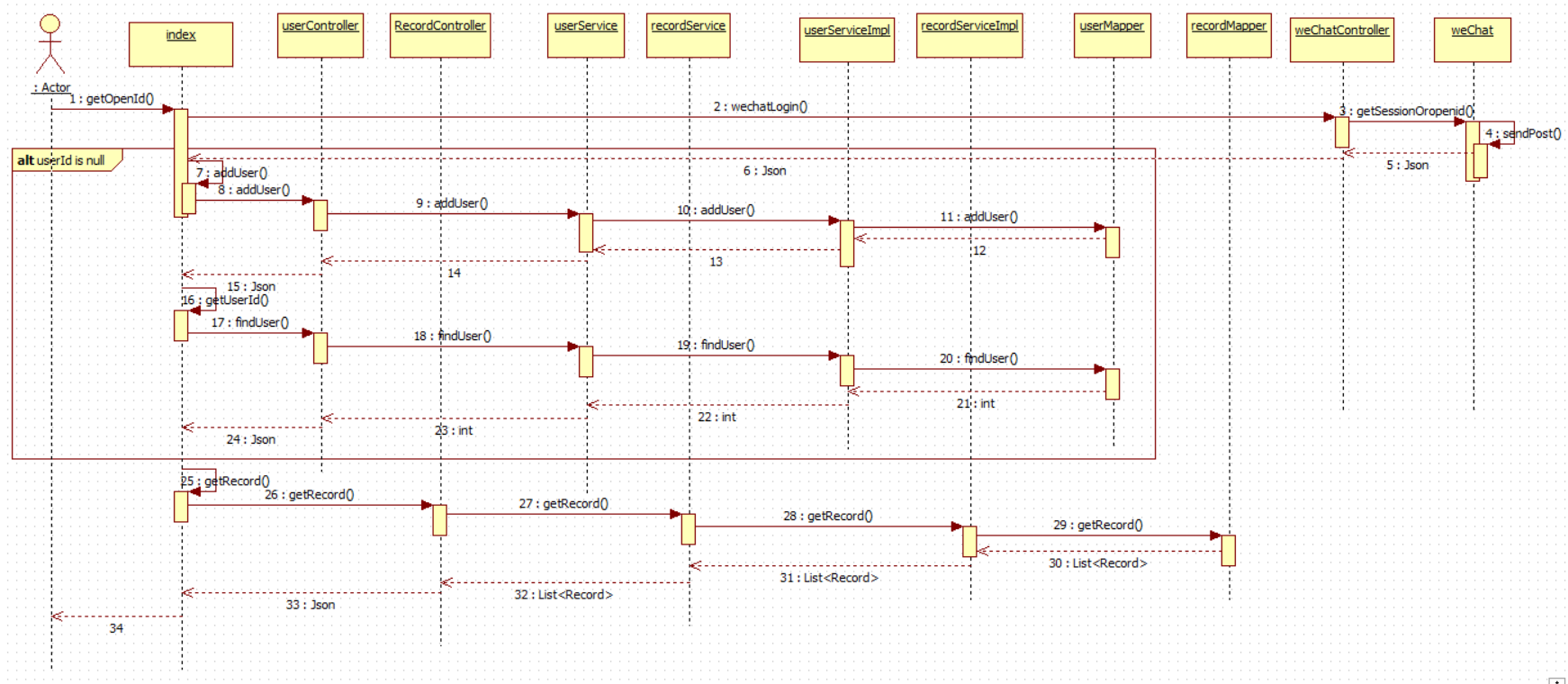
4.1、图像识别查询垃圾分类：



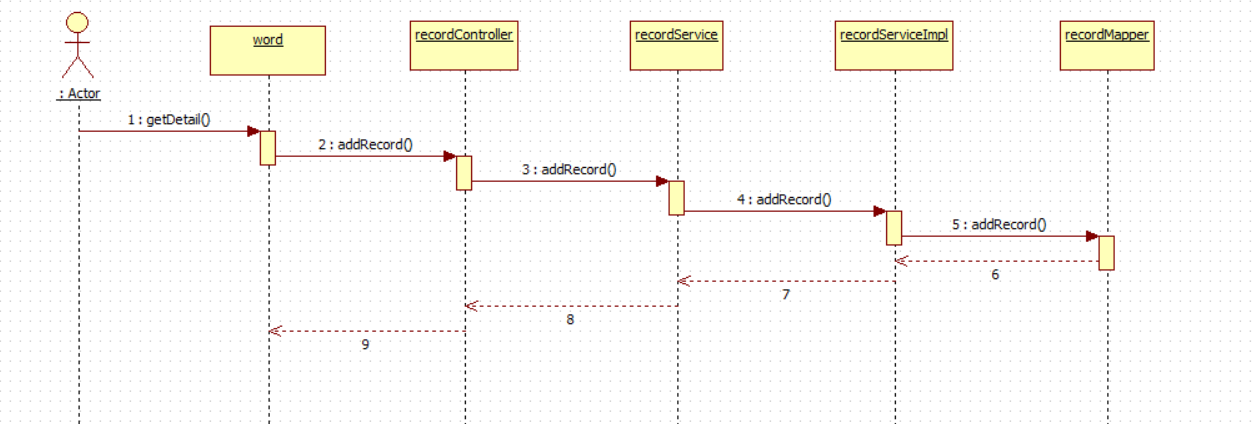
4.2、 文字模糊查询查询垃圾分类：



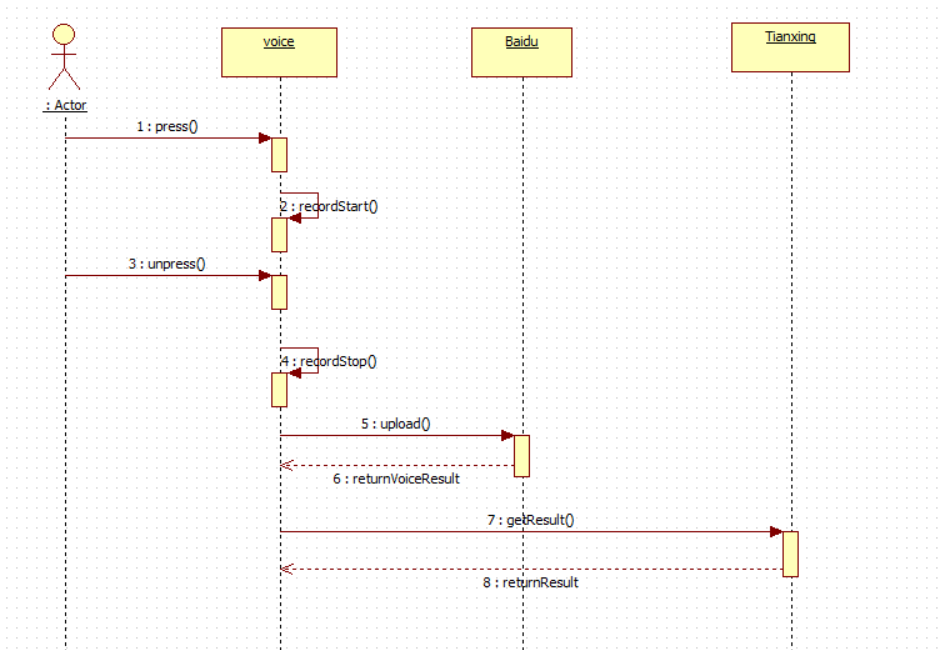
4.3：查询历史记录：



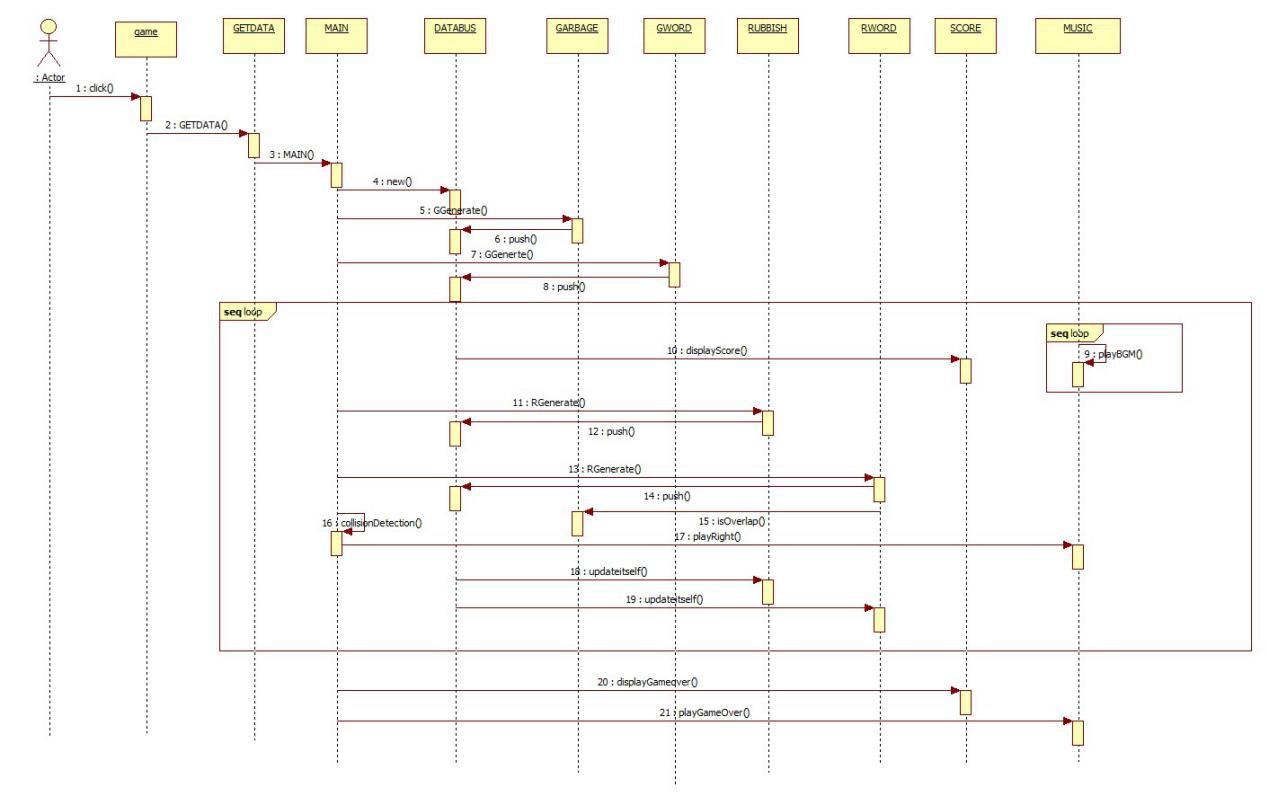
4.4、增加历史记录：



4.5语音识别



4.6小游戏



5、系统非功能设计

主色调：墨绿色

存储空间：因为小程序仅可占用资源少，为了减少系统所占的空间，系统仅会记录最新一次 的暂停游戏的记录，如果再次保存将会覆盖之前的记录。

响应时间：设计缓存页，查询过的垃圾将记住垃圾分类结果，不再调用垃圾分类接口查询。

可维护性：软件模块化设计以及注意代码规范性、类设计的低耦合。

可靠性：在识别物体时可能不准确，或者准确率难以提高，尽量返回多个识别结果供以选择 来提高结果的有效性。