## 垃圾分类小程序概要设计说明书

1. **概述**
2. 系统简介：

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。

1. 软件设计目标：
2. 科普垃圾分类知识
3. 使查询垃圾分类更加快速便捷
4. 使垃圾分类更加具有趣味性，增强人们垃圾分类的观念
5. 参考资料：

（1）《数据库系统概念（原书第六版）》（美）西尔伯沙茨等著 机械工业出版社 2012

（2）《Software Engineering》10th Edition ,Ian Sommerville 机械工业出版社 2017

（3）《Software Modeling & Design （UML,Cases,Patterns & Software Architecture)》Hassan Gomaa机械工业出版社 2017

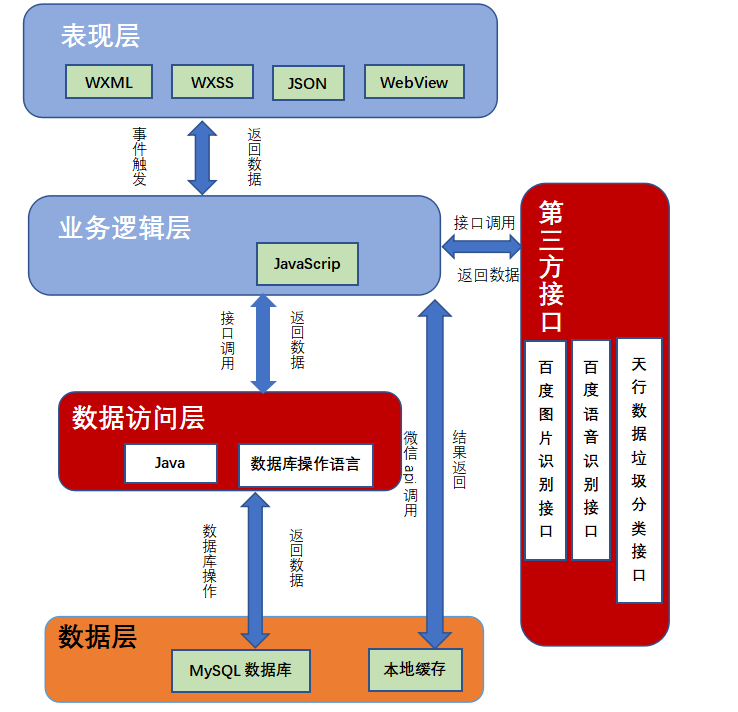
1. 修订版本记录 ：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修改内容 | 修改人 | 日期 |
| Version 1.0 | 概述、系统物理架构、系统数据模型 | 杨官梅 | 2020/03/09 |
| Version 2.0 | 系统逻辑架构、系统肺功能设计 | 文珮全 | 2020/03/11 |
| Version 2.1 | 系统逻辑架构说明 | 文珮全 | 2020/03/18 |
| Version2.2 | 系统物理架构说明、系统数据模型说明 | 杨官梅 | 2020/03/22 |
| Version2.3 | 系统逻辑架构、物理结构、数据模型更改补充 | 文珮全、杨官梅 | 2020/04/04 |

1. **术语表**
2. **设计概述**

系统结构设计

1. **系统逻辑架构（软件的技术架构）：**

系统分为表现层、业务逻辑层、抽象接口层、数据访问层以及数据层五层  
在表现层使用微信官方提供视图层描述语言WXML 、WXSS进行开发，使用JSON文件进行小程序的全局配置，使用WebView进行小游戏的渲染  
在业务逻辑层，使用JavaScript进行小程序/小游戏的后台逻辑控制，业务逻辑层与表现层之间得到交互是通过微信客户端实现的  
第三方接口，此处的接口包括微信官方提供的api、百度的语音识别等api、天性api（垃圾分类接口）以及自己定义的用于访问数据库的抽象接口  
在数据访问层，使用数据库操作语言实现的上述定义的抽象数据访问接口，而访问手机本地缓存则是通过微信提供的api直接调用数据  
在数据层，我们不仅将数据存在服务器端的数据库，也将利于微信分配的每个小程序10M的空间，将部分常用数据存储在手机本地

1. **系统物理架构 （物理设备架构图：软件如果部署到物理设备上）**

该微信小程序涉及的系统包括：租用的服务器（Java后端、MySQL数据库），微信小程序前端(TrashAway)，百度服务器（baidu图像识别、语音识别api）以及天行服务器（天行数据垃圾分类api）、微信服务器。

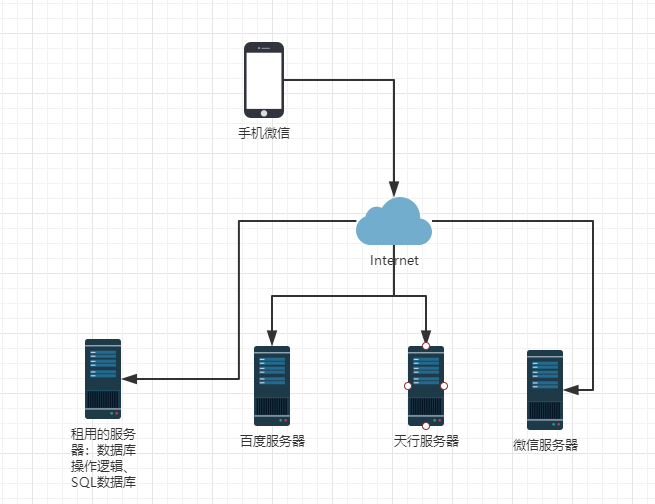
租用服务器配置说明：

系统：win10 64位操作系统

处理器：Intel®Core™i5-8550U CPU @ 1.8GHz 2.00GHz

内存：2.00GB

backEnd（java后端包括数据库操作逻辑）和Mysql数据库都部署在同一台计算机上，java后端（backEnd）通过数据库连接驱动连接Mysql数据库，垃圾分类微信小程序（TrashAway）在使用的过程中通过互联网internet访问到java后端（backEnd包括数据库操作逻辑）、百度图像识别、语音识别api以及天行数据垃圾分类api，微信服务端。



1. **系统数据模型（系统逻辑数据模型）**

3.1、系统逻辑数据模型：

3.1.1、微信小程序的数据模型

实体：用户

属性：用户id(后台服务器用来记录用户的唯一标识)、用户openId（微信用户唯一标识）

主键：用户id

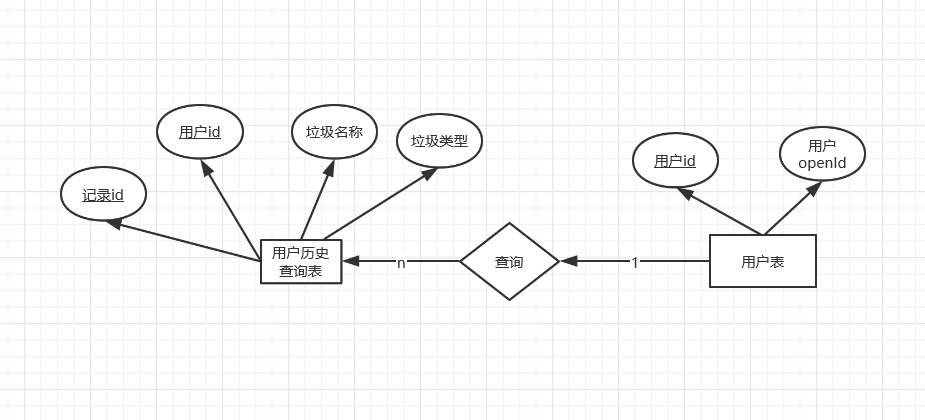
实体：用户历史查询记录

属性：记录id（后台服务器用来记录查询记录的唯一标识）、用户id（历史查询表关联用户表外键）、垃圾名称（查询过的垃圾名称）、垃圾类型（0代表可回收垃圾，1代表有害垃圾，2代表厨余垃圾，3代表其他垃圾）

主键：记录id

外键：用户id（关联用户表）

关系：用户和用户历史查询记录之间为一对多的关系，一个用户有多条历史查询记录



3.1.2、微信小游戏的数据模型

微信端不支持数据库，使用的是文件系统以及缓存机制

用户的最高历史得分：通过独一无二的key键直接对该数据进行改写读取，在缓存中是独立存在的

垃圾的文字以及对应的垃圾分类：通过微信的文件系统，以文件的形式储存在用户的手机当中，通过地址对文件进行定位读写，通过将数据转变为json字符串的形式写进文件，读取出来后再转变为json对象。json对象包括两个属性：垃圾、垃圾分类（以数字的形式进行定义及区分，例如0表示为可回收垃圾）。

3.2、系统数据字典：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据名称 | 数据业务需求描述 | 业务数据类型 | 所属表格 | 对应的字段 | 字段类型 |
| 用户Id | 后台为用户生成的唯一标识 | 自增整型数据 | user | userId | Int(10) |
| 用openId | 微信用户唯一标识 | 50位字符串 | user | openId | Varchar(20) |
| 记录Id | 后台为历史记录生成的唯一标识 | 自增整型数据 | record | recordId | Int(10) |
| 用户Id | 表外键：用户id | 自增整型数据 | record | userId | Int(10) |
| 垃圾名称 | 查询过的垃圾名称 | 50位字符串 | record | trashName | Varchar(100) |
| 垃圾类型 | 垃圾对应的类型：可回收垃圾、有害垃圾、厨余垃圾、其他垃圾 | 1位数字：0代表可回收垃圾，1代表有害垃圾，2代表厨余垃圾，3代表其他垃圾 | record | trashType | Int(4) |

1. **系统接口设计 ：**
2. 查询用户查询垃圾分类历史列表接口
3. 增加历史查询接口
4. 增加用户接口
5. 垃圾名称查询接口：查询存在的垃圾名称，以其作为游戏中掉落的垃圾的素材库
6. 查询垃圾的分类：查询当地的具体的垃圾分类种类，作为游戏中的垃圾桶分类依据

外部系统接口在此处要予以说明。

1. 百度图片识别接口：识别图片
2. 百度语音识别接口：识别语音并返回语音识别的结果
3. 查询垃圾分类接口：查询垃圾分类结果
4. 百度地图定位接口：调用接口判断用户当前所处位置，以此匹配该地区的垃圾分类结果
5. **非功能性设计：**

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。