直男杀手口红试色微信小程序

软件架构设计

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **修订说明** | **作者** |
| <2018-05-01> | <1.0> |  | 曹梦奇，卢锌原，徐迅，王池，王皓轩 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[1引言 1](#_Toc512965408)

[1.1编写目的 1](#_Toc512965409)

[1.2适用范围 1](#_Toc512965410)

[1.3定义 1](#_Toc512965411)

[1.4参考资料 1](#_Toc512965412)

[1.5概述 1](#_Toc512965413)

[2 目前软件系统体系架构 2](#_Toc512965414)

[3 软件系统架构设计目标 2](#_Toc512965415)

[4 建议的软件系统架构 2](#_Toc512965416)

[4.1 概述 2](#_Toc512965417)

[4.2 用例视图 3](#_Toc512965418)

[4.3 系统逻辑视图 5](#_Toc512965419)

[4.4系统运行视图 11](#_Toc512965420)

[4.5系统实现视图 11](#_Toc512965421)

[4.6系统物理视图 13](#_Toc512965422)

[4.7 边界条件设计 13](#_Toc512965423)

[4.8数据管理设计 19](#_Toc512965424)

[4.9 其他设计 20](#_Toc512965425)

软件需求规约

# 1引言

## 1.1编写目的

本文档目的在于为目标读者（软件工程课程指导教师与软件开发组成员）提供在现有系统的基础上，提出对期望系统的设计需求的说明。

## 1.2适用范围

此文档适用于我们开发的“直男杀手”这款软件应用程序，说明了系统期望架构和具体设计需求。

## 1.3定义

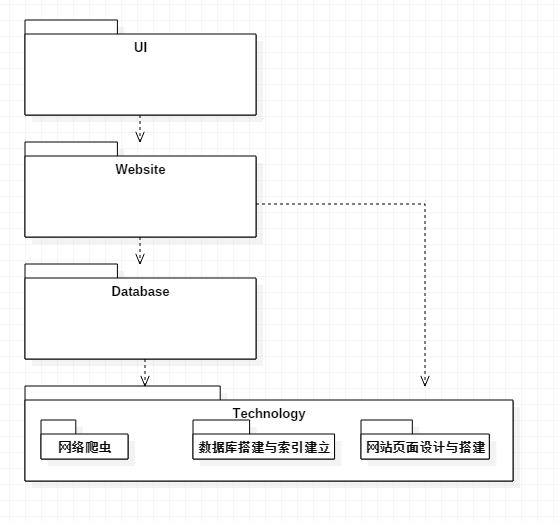
## 1.4参考资料

1. Perlibakas V. Distance measures for PCA-based face recognition[J]. Pattern Recognition Letters, 2004, 25(6):711-724.
2. Bruegge B, Dutoit A H. Object-Oriented Software Engineering Using UML, Patterns, and Java[M]. 清华大学出版社, 2011.
3. 王云, 周伯生. 标准建模语言UML简介[J]. 计算机应用研究, 1999(12):44-49.
4. Roger S. Pressman. Software Engineering: A Practitioner's Approach[M]. McGraw-Hill Book Co, 1982.
5. 赵本本, 殷旭东, 王伟. 基于Scrapy的GitHub数据爬虫[J]. 电子技术与软件工程, 2016(6):199-202.

## 1.5概述

在本文件中，第二部分介绍了我们现有的基于web端的软件系统体系架构，第三部分则阐述了我们期望对现有系统进行的改进目标，第四部分则详细说明了我们期望在微信小程序上实现的软件系统架构。

# 2 目前软件系统体系架构



原有系统架构图

我们原有的系统架构是建立在web端上的，而我们希望能够以微信小程序的形式实现该软件，同时增加web端没有的注册信息功能。

# 3 软件系统架构设计目标

我们软件将会收集用户自己所提供的信息，并将其存入数据库中，并且仅作为推荐算法的依据，不做他用，在最大程度上保护用户隐私。

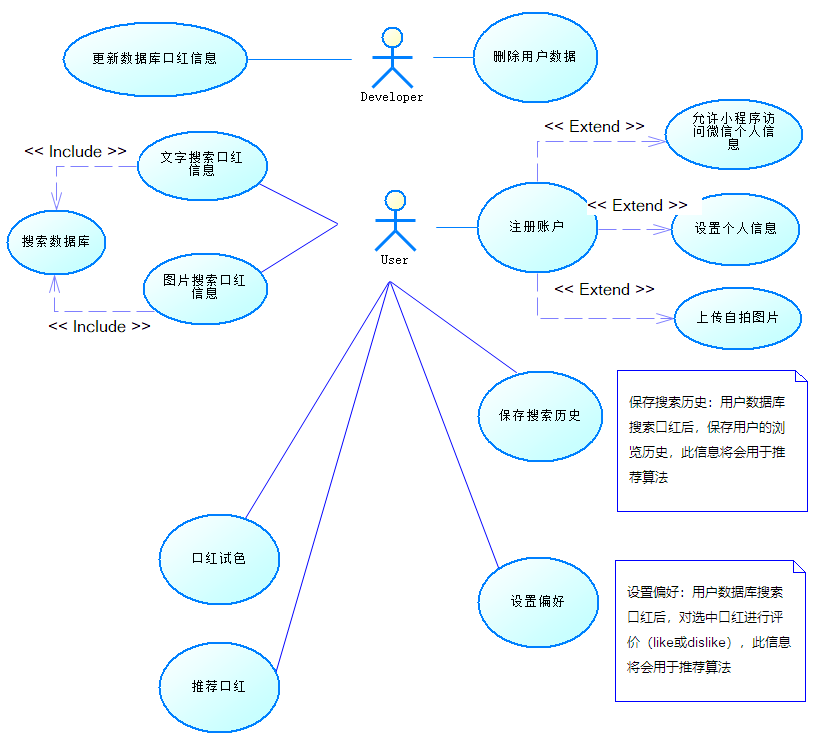
我们软件将会以微信小程序的形式来实现，用户可直接从微信中登陆我们的软件并且能够保留之前的注册信息。

# 4 建议的软件系统架构

## 4.1 概述

软件分为五大子系统：用户注册登录、数据库系统、口红试色系统、口红推荐系统和UI系统。

## 4.2 用例视图



|  |  |
| --- | --- |
| 用例名称 | **注册账户** |
| 参与者 | 用户 |
| 入口条件 | 用户初次进入小程序，允许小程序访问微信个人信息。 |
| 出口条件 | 用户按步骤完成所有设置操作，进入*搜索引擎初始界面*。 |
| 事件流 | 1. 用户进入*个人设置界面*。（页面上会出现相应的提示信息，以及软件的使用目的） 2. 用户选择性别*。* 3. 如果用户为女性，界面动态出现“请小公主上传自拍”button，用户上传图片。 4. 如果用户为男性，界面动态出现“请王子殿下上传公主的自拍”，用户上传图片。 5. 用户为图片中人物选择年龄。 6. 软件提示“设置完成！”进入*搜索引擎初始界面*。 |
| 质量需求 | 推荐口红功能后台实现 |

|  |  |
| --- | --- |
| 用例名称 | **图像搜索口红信息** |
| 参与者 | 用户 |
| 入口条件 | 软件正常工作 |
| 出口条件 | 用户获得搜索结果后离开软件 |
| 事件流 | 1. 用户选择并进入图像搜索界面 2. 用户上传想要搜索的图片 3. 软件解析图片并获取相关信息，调用搜索用例。并获得结果。 4. 将结果呈现在*搜索结果显示界面*上 |
| 质量需求 | 整个搜索过程现在30秒内实现 |

|  |  |
| --- | --- |
| 用例名称 | **文字搜索口红信息** |
| 参与者 | 用户 |
| 入口条件 | 软件正常工作 |
| 出口条件 | 用户获得搜索结果后离开软件 |
| 事件流 | 1. 用户选择并进入文字搜索界面 2. 用户输入想要搜索的文字描述 3. 软件分析文字并获取相关信息，调用搜索用例。并获得结果。 4. 将结果呈现在*搜索结果显示界面*上 |
| 质量需求 | 整个搜索过程现在30秒内实现 |

|  |  |
| --- | --- |
| 用例名称 | **口红试色** |
| 参与者 | 用户 |
| 入口条件 | 软件正常工作 |
| 出口条件 | 用户获得试妆结果后离开软件 |
| 事件流 | 1. 用户选择并进入试妆界面 2. 用户输入想要试妆口红的key值（每一个品牌——色号在本系统中都一一对应一个key值），并上传想要试妆并带有脸部的图片。   3、软件根据key值获得相关口红信息，并分析图片进行模拟试妆  4、将结果呈现在*用户试色界面*上 |
| 质量需求 | 整个试妆现在一分钟内实现 |

|  |  |
| --- | --- |
| 用例名称 | **推荐口红** |
| 参与者 | 用户 |
| 入口条件 | 用户已设置个人基本资料并上传女士人脸图片，进入*搜索引擎初始界面*点击*“查看推荐口红”button。* |
| 出口条件 | 用户获得推荐结果后返回其他界面 |
| 事件流 | 1. 用户点击*“查看推荐口红”button*。 2. 如果用户为男性，考虑到男性对口红的搜索结果缺乏专业性，软件根据用户上传图片的面部特征，在用户数据库中匹配与原图“年龄相近、唇形与面部颜色相似”的女性用户，调用此女性用户的推荐结果，将结果展示到*搜索结果显示界面*上*。* 3. 如果用户为女性，软件跟踪女性的浏览历史与偏好设置，确定受用户欢迎的品牌与价格；软件提取用户上传图片的面部特征，进而推算适合用户的口红色号；以“品牌+价格+色号”为关键词实现推荐算法，将结果展示到*搜索结果显示界面*上*。* |
| 质量需求 | 推荐口红功能后台实现 |

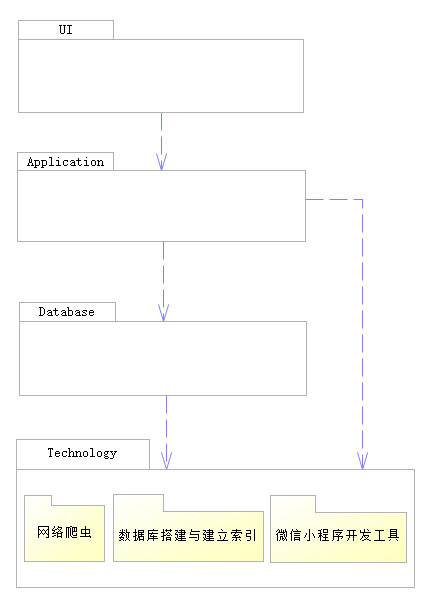
## 4.3 系统逻辑视图

1. 系统架构

本软件系统采用垂直架构，自上而下依次为UI层、应用层、数据存储层与技术核心层。

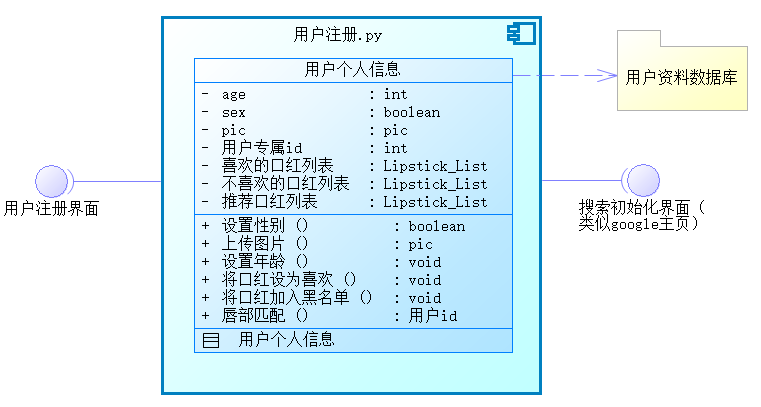
* UI层用于处理应用层请求，利用微信小程序进行.wxml开发。目前可实现的界面有：**用户注册界面**（设置性别、图片、年龄）、**数据库搜索初始界面**（类似www.google.cn的构图，提供文字搜索、图片搜索两种数据库搜索引擎，当用户访问数据库超过一定次数后，在页面上出现“猜你喜欢”button，表明已完成后台推荐搜索算法）、**搜索结果显示界面**（提供“按价格排序”功能，推荐口红列表也通过此页面显示）、**口红信息显示界面**（显示口红的品牌、色号、价格、购买链接；页面带有“like”与“dislike”button用于猜测用户偏好；页面带有“试色”button）、**口红试色界面**（根据用户上传图片进行唇识别与颜色匹配）。
* 应用层通过数据库资源与底层技术进行软件各功能（数据库搜索、口红试色、口红推荐）实现，将接口提供给UI层。
* 数据存储层搭建在云服务器上，依赖最底层技术核心，为应用层的功能实现提供数据库资源。
* 核心技术层利用外部技术与软硬件实现底层技术支持。

利用UML的package diagram给出本软件系统架构图如下：



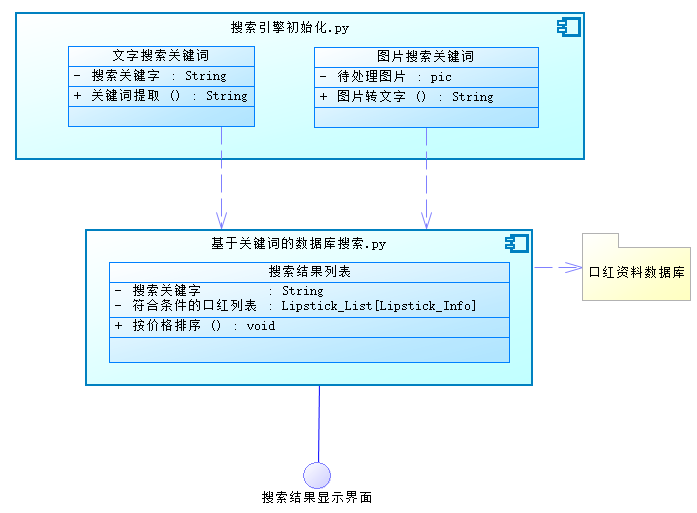
1. 子系统
2. 用户注册登录系统*（已确定开发）*

* 功能：记录每个用户的浏览特点，便于分析用户的喜好，推荐专属口红。
* 组件图：



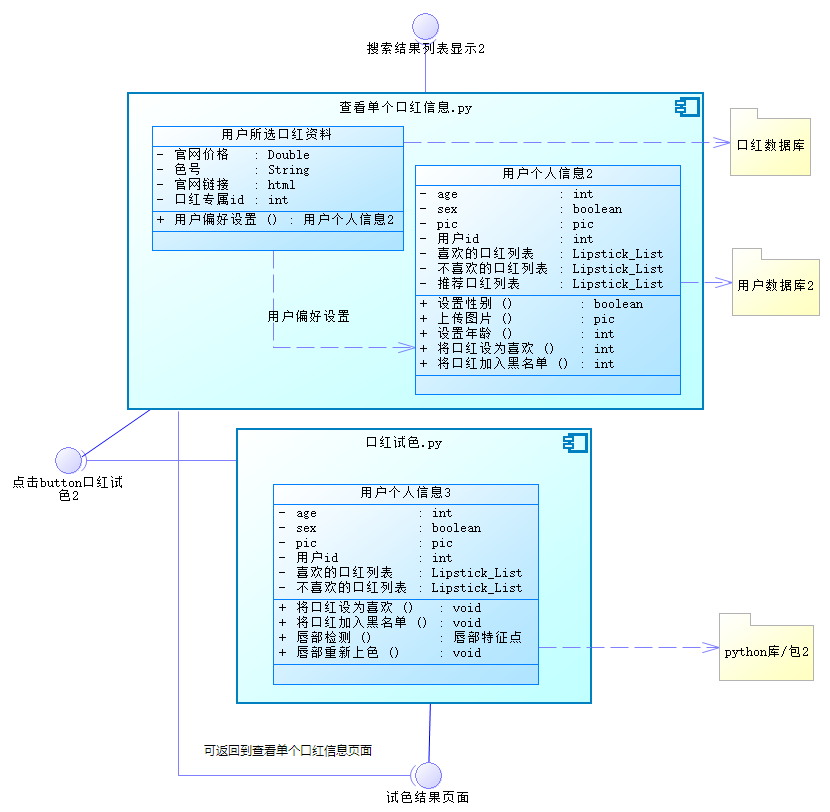
1. 数据库搜索系统*（已确定开发）*

* 功能：按照用户要求，基于关键字（文字，图片转文字）访问口红数据库。
* 组件图：



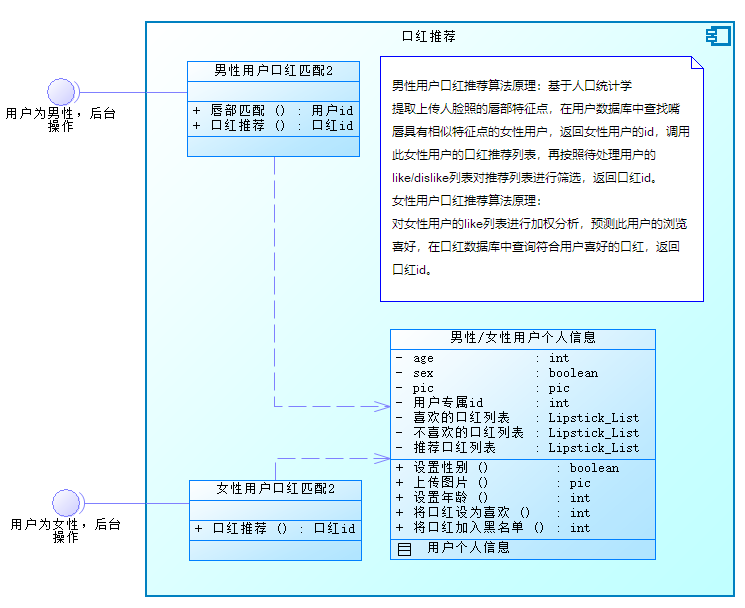
1. 口红试色系统*（已确定开发）*

* 功能：根据用户所选择的口红，对用户上传图片进行唇部识别与重新上色。
* 组件图：

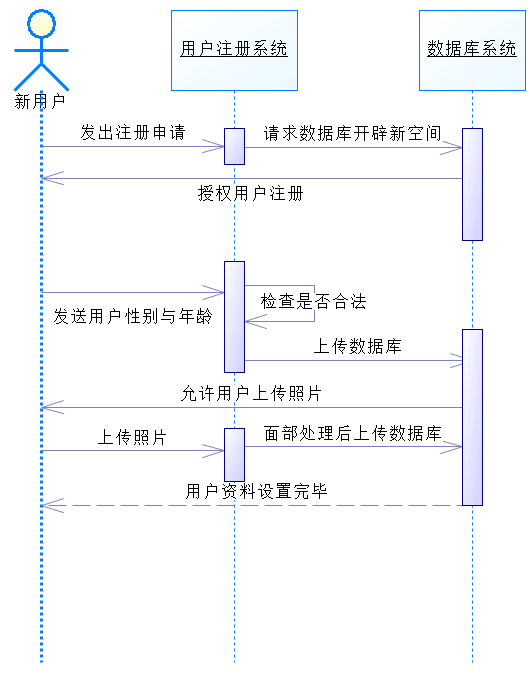


1. 口红推荐系统*（待确定开发）*

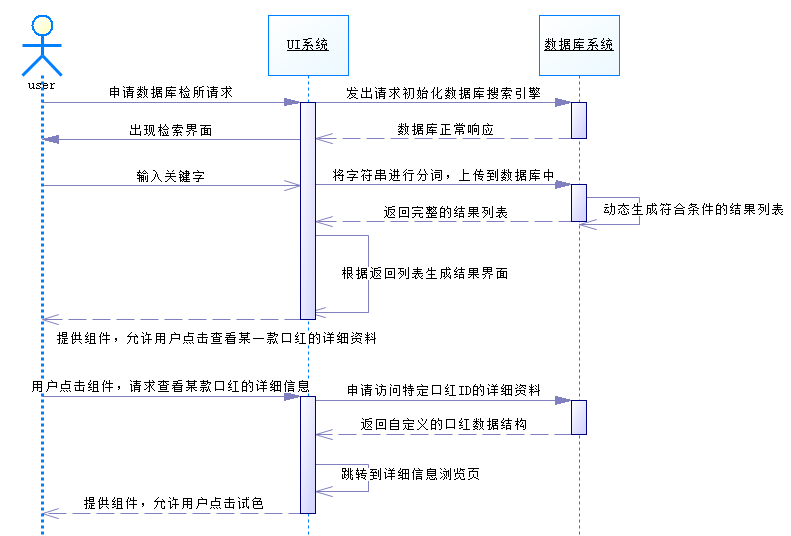
* 功能：根据用户的偏好设置（我们允许用户like或dislike某款口红）
* 组件图：



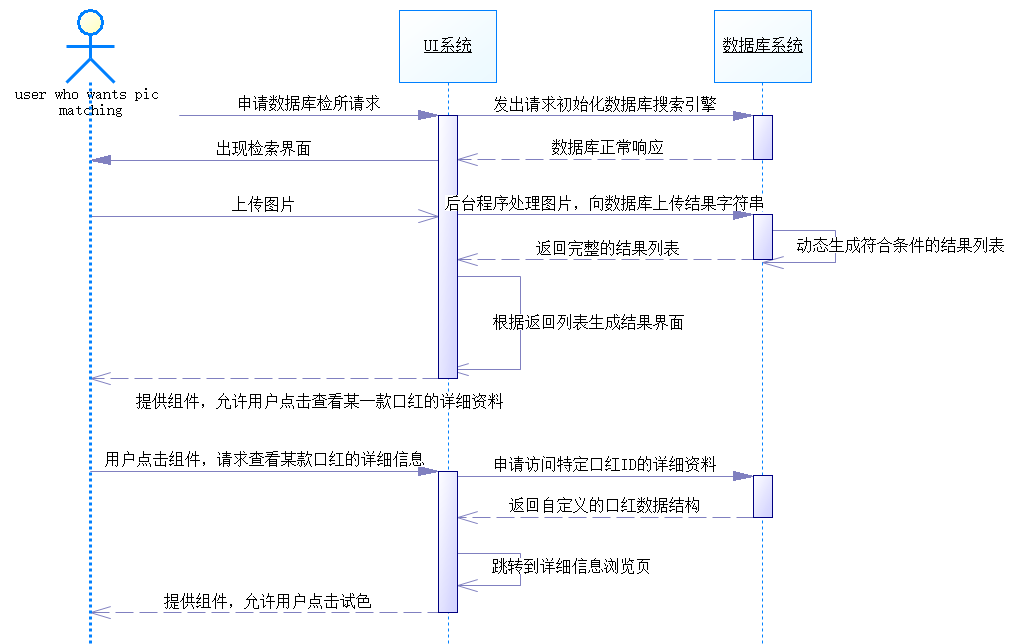
1. 用例实现（时序图）
2. 用例一：注册账户



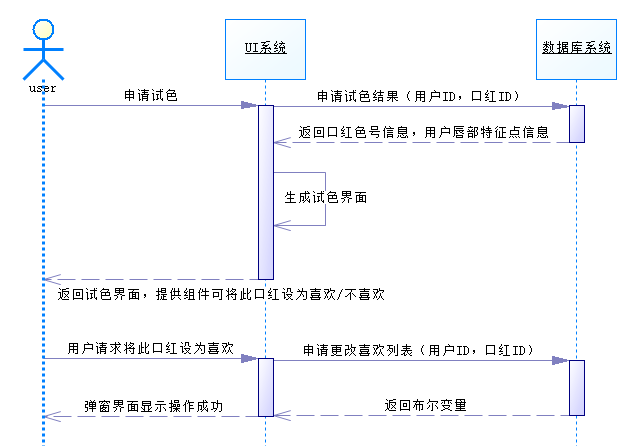
1. 用例二：文字搜索口红信息



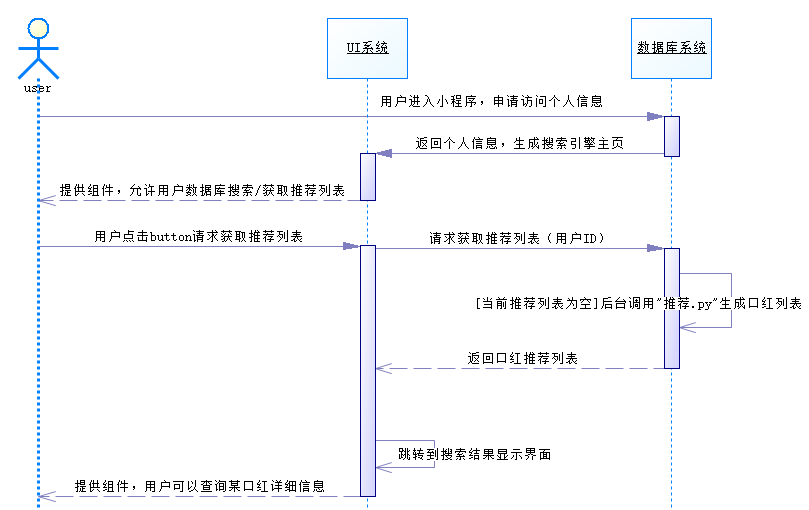
1. 用例三：图片搜索口红信息



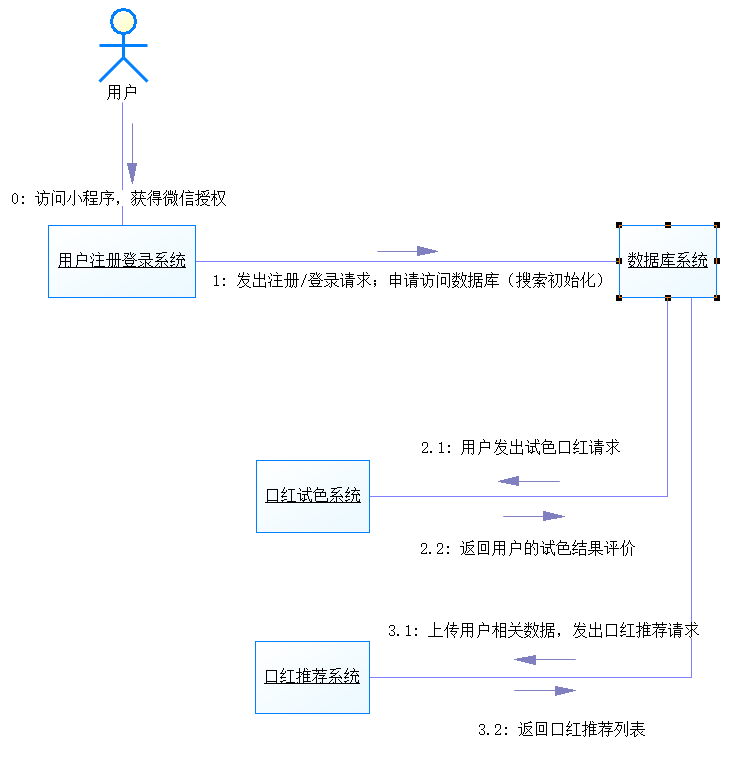
1. 用例四：口红试色



1. 用例五：口红推荐

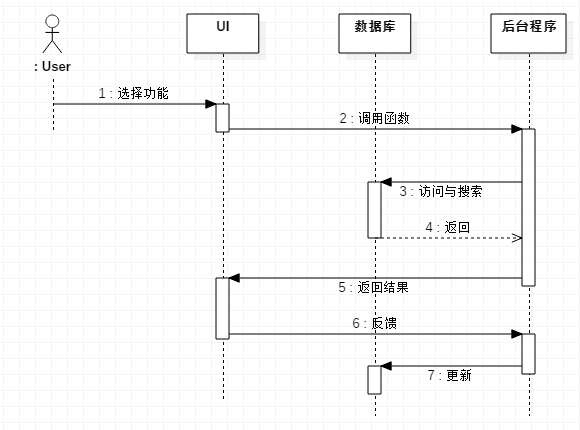


1. 子系统协作



## 4.4系统运行视图

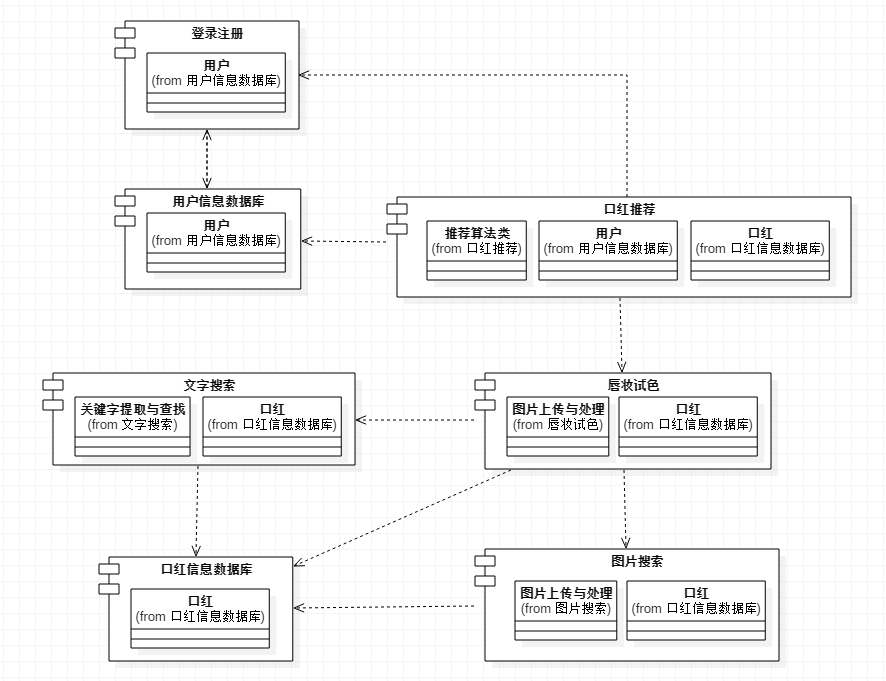
本系统的控制流由事件驱动，具体指用户在软件中选择不同的功能，系统呈现相应的结果，同时根据用户不同的操作，系统可能会对数据库中的信息进行添加和修改。



系统控制流

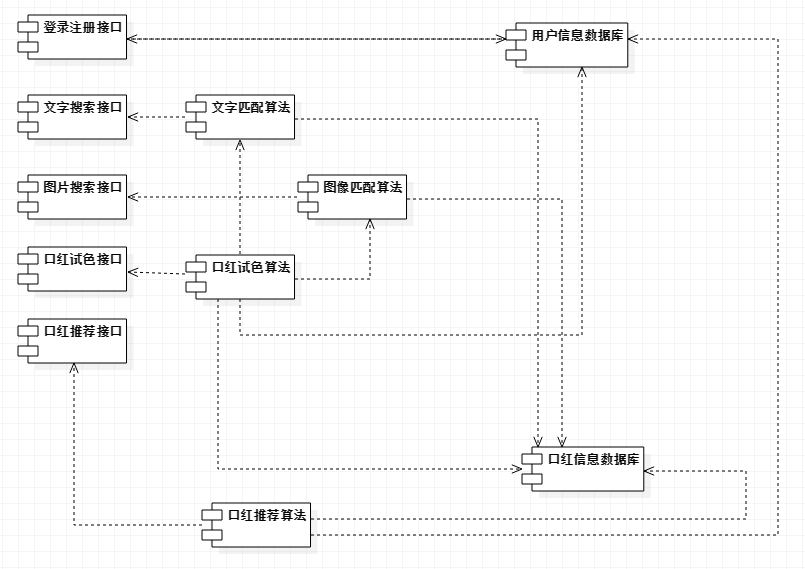
## 4.5系统实现视图

1. 系统开发模型：本系统主要在pyCharm软件和微信小程序服务器环境下开发，主要使用python语言。开发过程包括数据库建立和登录与搜索算法实现两个部分。前者包括用户信息数据库和口红信息数据库，后者包括登录注册、搜索与试色和口红推荐三个目录，其中搜索与试色包含文字搜索、图片搜索和唇妆试色三个源文件。



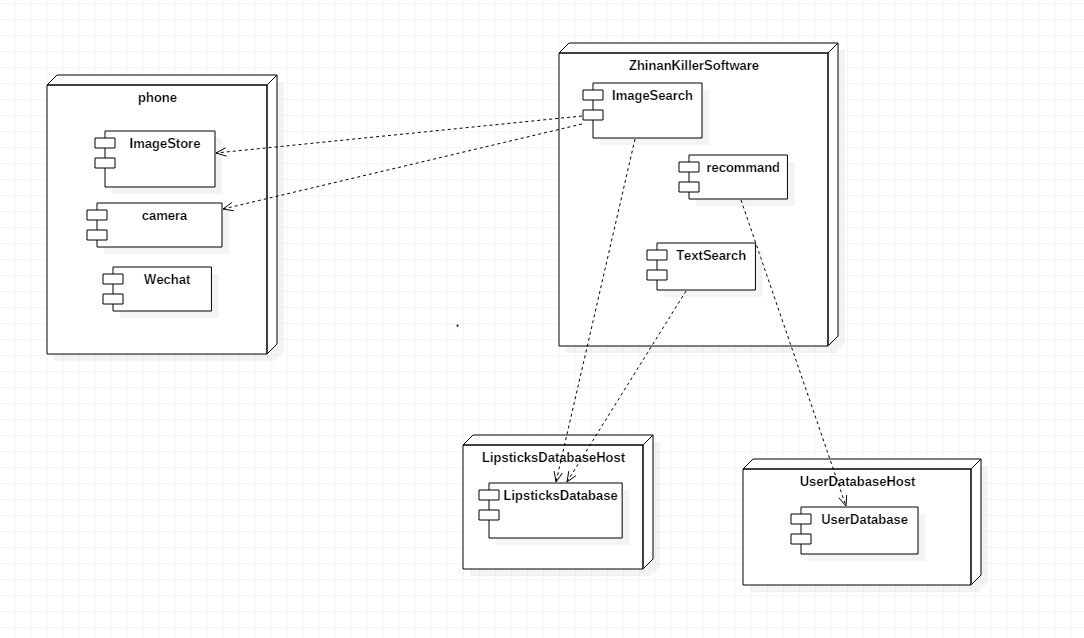
系统源文件组件图

1. 系统实现模型：本系统使用微信小程序作为软件实现平台，用户通过UI提供的接口选择不同功能，程序通过用户选择和数据库信息进行返回结果和更新数据库等操作。



系统各组件依赖关系

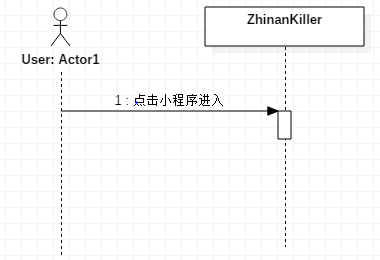
## 4.6系统物理视图



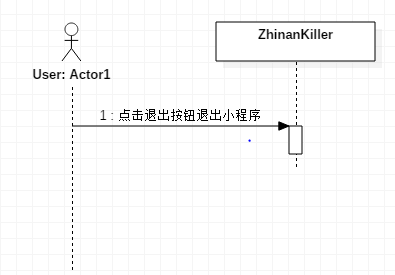
硬件要求：正常智能手机能上微信联网。

## 4.7 边界条件设计

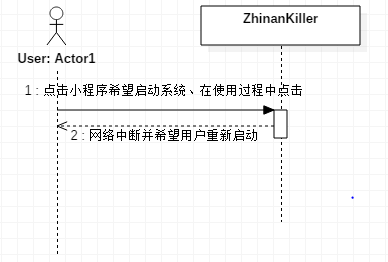
|  |  |
| --- | --- |
| 用例名称 | 启动小程序 |
| 参与者 | 使用者user |
| 入口条件 | 使用者打开微信并处于正常状态 |
| 出口条件 | 使用者登录（注册）成功并开始使用 |
| 事件流 | 1. 使用者打开微信并点击小程序，进入系统 2. 小程序启动并开始进行用户登录（注册） 3. 用户使用小程序 |
| 质量需求 | 无 |



|  |  |
| --- | --- |
| 用例名称 | 关闭小程序 |
| 参与者 | 使用者user |
| 入口条件 | 使用者处于正常使用小程序系统中 |
| 出口条件 | 使用者回到正常的微信聊天界面 |
| 事件流 | 1. 使用者点击退出按钮并退出小程序 2. 使用者回到正常聊天界面 |
| 质量需求 | 无 |



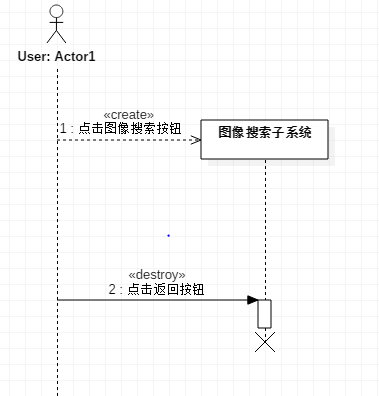
|  |  |
| --- | --- |
| 用例名称 | 网络中断错误处理 |
| 参与者 | 用户user |
| 入口条件 | 用户希望启动小程序同时手机处于断开网络状态或者在使用过程中网络中断 |
| 出口条件 | 用户回到微信聊天界面 |
| 事件流 | 1. 用户点击小程序但网络连接中断或用户在使用过程中网络中断 2. 系统给予反馈，给予退出按钮，用户点击退出 3. 用户回到微信聊天界面 |
| 质量需求 | 无 |



**图像搜索子系统**

|  |  |
| --- | --- |
| 用例名称 | 启动图像搜索 |
| 参与者 | 使用者user |
| 入口条件 | 使用者进入小程序主界面并点击图像搜索按钮 |
| 出口条件 | 使用者开始图像搜索 |
| 事件流 | 1. 用户进入了小程序，主界面点击图像搜索按钮 2. 用户进入图像搜索子系统并开始图像搜索 |
| 质量需求 | 无 |

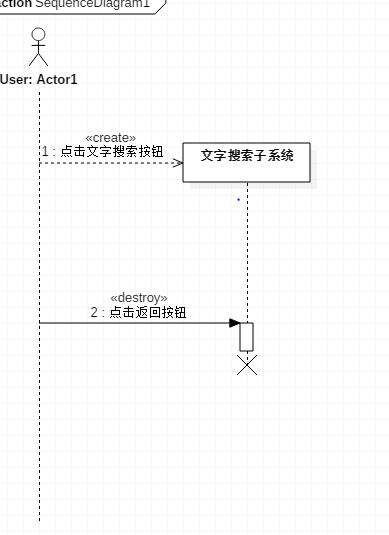
|  |  |
| --- | --- |
| 用例名称 | 关闭图像搜索 |
| 参与者 | 使用者user |
| 入口条件 | 使用者图像搜索完成并点击返回按钮 |
| 出口条件 | 使用者返回主页面 |
| 事件流 | 1、用户点击返回按钮返回到主页面 |
| 质量需求 | 无 |



**文字搜索子系统**

|  |  |
| --- | --- |
| 用例名称 | 启动文字搜索 |
| 参与者 | 使用者user |
| 入口条件 | 使用者进入小程序主界面并点击文字搜索按钮 |
| 出口条件 | 使用者开始文字搜索 |
| 事件流 | 1、用户进入了小程序，主界面点击文字搜索按钮  2、用户进入文字搜索子系统并开始文字搜索 |
| 质量需求 | 无 |

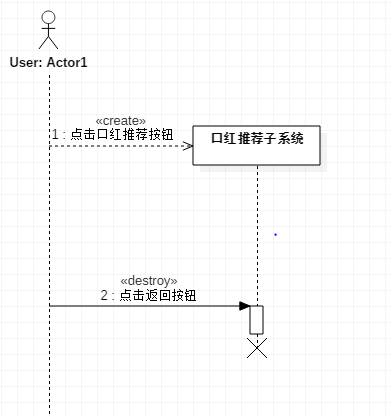
|  |  |
| --- | --- |
| 用例名称 | 关闭文字搜索 |
| 参与者 | 使用者user |
| 入口条件 | 使用者文字搜索完成并点击返回按钮 |
| 出口条件 | 使用者返回主页面 |
| 事件流 | 1、用户点击返回按钮返回到主页面 |
| 质量需求 | 无 |



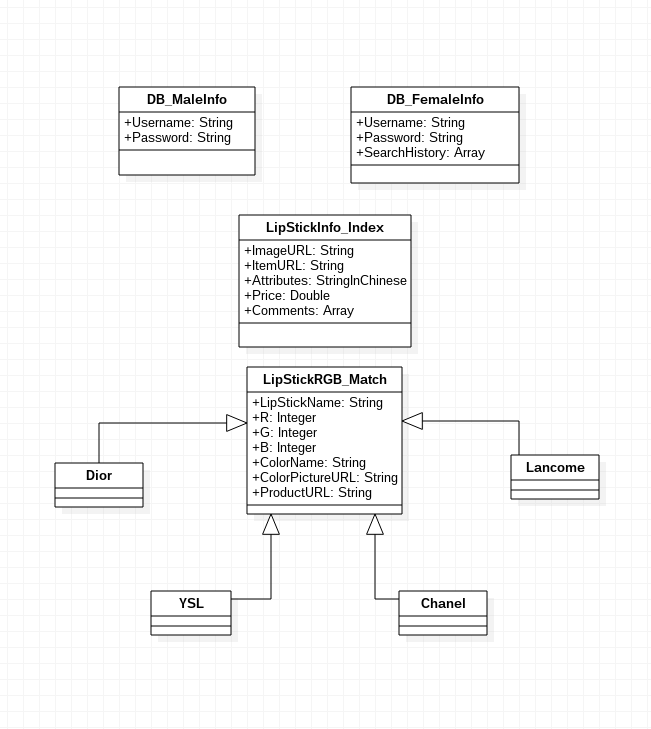
**口红推荐子系统**

|  |  |
| --- | --- |
| 用例名称 | 启动口红推荐 |
| 参与者 | 使用者user |
| 入口条件 | 使用者进入小程序主界面并点击文字搜索按钮 |
| 出口条件 | 使用者开始文字搜索 |
| 事件流 | 1、用户进入了小程序，主界面点击口红管推荐按钮  2、用户进入口红推荐子系统并开始口红推荐 |
| 质量需求 | 无 |

|  |  |
| --- | --- |
| 用例名称 | 关闭口红推荐 |
| 参与者 | 使用者user |
| 入口条件 | 使用者口红推荐完成并点击返回按钮 |
| 出口条件 | 使用者返回主页面 |
| 事件流 | 1、用户点击返回按钮返回到主页面 |
| 质量需求 | 无 |



## 4.8数据管理设计



我们需要保存的数据类型有两种：用户的数据与我们本地的数据。用户的数据包括用户的用户名与密码，以及每一位女性用户的部分搜索记录。我们之所以会在SQL数据库中保存这些用户的信息，是因为SQL数据库具有严谨的格式化的结构，能够很好地帮助我们将用户的数据进行有效的管理，并且数据库的访问速度很快，有利于我们对用户的操作进行很快的回应，改善用户体验。

本地的数据包括我们获取的口红产品信息，（包括源文件以及根据这些txt源文件所建立的索引），四种不同品牌口红的产品名称，色号，颜色图片URL以及产品图片URL。这些本地数据都会以文件系统的形式被保存下来。这样保存的原因是：1.虽然读取文件的时间较长，但是我们的数据量较小，故而所需要花费的时间还是很短的； 2. 我们会预先对这些数据进行处理，比如将产品信息处理成索引的形式，大大加快了访问计算的速度； 3. 数据库对图片的保存不友好，故而采用数据库不是一个很好的想法。

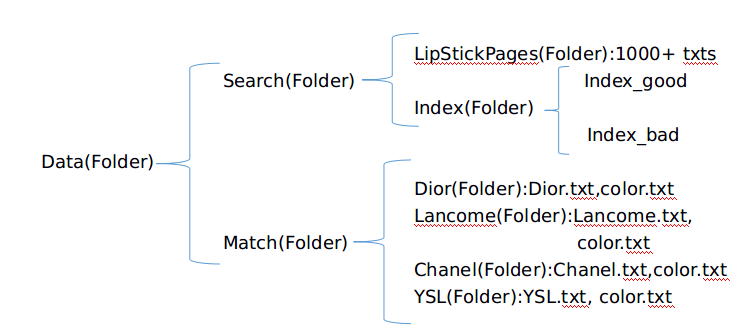
我们选择mySQL数据库作为我们数据库管理系统，女性用户表设计如下所示：

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Username | Password | History |
| 1 | XX | QQQ | [URL1, URL2] |
| 2 | XXX | WWW | [URL1] |

男性用户则如下表所示：

|  |  |  |
| --- | --- | --- |
| ID | Username | Password |
| 1 | XX | QQQ |
| 2 | XXX | WWW |

对于在本地存储的信息，我们采用的时文件系统，其目录结构如下所示：



## 4.9 其他设计

（1）访问控制和安全设计

采用表格列出不同的操作者对不同对象的权限；

|  |  |  |
| --- | --- | --- |
| 操作者 | 对象 | 权限 |
| 用户 | 口红数据库 | 无权限 |
| 用户 | 自己访问的历史记录 | 读写 |
| 用户 | 自己的用户信息 | 可读 |
| 后端数据库管理员 | 用户历史记录 | 可读 |
| 后端数据库管理员 | 口红数据库 | 读写 |
| 后端数据库管理员 | 用户信息 | 无权限 |

描述用户认证的方式；

用户在第一次使用该软件时，会注册账号，建立自己的用户名与密码，之后登陆时进行验证只需要输入自己的用户名与密码即可。

（2）可靠性设计

如果有特定的可靠性要求，给出在可靠性方面采取的特定设计方案。

数据库对于可靠性的要求很高，我们需要对数据库建立备份，使得数据库在受到损伤或信息的丢失时能够被恢复。