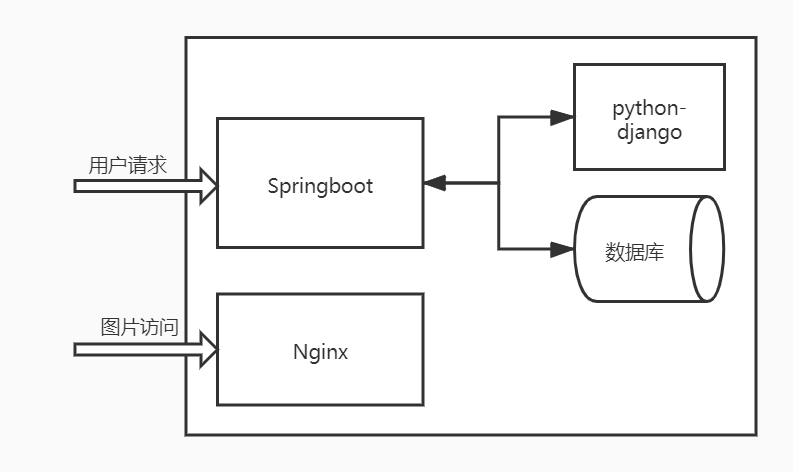
# 服务端

### 服务端的设计与实现

活死细菌在线计算系统服务端负责按照业务相关的需求，实现对用户账号的管理功能，以及对用户上传的活死细菌图片的计算处理功能。服务端采用Spring boot 2.5框架进行开发，使用maven构建项目，数据库ORM框架使用JPA，Reids数据库实现临时数据快速存取以及访问，账号注册采用邮箱激活方式实现，用户图片在线访问采用Nginx 配置静态资源访问实现。该部分主要从总体设计，数据库设计，以及api设计对项目进行阐述。

#### 服务端总体模块设计

服务端总共分为几大模块，首先是Springboot 基本后端服务处理，负责账号的注册登录，以及账号信息管理功能的实现，然后是使用python的 django形式对细胞技术算法进行封装，将该识别模块整合成http api模块对接收的图片进行处理并返回结果。然后是数据库部分，对用户数据进行存储，以及使用nginx的负载均衡技术，实现了用户以http请求获取存储在服务器的图片。服务端整体模块如下所示：



#### 服务端数据库表设计

数据库部分分为两个部分，详细介绍如下：

首先是Redis 键值型数据库，作为一个内存数据库，读写性能好，适合于一些临时的数据，然后又需要频繁读取的数据的存取。因此在该系统中，Redis模块负责以键值对的形式存储用户注册的激活码，并指定其激活码的有效时间，例如： <username,activecode> ,存活时间。

然后使用Mysql 事务性数据库管理该系统所涉及到的用户数据。为了满足该业务需求，该系统设计了用户表以及图片信息表，其中，用户信息表保存用户的基本信息，图片信息表保存用户上传图片的相关内容。表结构内容如下所示：

相关属性状态说明：

active: 默认-1，指定未激活状态，无法登录，大于-1，激活状态

相关状态属性说明：

Status:

0图片未识别，1 图片已进行细胞数识别

Save:

0 图片不在服务器进行永久保存，1图片可进行保存，不作为临时数据。

#### 服务端接口设计

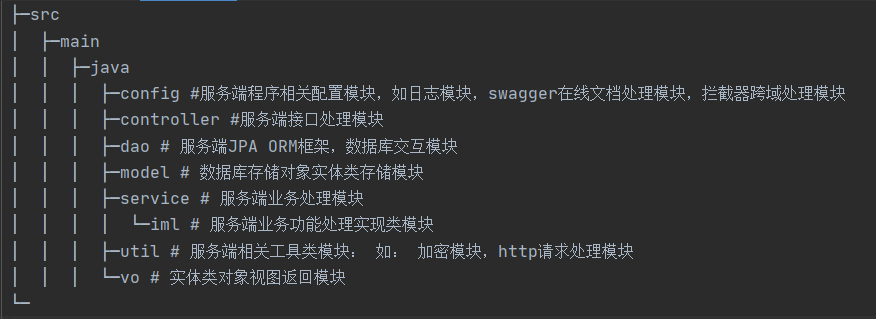
在接口设计方面，基于java servlet作为http请求处理的框架。同时为了方便前后端进行交接，使用swagger2 在线文档生成工具生成在线api文档，方便前端开发人员进行查询。

业务相关接口如下：

|  |  |
| --- | --- |
| 接口路径 | 接口作用 |
| /v2/api-docs | 作为swagger 文档请求接口，获取接口 api文档 |
| /picture/uploadPic | 上传待识别的细胞显微镜图片 |
| /picture/resultList | 获取该用户相关的所有的上传图片 |
| /picture/getPic | 根据图片id获取图片内容及其访问地址 |
| /picture/savePic | 上传pid，指定该图片为保存状态 |
| /user/register | 用户账户注册 |
| /user/login | 用户登录接口 |
| /user/activecode | 获取用户注册激活码 |
| /user/data | 获取某一个用户相关资料 |
| /user/update | 上传资料，更新用户信息 |

#### 服务端程序架构实现

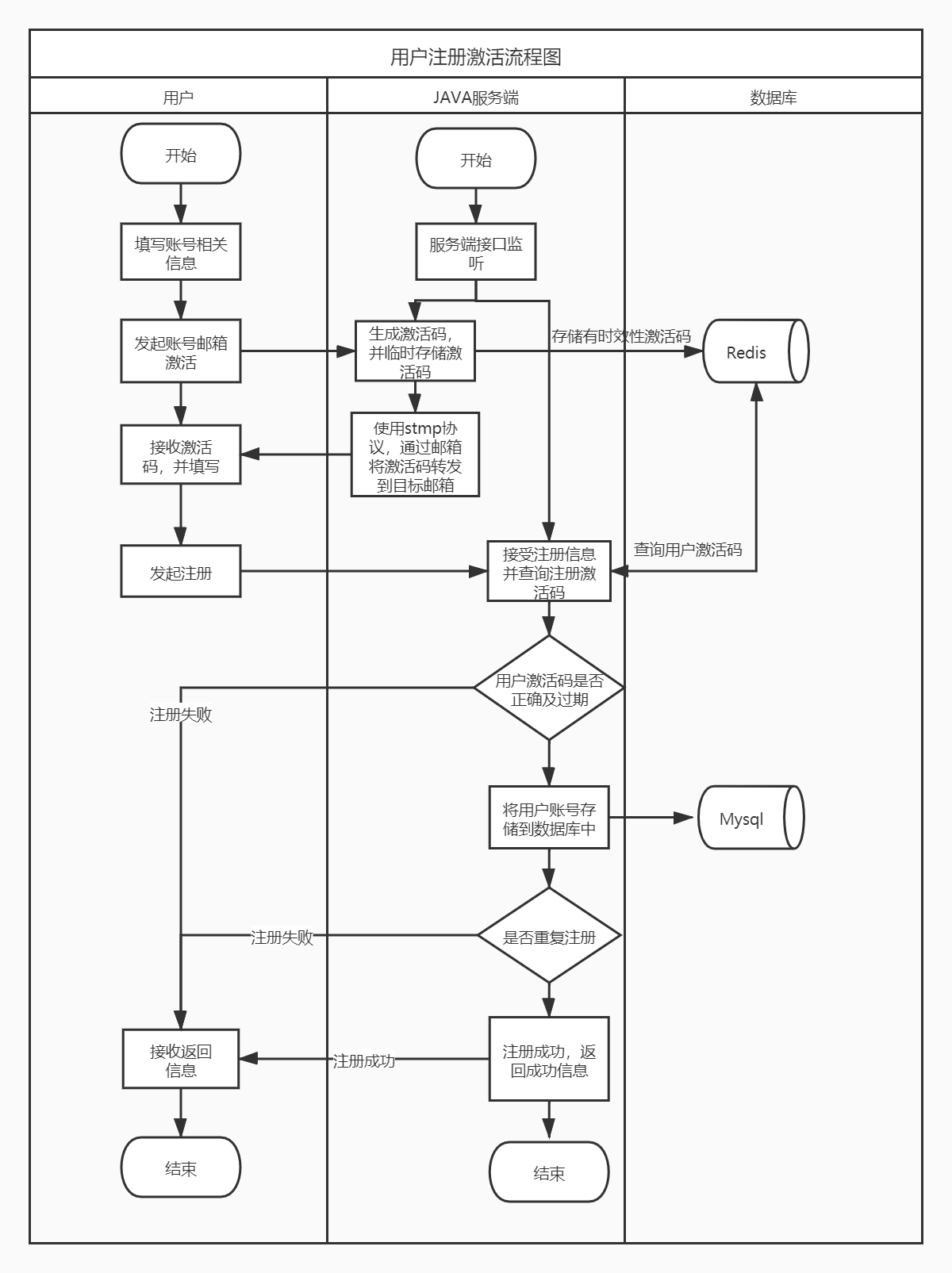
服务端程序分为Java Springboot程序以及基于django封装的识别算法api模块，以下将以文件结构树的形式对程序架构进行阐述：

spring boot程序架构树图

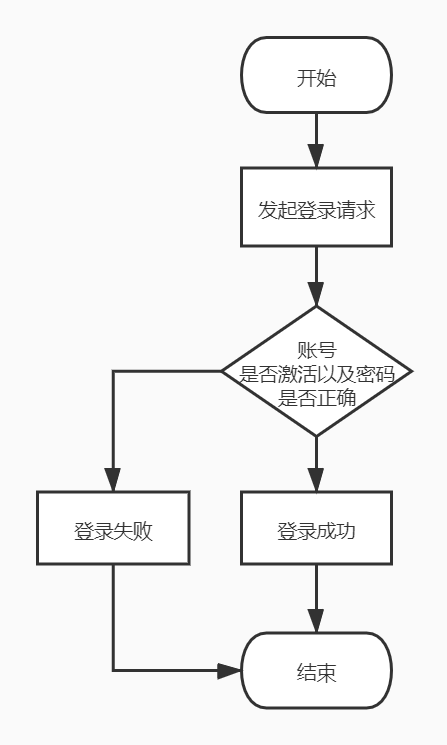
python django架构结构树图:

#### 服务端用户模块实现

用户账号处理主要分为两个部分，一个是注册功能，该功能通过使用邮箱注册的功能实现。Java通过使用java-mail库，指定系统邮箱向用户邮箱转发激活码邮件，然后用户再填写这个激活码以及账号相关信息进行激活，注册功能流程图如下：



在登录方面，服务端接收用户账号密码，处理要判断用户的身份，同时还要判断用户的账号是否激活。登录功能流程图如下：



#### 服务端细胞图片处理模块实现

用户上传图片后，服务端首先将保存在服务器本地，然后将文件的相关信息保存在数据库，在此处由于算法对图片技术需要数秒的时间，以防止单线程情况下还要对图片进行识别造成请求延迟过程。因此，这里采用多线程的形式，以异步的方式让识别进程在后台运行，最后再将计数结果更新到数据库。

此外，该算法使用python语言进行实现，与该系统Java服务端运行环境有冲突，因此，此处将使用django将该算法封装成python http api服务，只负责接收图片，并返回识别值，供Java后端进行图片时调用。

图片上传识别处理功能流程如下所示：