# 错误/异常处理设计

错误和异常的处理和反馈，会一定程度上影响软件的调试效率、性能表现甚至用户的使用体验。因此，落笔云烟在错误/异常处理方面进行了一定程度的规范。

### 错误/异常输出信息

表5.7.1-1 错误/异常输出信息表

|  |  |  |
| --- | --- | --- |
| 错误类型 | 错误位置 | 输出信息 |
| 网络错误 | 图片上传 | 图片上传失败 |
| 图片加载 | 图片下载失败 |
| 登录/注册 | 登录/注册异常，请检查是否已接入互联网 |
| 权限错误 | 登录失效 | 登录已过期，请重新登录 |
| 请求接口数据 | 请先进行登录！ |
| 数据错误 | 图片上传 | 无法检测到文字，请重新拍摄 |
| 用户输入错误 | 注册 | 两次密码输入不一致/密码强度弱 |
| 请求错误接口 | 请求接口数据 | 请求接口错误 |

除在客户端和后台输出错误信息外，对Nginx，flask与客户端后台均配置有错误日志系统；nginx日志用于收集错误请求，flask用于收集模型、算法以及数据库错误信息，方便调试时进行查看。同时通知相关线程释放资源，避免资源浪费；同时返回给接口调用者错误类型信息使其变更操作方式或通知用户，从而保证用户的良好交互性。

### 错误/异常处理对策

**表5.7.2-1 错误/异常处理对策表**

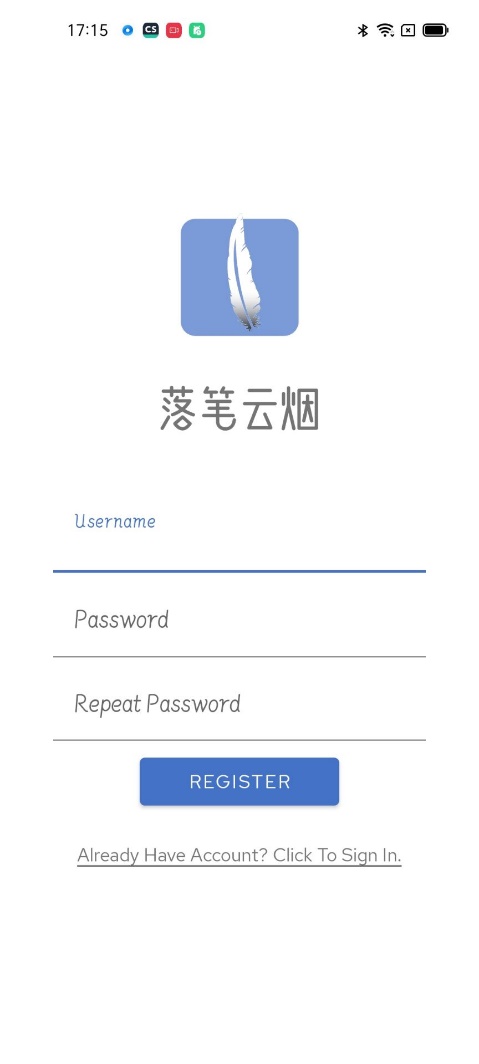
|  |  |  |
| --- | --- | --- |
| 错误类型 | 错误位置 | 处理对策 |
| 网络错误 | 图片上传 | 客户端提示用户接入网络，并抛出异常，集中进行异常捕获，并重启当前活动 |
| 图片加载 |
| 登录/注册 |
| 权限错误 | 登录失效 | 重新获取token |
| 请求接口数据 |
| 数据错误 | 图片上传 | 模型内部抛出异常，并捕获异常进行处理 |
| 数据格式错误 | 服务器抛出异常，并返回500错误 |
| 用户输入错误 | 注册 | 在客户端的数据模型中进行验证，并对用户进行即时提醒 |
| 请求错误接口 | 请求接口数据 | 服务器拒绝访问，返回404错误 |

除上述错误外，当进行其他敏感操作时，可能会引发预料之外的错误，此时处理不当可能会对数据库和用户数据造成较大的影响。因此，需要定期进行冗余备份，并提前设置好后台服务器宕机重启后的实例恢复、数据恢复等操作。

# UI设计

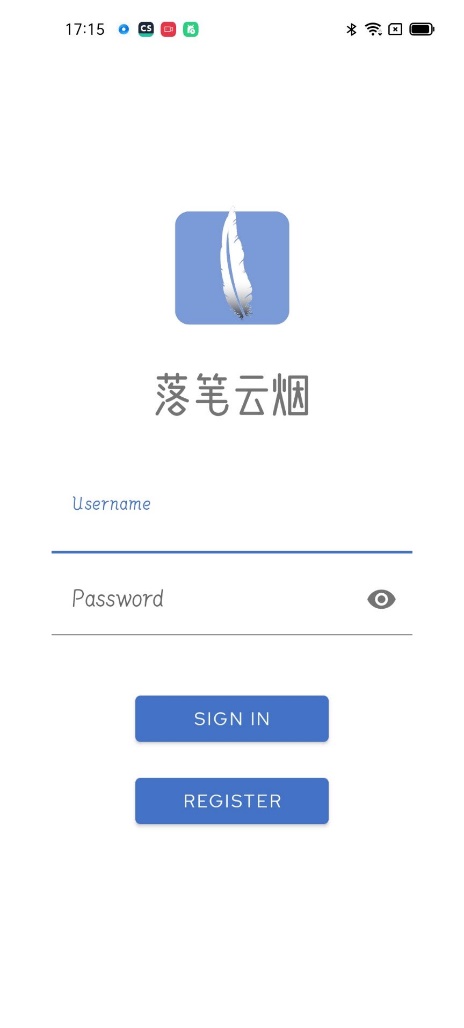
### 注册

为方便用户体验，落笔云烟App支持中英文+数字作为注册用用户名（**大于5位小于18位）**。基于安全考量，落笔云烟App要求用户**使用英文大小写、数字以及特殊符号至少四选三**作为密码**（大于5位小于18位）**，并采用两次输入保用户注册密码正确。



### 登录

用户可以使用刚刚注册的用户名、密码进行登录。出于安全考虑，落笔云烟在注册完成后不会自动填充密码，而要求用户继续输入一次密码，以确保账户安全。登录之后，除非其手动登出，App会在一段时间内保存用户登录状态，下次再次打开时会自动跳过登录步骤。



### 修改密码

进入应用后，用户若想修改密码，可以在侧边栏点击个人头像，然后选择修改密码，输入新密码对密码进行修改。



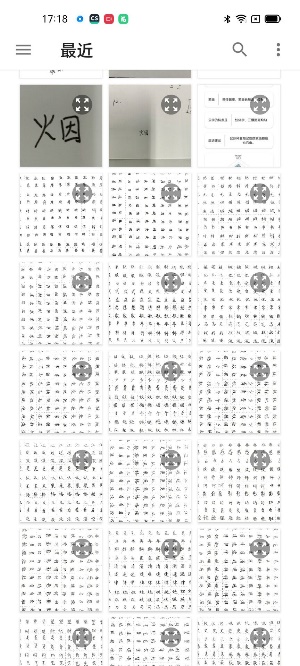
### 修改用户名、邮箱、手机

同时，进入应用后，用户若想修改用户名、邮箱、手机号等个人信息，都可以参照上述操作流程，在侧边栏点击个人头像，然后选择修改用户名、邮箱，输入新用户名、邮箱进行修改。



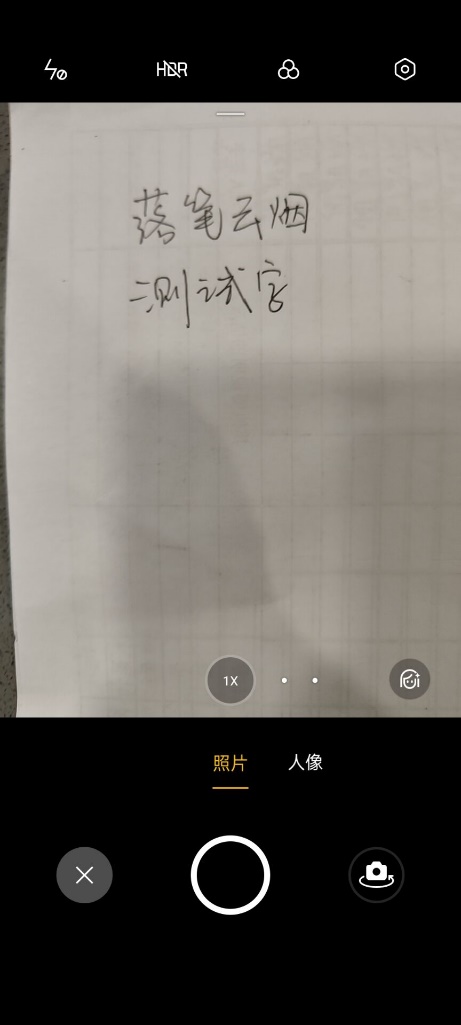
### 修改用户头像

在点击进入个人信息修改界面后，点击个人头像，可以调起系统相册，选择新的图片对当前头像进行修改。



### 拍照手写识别

用户可以点击主页的“拍照体验”按钮，直接调起系统相机对想要进行评测的手写内容进行拍照。



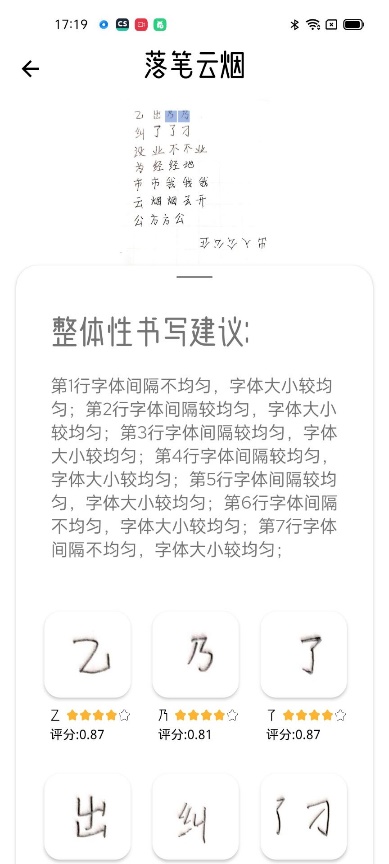
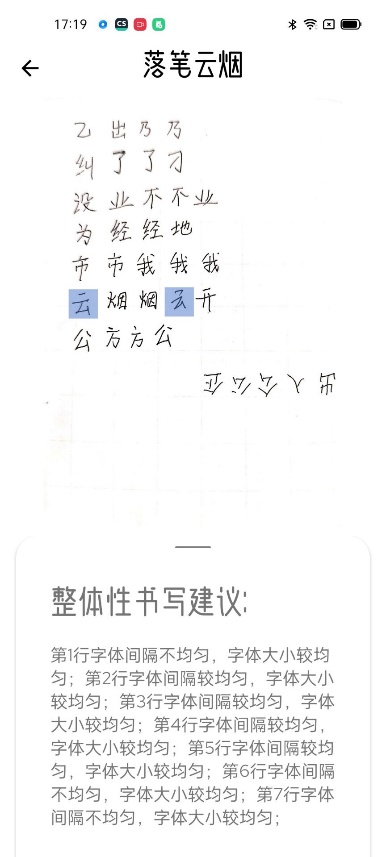
### 从相册选择图片识别

作为替代选择，用户可以点击主页的“从相册选择”按钮，调起系统相册选择相册中想要进行评测的手写内容图片。



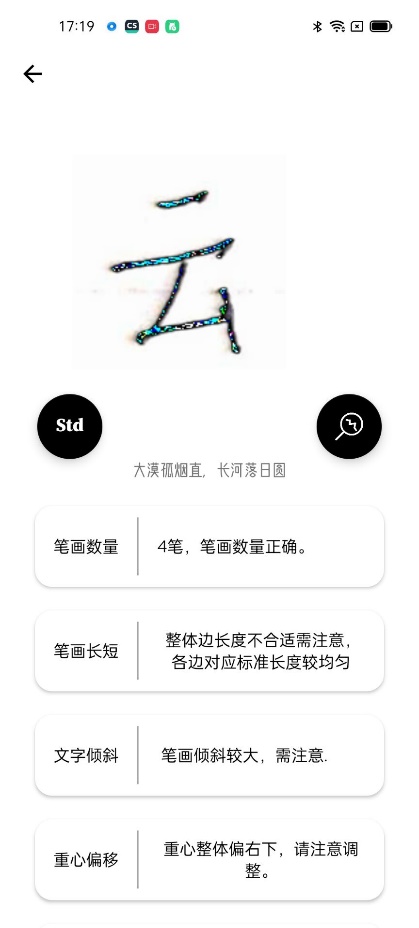
### 整体书写分析

从相册选择或者拍照之后，用户将进入整体书写分析界面，用户可以通过该界面获取整体书写情况评价，并选择下方识别出的不同单字，点击上方选择特定字的不同单字可以进入单字分析。



### 单字书写分析

用户进入单字书写分析后，可以具体查看系统对某个单字进行的详细分析评价结果，查看多维度分析雷达图，以及进行以下交互：



* **显示文字骨架**

点击右侧按钮可以显示App识别出的文字骨架，用户可以借此直观查看到文字的骨架信息，对文字的结构特点产生整体把握。再次点击可以关闭。



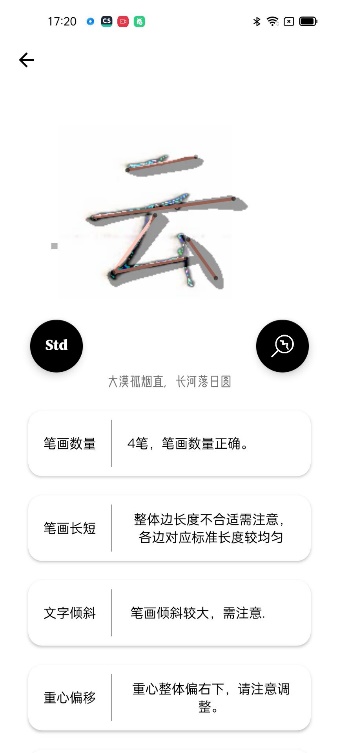
* **显示标准字图片**

点击左侧按钮可以显示当前字体的标准字图片，用户可以借助该功能将手写字与标准文字进行对比，迅速找出手写字的不足之处。再次点击可以关闭。



* **骨架渐变**

在开启文字骨架的前提下，**双击图片**可以将用户手写字骨架渐变成标准字骨架。借助动画可以更为直接地指导用户进行笔画的调整和间架结构不足之处的改进。



* **笔画分析**

在开启文字骨架的前提下，**单击**某个笔画的骨架，该笔画将高亮，在下方可以显示出对于该笔画的具体分析结果。



### 每日推荐

用户可以单击首页的每日推荐按钮，系统将根据用户特征信息给出每日10个推荐单字，用户可以点击进入单字查看书写指导，并直接点击指导卡片右上角的相机按钮调起系统相机对该字进行拍照评测。



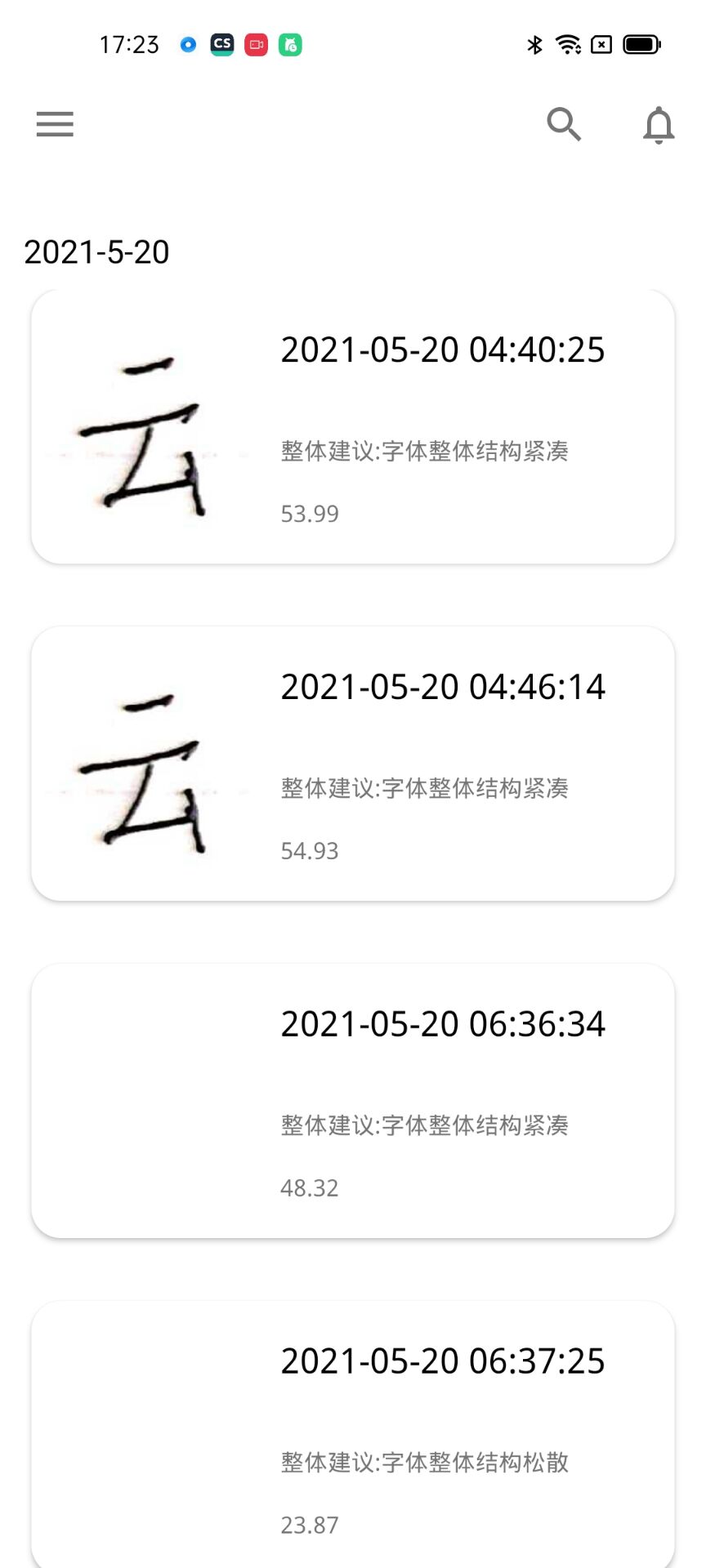
### 趣味天梯

用户可以单击首页的趣味天梯按钮，从而查看系统提供的难度从浅入深的若干关卡（目前设置有三个天梯段位），当用户将某段位下的所有字练习完成后，将获得该段位的成就，并开启下一段位的练习权限。



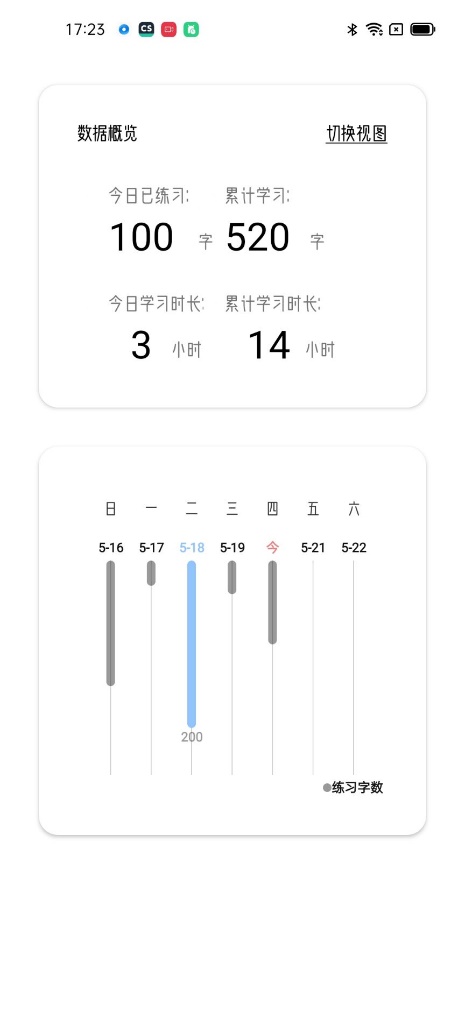
### 历史记录

用户可以点击首页的历史记录按钮，查看自己曾经进行的评测记录。点击某条历史记录可以直接重新进入该次评测的流程，方便用户随时调阅自己的历史书写情况。



### 深度追踪

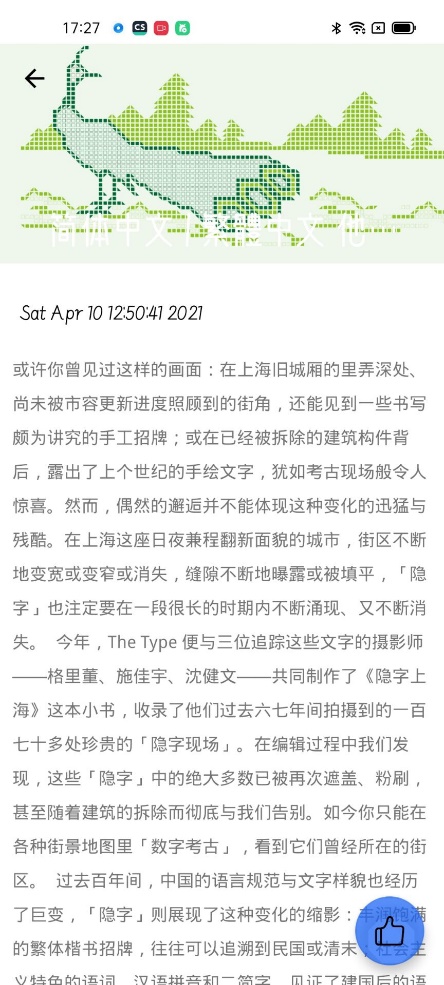
用户可以点击首页的深度追踪按钮，查看自己的练习时间、练习数量统计信息，以及每日的签到信息等，方便用户了解自己每日练习情况；点击某一天可以查看自己该天的练习字数；点击**切换视图**按钮可以切换周、月视图，方便用户查看自己的长期的练习情况





### 文字故事

用户可以在首页下滑查看文字故事，了解文字的相关知识以及历史典故等。



### 通知消息

用户可以点击右上角的消息按钮或者滑出侧边栏，查看系统通知。

### 选择字体

用户可以打开侧边栏，点击**选择字体**按钮，重新选择自己想要练习的标准字体，以满足不同用户的练习需求。



### 退出登录

用户可以打开侧边栏，点击退出登录，即可退出当前账号。



# 数据管理

### 静态数据

静态数据，又称“横截面数据”，是指在运行过程中主要作为控制或参考用的数据,它们在很长的一段时间内不会变化，一般不随运行而变。在此，我们用到的静态数据主要分为以下两方面：

**表4.1.1-1 静态数据表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 名称 | 设置值 | 定义 | 格式 | 类型 |
| 服务器操作部分 | 模型地址 | /model\_path/.. | 用于保存训练后的模型的地址 | String | String |
| sql | ‘select \* from ...’ | 数据库操作的固有命令，用于直接处理信息 | String | String |
| finalPath | /image/../.. | 用于保存用户上传的图像信息 | String | String |
| …… |  |  |  |  |

### 动态数据

**表4.1.2-1 动态数据表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 名称 | 定义 | 格式 | 类型 |
| 服务器操作部分 | 用户个人信息 | 用于保存用户个人设置及隐私信息 | struct | struct |
| 服务器流量 | 用于保存当日访问人数 | integer | integer |
| 用户字体分数 | 用于保存用户拍照字体历史得分数据 | double | double |
| 用户天梯分数 | 用于保存用户天梯得分数据 | String | String |
| …… |  |  |  |

### 数据词典

**表4.1.3-1 用户信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 允许空值 | 默认值 |
| id | 用户id | int(10) UNSIGNED | No |  |
| name | 用户名 | varchar(255) | No |  |
| email | 用户邮箱 | varchar(255) | No |  |
| email\_verified\_at | 邮箱验证时间 | timestamp | Yes |  |
| password | 用户密码 | varchar(255) | No |  |
| remember\_token | 找回密码令牌 | varchar(100) | Yes |  |
| created\_at | 创建时间 | timestamp | Yes |  |
| updated\_at | 更新时间 | timestamp | Yes |  |
| avatar\_url | 头像地址 | varchar(255) | No | /avatar/default\_avatar.png' |
| phone | 手机号 | varchar(255) | No |  |
| birthday | 生日 | date | No | 1901-01-01' |
| gender | 性别 | tinyint(1) | No | 0 |
| structure | 用户字体布局得分 | int(11) | Yes |  |

表4.1.3-2 用户上传字体信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 允许空值 | 默认值 |
| word\_id | 字体id | int(10) UNSIGNED | No |  |
| user\_id | 对应用户id | int(10) UNSIGNED | No |  |
| user\_img\_path | 用户字体图像存储 | varchar(255) | No |  |
| grade | 用户得分 | double | No | 0 |
| date | 字体得分日期 | date | No | 1901-01-01' |
| YB | 运笔得分 | int(11) | No | 0 |
| DX | 大小得分 | int(11) | No | 0 |
| JG | 结构得分 | int(11) | No | 0 |
| ZX | 中心得分 | int(11) | No | 0 |
| JD | 角度得分 | int(11) | No | 0 |
| style | 该字体风格 | int(11) | Yes |  |
| img\_path | 标准字体图像存储 | varchar(255) | No |  |

表4.1.3-3 天梯信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 允许空值 | 默认值 |
| user\_id | 用户id | int(10) UNSIGNED | No |  |
| time | 赛季时间 | varchar(255) | No |  |
| point | 用户该赛季点数 | int(11) | No | 1200 |

表4.1.3-4找回密码表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 允许空值 | 默认值 |
| email | 验证的邮箱 | varchar(255) | No |  |
| token | 找回密码令牌 | varchar(255) | No |  |
| created\_at | 创建时间 | timestamp | Yes |  |

# 框架

落笔云烟后端利用Flask+Celery+Redis+MongoDB组合作为业务处理逻辑，采用Nginx+Gunicorn进行部署，使用pytorch进行模型的构建。

* 采用Flask的原因

1. Flask框架为轻量级 Web 应用框架，其有着极强的扩展性，对比Django等框架，Flask拥有更高的灵活性适合自己设计代码框架。
2. 由于本项目需要使用多种自实现深度学习模型，使用Python应用框架便于加载模型，结合flask扩展性强的特点可以轻松引入。

* 使用Redis的优势

1. 对于纯静态数据，直接使用URL进行缓存，在之后处理中可直接返回数据，无需访问磁盘。
2. 采用Redis的另一个目的便是作为Celery任务队列的中间件。

落笔云烟前端采用RxJava+Retrofit作为网络请求框架，采用GSON作为数据实体处理库，采用Glide作为图片加载框架，UI采用了swiperefreshlayout、stickyDecoration、calendarview、RollingText、MPAndroidChart等社区组件，MVVM调用到了官方架构组件Lifecycle、Viewmodel、Livedata、Paging、Navigation以及Data。

* 采用Retrofit的优点：

1. 可以配置不同HTTP client来实现网络请求，如okhttp、httpclient等
2. 将接口的定义与使用分离开来，实现结构。
3. 支持多种返回数据解析的Converter可以快速进行数据转换。
4. Retrofit和RxJava集成的很好
5. 因为容易和RxJava结合使用，所以对于异步请求，同步请求也不需要做额外的工作。

* 采用官方MVVM组件库的优点：

1. ViewModel：因设备配置改变导致 Activity 重建时，无需从 Model 中再次加载数据，减少了 IO 操作
2. LiveData：更新 UI 时，不用再关注生命周期问题
3. Data Binding： 可以有效减少模板代码的编写，而且目前已经支持双向绑定