

**简 易 Blog**

软件使用说明书

|  |  |
| --- | --- |
| 学号 | 姓名 |
| 3119305851 | 王超勇 |
| 3119305845 | 何 苗 |
| 3119305832 | 张越皖 |
| 3119305848 | 王嘉康 |

**目 录**

[1.数据库设计 3](#_Toc10867)

[1.1 MongoDB介绍 3](#_Toc6562)

[1.2 MongoDB特点 3](#_Toc21011)

[1.3 软件数据库设计 4](#_Toc21873)

[2.单元测试设计 5](#_Toc18741)

[3.类设计 6](#_Toc10364)

[4. 本地接口设计 8](#_Toc32354)

[5. restful网络接口设计 9](#_Toc30514)

[6. 软件安装及使用说明设计 10](#_Toc23594)

[6.1进入App: 10](#_Toc16957)

[6.2百度AI基础功能: 11](#_Toc1923)

[6.3扩展功能: 13](#_Toc28130)

**1.数据库设计**

**1.1 MongoDB介绍**

MongoDB 是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。他支持的数据结构非常松散，是类似 json 的 bson 格式，可以存储比较复杂的数据类型。MongoDB数据结构是由数据库（database）、集合（collection）、文档（document）三部分组成的。

其中，对于开发人员主要面对的是collection和document。其中数据文档（document）是数据载体，其数据格式定义如：{文档名:文档内容, 文档名:文档内容, 文档名:文档内容,….}

**1.2 MongoDB特点**

(1) 模式自由。支持动态查询、完全索引，可以方便地对文档中内嵌的对象及数组进行检索。

(2)面向文档存储。开发人员可以将现有的序列文档直接导入到MogonDB数据集中。

(3)高效的存储机制。MogonDB对数据并没有特殊的限制，任何数据（包括大型的图片、视频等）保存到MogonDB数据库中，最终都将被转换为二进制数据，从而在一定程度上保证读写的搞笑，以及确保数据完整、安全等。

(4)支持复制和故障恢复。提供了主-从、主-主模式的数据复制及服务器之间的数据复制。

(5)自动分片并支持云级别的伸缩性。支持水平的数据库集群，可以动态添加额外的服务器。

(6)支持Python、PHP、Ruby、Java、C、C#、JavaScript、Perl及C++语言的驱动程序。

**1.3 软件数据库设计**

数据库中有三个Collection，其具体结构如下表所示：

|  |  |
| --- | --- |
| Databases（数据库） | Collection（集合） |
| chatapp | User、friendlist、blog |

对于User、friendlist、blog这三个Collection，其目前为止，存的数据内容如下：

(1)Collection User:

{ "\_id" : ObjectId("5dc6940117d247ce57c8e88a"), "name" : "wcy", "password" : "123456" }

{ "\_id" : ObjectId("5dc6b9633e6c736aa6b19f9c"), "name" : "wjk", "password" : "123456" }

{ "\_id" : ObjectId("5dc7892fa8a1f86d4f189d7e"), "name" : "qwer", "password" : "123456" }

{ "\_id" : ObjectId("5dc78f7da8a1f86d4f189d7f"), "name" : "zyw", "password" : "123456" }

{ "\_id" : ObjectId("5dc78f83a8a1f86d4f189d80"), "name" : "hm", "password" : "123456" }

1. Collection friendlist:

{ "\_id" : ObjectId("5dcbebb69bf8ff808f3b55f5"), "name" : "wjk", "friends" : [ "wcy", "zyw", "hm" ] }

{ "\_id" : ObjectId("5dcbebc69bf8ff808f3b55f6"), "name" : "zyw", "friends" : [ "wcy", "wjk", "hm" ] }

{ "\_id" : ObjectId("5dcbebd49bf8ff808f3b55f7"), "name" : "hm", "friends" : [ "wcy", "wjk", "zyw" ] }

{ "\_id" : ObjectId("5dcab28df59605fc47b7403b"), "name" : "wcy", "friends" : [ "hm", "wjk" ] }

1. Collection blog:

{ "\_id" : ObjectId("5dcd3843d09eef02b14cfff7"), "name" : "all", "record" : [ { "time" : "201911141934", "content" : "时间是1934" }, { "time" : "201911141944", "content" : "时间是1944" } ] }

**2.单元测试设计**

|  |  |  |
| --- | --- | --- |
| 动作 | 理论返回数据 | 实际返回数据 |
| 登录传入错误用户名 | invalidName | invalidName |
| 登录传入正确用户名和密码 | loginSuccess | loginSuccess |
| 账户注册时传入已有用户 | invalidName | invalidName |
| 账户注册时传入有效用户名 | registerSuccess | registerSuccess |
| 人脸注册时 | 返回注册信息 | 返回注册信息 |
| 人脸登录时没有该用户 | 返回信息中score小于80 | 返回信息中score小于80 |
| 人脸登录时有该用户 | 返回信息中score大于80 | 返回信息中score大于80 |
| 点击发表微博 | submitSuccess | submitSuccess |
| 点击修改微博 | alterSuccess | alterSuccess |
| 点击删除微博 | deleteSuccess | deleteSuccess |
| 添加好友：添加不存在的用户 | invalidName | invalidName |
| 添加好友：添加已是好友的用户 | alreadyExistence | alreadyExistence |
| 添加好友：添加不是好友的用户 | addSuccess | addSuccess |
| 删除好友 | deleteSuccess | deleteSuccess |
| 查看好友微博 | 好友微博内容 | 好友微博内容 |
| 修改密码 | alterSuccess | alterSuccess |
| 人脸评测 | 返回人脸评测信息 | 返回人脸评测信息 |
| 多脸检测 | 返回人脸库中所有匹配信息 | 返回人脸库中所有匹配信息 |
| 人脸对比 | 返回相似度 | 返回相似度 |

**3.类设计**

|  |  |
| --- | --- |
| 类名 | 功能 |
| MainActivity | 进入APP的首页面（包括人脸登录，账户登录，注册） |
| AccountLogin | 账户登录页面 |
| Register | 账户注册和人脸注册页面 |
| HomeActivity | 登录成功后的首页面，承载四个碎片 |
| mainFragment | 第一个碎片（默认打开，显示所有人的微博） |
| MyBlog | 第二个碎片（实现发表微博，显示自己的微博） |
| chatContacts | 第三个碎片（好友展示） |
| chatMy | 第四个碎片（我的基本信息，修改密码） |
| friendInformation | 好友信息页面（实现好友删除，进入好友微博） |
| AddFriendActivity | 实现添加好友 |
| AlterBlog | 实现微博修改和删除 |
| FriendBlog | 好友微博展示页面 |
| Logining | 记录当前登录的用户 |
| Utils | 方法类（获取token方法，转化为base64方法） |
| FaceAIActivity | 百度AI功能入口页面 |
| DetectActivity | 人脸检测 |
| FaceAddActivity | 人脸上传 |
| MatchActivity | 人脸匹配 |
| MultiSearchActivity | 多脸匹配检测 |
|  |  |
| JsonAccessToken | 解析获取百度token |
| JsonIdentifyResult | 解析百度返回人脸搜索信息 |
| JsonTypeAddface | 解析百度返回人脸增加信息 |
| JsonFindUserBlog | 解析微博内容 |
| JsonFriends | 解析好友内容 |

1. **本地接口设计**

|  |  |
| --- | --- |
| 动作 | 结果 |
| 登录传入错误用户名 | 提示用户名不存在 |
| 登录传入正确用户名错误密码 | 提示密码不正确 |
| 登录传入正确用户名和密码 | 登陆成功 |
| 账户注册时用户名或密码为空 | 提示对应项不能为空 |
| 账户注册时传入已有用户 | 提示该用户名已存在 |
| 账户注册时传入有效用户名 | 该用户注册成功 |
| 人脸登录时没有该用户 | 提示该用户不存在 |
| 人脸登录时有该用户 | 登陆成功 |
| 人脸注册时用户名为空 | 提示用户名不能为空 |
| 人脸注册时未拍摄照片 | 提示照片为空 |
| 人脸注册时输入用户名并拍摄照片 | 该用户注册成功 |
| 点击刷新微博 | 显示新发表的微博并消除已删除的微博 |
| 点击发表微博时文本框为空 | 提示内容不能为空 |
| 点击发表微博时文本框不为空 | 发表成功 |
| 点击修改微博时文本框为空 | 提示内容不能为空 |
| 点击修改微博时文本框不为空 | 修改成功 |
| 点击删除微博 | 删除该微博 |
| 添加好友：添加不存在的用户 | 提示该用户不存在 |
| 添加好友：添加已是好友的用户 | 跳转到该好友的微博主页 |
| 添加好友：添加不是好友的用户 | 添加好友成功 |
| 删除好友 | 删除该好友 |
| 查看好友微博 | 查看该好友发布的全部微博 |
| 修改密码时文本框为空 | 提示内容不能为空 |
| 修改密码时文本框不为空 | 修改成功 |
| 添加人脸 | 实现拍照、查看相册、将人脸照上传到人脸库 |
| 人脸评测 | 通过拍照检测出人脸的基本信息 |
| 多脸检测 | 检测一张包含多个人脸的照片并与人脸库匹配 |
| 人脸对比 | 导入两张人脸以检测相似度 |

1. **restful网络接口设计**

|  |  |
| --- | --- |
| 接口 | 功能 |
| 114.55.33.227:8000/login | 登录 |
| 114.55.33.227:8000/register | 注册 |
| 114.55.33.227:8000/alterPassword | 修改密码 |
| 114.55.33.227:8000/findFriends | 查找好友 |
| 114.55.33.227:8000/addFriends | 添加好友 |
| 114.55.33.227:8000/deleteFriends | 删除好友 |
| 114.55.33.227:8000/submitBlog | 发表博客 |
| 114.55.33.227:8000/deleteBlog | 删除博客 |
| 114.55.33.227:8000/findBlog | 查找博客 |
| 114.55.33.227:8000/alterBlog | 修改博客 |

1. **软件安装及使用说明设计**

**6.1进入App:**

点击app进入后如图1，点击“人脸登录”进行拍照登录；点击“账号密码登录”跳转到如图二所示界面；点击“注册”跳转到图三所示界面进行注册。

图1 图2

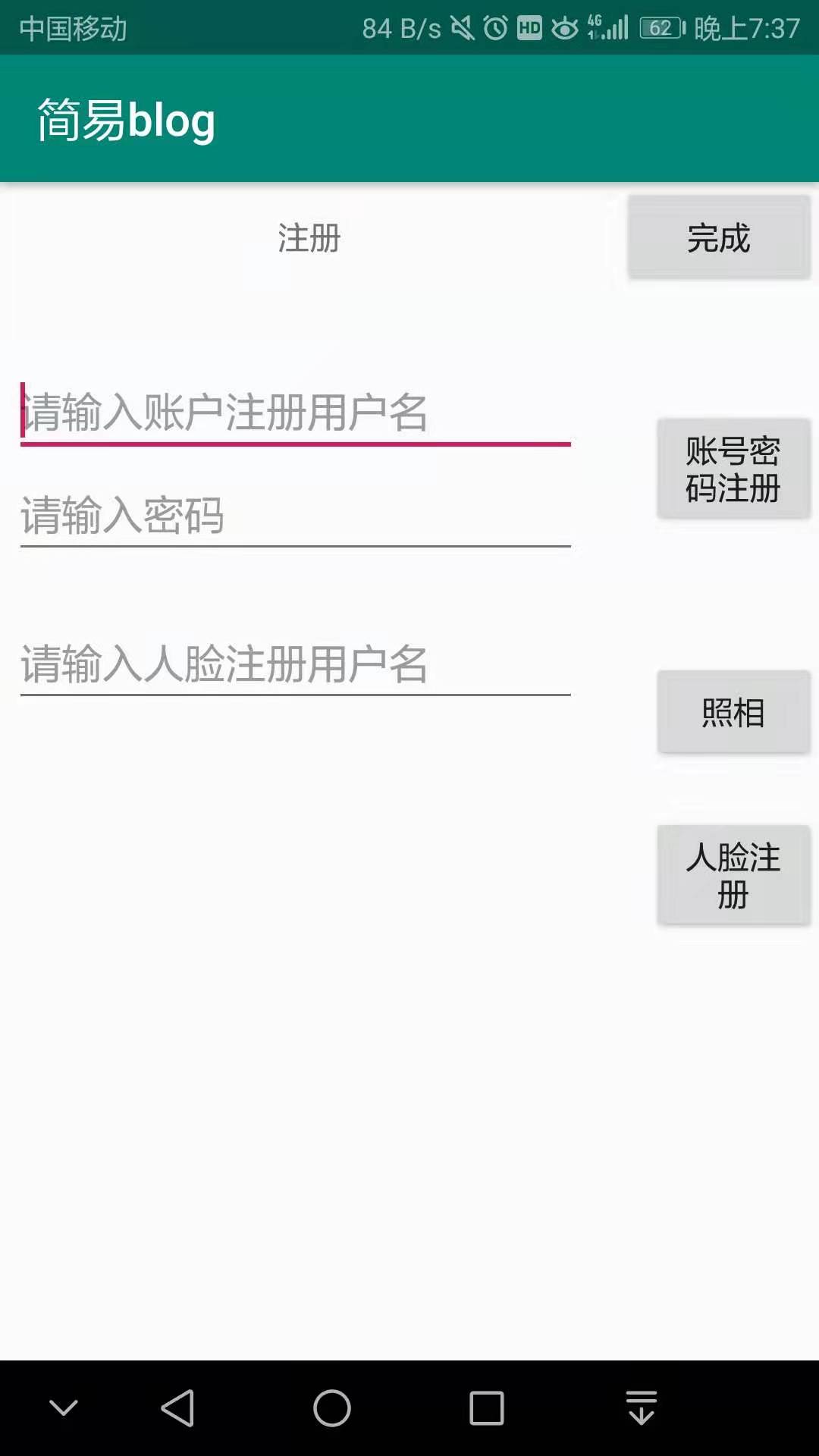


图3

**6.2百度AI基础功能:**

进入界面后，点击底部导航栏的“me”进入图4所示界面，在这个界面可以进行密码修改。同时，点击“基础功能展示”进入图5所示界面。这个界面将展示四个百度AI基础功能：添加人脸、人脸评测、多脸检测、人脸对比。这四个功能分别是图6、图7、图8、图9。

图4 图5

图6 图7

图8 图9

**6.3扩展功能:**

扩展功能部分做的是简单的微博功能。微博显示以及刷新页面如图10所示。在图11所示界面，进行发布微博。在图12所示页面，进行好友添加。

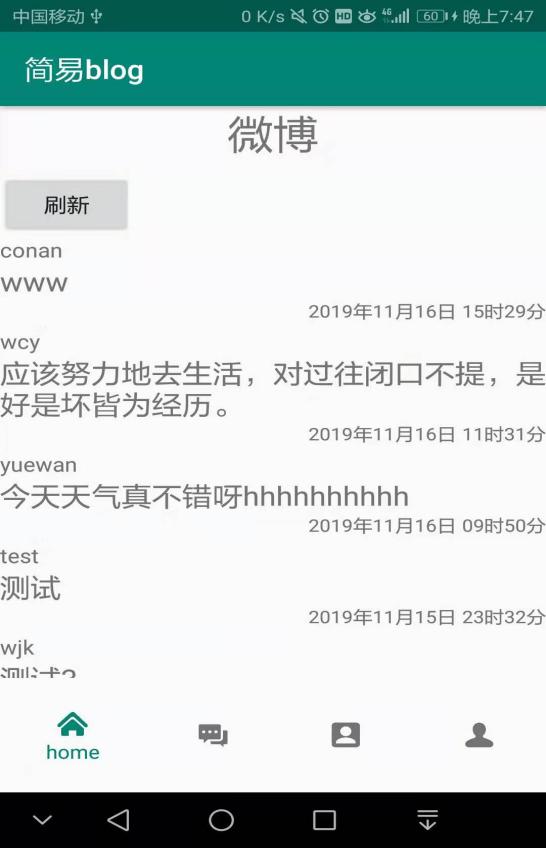
 

图10 图11 

图12