**“佳缘”线上婚恋交友平台**

**系统设计文档**

**版本：V3.0**



**课程名称：**软件工程实验

**项目名称：**“佳缘”线上婚恋交友平台

**所在学校：**广州大学

**所在学院：**经济与统计学院

**所在班级：**数据201班

**项目成员：**2009700003 易鸿

2064700031 罗玮

2064700020 黄颖

**指导老师：**麦红老师

**编写日期：**2022年6月5日

**目 录**

[1 引言 1](#_Toc3467)

[1.1 编写目的 1](#_Toc26051)

[1.2 项目背景 1](#_Toc15013)

[1.3 术语缩写 1](#_Toc27986)

[1.4 参考资料 2](#_Toc9089)

[1.5 设计约束 2](#_Toc12222)

[1.6 设计策略 2](#_Toc29277)

[1.7 修订历史记录 3](#_Toc22898)

[2 任务概述 4](#_Toc10379)

[2.1 目标 4](#_Toc15015)

[2.2 需求概述 4](#_Toc8398)

[3 概要设计 5](#_Toc2263)

[3.1 总体模块设计 5](#_Toc29041)

[3.2 模块功能 6](#_Toc17652)

[3.3 数据库设计 9](#_Toc5161)

[4 详细设计 12](#_Toc13376)

[4.1 时序设计 12](#_Toc10761)

[4.2 算法设计 18](#_Toc2293)

[4.3 界面设计 32](#_Toc9228)

[4.4 接口设计 41](#_Toc288)

[4.5 运行设计 41](#_Toc7431)

[5 系统出错处理设计 43](#_Toc13658)

[5.1 出错信息 43](#_Toc22521)

[5.2 数据保护措施 43](#_Toc24132)

[5.3 补救措施 43](#_Toc11799)

[6 安全设计 44](#_Toc20214)

[7 维护设计 45](#_Toc21000)

**1 引言**

* 1. **编写目的**

本概要说明书在完成需求分析阶段后，根据《“佳缘”线上婚恋交友平台需求规格说明书》，根据系统用户对本系统需求，将需求转换为实际程序模块，解决该过程出现的问题。

本说明书也可为之后的软件测试以及软件维护阶段提供参考，以了解设计过程中的模块设计结构，为系统测试人员提供测试依据，便于找出设计阶段的不足或错误。

* 1. **项目背景**

本项目（“佳缘”线上婚恋交友平台）在调查、参考已有平台功能，明确用户及管理员所需功能的前提下，进行进一步详细的系统设计。

* 1. **术语缩写**

“佳缘”线上婚恋交友平台：对有婚恋交友需求的用户提供信息匹配，以此满足其需求的线上平台。

Python：提供高效的高级数据结构，面向对象编程的脚本和快速开发应用的编程语言

IPO：Input&Process&Output(输入、处理、输出)

My SQL：一个关系型数据库管理系统(DBMS)。

API：Application Programming Interface,一些预先定义的函数。

Def：预先自定义的函数。

本文用到的术语符合国家标准《软件工程术语（GB/T11475-1995）》 。

与本文直接相关的国家标准包括：（中国标准出版社 1996 年）

GB8566-1995 软件生存期过程

GB8567-88 计算机软件产品开发文件编制指南

GB8567-88 计算机软件数据库设计说明编制指南

GB/T11457-1995 软件工程术语

* 1. **参考资料**

[1]张海藩,牟永敏.软件工程导论(第 6 版)[M].北京:清华大学出版社，2013

[2][Baron Schwartz](https://book.jd.com/writer/Baron%20Schwartz_1.html)，[Peter Zaitsev](https://book.jd.com/writer/Peter%20Zaitsev_1.html)，[Vadim Tkachenko](https://book.jd.com/writer/Vadim%20Tkachenko_1.html)著，[宁海元](https://book.jd.com/writer/%E5%AE%81%E6%B5%B7%E5%85%83_1.html)，[周振兴](https://book.jd.com/writer/%E5%91%A8%E6%8C%AF%E5%85%B4_1.html)，[彭立勋](https://book.jd.com/writer/%E5%BD%AD%E7%AB%8B%E5%8B%8B_1.html)等译.高性能My SQL（第3版）[M].北京:电子工业出版社，2013

[3]《“佳缘”线上婚恋交友平台需求分析规格说明版本：V1.0》

* 1. **设计约束**

1. 需求约束
2. 能够在PC机上访问平台；
3. 用户使用不受操作系统限制；
4. 页面简单，指引明确，操作方便快捷；
5. 具有较高的正确性、鲁棒性、健壮性、可维护性、可靠性、安全性等。
6. 隐含约束
7. 用户必须具备一定的智能设备使用能力和相关平台使用经验
8. 受服务器限制，用户有数据库等错误提醒常识
   1. **设计策略**
9. 拓展策路：本系统的软件设计采用模块化体系结构，新功能的引入不影响原有的功能模块，具有良好的可扩展性。
10. 复用策略：tkinter窗口、信息查询等部分通用化设计，便于复用。
11. 安全性：系统中涉及到个人信息安全，系统应具备统一、完善的多级安全机制，以保证系统的安全性。
12. 易用性：系统应提供友好的操作界面，能适配手机终端和计算机终端，方便用户使用。
13. 灵活性：业务本身具有较多的灵活性，在系统设计时，设计上要充分考虑到灵活性。
    1. **修订历史记录**

**表1-1 版本更新记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 日期 | 版本 | 说明 | 作者 |
| 1 | 2022年5月4日 | v1.0 | 系统设计文档1.0版本 | 罗玮 |
| 2 | 2022年5月7日 | V2.0 | 算法部分更新、数据库等设计更新 | 易鸿 |
| 3 | 2022年6月5日 | V3.0 | 软件部分功能的增加、更新 | 罗玮 |

**2 任务概述**

* 1. **目标**

搭建一个以真实、准确的数据为依据，以现代计算机技术为方法的线上婚恋交友平台。交互界面美观简洁，用户操作提示信息明确，能够指导用户以简单的操作使用该平台。

对用户填写的“理想型”条件，建立独属于每一个不同用户喜爱的另一半的模型，用于在该平台的所有用户中匹配交友。

* 1. **需求概述**

本平台的设计目标是一个面向广大适婚人群的婚恋交友平台。本平台应能使用户进行自己注册、登录账号，查看、修改、管理个人信息，在平台内查询目标条件的用户、匹配个人理想型模型等基本操作。系统设计应能够使用户管理个人信息。平台根据用户在系统中的角色不同，具有不同的权限。平台应具有操作指向简单、安全性高、推送准确、用户界面简洁大方等特色，为用户提供使用的便利。

**3 概要设计**

* 1. **总体模块设计**

本平台拥有如下功能：

* + - 1. 新用户填写相应信息注册一个本平台的用户账号，用户使用账号登录到平台后进入用户界面；
      2. 在用户界面，用户可以查看、修改个人信息，查看浏览记录，对部分对象进行打分并匹配符合用户理想型的对象人群，检索特定条件的目标人群，意见反馈；
      3. 平台管理员可查看、管理用户信息，查看用户浏览记录，查看用户反馈。
      4. 最后，用户都可以自由退出本系统。



**图3-1 总体模块图**



**图 3-2 用户管理模块**



**图 3-3 管理员管理模块**

* 1. **模块功能**

**3.2.1 登录模块功能**

1. 登录界面设计：整个界面设计能够实现用户的注册、登录、退出登录。
2. 用户登录：用户打开平台的登录页面，在“用户名”和“密码”框中输入注册时设置的用户名和密码，并点击“登录”键。输入正确后，可进入平台的用户界面，若用户名输入错误，则提示“您还没有注册，请赶紧注册，然后遇见姻缘！”；若密码输入错误，则提示“密码错误，请重新登录！”。
   * 1. 退出登录：在登录页面，点击页面右上角的页面关闭键，关闭对话框，退出登录。
     2. 用户注册：若普通用户想要查看平台的相关信息，需注册个人账号并登录，才能进行检索等功能的操作。注册时单击“注册”键进入注册界面。在注册过程中，对某些关键字进行了规定：①用户名、密码、性别、年龄、身高、为必填项目；②用户名长度必须小于20个字符，密码必须小于15个字符手机号长度必须为11个字符；③若注册时输入的用户名已注册，则提示“用户已注册，请重新注册！”；④“密码”与“验证”需输入完全相同的字符。输入完毕后，单击“注册”键，完成注册，若需取消注册，则单击“取消”键终止注册。
     3. 忘记密码：若普通用户登录时忘记密码，点击页面的“忘记密码”，即弹出验证用户信息页面，用户需输入用户名与手机号验证个人身份。

**3.2.2 普通用户模块功能**

（1）修改用户信息：在用户管理员界面单击左侧菜单栏的“个人中心”键即可进入修改用户信息界面。在界面上方显示当前用户的信息。用户在“请选择待修改条件”框中选择需要修改的信息类别，并在下方“请输入修改值”框中填写修改后的值，最后点击“提交”键，提示“修改成功”。

（2）模型匹配训练：单击用户界面左侧菜单栏的“匹配训练”键，即可弹出显示多个个人信息的窗口，用户根据不同的个人信息，选择该条件下的好感度等级，由此生成专属于用户个人的婚恋对象模型。

（3）查看历史记录：点击用户界面左侧菜单栏“历史记录”，即可弹出用户的检索历史记录界面，内容包括被检索用户的姓名、性别、爱好等个人信息。

（4）意见反馈：单击用户界面左侧菜单栏的“意见反馈”键，即可进入用户意见反馈界面。用户在“用户名”框中填写个人用户名，在“反馈主题”框填写需要反馈给平台的内容主题，在“反馈内容”框中填写需要反馈给平台的具体内容，最后点击“提交”键，完成反馈。

（5）个人评分：单击用户界面右侧导航栏的“个人评分”键，即可弹出对设定条件的个人评分界面。在界面输入框中，用户依次输入设定的个人信息，并选择对其的好感度，提高婚恋对象模型对用户的匹配度。

（6）模式匹配：单击用户界面右侧导航栏的“模式匹配”键，即可弹出模式匹配操作窗口。在“请选择搜索条件”框中选择匹配条件，在“请输入搜索值”框中输入该匹配条件的具体值，点击“提交”键，完成模式匹配，并弹出显示匹配出的用户信息的窗口，双击用户即可发送添加好友请求。

（7）检索：单击用户界面右侧导航栏的“检索ta”键，即可弹出检索用户操作界面。在“请选择搜索条件”框中选择检索条件，在“请输入搜索值”框中输入该搜索条件的具体值，点击“确定”键，完成检索操作，并弹出显示检索出的用户信息的窗口。

（8）查看平台信息：点击用户界面左侧菜单栏的“介绍文档”、“隐私政策”、“用户协议”、“版本更新”、“帮助文档”任意一个操作键，即可弹出显示相关内容的窗口。

（9）查看好友列表：点击用户界面右侧导航栏的“我的好友”键，即可弹出用户的好友列表窗口，鼠标左键双击好友用户名可以进行聊天操作，鼠标右键双击删除好友；点击窗口的“待ta确认”键，即显示发送添加好友请求后，暂未被对方通过验证的好友列表，鼠标左键双击即可删除该条好友申请；点击窗口的“待你通过”键，即可显示向当前用户发送好友申请的好友列表，鼠标右键双击即可通过该好友申请，鼠标左键双击即可删除该好友申请。

**3.2.3 管理员模块功能**

（1）修改管理员信息：在管理员界面单击左侧菜单栏的“个人中心”键即可进入修改管理员信息界面。在界面上方显示管理员的信息。管理员在“请选择待修改条件”框中选择需要修改的信息类别，并在下方“请输入修改值”框中填写修改后的值，最后点击“提交”键，提示“修改成功”。

（2）查看用户信息：点击管理员界面的“用户信息”键，即可进入用户信息查看界面，包括平台用户的姓名、性别、爱好等个人信息。

（3）查看用户搜索记录：点击管理员界面的“用户记录”键，即可。

（4）修改用户信息：点击管理员界面的“用户修改”键，即可进入修改用户信息界面。在“请选择待修改人条件”框中，选择需要修改的用户的条件，在“请输入修改值”框中输入修改的值，最后点击“提交”键，完成修改。

（5）查看用户意见反馈：点击管理员界面左侧菜单栏的“意见反馈”键，即可弹出用户的意见反馈及其对应用户。

* + 1. 审核用户注册申请：点击管理员界面左侧菜单栏的“注册申请”键，查看用户注册申请，对注册信息进行审核，点击通过审核的注册申请，并点击“确定”键完成通过用户注册申请流程。
    2. 查看用户通过记录：点击管理员界面左侧菜单栏的“待办情况”键，查看用户的申请注册记录。
  1. **数据库设计**

1. **数据库实体**
2. 用户实体的属性有用户名、密码、性别、爱好、年龄、月收入、身高、月花销、好评、联系方式。每一名用户有唯一的 用户名。
3. 管理员实体的属性有 id、性别、联系方式、管理员密码。每一位管理员有唯一的 id。
4. 测试实体的属性有发起人、年龄、月收入、身高、月花销、好评、爱好、好感度。每一个测试实体有唯一的发起人。
5. 匹配结果实体的属性有用户名、被匹配者、性别、爱好、年龄、月收入、身高、月花销、好评。每一个匹配结果实体有唯一的用户名。
6. 浏览记录实体的属性有发起人、时间、用户名、性别、爱好、年龄、月收入、身高、月花销、好评、联系方式。每一个浏览记录实体有唯一的发起人。
7. 反馈实体的属性有用户名、时间、主题、反馈正文。每一个反馈实体有唯一的用户名。
8. **数据库字段**
9. 用户信息表

**表3-1 用户信息表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 小数点 | 不是null | 虚拟 | 键 | 注释 |
| name | varchar | 20 | 0 | 不是Null | 0 | Key |  |
| sex | varchar | 5 | 0 | 不是Null | 0 | 0 |  |
| hobby | varchar | 30 | 0 | 无限制 | 0 | 0 |  |
| age | varchar | 3 | 0 | 不是Null | 0 | 0 |  |
| salary | varchar | 10 | 0 | 无限制 | 0 | 0 |  |
| height | varchar | 4 | 0 | 不是Null | 0 | 0 |  |
| cost | varchar | 10 | 0 | 无限制 | 0 | 0 |  |
| good | varchar | 12 | 0 | 无限制 | 0 | 0 |  |
| num | varchar | 12 | 0 | 无限制 | 0 | 0 |  |
| key | varchar | 15 | 0 | 不是Null | 0 | Key |  |

1. 管理员信息表

**表3-2 管理员信息表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 小数点 | 不是null | 虚拟 | 键 | 注释 |
| name | varchar | 20 | 0 | 不是Null | 0 | Key |  |
| sex | varchar | 5 | 0 | 不是Null | 0 | 0 |  |
| num | varchar | 12 | 0 | 无限制 | 0 | 0 |  |
| key | varchar | 15 | 0 | 不是Null | 0 | Key |  |

1. 历史记录表

**表3-3 历史记录表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 小数点 | 不是null | 虚拟 | 键 | 注释 |
| initiator | varchar | 20 | 0 | 不是Null | 0 | 0 |  |
| name | varchar | 20 | 0 | 不是Null | 0 | 0 |  |
| sex | varchar | 5 | 0 | 不是Null | 0 | 0 |  |
| hobby | varchar | 30 | 0 | 无限制 | 0 | 0 |  |
| age | varchar | 3 | 0 | 无限制 | 0 | 0 |  |
| salary | varchar | 10 | 0 | 无限制 | 0 | 0 |  |
| height | varchar | 4 | 0 | 无限制 | 0 | 0 |  |
| cost | varchar | 10 | 0 | 无限制 | 0 | 0 |  |
| good | varchar | 12 | 0 | 无限制 | 0 | 0 |  |
| num | varchar | 12 | 0 | 无限制 | 0 | 0 |  |

1. 意见反馈表

**表3-4 意见反馈表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 小数点 | 不是null | 虚拟 | 键 | 注释 |
| topic | varchar | 20 | 0 | 不是Null | 0 | 0 |  |
| time | datetime | 6 | 0 | 无限制 | 0 | 0 |  |
| user\_name | varchar | 20 | 0 | 无限制 | 0 | 0 |  |
| feedback | varchar | 255 | 0 | 不是Null | 0 | 0 |  |

**4 详细设计**

**4.1 时序设计**

**4.1.1 注册登录模块设计**

1. 注册

**图4-1 注册时序图**

1. 登录

**图4-2 登录时序图**

**4.1.2 账户管理模板设计**

1. 修改信息

**图4-3 修改信息时序图**

**4.1.3 功能管理模块设计**

1. 个人评分



**图4-4 个人评分时序图**

1. 模式匹配



**图4-5 模式匹配时序图**

1. 检索功能



**图4-6 检索功能时序图**

1. 问题反馈



**图4-7 问题反馈时序图**

1. 反馈处理



**图4-8 反馈处理时序图**

1. 浏览记录



**图4-9 浏览记录时序图**

**4.2 算法设计**

### 4.2.1 用户模块

**（1） 注册**

1） 功能：新增注册用户

A.封装表单数据到后台数据库

B.后端校验，防范各类越过前端校验的非法输入和操作，保障系统安全

C.校验成功后通过新增数据库记录来新增用户

D.返回注册信息给前端页面

2） 输入：用户名、密码、性别、爱好、年龄、月收入、身高、月花销、好评、联系方式

3） 输出：注册成功与否，成功则跳转登录页面；注册失败返回注册失败的原因

4） 程序流程图：



**图4-10 用户注册程序流程图**

1. 接口：
2. 调用方法：python和sql语句
3. 调用形式：def usr\_sign\_up()
4. 传入参数：entry\_new\_name、new\_pwd、new\_sex、new\_hobby、new\_age、new\_salary、new\_height、new\_cost、new\_good、new\_num
5. 传出参数：messagebox.showinfo(注册成败标志)、Error(错误信息)、user(封装好的user的对象)
6. 测试要点：

**表4-1 注册功能测试要点**

|  |  |
| --- | --- |
| 测试项 | 具体要点 |
| 前端校验是否正常 | 用户名格式错误 |
| 用户名已经被注册 |
| 用户密码不满足长度 |
| 两次密码不一致 |
| 后端校验是否正常 | 用户名格式错误 |
| 用户名已经被注册 |
| 用户密码不满足长度 |
| 两次密码不一致 |
| 前端是否能显示注册错误信息 | |
| 合法用户是否成功添加到数据库 | |

**（2） 登录**

1） 功能：用户登录

A. 验证表单数据到数据库

B. 后端校验，防范各类越过前端校验的非法输入和操作，保障系统安全

C. 校验成功后用户成功登录

D. 返回用户信息给前端页面，进入用户主页面

2） 输入：用户名、登录密码

3） 输出：如果登录成功，则跳转到主页页面；登录失败返回登 录失败的原因

4） 程序流程图：



**图4-11 用户登录程序流程图**

1. 接口：
2. 调用方法：python和sql语句
3. 调用形式：usr\_login()
4. 传入参数：usr\_name、usr\_pwd
5. 传出参数：messagebox.showinfo(注册成败标志)、Error(错误信息)
6. 测试要点：

**表4-2 登录功能测试要点**

|  |  |
| --- | --- |
| 测试项 | 具体要点 |
| 前端校验是否正常 | 用户名或密码错误 |
| 用户名未注册 |
| 后端校验是否正常 | 用户名或密码错误 |
| 用户名未注册 |
| 前端是否能显示注册错误信息 | |
| 合法用户在数据库是否成功验证 | |

### 4.2.2 评分模块

**（1） 个人评分**

1） 功能：对待评分用户输入相关信息后打分

A. 验证表单数据到数据库

B. 后端校验，防范各类越过前端校验的非法输入和操作

C. 校验成功后信息进行匹配测试

D. 返回打分信息给前端页面

2） 输入：年龄、月收入、身高、月花销、好评、爱好

3） 输出：评分与否，成功则跳转评分结果页面；失败返回失败的原因

4） 程序流程图：



**图4-12 用户评分程序流程图**

1. 接口：
2. 调用方法：python和sql语句
3. 调用形式：def classuse()
4. 传入参数：var\_age、var\_salary、var\_height、var\_buy、var\_comment
5. 传出参数：classifierResult(好感度结果)、Exception(错误信息)
6. 测试要点：

**表4-3 登录功能测试要点**

|  |  |
| --- | --- |
| 测试项 | 具体要点 |
| 前端校验是否正常 | 匹配信息填写不规范 |
| 中途退出等影响 |
| 后端校验是否正常 | 匹配信息填写不规范 |
| 中途退出等影响 |
| 前端是否显示能显示错误信息 | |
| 待评分信息在个人模型数据库是否成功匹配 | |

### 4.2.3 匹配模块

**（1） 模式匹配**

1） 功能：对用户输入相关条件后进行模型识别后的匹配

A. 验证表单数据到数据库

B. 后端校验，防范各类越过前端校验的非法输入和操作

C. 校验成功后信息进行检索匹配测试

D. 返回匹配信息给前端页面

2） 输入：匹配条件、匹配条件信息

3） 输出：匹配成功与否，成功则跳转匹配结果页面；匹配失败返回匹配失败的原因

4） 程序流程图：



**图4-13 用户匹配程序流程图**

1. 接口：
2. 调用方法：python和sql语句
3. 调用形式：def match()
4. 传入参数：var（匹配条件）、Combobox（条件列表）、var\_zhi（匹配条件信息）
5. 传出参数：result2（匹配识别结果）、Exception（错误信息）、messagebox.askquestion（无此人结果信息）
6. 测试要点：

**表4-4 匹配功能测试要点**

|  |  |
| --- | --- |
| 测试项 | 具体要点 |
| 前端校验是否正常 | 匹配条件填写选择不规范 |
| 匹配条件信息填写不合法 |
| 后端校验是否正常 | 匹配条件填写选择不规范 |
| 匹配条件信息填写不合法 |
| 前端是否显示能显示错误信息 | |
| 待匹配信息在个人模型数据库是否成功匹配 | |
| 个人模型是否建立可使用 | |

### 4.2.4 检索模块

**（1） 检索功能**

1） 功能：对用户输入相关条件后进行检索

A. 验证表单数据到数据库

B. 后端校验，防范各类越过前端校验的非法输入和操作

C. 校验成功后按检索信息进行检索

D. 返回检索信息给前端页面

2） 输入：检索条件、检索条件信息

3） 输出：检索成功与否，成功则跳转检索结果页面；检索失败返回检索失败的原因

4） 程序流程图：



**图4-14 用户检索程序流程图**

1. 接口：
2. 调用方法：python和sql语句
3. 调用形式：def matchup()
4. 传入参数：var(检索条件)、var\_zhi（检索条件信息）
5. 传出参数：result3(检索用户结果)、Exception（错误信息）、messagebox.askquestion（无此人结果信息）
6. 测试要点：

**表4-5 检索功能测试要点**

|  |  |
| --- | --- |
| 测试项 | 具体要点 |
| 前端校验是否正常 | 检索条件填写不规范 |
| 检索条件信息填写不合法 |
| 后端校验是否正常 | 检索条件填写不规范 |
| 检索条件信息填写不合法 |
| 前端是否显示能显示错误信息 | |
| 待检索信息在用户数据库是否成功匹配 | |

### 4.2.5 反馈模块

**（1） 反馈问题**

1） 功能：提交反馈

A. 填写反馈信息

B. 后端校验用户信息，防范各类越过前端的非法输入和违规操作

C. 校验成功后上传到反馈信息数据库

D. 返回成功/失败提示信息给前端页面

2） 输入：用户名（自动识别）、密码（自动识别）、反馈信息主题、反馈信息

3） 输出：反馈成功与否，成功则弹出成功提示页面；反馈失败则返回反馈失败的原因

4） 程序流程图：



**图4-15 用户反馈程序流程图**

1. 接口：
2. 调用方法：python和sql语句
3. 调用形式：def viewword()
4. 传入参数：var\_topic(反馈主题)、var\_text（反馈信息）
5. 传出参数：messagebox.showinfo（成功提示）、Error（错误信息）
6. 测试要点：

**表4-5 反馈功能测试要点**

|  |  |
| --- | --- |
| 测试项 | 具体要点 |
| 前端校验是否正常 | 反馈主题填写不规范 |
| 反馈信息填写不合法 |
| 后端校验是否正常 | 反馈主题填写不规范 |
| 反馈信息填写不合法 |
| 前端是否显示能显示错误信息 | |
| 反馈成功的信息在反馈信息数据库是否成功汇总 | |

### 4.2.6 记录模块

（1） 浏览记录

1） 功能：查看用户浏览记录

A. 进入浏览记录页面

B. 后端校验用户信息，防范违规操作

C. 校验成功后返回浏览记录给前端页面

2） 输入：用户名、密码

3） 输出：查看浏览记录成功与否，成功则跳转浏览记录页面；失败则返回失败的原因

4） 程序流程图：

****

**图4-16 用户记录程序流程图**

1. 接口：
2. 调用方法：python和sql语句
3. 调用形式：def history()
4. 传入参数：usr\_name、usr\_pwd
5. 传出参数：result1（浏览记录）、Error（错误信息）
6. 测试要点：

**表4-5 反馈功能测试要点**

|  |  |
| --- | --- |
| 测试项 | 具体要点 |
| 前端校验是否正常 | 用户/密码填写不合法 |
| 前端校验是否正常 | 用户/密码填写不合法 |
| 前端是否显示能显示错误信息 | |
| 请求在历史记录数据库是否成功查询 | |

**4.3 界面设计**

### 登录模块界面设计

（1）登录界面设计



**图4-17 登录界面**

（2）注册界面设计



**图4-18 注册界面**

### 普通用户模块界面设计

（1）用户界面设计

**图4-19 用户主界面**

（2）个人中心界面设计



**图4-20 个人中心界面**

1. 匹配训练界面设计



**图4-21 匹配训练界面**

1. 历史记录界面设计



**图4-22 历史记录界面**

1. 意见反馈界面设计

**图4-23 意见反馈界面**

1. 个人评分界面设计

**图4-24 个人评分界面**

1. 模式匹配界面设计

**图4-25 模式匹配界面**

1. 检索ta界面设计

**图4-26 检索界面**

1. 我的好友界面设计

**图 4-27我的好友界面**

### 管理员模块界面设计

1. 管理员界面设计

**图4-28 管理员界面**

1. 个人中心界面设计



**图4-29 管理员个人中心界面**

1. 用户信息界面设计

**图4-30 用户信息界面**

1. 用户记录界面设计



**图4-31 用户记录界面**

（5）用户修改界面设计

**图4-32 用户修改界面**

（6）意见反馈界面



**图4-33 意见反馈界面**

（7）待办情况、交友情况界面设计

**图4-34 待办情况、交友情况界面**

（8）注册申请界面设计

**图4-35 注册申请界面**

**4.4 接口设计**

### 4.4.1 外部接口

(1) 服务器程序可使用Python提供的对ySQL的接口，进行对数据库的所有访问。

(2) 服务器程序上可使用MySQL的对数据库的备份命令，以做到对数据的保存。

### 4.4.2 内部接口

内部接口方面，各模块之间采用函数调用、参数传递、返回值的方式进行信息传递。接口传递的信息将是以数据结构封装了的数据，以参数传递或返回值的形式在各模块间传输。

**4.5 运行设计**

### 4.5.1 运行控制

根据事务处理类型的不同，把事务划分为两类：交互类业务、查询类业务。在本方案中，相关的性能指标如下：

1. 交互类业务

交互类业务是指平时工作中在系统中进行的业务处理，如注册、修改、删除等操作。

1. 平均响应时间：2-5(秒)
2. 峰值响应时间：8-15(秒)
3. 查询类业务

如信息查询、浏览记录等。查询业务由于受到查询的复杂程度、查询的数据量大小等因素的影响，需要根据具体情况而定，在此给出一个参考范围。

1. 简单查询平均响应时间：2-10(秒)
2. 复杂查询平均响应时间：10-60(秒)

**5 系统出错处理设计**

1. **出错信息**

出错信息如表所示。

**表5-1 出错信息记录表**

|  |  |  |
| --- | --- | --- |
| 故障描述 | 输出信息形式 | 处理方法 |
| 用户名和密码出错 | 弹窗提示 | 提示输入不正确，并返回 |
| 注册用户名已存在 | 弹窗提示 | 提示用户名已注册，并返回 |
| 用户名、密码等必填未填写 | 弹窗提示 | 提示输入不完整，并返回 |
| 修改信息为空或不合法 | 弹窗提示 | 提示修改不合法，并返回 |
| 检索值为空 | 弹窗提示 | 提示条件选择不能为空，并返回 |
| 密码与确认密码不同 | 弹窗提示 | 提示两次输入的密码要一致 |
| 条件选择为空 | 弹窗提示 | 提示条件选择不能为空，并返回 |
| 用户名长度为超过20个字符 | 原页面显示 | 提示长度必须小于20个字符，并重新输入 |
| 密码长度超过15个字符 | 原页面显示 | 提示长度必须小于15个字符，并重新输入 |

1. **数据保护措施**

为了预防系统故障或者遭遇黑客攻击造成数据丢失，采用以下方法保护数据：

1. 备份：使用附加存储设备备份数据，备份频率为每日一次。
2. 恢复及再启动：如果数据造成丢失，可使用备份数据还原。
3. **补救措施**

为了预防系统故障或者遭遇黑客攻击造成的丢失数据，将采用以下三种方法保护数据：

1. 后备技术：双机备份。
2. 降效技术：通过数据库的第三方工具（如 MySQL等）对数据库备份.
3. 设置数据库的自动备份，防止人工备份的不及时。

**6 安全设计**

为每个操作人员设置账号及口令，规定每一级操作人员的权限。

**7 维护设计**

由于此项目比较小，除了严格按照软件工程的科学方法开发软件、认真建立文档、编码阶段写详细的注释外，没有特殊的维护设计，如：设置维护模块等。

本系统没有采用单独的维护设计模块，在做开发技术选择的时候采用易于维

护的开发技术，所以系统采用数据库字段和应用界面中的控件绑定的技术，这样

有利于今后的系统维护。