文档编号：BMS-JS-0002

实验室人员管理系统

设计说明书

**目录**

[**一、引言** 3](#_Toc119252364)

[1.1 编写目的 3](#_Toc119252365)

[1.2 背景 3](#_Toc119252366)

[1.3 参考资料 3](#_Toc119252367)

[**二、系统数据结构设计** 3](#_Toc119252368)

[2.1 逻辑结构设计要点 3](#_Toc119252369)

[2.2 物理结构设计要点 6](#_Toc119252370)

[2.3 数据结构与程序的关系 6](#_Toc119252371)

[**三、系统体系结构设计** 6](#_Toc119252372)

[3.1 系统体系结构描述 6](#_Toc119252373)

[3.2 体系结构图 7](#_Toc119252374)

[**四、构件及接口设计** 8](#_Toc119252375)

[4.1 账号管理 8](#_Toc119252376)

[4.2 读者书籍操作 9](#_Toc119252377)

[4.3 读者信息管理 10](#_Toc119252378)

[4.4 管理员图书管理 11](#_Toc119252379)

[4.5 管理员信息管理 14](#_Toc119252380)

[4.6 数据操作 15](#_Toc119252381)

[4.7 读者界面 18](#_Toc119252382)

[4.8 管理员界面 18](#_Toc119252383)

[**五、界面设计** 20](#_Toc119252384)

[5.1 登陆界面 20](#_Toc119252385)

[5.2 注册界面 20](#_Toc119252386)

[5.3 用户主页 21](#_Toc119252387)

[5.4 用户信息界面 21](#_Toc119252388)

[5.5用户修改密码界面 22](#_Toc119252389)

[5.6用户修改用户名界面 23](#_Toc119252390)

[5.6用户借还书界面 23](#_Toc119252391)

[5.7用户预约图书界面 25](#_Toc119252392)

[5.8管理员主界面 26](#_Toc119252393)

[5.9管理员类别管理界面 27](#_Toc119252394)

[5.10管理员增加类别界面 27](#_Toc119252395)

[5.11管理员删除类别界面 28](#_Toc119252396)

[5.12管理员修改类别界面 29](#_Toc119252397)

[5.13管理员图书管理界面 30](#_Toc119252398)

[5.14管理员增加图书界面 31](#_Toc119252399)

[5.15管理员删除图书界面 32](#_Toc119252400)

[5.16管理员修改图书界面 33](#_Toc119252401)

[5.17管理员管理用户信息界面 34](#_Toc119252402)

[5.18管理员管理信息界面 36](#_Toc119252403)

评审意见：

缺点：

1. **引言**

1.1 编写目的

编写本设计说明书的目的是详细说明实验室人员管理系统的结构和内部设计，基于需求规格说明书，提供详细的程序和接口设计，为本系统程序开发提供直接的支持。

本设计说明书主要面向系统分析人员、编码人员和以后的系统维护人员编写。作为编码人员开发系统的根本依据，以及维护人员在系统维护阶段对系统实施维护的参考资料。

1.2 背景

随着科技的不断进步和实验室研究的广泛应用，以及大型实验室的组建，不同指导老师的学生可能在同一实验室的不同工位使用归属不同的设备，给实验室人员管理带来了更复杂的挑战。传统的人工管理方式已经无法满足实验室的需求，因此需要一个高效、智能的实验室人员管理系统。该系统可以方便地帮助实验室管理员查看和管理学生的实验室工位，更好地管理实验室设备，提高实验室的使用效率和工作质量。这有助于提升实验室的科研水平和竞争力。

基于此，本团队拟开发一款边界高效的实验室人员管理软件，以达到如下目的：实验室管理员只需将数据输入到系统的数据库中即可。由于数据库具有较大的存储容量和稳定性，适合长期保存且不易丢失，为信息量较大的实验室提供了便捷、快速的操作方式。我们的系统具备快速运行、高安全性和良好的稳定性的优点，并且支持修改功能，能够快速查询实验室所需的工位信息。

1.3 参考资料

张海藩：《软件工程导论》清华大学出版社 2008年2月第五版

肖刚：《实用软件文档写作》清华大学出版社 2005年2月

French P W. Managed realignment–the developing story of a comparatively new approach to soft engineering[J]. Estuarine, Coastal and Shelf Science, 2006, 67(3): 409-423.

1. **系统数据结构设计**

2.1 逻辑结构设计要点

实验室人员管理系统 E-R 图如图2.1.1所示：

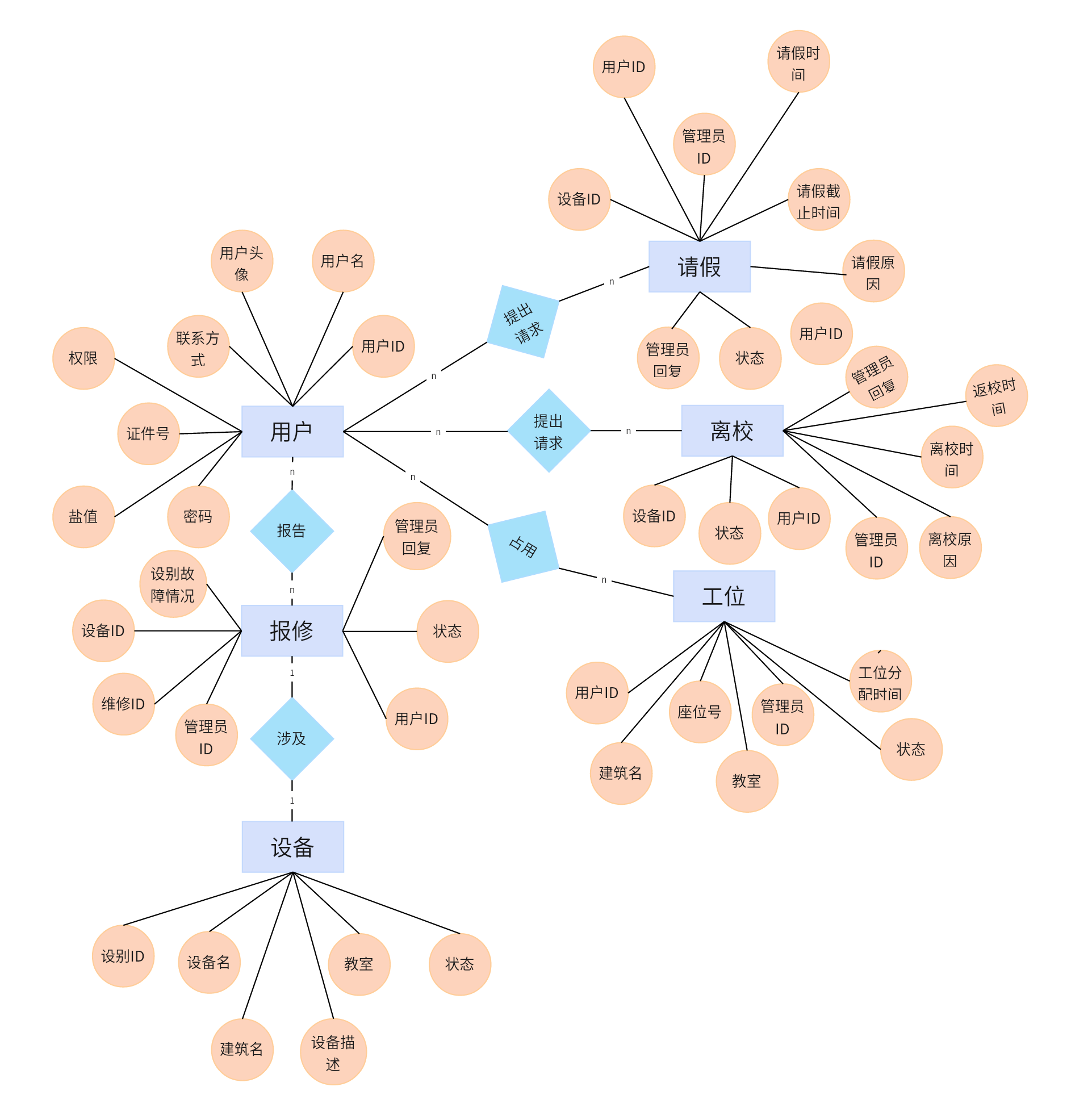


图2.1.1 实验室人员管理系统 E-R 图

具体表结构的设计如下：

**表2.1.1 用户信息表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **用户信息** | | | | | | |
| **字段名** | **说明** | **数据类型** | **数据长度** | **可否为空** | **默认值** | **备注** |
| **user\_id** | 用户id | Unsigned Int | 10位固定 | No | 自增 | 主键 |
| **user\_name** | 姓名 | Char | 32位固定 | No | 萌新 |  |
| **user\_img** | 用户头像 | Char | 32位固定 | No | 0 |  |
| **user\_license** | 证件号 | Char | 16位固定 | No | 0000000000000000 |  |
| **user\_phone** | 联系方式 | Char | 16位固定 | No | 无 |  |
| **user\_email** | 用户邮箱 | Char | 32位固定 | No | 无 |  |
| **user\_salt** | 盐 | Char | 64位固定 | No | 无 | 加密用盐 |
| **user\_passwd** | 加密密码 | Char | 64位固定 | No | 无 | SHA256加密密码 |
| **user\_permissions** | 用户权限 | Int | 11位固定 | No | 0 | 0表示用户, 1表示管理员 |

**表2.1.2 工位信息表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **工位信息** | | | | | | |
| **字段名** | **说明** | **数据类型** | **数据长度** | **可否为空** | **默认值** | **备注** |
| **workstation\_id** | 设备id | Unsigned Int | 10位固定 | No | 自增 | 主键 |
| **user\_id** | 用户id | Unsigned Int | 32位固定 | No | 无 | 外键 |
| **workstation\_building** | 建筑名 | Char | 16位固定 | No | 江安综合楼A座 |  |
| **workstation\_classroom** | 教室号 | Char | 16位固定 | No | A101 |  |
| **workstation\_seat** | 座位号 | Int | 11位固定 | No | 1 |  |
| **workstation\_time** | 分配时间 | DATETIME | 8位固定 | No | NOW() |  |
| **workstation\_state** | 工位状态 | Int | 11位固定 | No | 0 | 0可用, 1已用, -1不可用 |

**表2.1.3 设备信息表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **设备信息** | | | | | | |
| **字段名** | **说明** | **数据类型** | **数据长度** | **可否为空** | **默认值** | **备注** |
| **equipment\_id** | 设备id | Unsigned Int | 10位固定 | No | 自增 | 主键 |
| **equipment\_name** | 设备名 | Char | 32位固定 | No | H100 80GB显卡 |  |
| **equipment\_introduction** | 设备描述 | VarChar | 128位变长 | No | 第一张商用显卡 |  |
| **equipment\_building** | 设备所在建筑 | Chat | 16位固定 | No | 江安综合楼A座 |  |
| **equipment\_classroom** | 设备所在教室 | Char | 16位固定 | No | A101 |  |
| **equipment\_state** | 设备状态 | Int | 11位固定 | No | 0 | 0报修中, 1可使用, -1损坏 |

**表2.1.4 请假信息表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **请假信息** | | | | | | |
| **字段名** | **说明** | **数据类型** | **数据长度** | **可否为空** | **默认值** | **备注** |
| **askforleave\_id** | 请假id | Unsigned Int | 10位固定 | No | 自增 | 主键 |
| **user\_id** | 用户id | Unsigned Int | 10位固定 | No | 无 | 外键 |
| **admin\_id** | 管理员id | Unsigned Int | 10位固定 | Yes | 无 | 外键 |
| **askforleave\_reason** | 请假原因 | VarChar | 512位变长 | No | 申请请假 |  |
| **askforleave\_time** | 请假时间 | DATETIME | 8位固定 | No | 无 | 应早于截止时间 |
| **askforleave\_back** | 截止时间 | DATETIME | 8位固定 | No | 无 | 应晚于请假时间 |
| **askforleave\_reply** | 管理员回复 | VarChar | 512位变长 | No | 已收到,允许本次请假申请 |  |
| **askforleave\_state** | 审核状态 | Int | 11位固定 | No | 0 | 0待核, 1批准, -1不允许 |

**表2.1.5 离校信息表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **离校信息** | | | | | | |
| **字段名** | **说明** | **数据类型** | **数据长度** | **可否为空** | **默认值** | **备注** |
| **leaveschool\_id** | 离校id | Unsigned Int | 10位固定 | No | 自增 | 主键 |
| **user\_id** | 用户id | Unsigned Int | 10位固定 | No | 无 | 外键 |
| **admin\_id** | 管理员id | Unsigned Int | 10位固定 | Yes | 无 | 外键 |
| **askforleave\_reason** | 离校原因 | VarChar | 512位变长 | No | 申请离校 |  |
| **askforleave\_time** | 离校时间 | DATETIME | 8位固定 | No | 无 | 应早于返校时间 |
| **askforleave\_back** | 返校时间 | DATETIME | 8位固定 | No | 无 | 应晚于离校时间 |
| **askforleave\_reply** | 管理员回复 | VarChar | 512位变长 | No | 已收到,允许本次离校申请 |  |
| **askforleave\_state** | 审核状态 | Int | 11位固定 | No | 0 | 0待核, 1批准, -1不允许 |

**表2.1.6 报修信息表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **报修信息** | | | | | | |
| **字段名** | **说明** | **数据类型** | **数据长度** | **可否为空** | **默认值** | **备注** |
| **repair\_id** | 离校id | Unsigned Int | 10位固定 | No | 自增 | 主键 |
| **user\_id** | 用户id | Unsigned Int | 10位固定 | No | 无 | 外键 |
| **admin\_id** | 管理员id | Unsigned Int | 10位固定 | Yes | 无 | 外键 |
| **equipment\_id** | 设备id | Unsigned Int | 10位固定 | No | 无 | 外键 |
| **repair\_reason** | 故障现象 | DATETIME | 8位固定 | No | 无 | 应早于返校时间 |
| **repair\_reply** | 管理员回复 | DATETIME | 8位固定 | No | 无 | 应晚于离校时间 |
| **repair\_state** | 管理员回复 | VarChar | 512位变长 | No | 已收到,允许本次离校申请 |  |

2.2 物理结构设计要点

实验室成员和管理人员在本系统中进行操作和管理时，都可以通过直观的图形界面轻松访问数据，无需关心底层存储的物理关系。由于本系统的数据设计简明且体积较小，因此不需要过多关注数据的物理结构分布

2.3 数据结构与程序的关系

本系统采用多种数据结构，并结合合适的算法，旨在优化程序的几个关键方面，包括简洁性、可读性和高效性。通过巧妙地选择和应用数据结构，实现程序设计中更清晰、高效和易维护的代码

1. **系统体系结构设计**

3.1 系统体系结构描述

本图书管理系统包含以下构件：

1. 账号管理

2. 用户信息管理

3. 用户申请操作

4. 管理员审批管理

5. 管理员用户信息管理

6. 数据操作

7. 读者界面

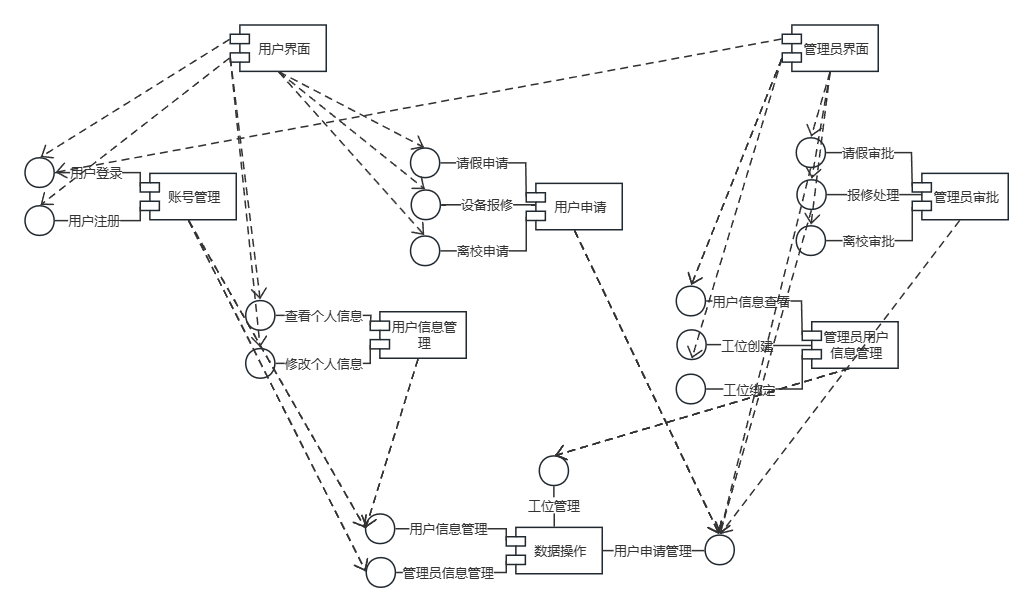
8. 管理员界面

以下是构件的简介：

1. **账号管理：**该构件用于处理用户的注册和用户/管理员登录事务。注册功能允许用户输入其邮箱、姓名、证件号、密码等信息。系统将验证这些信息是否符合规定标准，并向界面发送相应信息（注册成功、注册失败）。注册成功后，系统将进行界面跳转；注册失败则会返回错误信息。登录功能将通过数据库获取信息，将用户/管理员输入的登录信息与数据库中的信息进行比对，以检查用户/管理员填写的信息是否正确。系统将向界面发送相应信息（登录成功、登录失败），成功则进行界面跳转，失败则反馈相应错误信息。
2. **用户信息管理：**该模块用于处理用户/管理员查看和修改个人信息（如手机号、用户名、头像等）的请求。系统将验证用户/管理员输入信息是否正确，若验证成功，则允许修改信息。随后，系统将访问数据库，返回修改的结果，并进行界面跳转。若验证失败，则系统会反馈相应错误信息。
3. **用户申请操作构件：**该构件用于响应并处理用户请假/离校/报修申请和历史记录查看，系统接收到用户发出的请求后，通过访问数据库对相应的数据进行查看、编写、删除和修改，并返回操作是否成功。
4. **管理员审批管理：**该构件用于响应并处理管理员对用户申请的相关操作，包括对用户申请请假/离校的审批答复，以及对用户申请报修设备的处理和审批答复，访问数据库进行对相应数据记录进行修改，并返回操作是否成功以及跳转至相应结果界面。
5. **管理员用户信息管理：**该构建用于相应并处理管理员对于用户的管理，比如查看用户信息和设立工位并为指定用户分配/取消工位，所有信息请求通过服务器对数据库进行访问和修改创建，并返回操作是否成功以及跳转至相应的结果界面。
6. **用户界面构件：**该模块旨在为用户提供系统交互的界面。用户可以通过该模块输入注册、登录信息，并进行用户信息修改、请假/离校申请、设备报修，信息查看等操作。此外，该模块也负责将用户的操作请求传递给其他模块，并在界面上显示其他模块返回的信息，并将其他构件返回的信息显示在界面上以反馈给用户。
7. **管理员界面构件：**该模块旨在为实验室管理员用户提供系统交互的界面。管理员可以通过该模块审批用户请求，查看用户信息，分配工位等操作。此外，该模块也负责将管理员操作请求传递给其他模块，并在界面上显示其他模块返回的信息，以便向管理员提供反馈。

3.2 体系结构图

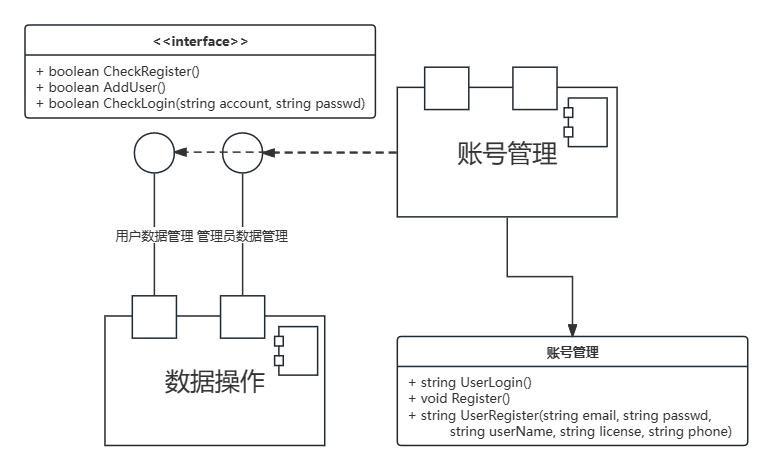
实验室人员管理系统体系结构图如图3.2.1所示



**图3.2.1 实验室人员管理系统体系结构图**

1. **构件及接口设计**

4.1 账号管理



**图4.1 账号管理构件及其接口示意图**

4.1.1 账号管理构件详细设计

**表4.1.1 账号管理构件详细设计表**

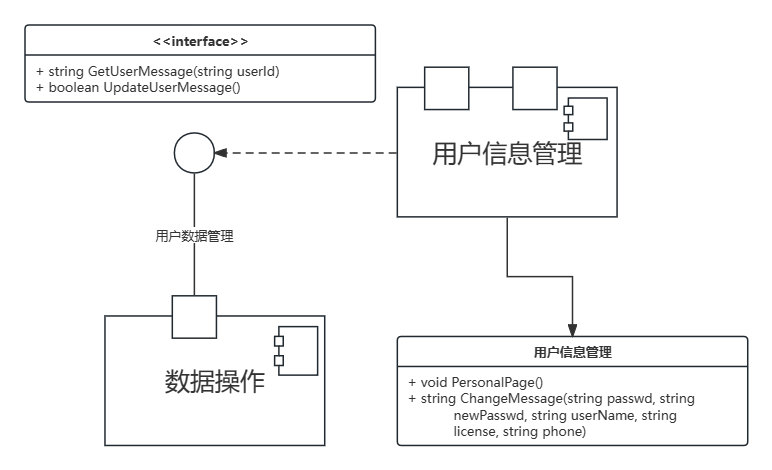
|  |  |  |
| --- | --- | --- |
| 构件名称 | 方法 | 方法描述 |
| 账号管理 | string UserLogin() | 获取用户在登录界面输入的账号和密码信息 |
| void Register() | 用户点击注册按钮，系统跳转至注册界面 |
| string UserRegister(string  email, string passwd, string userName, string license,  string phone) | 获取用户在注册界面输入的游戏、密码、用户名、证件号、手机号等信息 |

4.1.2 账号管理接口详细设计

**表4.1.2 账号管理接口详细设计表**

|  |  |  |
| --- | --- | --- |
| 接口名称 | 方法 | 方法描述 |
| 检测注册格式 | boolean CheckRegister() | 检查用户注册时输入的信息是否符合格式 |
| 增加用户 | boolean AddUser() | 根据用户输入的注册信息在数据库中增加新的用户 |
| 检查密码 | boolean CheckLogin(string account, string passwd) | 判断数据库中是否存在该账户且账户密码是否正确，正确则返回用户id |

4.2 用户信息管理



**图4.2 用户信息管理构件及其接口示意图**

4.2.1 用户信息管理构件详细设计

**表4.2.1 用户信息管理构件详细设计表**

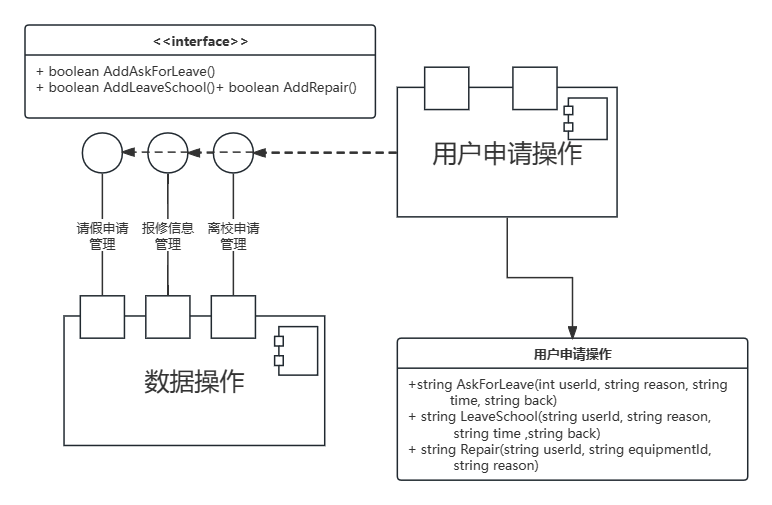
|  |  |  |
| --- | --- | --- |
| 构件名称 | 方法 | 方法描述 |
| 账号管理 | void PersonalPage() | 用户点击个人信息按钮，系统跳转至个人信息界面 |
| string ChangeMessage(string  passwd, string newPasswd, string userName, string  license, string phone) | 用户在个人信息界面输入新的用户信息，并申请修改用户信息 |

4.2.2 用户信息管理接口详细设计

**表4.2.2 用户信息管理接口详细设计表**

|  |  |  |
| --- | --- | --- |
| 接口名称 | 方法 | 方法描述 |
| 获取用户信息 | string GetUserMessage  (string userId) | 用户跳转个人信息页面后获取用户的信息进行展示 |
| 修改用户信息 | boolean UpdateUserMessage() | 根据用户输入的个人信息在数据库中修改用户信息 |

4.3 用户申请操作



**图4.3 用户申请操作构件及其接口示意图**

4.3.1 用户申请操作构件详细设计

**表4.3.1 用户申请操作构件详细设计表**

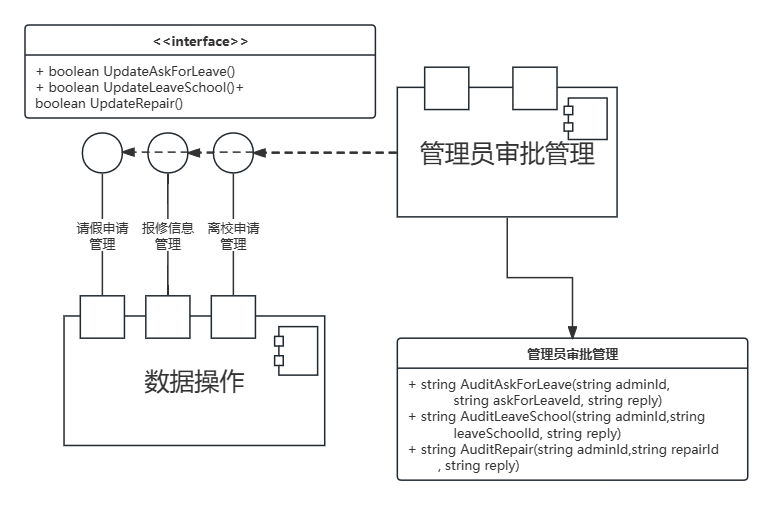
|  |  |  |
| --- | --- | --- |
| 构件名称 | 方法 | 方法描述 |
| 用户申请操作 | string AskForLeave(int userId, string reason, string time, string back) | 用户填写请假信息并向管理员发出申请 |
| string LeaveSchool(string userId, string reason, string time ,string back) | 用户填写离校信息并向管理员发出申请 |
| string Repair(string userId, string equipmentId, string reason) | 用户为指定设备填写报修请求并发送给管理员进行维护 |

4.3.2 用户申请操作接口详细设计

**表4.3.2 用户申请操作接口详细设计表**

|  |  |  |
| --- | --- | --- |
| 接口名称 | 方法 | 方法描述 |
| 添加请假信息 | boolean AddAskForLeave() | 根据用户输入的请假申请，在数据库增加新的请假记录 |
| 添加离校信息 | boolean AddLeaveSchool() | 根据用户输入的离校申请，在数据库增加新的离校记录 |
| 添加报修信息 | boolean AddRepair() | 根据用户输入的报修信息在数据库增加新的报修单，并修改设备状态 |

4.4 管理员审批管理



**图4.4 管理员审批管理构件及其接口示意图**

4.4.1 管理员审批管理详细设计

**表4.4.1 管理员审批管理构件详细设计表**

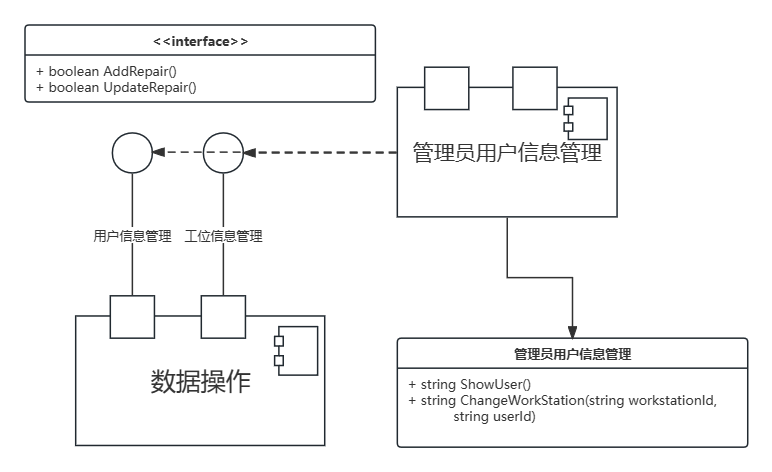
|  |  |  |
| --- | --- | --- |
| 构件名称 | 方法 | 方法描述 |
| 管理员审批 | string AuditAskForLeave(string adminId,string askForLeaveId  , string reply) | 管理员审批用户请假请求，并留言答复 |
| string AuditLeaveSchool(string adminId,string leaveSchoolId  , string reply) | 管理员审批用户离校请求，并留言答复 |
| string AuditRepair(string adminId,string repairId  , string reply) | 管理员检查报修信息，留言答复或恢复设备状态完成维修 |

4.4.2 管理员审批管理接口详细设计

**表4.4.2 管理员审批管理接口详细设计表**

|  |  |  |
| --- | --- | --- |
| 接口名称 | 方法 | 方法描述 |
| 添加请假信息 | boolean UpdateAskForLeave() | 在数据库中修改请假状态和添加管理员回复 |
| 添加离校信息 | boolean UpdateLeaveSchool() | 在数据库中修改请假状态和添加管理员回复 |
| 添加报修信息 | boolean UpdateRepair() | 在数据库中添加管理员回复，或修改维修状态和设备状态 |

4.5 管理员用户信息管理



**图4.5 管理员用户信息管理构件及其接口示意图**

4.5.1 管理员用户信息管理详细设计

**表4.5.1 管理员用户信息管理构件详细设计表**

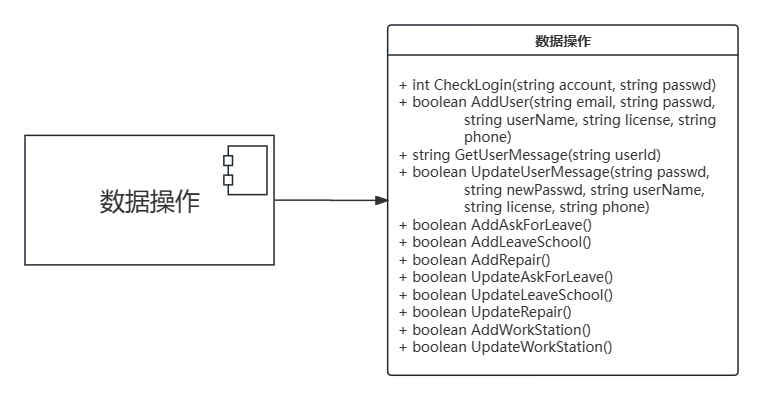
|  |  |  |
| --- | --- | --- |
| 构件名称 | 方法 | 方法描述 |
| 管理员用户信息管理 | string ShowUser() | 管理员查看用户的信息 |
| string ChangeWorkStation(string workstationId,string userId) | 管理员修改工位上绑定的用户 |

4.5.2 管理员用户信息管理接口详细设计

**表4.5.2 管理员用户信息管理接口详细设计表**

|  |  |  |
| --- | --- | --- |
| 接口名称 | 方法 | 方法描述 |
| 添加工位 | boolean AddRepair() | 在数据库中添加管理员所设定的工位 |
| 修改用户的工位 | boolean UpdateRepair() | 在数据库中修改工位所绑定的用户，并更新工位的分配时间 |

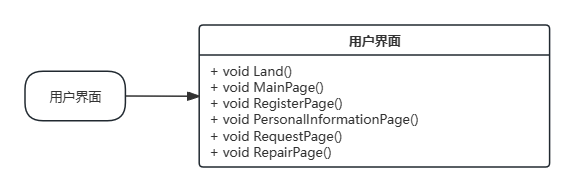
4.6 数据操作



**表4.6.2 数据操作接口详细设计表**

|  |  |  |
| --- | --- | --- |
| 接口名称 | 方法 | 方法描述 |
| 检查密码 | int CheckLogin(string account, string passwd) | 判断数据库中是否存在该账户且账户密码是否正确,正确则返回用户id |
| 增加用户 | boolean AddUser(string  email, string passwd, string userName, string license,  string phone) | 根据用户输入的注册信息在数据库中增加新的用户 |
| 获取用户信息 | string GetUserMessage(string userId) | 用户跳转个人信息页面后从数据库获取用户的个人信息用以进行展示 |
| 修改用户信息 | boolean UpdateUserMessage(string  passwd, string newPasswd, string userName, string  license, string phone) | 根据用户输入的个人信息在数据库中修改用户信息 |
| 添加请假信息 | boolean AddAskForLeave() | 根据用户输入的请假申请，在数据库增加新的请假记录 |
| 添加离校信息 | boolean AddLeaveSchool() | 根据用户输入的离校申请，在数据库增加新的离校记录 |
| 添加报修信息 | boolean AddRepair() | 根据用户输入的报修信息在数据库增加新的报修单，并修改设备状态 |
| 添加请假信息 | boolean UpdateAskForLeave() | 在数据库中修改请假状态和添加管理员回复 |
| 添加离校信息 | boolean UpdateLeaveSchool() | 在数据库中修改请假状态和添加管理员回复 |
| 添加报修信息 | boolean UpdateRepair() | 在数据库中添加管理员回复，或修改维修状态和设备状态 |
| 添加工位 | boolean AddWorkStation() | 在数据库中添加管理员所设定的工位 |
| 修改用户的工位 | boolean UpdateWorkStation() | 在数据库中修改工位所绑定的用户，并更新工位的分配时间 |

4.7 用户界面



|  |  |  |
| --- | --- | --- |
| 构件名称 | 方法 | 方法描述 |
| 读者界面 | void Land() | 显示登录界面 |
| void MainPage() | 显示主界面 |
| void RegisterPage() | 显示注册界面 |
| void PersonalInformationPage() | 用户登录后调用，显示个人信息界面 |
| void RequestPage() | 用户在主页面点击按钮后，跳转至用户请假/离校申请界面 |
| void RepairPage() | 用户在主页面点击按钮后，跳转至报修界面 |

4.8 管理员界面

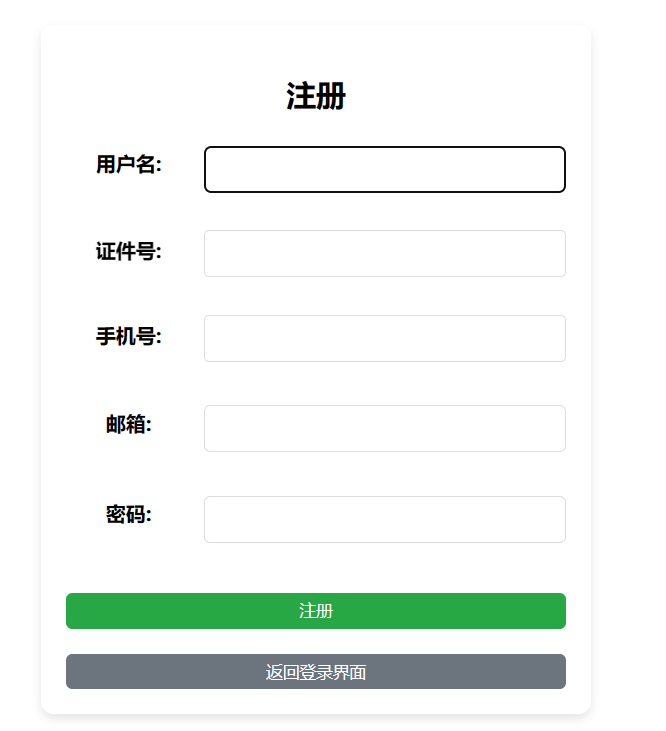


|  |  |  |
| --- | --- | --- |
| 构件名称 | 方法 | 方法描述 |
| 管理员界面 | void AdminPage() | 管理员用户登录后显示管理员界面 |
| void WorkStationPage() | 管理员在主界面点击后，跳转至工位界面，进行工位的创建和分配 |
| void UserManagementPage() | 管理员用户管理界面,管理员在主界面点击后可以进入此界面，进行用户信息查看和查询 |
| void AuditPage() | 管理员在主界面点击按钮后跳转至审批界面，对用户的请假/离校请求进行审批，或对报修进行处理 |

1. **界面设计**

5.1 注册界面

此界面需要用户输入用户名、证件号、手机号、邮箱和密码，然后通过点击下方注册按钮进行注册。



5.2 登录界面

此界面需要用户输入邮箱和密码，然后通过点击下方登录按钮进行登录。

5.3 用户主页

这个界面可以点击修改个人信息、请假申请、离校申请、设备报修。



5.4 请假界面

这个界面会显示以往的请假记录，可以填写请假原因，开始日期，结束日期后提交请假申请。



5.5 离校申请界面

这个界面会显示以往的离校记录，可以填写离校原因，开始日期，结束日期后提交离校申请。



5.6 报修界面

这个界面会显示以往的报修记录，可以填写设备和故障现象提交报修申请。



5.7 管理员界面

这个界面可以跳转审批界面，查看用户信息，工位管理。



5.8 审批界面

这个界面可以查看请假审批和离校审批并且进行批准和拒绝。



5.9 管理员用户信息查看界面

这个界面管理员可以查看用户信息。



5.10 工位管理界面

这个界面可以进行工位的修改，创建和删除。

