系统详细设计文档

系统详细设计文档
1. 引言
2. 总体设计
    2.1 需求总体描述
    2.2 概要设计总览
3. 开发环境
4. 功能函数设计
    管理员模块（admin.py）
    管理员测试模块（test_admin.py）
    图书查询模块（book_search.py）
    逾期图书查询模块（overdue_books.py ）
    读者信息查询模块（reader_query.py）
    登录注册模块（login.py）
    借书还书模块（borrowbook.py）

1. 引言

该项目开发的软件为图书馆管理系统，以应对当前阅读人数激增和信息爆炸增长的挑战。设计旨在实现图书管理的自动化和准确化，适用于各类教育单位（如学校、学院等）的学生信息管理需求。

在信息管理系统迅速发展的今天，各企事业单位纷纷引入信息管理软件来应对不断增长的信息管理需求。尽管商业化学生信息管理软件层出不穷，本系统独立开发，致力于提供简洁明了、功能齐全且易于操作的解决方案。

图书管理系统对于教育单位至关重要，不仅对图书馆决策者和管理者具有重要意义，而且实现了包括图书信息增删改查在内的完整功能，并提供用户端管理支持。

关键字：信息管理系统、模块设计、软件工程。

2. 总体设计

2.1 需求总体描述

系统需求围绕图书管理、用户（读者和管理员）管理、借阅管理等主要功能展开。

学生信息管理系统的功能总结起来，共需要以下几个方面：

1、注册、登录功能

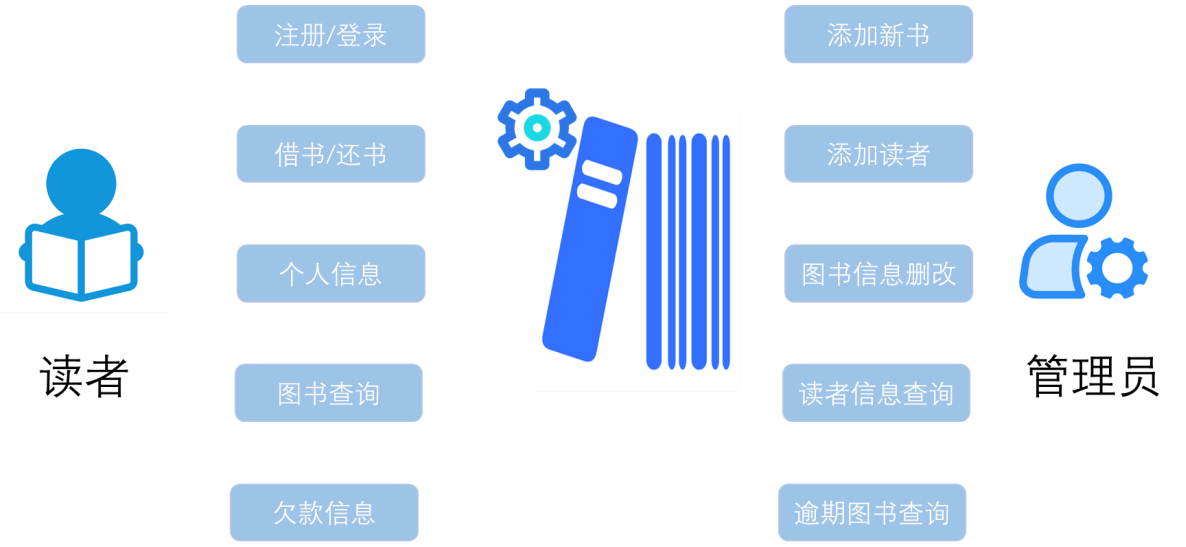
具有账号的操作者登录使用系统，且管理员和普通用户功能点不一样

2、图书信息管理

管理所有图书的基本信息，包括增加、修改、删除等，也可以根据各种条件查询出需要的信息。

3、借阅信息可视化

可视化显示所有的借阅信息，方便管理者管理



2.2 概要设计总览

1. **用户需求分析**:
   - 确定用户（管理员、读者）的基本需求。
   - 分析用户操作流程和功能需求。
2. **系统设计**:
   - 确定系统架构，通常包括前端展示层、后端逻辑层和数据库层。
3. **技术选型**:
   - 前端：使用 HTML 构建用户界面。
   - 后端：使用 Flask 作为 Web 服务器框架。
   - 数据库连接：使用 Pyodbc 作为 Python 与数据库之间的接口。
4. **数据库设计**:
   - 根据需求分析结果，设计数据库模型。
   - 创建必要的表结构，如用户表、图书表、借阅记录表等。
5. **前端页面开发**:
   - 使用 HTML、CSS 和 JavaScript 构建系统的前端页面。
   - 实现用户交互界面，如登录、图书搜索、借阅操作等。
6. **后端逻辑开发**:
   - 使用 Flask 框架编写后端逻辑。
   - 实现用户认证、图书管理、借阅管理等功能。
7. **数据库连接与操作**:
   - 利用 Pyodbc 库在 Flask 应用中连接数据库。
   - 执行 SQL 语句进行数据的增删改查操作。
8. **系统测试**:
   - 对系统进行单元测试、集成测试和系统测试。
   - 确保所有功能按预期工作，修复发现的问题。
9. **文档编写**:
   - 编写系统使用文档和开发文档。

- 确保用户和开发人员能够理解和使用系统。

3. 开发环境

数据库系统

SQL Server 2019

开发工具

PyCharm2023.2

其他支持要素

PowerDesigner: 绘制物理数据模型

飞书：项目管理、协同合作

4. 功能函数设计

管理员模块（admin.py）

- **add_reader函数**

  功能： 添加读者信息

  参数： library_card_number, name=None, gender=None, title=None, available_quantity=10, borrowed_quantity=0, department=None, contact_number=None

  调用此函数时library_card_number一定要给出，其余可以省略。

  其中available_quantity(可借数量)默认为10，borrowed_quantity(已借书数量)默认为0、

  返回值：

  - 成功

    ```
    {
        "success":True,
        "data":{
                "library_card_number": library_card_number,
                "name": name,
                "gender": gender,
                "title": title,
                "available_quantity": available_quantity,
                "borrowed_quantity": borrowed_quantity,
                "department": department,
                "contact_number": contact_number
                }
    }
    ```

  - 失败

```
except pyodbc.DatabaseError as e:
    ...
    return error(301, '添加读者失败: ' + str(e))
except Exception as e:
    return error(401, "错误: " + str(e))
```

- delete_reader函数

  功能：删除指定读者信息

  参数：library_card_number

  返回值：

  - 成功

    ```
    {
        "success":True,
        "data":{"library_card_number":library_card_number}
    }
    ```

  - 失败

    ```
    except pyodbc.DatabaseError as e:
        ...
        return error(301,'删除读者信息失败:' + str(e))
    except Exception as e:
        return error(401,"错误"+str(e))
    ```

- print_all_reader_info函数

  功能：打印整个reader_info表格

  参数：无

  返回值：

  - 成功

    ```
    {
        "success":True,
        "data":{"readers":[数组元素]}
    }
    # 数组元素
    {
    "library_card_number": library_card_number,
    "name": name,
    "gender": gender,
    "title": title,
    "available_quantity": available_quantity,
    "borrowed_quantity": borrowed_quantity,
    "department": department,
    "contact_number": contact_number
    }
    ```

    其中all_reader_info是一个字典类型，内容为reader_info表格

  - 失败

```
except pyodbc.DatabaseError as e:
    ...
    return error(301,'删除读者信息失败:' + str(e))
except Exception as e:
    return error(401,"错误"+str(e))
```

- get_reader_info函数

  功能：打印指定读者的信息

  参数：library_card_number

  返回值：

  - 成功

    ```
    {
        "success":True,
        "data":{
                "library_card_number": library_card_number,
                "name": name,
                "gender": gender,
                "title": title,
                "available_quantity": available_quantity,
                "borrowed_quantity": borrowed_quantity,
                "department": department,
                "contact_number": contact_number
                }
    }
    ```

    其中reader_info为一个字典类型，内容为借书证号为library_card_number的读者信息

  - 失败

    ```
        return error(20, "未找到该借书证号对应的读者信息")
    except pyodbc.DatabaseError as e:
        ...
        return error(301,'删除读者信息失败:' + str(e))
    except Exception as e:
        return error(401,"错误"+str(e))
    ```

- get_overdue_books函数

  功能：查询所有到期未归还的图书信息

  参数：无

  返回值：

  - 成功

```
{
    "success":True,
    "data":{"overdue_books":[数组元素]}
}
# 数组元素
{
"borrow_id": borrow_id,
"library_card_number": library_card_number,
"ISBN": ISBN,
"borrow_date": borrow_date,
"due_date": due_date,
"return_date":return_date
}
```

其中overdue_books_info是一个字典类型，内容为所有到期未归还的图书信息

- 失败

```
except pyodbc.DatabaseError as e:
    ...
    return error(301,'删除读者信息失败:' + str(e))
except Exception as e:
    return error(401,"错误"+str(e))
```

- get_reader_fines函数

  功能：查询所有读者的欠款状况

  参数：无

  返回值：

  - 成功

```
{
    "success":True,
    "data":{"readers_fines":[数组元素]}
}
# 数组元素
{
"library_card_number": library_card_number,
"name": name,
"total_fine": float(total_fine) if total_fine is not None else 0.0,

}
```

其中reader_fines_info是一个字典类型，内容为读者欠费状况

- 失败

```
except pyodbc.DatabaseError as e:
    ...
    return error(301,'删除读者信息失败:' + str(e))
except Exception as e:
    return error(401,"错误"+str(e))
```

管理员测试模块（test_admin.py）

- 运行test_admin_reader()的参考结果

```
Initial reader info:
Connection successful
{'library_card_number': 2, 'name': '张三', 'gender': '男', 'title': '学生', 'available_quantit
y': 10, 'borrowed_quantity': 0, 'department': '计算机科学与技术', 'contact_number': '123456789
01'}

Adding a new reader:
Connection successful
{'library_card_number': 1, 'name': '张三', 'gender': '男', 'title': '教授', 'available_quantit
y': 10, 'borrowed_quantity': 0, 'department': None, 'contact_number': '12345678901'}

Reader info after adding a new reader:
Connection successful
{'library_card_number': 1, 'name': '张三', 'gender': '男', 'title': '教授', 'available_quantit
y': 10, 'borrowed_quantity': 0, 'department': None, 'contact_number': '12345678901'}
{'library_card_number': 2, 'name': '张三', 'gender': '男', 'title': '学生', 'available_quantit
y': 10, 'borrowed_quantity': 0, 'department': '计算机科学与技术', 'contact_number': '123456789
01'}

Updating reader info:
Connection successful
读者信息更新成功
{'library_card_number': 1, 'name': '李四', 'gender': None, 'title': '副教授', 'available_quant
ity': 10, 'borrowed_quantity': 0, 'department': None, 'contact_number': '09876543210'}

Reader info after updating:
Connection successful
{'library_card_number': 1, 'name': '李四', 'gender': '男', 'title': '副教授', 'available_quant
ity': 10, 'borrowed_quantity': 0, 'department': None, 'contact_number': '09876543210'}
{'library_card_number': 2, 'name': '张三', 'gender': '男', 'title': '学生', 'available_quantit
y': 10, 'borrowed_quantity': 0, 'department': '计算机科学与技术', 'contact_number': '123456789
01'}


Getting reader info for library_card_number=1:
Connection successful
{'library_card_number': 1, 'name': '李四', 'gender': '男', 'title': '副教授', 'available_quant
ity': 10, 'borrowed_quantity': 0, 'department': None, 'contact_number': '09876543210'}

Getting reader info for library_card_number=200 (non-existent):
Connection successful
{'success': False, 'status': 20, 'message': '未找到该借书证号对应的读者信息'}

Deleting reader with library_card_number=1:
Connection successful
{'success': True, 'data': {'library_card_number': 1}}

Reader info after deleting the reader:
Connection successful
{'library_card_number': 2, 'name': '张三', 'gender': '男', 'title': '学生', 'available_quantit
y': 10, 'borrowed_quantity': 0, 'department': '计算机科学与技术', 'contact_number': '123456789
01'}
```

添加读者信息

验证

更新读者信息

验证

查询指定读者信息(存在)

查询指定读者信息(不存在)

删除读者信息

验证

- 运行test_admin_book_borrow()参考结果

首先插入几条示例借阅信息(已更新在github仓库中的sql文件中)

```
-- 插入借阅信息
INSERT INTO borrow_info (library_card_number, ISBN, borrow_date, due_date,
return_date, fine)
VALUES
(2, '978-3-16-148410-0', '2023-01-01', '2023-01-15', '2023-01-14', 0.00),--过期已
还
(2, '978-0-12-345678-9', '2023-02-01', '2023-02-15', NULL, 12.00),--过期未还
(2, '978-1-23-456789-0', '2023-01-01', '2025-01-15', NULL, 0.00) --未过期且未还
```

```
Connection successful
{'borrow_id': 4, 'library_card_number': 2, 'ISBN': '978-0-12-345678-9', 'borrow_date': '2023-0
2-01', 'due_date': '2023-02-15', 'return_date': None}
Connection successful
{'library_card_number': 2, 'name': '张三', 'total_fine': 12.0}
```

只有过期未还的借阅信息需要被打印出来，结果相符。

图书查询模块（book_search.py）

- **search_books 函数**

功能： 搜索图书，搜索关键词可以是书籍的ISBN号、作者、出版社和书名

参数： 输入字符串，可以是 ISBN，author，publisher 或者 book_title

返回值：

- 成功

```
for book in overdue_books:
    overdue_books_data.append({
        "borrow_id": book.borrow_id,
        "library_card_number": book.library_card_number,
        "ISBN": book.ISBN,
        "borrow_date": book.borrow_date,
        "due_date": book.due_date,
        "overdue_days": book.overdue_days,
        "fine": book.fine
     })
return success(books)
```

- 失败

```
if not results:
    return error(404, "图书信息未找到")

except pyodbc.DatabaseError as e:
    ...
    return error(301, '查询失败: ' + str(e))
except Exception as e:
    return error(401, "错误: " + str(e))
```

运行测试：

查找成功：

```
{'success': True, 'data': [{'ISBN': '978-3-16-148410-0', 'book_title': '图书1', 'publisher': '出版社A', 'author': '作者A', 'total_quantity': 5, 'available_quantity': 5, 'is_borrowable': True}]}
```

没有图书记录：`{'success': False, 'status': 404, 'message': '图书信息未找到'}`

逾期图书查询模块（overdue_books.py ）

- **get_overdue_books 函数**

  功能：查询未归还图书信息

  参数： library_card_number, name=None, gender=None, title=None, available_quantity=10, borrowed_quantity=0, department=None, contact_number=None

  调用此函数时library_card_number一定要给出，其余可以省略。

  其中available_quantity(可借数量)默认为10，borrowed_quantity(已借书数量)默认为0、

  返回值：

  - 成功

    ```
    for row in results:
        books.append({
            "ISBN": row.ISBN,
            "book_title": row.book_title,
            "publisher": row.publisher,
            "author": row.author,
             "total_quantity": row.total_quantity,
            "available_quantity": row.available_quantity,
            "is_borrowable": row.is_borrowable
    })
    return success(books)
    ```

  - 失败

    ```
    except pyodbc.DatabaseError as e:
        ...
        return error(301, '查询失败: ' + str(e))
    except Exception as e:
        return error(401, "错误: " + str(e))
    ```

  运行测试：

  查找成功：

  ```
  {'success': True, 'data': [{'borrow_id': 2, 'library_card_number': 2, 'ISBN': '978-0-12-345678-9', 'borrow_date': '2023-02-01', 'due_date': '2023-02-15', 'overdue_days': 490, 'fine': Decimal('12.00')}]}
  ```

读者信息查询模块（reader_query.py）

- **get_reader_info 函数**

  功能： 查询读者个人信息和读者借书信息

  参数： 输入 library_card_number

  返回值：

  - 成功

```
reader_data = {
    "library_card_number": reader_info.library_card_number,
    "name": reader_info.name,
    "gender": reader_info.gender,
    "title": reader_info.title,
    "available_quantity": reader_info.available_quantity,
    "borrowed_quantity": reader_info.borrowed_quantity,
    "department": reader_info.department,
    "contact_number": reader_info.contact_number
    }
...
for book in borrowed_books:
    borrowed_books_data.append({
        "borrow_id":book.borrow_id,
        "ISBN": book.book_ISBN,
        "book_title": book.book_title,
        "author": book.author,
        "borrow_date": book.borrow_date,
        "due_date": book.due_date,
        "return_date": book.return_date,
        "fine": book.fine
     })

return success({
    "reader_info": reader_data,
    "borrowed_books": borrowed_books_data,
})
```

  - 失败

```
if not reader_info:
    return error(404, "读者信息未找到")

except pyodbc.DatabaseError as e:
        cursor.rollback()
        cursor.close()
        cnxn.close()
        return error(301, '查询失败: ' + str(e))
except Exception as e:
        return error(401, "错误: " + str(e))
```

  运行测试：

  查询成功：

{'success': True, 'data': {'reader_info': {'library_card_number': 2, 'name': '张三', 'gender': '男', 'title': '学生', 'available_quantity': 10, 'borrowed_quantity': 0, 'department': '计算机科学与技术', 'contact_number': '12345678901'}, 'borrowed_books': [{'ISBN': '978-0-12-345678-9', 'book_title': '图书2', 'author': '作者B', 'borrow_date': '2023-02-01', 'due_date': '2023-02-15', 'return_date': None, 'fine': Decimal('12.00')}, {'ISBN': '978-1-23-456789-0', 'book_title': '图书3', 'author': '作者C', 'borrow_date': '2023-01-01', 'due_date': '2025-01-15', 'return_date': None, 'fine': Decimal('0.00')}]}}

没有该读者记录： `{'success': False, 'status': 404, 'message': '读者信息未找到'}`

登录注册模块（login.py）

- **register 函数**

  功能：进行注册

  参数：name->要注册的姓名，password->要注册的密码，title->职称，phone_number->电话号码，department->系别，gender->性别，is_root->是否为管理员

  返回值：

  - 成功

```
{
  "success":True,
  "data":account
}
```

  - 失败（代码与说明）
    - 1 :姓名为空错误
    - 2 :密码为空错误
    - 3 :姓名长度过长
    - 4 :密码长度过长
    - 10:权限设置错误
    - 20:身份长度过长
    - 21:性别不为男也不为女
    - 22:系别长度过长
    - 23:电话长度过长
    - 301:数据库操作错误
    - 401:意外错误
- **login函数**

  功能：登录

  参数：account->账号，password->密码

  返回值：

  - 成功

```
{
  "success":True,
  "data":{
    "role":role,
    "name":name
  }
}
```

- 失败
  - 1： 用户名或密码错误
  - 301：数据库部分错误
  - 401：意外错误
  -

借书还书模块（borrowbook.py）

- **borrow_book函数**

  功能：借书

  参数：library_card_number->借书号/账号,isbn->借书的isbn号,due_date->到期时间,begin_date->起始时间，默认为当天

  返回值：

  - 成功

```
{
  "success":True,
  "data":{
          "library_card_number":library_card_number,
          "isbn":isbn,
          "begin_date":begin_date,
          "due_date":due_date
      }
}
```

  - 失败
    - 1： isbn不存在错误
    - 2： library_card_number不存在错误
    - 3： 借书超过上限错误
    - 4： 有未交罚金或有未在时间内归还的图书
    - 5： 重复借同一本书
    - 6： 书籍不可借
    - 301：数据库操作错误
    - 401：意外错误
- **return_book函数**
  功能：还书，并提醒快过期和已经过期的书单

  参数：library_card_number->借书号/账号,isbn->借书的isbn号

  返回值：

  - 成功

```
{
  "success":True,
  "data":[
    [isbn,due_date,fine],
    [isbn,due_date,fine],
    ...
  ]
}
```

- 失败
  - 1： 借书记录不存在错误
  - 301：数据库操作错误
  - 401：意外错误