

信息工程学院 2023-2024 第一学期 集中实践报告

——《课程设计 I》

图书管理系统开发文档

学生姓名：_____何炜_____

学生学号：_____2022010611_____

学生姓名：_____和广洋_____

学生学号：_____2022060407_____

学生姓名：_____杜志成_____

学生学号：_____2022010610_____

所在班级：_____计科 226_____

所学专业：_____计算机科学与技术_____

完成时间：_____2023-12_____

图书管理系统

摘 要

这项目实现了一个图书馆管理系统，分为用户和管理员两个登录模式。用户能够浏览图书信息、借阅和归还图书，而管理员具备对图书信息的增、删、改、查等管理权限。系统采用 CSV 文件存储用户和图书信息，通过 Python 编写核心逻辑，包括用户注册、登录，图书管理和相应交互功能。通过命令行界面，用户可以方便地执行各项操作，实现了一个基本的图书馆管理系统。

关键词： 图书馆管理系统、用户注册、用户登录、管理员权限、图书信息、CSV 文件、图书借阅、图书归还。

目录

摘 要	2
1 需求分析	3
1.1 系统目标	4
1.2 用户类型	4
1.3 功能需求	4
1.4 文件存储	4
2 系统功能结构	4
2.1 文件架构	5
2.2 模块说明	6
2.2.1 main 模块	6
3 系统实现	6
3.1 模块说明	7
3.1.1 book 模块	7
3.1.2 login 模块	9
4 系统测试	10
4.1 登录界面	11
4.2 注册页面	11
4.2 普通用户界面	12
4.3 管理员界面	13
5 结论	14

1 需求分析

1.1 系统目标

该系统的主要目标是实现一个简单的图书馆管理系统，包括用户登录、图书管理（列出、增加、删除、更新、借阅和归还书籍）、管理员权限等功能。

1.2 用户类型

系统支持两种用户类型：普通用户（读者）和管理员。普通用户可以浏览图书、借阅和归还书籍，而管理员则具有对图书进行增加、删除、更新等管理权限。

1.3 功能需求

系统需要实现以下主要功能：

用户登录和注册

列出所有书籍信息

借阅和归还书籍

管理员权限下的图书管理（新增、删除、修改）

1.4 文件存储

在系统的实现中，我们选择使用 CSV 文件作为用户信息和图书信息的存储载体。具体而言，用户信息存储在名为 `users.csv` 的文件中，而图书信息则存储在名为 `books.csv` 的文件中。这种文件存储方式具有一定的优势和方便之处。

users.csv 文件：

文件用途：存储用户的基本信息，包括用户名、密码、姓名、班级名和电话号码。

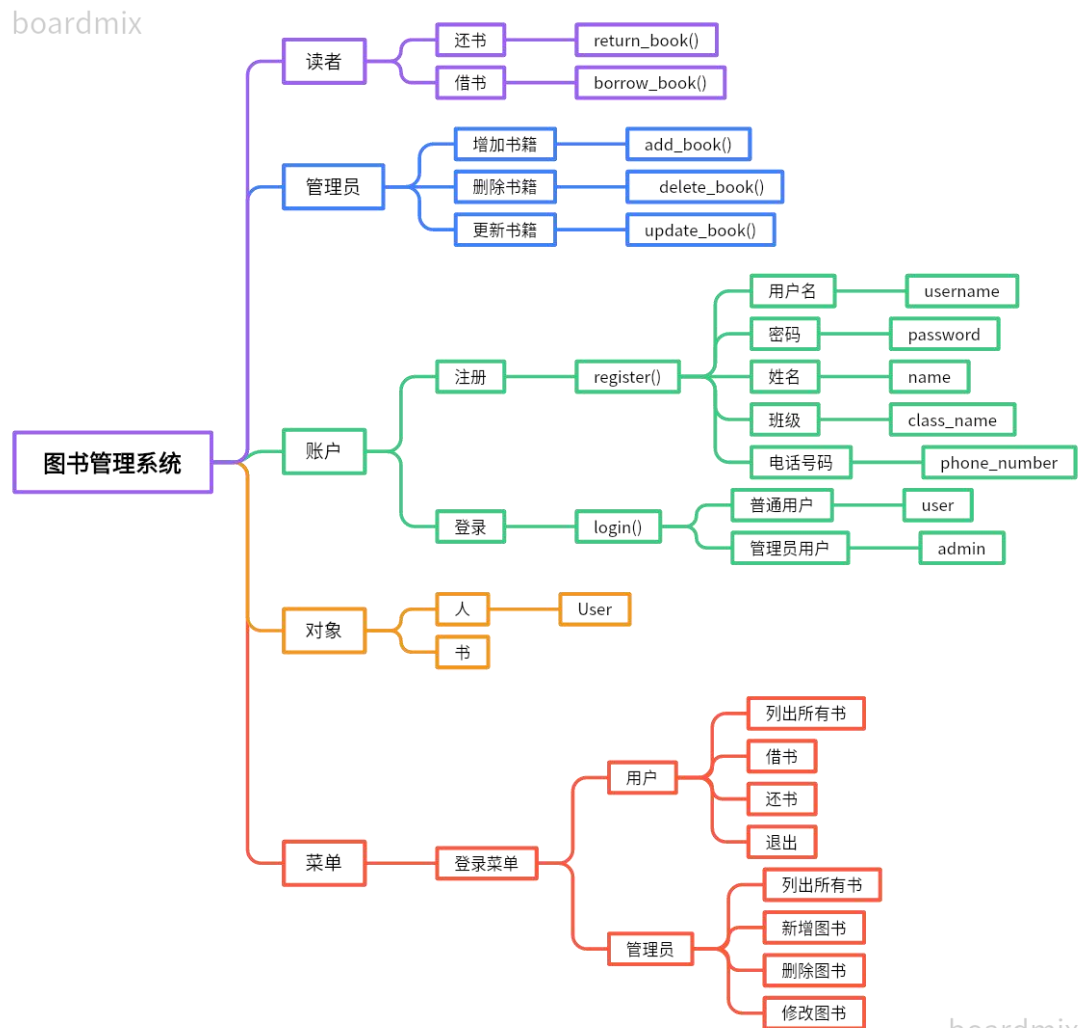
数据结构：每一行代表一个用户的信息，不同字段使用逗号分隔。文件第一行为表头，包括 "UID"、"PSK"、"Name"、"cid" 和 "Phone" 等字段。

books.csv 文件：

文件用途：存储图书馆中书籍的基本信息，包括书名、作者和数量。

数据结构：每一行代表一本书的信息，不同字段使用逗号分隔。文件第一行为表头，包括 "Title"、"Author" 和 "Num" 等字段。

2 系统功能结构



2.1 文件架构

系统的文件架构包括以下文件：

main.py: 主程序模块

login.py: 用户登录和注册模块

book.py: 图书管理模块

users.csv: 存储用户信息的 CSV 文件

books.csv: 存储图书信息的 CSV 文件

2.2 模块说明

2.2.1 main 模块

```
(1) check_file_exists()
def check_file_exists():
    user_csv_path = os.path.join(os.path.dirname(os.path.abspath(__file__)),
    'users.csv')
    books_csv_path = os.path.join(os.path.dirname(os.path.abspath(__file__)),
    'books.csv')
    if not os.path.isfile(user_csv_path):
        with open(user_csv_path, 'w', newline='') as f:
            writer = csv.writer(f)
            writer.writerow(["UID", "PSK", "Name", "cid", "Phone"])
    if not os.path.isfile(books_csv_path):
        with open(books_csv_path, 'w', newline='') as f:
            writer = csv.writer(f)
            writer.writerow(["Title", "Author", "Num"])
```

检查用户信息和图书信息的 CSV 文件是否存在，如果不存在，则创建对应的 CSV 文件并写入表头。

3 系统实现

3.1 模块说明

3.1.1 book 模块

(1) list_books():

打开图书 CSV 文件，以只读方式读取并打印所有书籍的信息。

```
def list_books():
```

```
    with open(books_csv_path, 'r') as f:
```

```
        reader = csv.reader(f)
```

```
        for row in reader:
```

```
            print(row)
```

(2) add_book():

用户输入书名、作者和数量，将这些信息追加到图书 CSV 文件中。

```
def add_book():
```

```
    book_name = input("请输入书名： ")
```

```
    book_author = input("请输入作者： ")
```

```
    book_quantity = input("请输入数量： ")
```

```
    with open(books_csv_path, 'a', newline="") as f:
```

```
        writer = csv.writer(f)
```

```
        writer.writerow([book_name, book_author, book_quantity])
```

(3) delete_book():

用户输入要删除的书名，删除图书 CSV 文件中对应的书籍信息。

```
def delete_book():
```

```
    book_name = input("请输入要删除的书名： ")
```

```
    lines = []
```

```
    with open(books_csv_path, 'r') as f:
```

```
        reader = csv.reader(f)
```

```
        lines = [row for row in reader if row[0] != book_name]
```

```
    with open(books_csv_path, 'w', newline="") as f:
```

```
        writer = csv.writer(f)
```

```
        writer.writerows(lines)
```

(4) update_book():

用户输入要更新的书名，然后输入新的作者和数量，更新图书 CSV 文件中对应书籍的信息。

```

def update_book():
    book_name = input("请输入要更新的书名： ")
    lines = []
    with open(books_csv_path, 'r') as f:
        reader = csv.reader(f)
        lines = [row for row in reader if row[0] != book_name]
    book_author = input("请输入新的作者： ")
    book_quantity = input("请输入新的数量： ")
    lines.append([book_name, book_author, book_quantity])
    with open(books_csv_path, 'w', newline="") as f:
        writer = csv.writer(f)
        writer.writerows(lines)

(5) borrow_book():
    用户输入要借阅的书名，从图书 CSV 文件中找到对应书籍，如果数量
    大于 0，则借阅成功并减少数量；否则，提示该书已被借完。

def borrow_book():
    book_name = input("请输入要借阅的书名： ")
    lines = []
    with open(books_csv_path, 'r') as f:
        reader = csv.reader(f)
        lines = [row for row in reader]
    for row in lines:
        if row[0] == book_name:
            if int(row[2]) > 0:
                row[2] = str(int(row[2]) - 1)
                print("借书成功！ ")
            else:
                print("这本书已经被借完了。")
            break
    with open(books_csv_path, 'w', newline="") as f:
        writer = csv.writer(f)
        writer.writerows(lines)

```


(6) return_book():

用户输入要归还的书名，系统检查图书馆中是否存在该书籍。如果存在，则将该书的数量加 1，表示图书归还成功，并提示用户还书成功。如果该书不存在，则系统会提示用户该书不存在。最后，系统将更新后的图书信息写入 CSV 文件。

```
def return_book():
    book_name = input("请输入要归还的书名： ")
    lines = []
    with open(books_csv_path, 'r') as f:
        reader = csv.reader(f)
        lines = [row for row in reader]
    for row in lines:
        if row[0] == book_name:
            row[2] = str(int(row[2]) + 1)
            print("还书成功！ ")
            break
    with open(books_csv_path, 'w', newline="") as f:
        writer = csv.writer(f)
        writer.writerows(lines)
```

3.1.2 login 模块

(1) register():

- 获取用户输入的用户名、密码、姓名、班级和电话号码。
- 创建新用户对象，并将其信息追加到用户 CSV 文件中。
- 返回新用户对象。

```
def register():
    new_user = User(username, password, name, class_name,
phone_number)
    with open(os.path.join(os.path.dirname(os.path.abspath(__file__)), 'users.csv'),'a',
newline="") as f:
        writer = csv.writer(f)
        writer.writerow([new_user.username, new_user.password,
new_user.name, new_user.class_name, new_user.phone_number])
    return new_user
```

(2) login():

- 获取用户输入的用户名和密码。
 - 如果用户名为 'admin' 且密码为 '123456', 返回 'admin'。
 - 打开用户 CSV 文件, 遍历文件内容, 如果输入的用户名和密码正确, 返回 'user'。
 - 如果用户名和密码不正确, 返回 None。

def login():

```
    username = input("请输入用户名: ")
    password = input("请输入密码: ")
    if username == 'admin' and password == '123456':
        return 'admin'
    with open(os.path.join(os.path.dirname(os.path.abspath(__file__)), 'users.csv'),
'r') as f:
        reader = csv.reader(f)
        for row in reader:
            if row[0] == username and row[1] == password:
                return 'user'
    return None
```

4 系统测试

4.1 登录界面

欢迎来到图书馆

—— 用户操作 ——

1. 列出所有书
2. 借书
3. 还书
4. 退出

—— 管理员操作 ——

1. 列出所有书
2. 新增图书
3. 修改图书
4. 删除图书
5. 退出

1 登录 2 注册 3 退出

请选择：

4.2 注册页面

```
UID, Col 3: Name ,Phone
1,1,1,1,1
USER1,123456,John,226,18888888888
```

1 登录 2 注册 3 退出

请选择：2

请输入用户名：dzcnb666

请输入密码：ddd

请输入姓名：ddzc

请输入班级：jk26

请输入电话号码：55555

```
UID,PSK,Name,cid,Phone
1,1,1,1,1
USER1,123456,John,226,18888888888
dzcnb666,ddd,ddzc,jk26,55555
```

4.2 普通用户界面

```
1 登录 2 注册 3 退出  
  
请选择: 1  
请输入用户名: 1  
请输入密码: 1  
登录成功!  
1. 列出所有的书  
2. 借书  
3. 还书  
4. 退出  
请选择: █
```

功能 1 列出书籍展示

```
1. 列出所有的书  
2. 借书  
3. 还书  
4. 退出  
请选择: 1  
['Title', 'Author', 'Num']  
['tree', 'ok', '9']  
['tree', 'ok', '41']
```

功能 2 借书展示

```
1. 列出所有的书  
2. 借书  
3. 还书  
4. 退出  
请选择: 2  
请输入要借阅的书名: tree  
借书成功!
```

功能 3 还书展示

```
Title, Author, Num  
tree, ok, 9  
tree, ok, 41  
  
1. 列出所有的书  
2. 借书  
3. 还书  
4. 退出  
请选择: 3  
请输入要归还的书名: tree  
还书成功!  
  
Title, Author, Num  
tree, ok, 10  
tree, ok, 41
```

4.3 管理员界面

默认账户 admin 密码 123456

```
1 登录 2 注册 3 退出

请选择: 1
请输入用户名: admin
请输入密码: 123456
管理员登录成功!
1. 列出所有的书
2. 新增图书
3. 修改图书
4. 删除图书
5. 退出
请选择: █
```

功能 1 列出所有书

```
1. 列出所有的书
2. 新增图书
3. 修改图书
4. 删除图书
5. 退出
请选择: 1
['Title', 'Author', 'Num']
['tree', 'ok', '8']
['tree', 'ok', '41']
```

功能 2 新增图书

```
1. 列出所有的书
2. 新增图书
3. 修改图书
4. 删除图书
5. 退出
请选择: 2
请输入书名: nb666
请输入作者: ddzc
请输入数量: 10

Title,Author,Num
tree,ok,8
tree,ok,41
nb666,ddzc,10
```

功能 3 修改图书

```
1. 列出所有的书
2. 新增图书
3. 修改图书
4. 删除图书
5. 退出
请选择: 3
请输入要更新的书名: nb666
请输入新的作者: ddzc
请输入新的数量: 100
```

```
Title,Author,Num
tree,ok,8
tree,ok,41
nb666,ddzc,100
```

功能 4 删除图书

```
Title,Author,Num
tree,ok,8
tree,ok,41
nb666,ddzc,100
```

```
1. 列出所有的书
2. 新增图书
3. 修改图书
4. 删除图书
5. 退出
请选择: 4
请输入要删除的书名: nb666
```

```
Title,Author,Num
tree,ok,8
tree,ok,41
```

5 结论

在完成本次任务中，我们成功地设计并实现了一个简单的图书管理系统，该系统包括用户注册、登录以及图书的增加、删除、更新、借阅和归还等功能。通过这个项目，我们深入理解了 Python 中文件操作、类的使用以及 CSV 文件的读写等方面的知识。我们学会了如何组织和管理代码，提高了问题解决能力和编码实践经验。在已实现功能方面，系统能够准确处理用户的注册和登录请求，保证用户信息的安全性。图书管理方面，系统支持书籍的增加、删除、更新以及借阅和归还等基本操作，有效管理图书馆的图书信息。

然而，系统仍存在一些需要改进和完善的地方。例如，对于输入的合法性和错误处理方面可以加强，系统还可以添加更多的辅助功能，提升用户体验。未来，我计划进一步优化系统代码，提高系统的稳定性和可靠性。通过这个项目，我不仅学到了具体的技术知识，也提高了对软件开发流程的理解和把控能力。

附录:

main.py:

```
import login

import book as book

import os

import csv

def check_file_exists():

    user_csv_path = os.path.join(os.path.dirname(os.path.abspath(__file__)), 'users.csv')

    books_csv_path = os.path.join(os.path.dirname(os.path.abspath(__file__)), 'books.csv')

    if not os.path.isfile(user_csv_path):

        with open(user_csv_path, 'w', newline='') as f:

            writer = csv.writer(f)

            writer.writerow(["UID", "PSK", "Name", "cid", "Phone"])

    if not os.path.isfile(books_csv_path):

        with open(books_csv_path, 'w', newline='') as f:

            writer = csv.writer(f)

            writer.writerow(["Title", "Author", "Num"])

def main():

    check_file_exists()

    while True:

        print(
"""

        欢迎来到图书馆

        ┌────────── 用户操作 ─────────┐
        | 1. 列出所有书                |
        | 2. 借书                      |
        | 3. 还书                      |
        | 4. 退出                      |
        └────────────────────────────────┘

        ┌────────── 管理员操作 ─────────┐
        | 1. 列出所有书                |
        | 2. 新增图书                  |
        └────────────────────────────────┘
"""
)
```

```

| 3. 修改图书          |
| 4. 删除图书          |
| 5. 退出              |
|_____|

1 登录 2 注册 3 退出
"""
)

choice = input("请选择: ")

if choice == '1':
    user_type = login.User.login()
    if user_type == 'user':
        print("登录成功! ")
        while True:
            print("1. 列出所有的书")
            print("2. 借书")
            print("3. 还书")
            print("4. 退出")
            choice = input("请选择: ")
            if choice == '1':
                book.list_books()
            elif choice == '2':
                book.borrow_book()
            elif choice == '3':
                book.return_book()
            elif choice == '4':
                break
    elif user_type == 'admin':
        print("管理员登录成功! ")
        while True:
            print("1. 列出所有的书")
            print("2. 新增图书")
            print("3. 修改图书")
            print("4. 删除图书")
            print("5. 退出")

```



```

        choice = input("请选择: ")

        if choice == '1':

            book.list_books()

        elif choice == '2':

            book.add_book()

        elif choice == '3':

            book.update_book()

        elif choice == '4':

            book.delete_book()

        elif choice == '5':

            break

    else:

        print("登录失败! ")

elif choice == '2':

    user = login.User.register()

elif choice == '3':

    break

if __name__ == "__main__":

    main()

```

login.py:

```

import csv

import os

class User:

    def __init__(self, username, password, name, class_name, phone_number):

        self.username = username

        self.password = password

        self.name = name

        self.class_name = class_name

        self.phone_number = phone_number

    def register():

        username = input("请输入用户名: ")

        password = input("请输入密码: ")

        name = input("请输入姓名: ")

        class_name = input("请输入班级: ")

```

```

        phone_number = input("请输入电话号码: ")

        new_user = User(username, password, name, class_name, phone_number)

        with open(os.path.join(os.path.dirname(os.path.abspath(__file__)), 'users.csv'), 'a', newline='')
as f:

            writer = csv.writer(f)

            writer.writerow([new_user.username,          new_user.password,          new_user.name,
new_user.class_name, new_user.phone_number])

            return new_user

def login():

    username = input("请输入用户名: ")

    password = input("请输入密码: ")

    if username == 'admin' and password == '123456':

        return 'admin'

    with open(os.path.join(os.path.dirname(os.path.abspath(__file__)), 'users.csv'), 'r') as f:

        reader = csv.reader(f)

        for row in reader:

            if row[0] == username and row[1] == password:

                return 'user'

    return None

```

Book.py:

```

import csv

import os

books_csv_path = os.path.join(os.path.dirname(os.path.abspath(__file__)), 'books.csv')

def list_books():

    with open(books_csv_path, 'r') as f:

        reader = csv.reader(f)

        for row in reader:

            print(row)

def add_book():

    book_name = input("请输入书名: ")

    book_author = input("请输入作者: ")

    book_quantity = input("请输入数量: ")

    with open(books_csv_path, 'a', newline='') as f:

        writer = csv.writer(f)

```

```

        writer.writerow([book_name, book_author, book_quantity])

def delete_book():
    book_name = input("请输入要删除的书名: ")
    lines = []
    with open(books_csv_path, 'r') as f:
        reader = csv.reader(f)
        lines = [row for row in reader if row[0] != book_name]
    with open(books_csv_path, 'w', newline='') as f:
        writer = csv.writer(f)
        writer.writerows(lines)

def update_book():
    book_name = input("请输入要更新的书名: ")
    lines = []
    with open(books_csv_path, 'r') as f:
        reader = csv.reader(f)
        lines = [row for row in reader if row[0] != book_name]
    book_author = input("请输入新的作者: ")
    book_quantity = input("请输入新的数量: ")
    lines.append([book_name, book_author, book_quantity])
    with open(books_csv_path, 'w', newline='') as f:
        writer = csv.writer(f)
        writer.writerows(lines)

def borrow_book():
    book_name = input("请输入要借阅的书名: ")
    lines = []
    with open(books_csv_path, 'r') as f:
        reader = csv.reader(f)
        lines = [row for row in reader]
    for row in lines:
        if row[0] == book_name:
            if int(row[2]) > 0:
                row[2] = str(int(row[2]) - 1)
                print("借书成功!")
            else:

```

```

        print("这本书已经被借完了。")

        break

    with open(books_csv_path, 'w', newline='') as f:
        writer = csv.writer(f)
        writer.writerows(lines)

def return_book():
    book_name = input("请输入要归还的书名: ")
    lines = []

    with open(books_csv_path, 'r') as f:
        reader = csv.reader(f)
        lines = [row for row in reader]

    for row in lines:
        if row[0] == book_name:
            row[2] = str(int(row[2]) + 1)
            print("还书成功! ")
            break

    with open(books_csv_path, 'w', newline='') as f:
        writer = csv.writer(f)
        writer.writerows(lines)

```