中山大學

SUN YAT-SEN UNIVERSITY

软件工程项目报告

基于 cursor 辅助的图书管理软件开发

姓 名 饶鹏程

学 号 23336201

学 院 计算机学院

专 业 计算机科学与技术

2025 年 4 月 12 日

目录

一 项目介绍

本软件是一款基于 cursor ai agent 辅助开发的图书管理软件，具备添加图书、多种方式查找图书、借阅或归还图书、删除图书等基本功能。支持.txt、.pdf、.epub 等多种格式的图书上传，可以添加自定义封面。本项目的 github 仓库地址：https://github.com/seevilador/library

（一） 软件结构

本软件采用 Flask 框架开发，采用 MVC 架构模式，主要包含以下几个部分：

- **Model**：使用 SQLAlchemy ORM 框架实现，包含 Book 数据，存储图书的所有属性和关系，包括：
  - 基本属性：标题、作者、ISBN、出版日期、分类、描述
  - 文件信息：电子书文件路径、封面图片路径
  - 借阅信息：借出状态、借阅者、借阅日期、归还日期
  - 时间戳：创建时间、更新时间
- **View**：使用 HTML 实现，位于 templates 目录下，包括：
  - index.html：首页，显示图书列表和搜索功能
  - 其他页面
- **Controller**：在 app.py 中实现，包含主要功能：
  - 图书管理和编辑：添加、编辑、删除图书
  - 借阅管理：借出、归还图书
  - 文件处理：上传、下载电子书和封面图片
  - 搜索图书：支持按标题、作者、ISBN、分类搜索
- 静态资源：位于 static 目录，包括：
  - CSS 文件
  - JS 脚本
  - 图片（封面）

- 数据存储：
  - 数据库：使用 SQL 存储图书信息和属性
  - 文件：存储电子书文件和封面图片

软件的文件结构如下：

```
library/
├── app.py                # 主程序
├── database.py           # 数据库
├── requirements.txt      # 依赖包列表
├── templates/            # HTML
├── static/               # 静态资源
│   ├── css/              # 样式
│   ├── js/               # JS文件
│   └── covers/           # 封面图片
├── uploads/              # 上传的电子书文件
└── .specstory/           # 开发历史记录
```

二　软件开发

（一）　开发流程概述

本软件是基于 ai agent 的代码实现功能，由开发者任务驱动开发的软件。抽象的开发过程为开发者分析软件功能，得出开发需求并提供给 ai 编程实现，开发者再对代码进行测试并提出改进或新需求的交替实现。

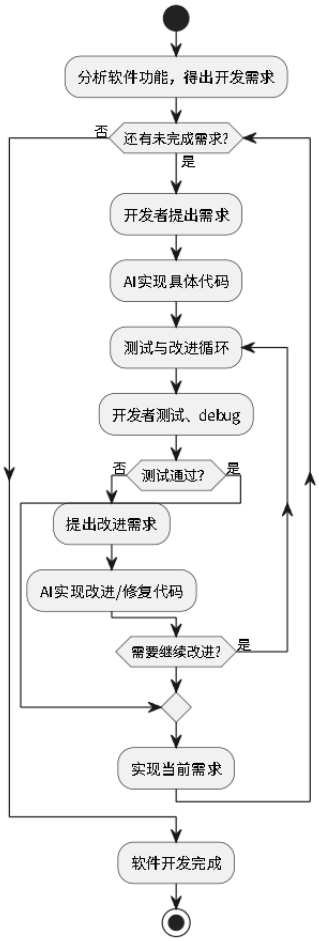

图 1: 软件开发流程图

与 ai agent 对话辅助开发的完整历史存放在\.specstory\history 中。

三　功能实现

（一）　添加图书功能

添加图书功能主要通过以下代码实现：

```
@app.route('/add', methods=['POST'])
def add_book():
    try:
        # 获取表单数据
        title = request.form.get('title')
        author = request.form.get('author')
        isbn = request.form.get('isbn')
```

```
        publish_date_str = request.form.get('publish_date')
        category = request.form.get('category')
        description = request.form.get('description')

        # 处理图书文件
        file_path = None
        if 'file' in request.files:
            file = request.files['file']
            if file.filename != '' and allowed_file(file.
               filename):
                filename = secure_filename(file.filename)
                file_path = os.path.join(app.config['UPLOAD_
                   FOLDER'], filename)
                file.save(file_path)

        # 处理封面图片
        cover_path = None
        if 'cover' in request.files:
            cover = request.files['cover']
            if cover.filename != '' and allowed_image(cover.
               filename):
                cover_filename = secure_filename(cover.filename
                   )
                cover_path = f"covers/{cover_filename}"
                full_cover_path = os.path.join(app.config['
                   COVER_FOLDER'], cover_filename)
                cover.save(full_cover_path)

        # 创建图书对象并保存到数据库
        if title and author:
            book = Book(
                title=title,
```

```
                author=author,
                isbn=isbn,
                publish_date=datetime.strptime(publish_date_str
                   , '%Y-%m-%d').date() if publish_date_str
                   else None,
                category=category,
                description=description,
                file_path=file_path,
                cover_path=cover_path
            )
            db.session.add(book)
            db.session.commit()
            flash('图书添加成功！')
    except Exception as e:
        logger.error(f"添加图书时出错: {str(e)}")
        flash('添加图书时出错，请查看日志')

    return redirect(url_for('index'))
```

该功能的主要实现步骤包括：

1. 接收表单数据：

```
    title = request.form.get('title')
    author = request.form.get('author')
    isbn = request.form.get('isbn')
```

通过 Flask 的 request 对象获取表单提交的图书基本信息，包括标题、作者、ISBN 等字段。

2. 处理文件上传：

```
    if 'file' in request.files:
        file = request.files['file']
        if file.filename != '' and allowed_file(file.
           filename):
            filename = secure_filename(file.filename)
```

```
            file_path = os.path.join(app.config['UPLOAD_
                FOLDER'], filename)
            file.save(file_path)
```

使用 Flask 的文件上传功能处理电子书文件，通过 secure_filename 确保文件名安全，并将文件保

3. 处理封面图片：

```
    if 'cover' in request.files:
        cover = request.files['cover']
        if cover.filename != '' and allowed_image(cover.
            filename):
            cover_filename = secure_filename(cover.filename
                )
            cover_path = f"covers/{cover_filename}"
            full_cover_path = os.path.join(app.config['
                COVER_FOLDER'], cover_filename)
            cover.save(full_cover_path)
```

类似地处理封面图片上传，使用相对路径存储图片引用。

4. 数据验证和保存：

```
    if title and author:
        book = Book(
            title=title,
            author=author,
            isbn=isbn,
            publish_date=datetime.strptime(publish_date_str
                , '%Y-%m-%d').date() if publish_date_str
                else None,
            category=category,
            description=description,
            file_path=file_path,
            cover_path=cover_path
        )
```

```
        db.session.add(book)
        db.session.commit()
```

验证必填字段（标题和作者），创建 Book 对象并保存到数据库。

5. 错误处理：

```
    try:
        # ... 主要代码 ...
    except Exception as e:
        logger.error(f"添加图书时出错: {str(e)}")
        flash('添加图书时出错，请查看日志')
```

使用 try-except 捕获可能的异常，记录错误日志并显示用户友好的提示信息。

（二） 借阅图书功能

借阅图书功能通过以下代码实现：

```
@app.route('/borrow_book/<int:id>')
def borrow_book(id):
    try:
        book = Book.query.get_or_404(id)

        # 检查图书是否已被借出
        if book.is_borrowed:
            flash('该图书已被借出！')
            return redirect(url_for('index'))

        # 更新借阅信息
        book.is_borrowed = True
        book.borrower = request.args.get('borrower', '未知借阅
           者')
        book.borrow_date = datetime.utcnow()
        book.return_date = datetime.utcnow() + timedelta(days
           =30)  # 默认借阅30天
```

```
        db.session.commit()
        flash('借阅成功！')
    except Exception as e:
        logger.error(f"借阅图书时出错: {str(e)}")
        flash('借阅失败，请重试！')

    return redirect(url_for('index'))
```

借阅功能的主要实现步骤：

1. 获取图书信息：

```
    book = Book.query.get_or_404(id)
```

通过图书 ID 查询数据库，如果图书不存在则返回 404 错误。

2. 状态检查：

```
    if book.is_borrowed:
        flash('该图书已被借出！')
        return redirect(url_for('index'))
```

检查图书是否已被借出，如果已借出则显示提示信息并返回首页。

3. 更新借阅信息：

```
    book.is_borrowed = True
    book.borrower = request.args.get('borrower', '未知借阅
       者')
    book.borrow_date = datetime.utcnow()
    book.return_date = datetime.utcnow() + timedelta(days
       =30)
```

设置借阅状态、借阅者信息、借阅日期和预计归还日期（默认 30 天）。

4. 保存更改：

```
    db.session.commit()
```

将更改提交到数据库。

（三） 删除图书功能

删除图书功能通过以下代码实现：

```
@app.route('/delete_book/<int:id>')
def delete_book(id):
    try:
        book = Book.query.get_or_404(id)

        # 检查图书是否已借出
        if book.is_borrowed:
            flash('该图书已被借出，无法删除！请先归还图书。')
            return redirect(url_for('index'))

        # 删除相关文件
        if book.file_path and os.path.exists(book.file_path):
            os.remove(book.file_path)

        if book.cover_path:
            cover_path = os.path.join(app.config['COVER_FOLDER'
               ], os.path.basename(book.cover_path))
            if os.path.exists(cover_path):
                os.remove(cover_path)

        # 从数据库中删除记录
        db.session.delete(book)
        db.session.commit()
        flash('图书删除成功！')
    except Exception as e:
        logger.error(f"删除图书时出错: {str(e)}")
        flash('删除失败，请重试！')
```

```
    return redirect(url_for('index'))
```

删除功能的主要实现步骤：

1. 安全检查：

```
    if book.is_borrowed:
        flash('该图书已被借出，无法删除！请先归还图书。')
        return redirect(url_for('index'))
```

确保图书未被借出才能进行删除操作。

2. 文件清理：

```
    if book.file_path and os.path.exists(book.file_path):
        os.remove(book.file_path)

    if book.cover_path:
        cover_path = os.path.join(app.config['COVER_FOLDER'
           ], os.path.basename(book.cover_path))
        if os.path.exists(cover_path):
            os.remove(cover_path)
```

删除相关的电子书文件和封面图片。

3. 数据库操作：

```
    db.session.delete(book)
    db.session.commit()
```

从数据库中删除图书记录。

四　效果展示

使用编译形成的 app.exe 或者执行命令”python app.py” 运行软件，然后在浏览器中访问：http://localhost:5000进入软件的 webui 界面。

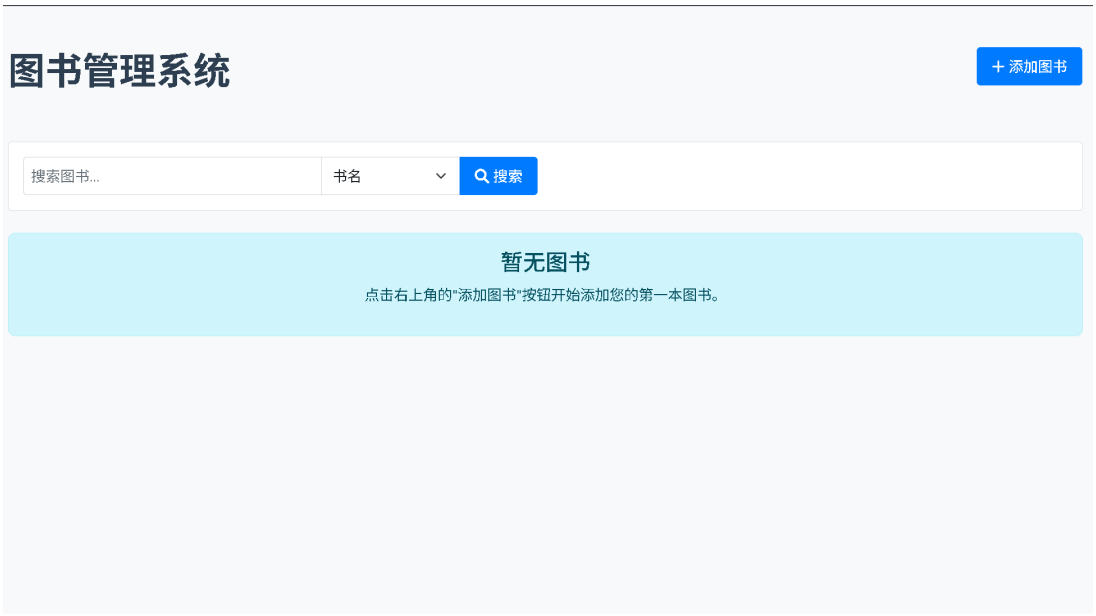

图 2: 起始页面

单击右上角的” 添加图书”，即可添加图书。如果不配置封面则使用存放在\cover 中的默认封面。

图 3: 添加图书

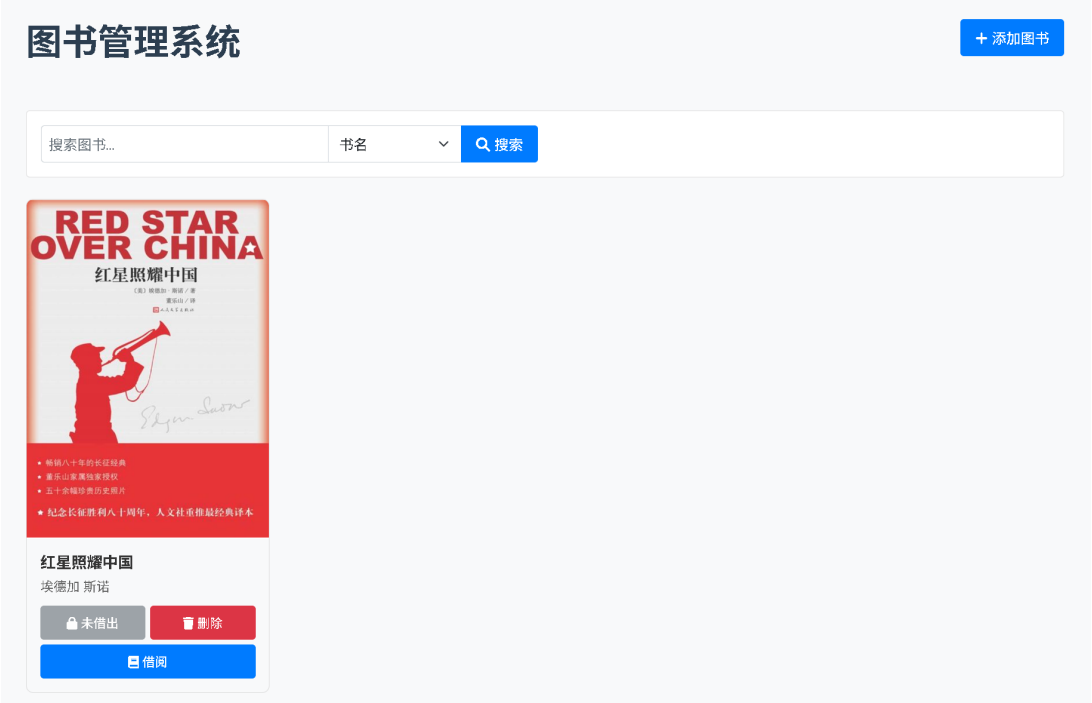

图 4: 添加图书后效果

添加图书后，按照属性搜索图书

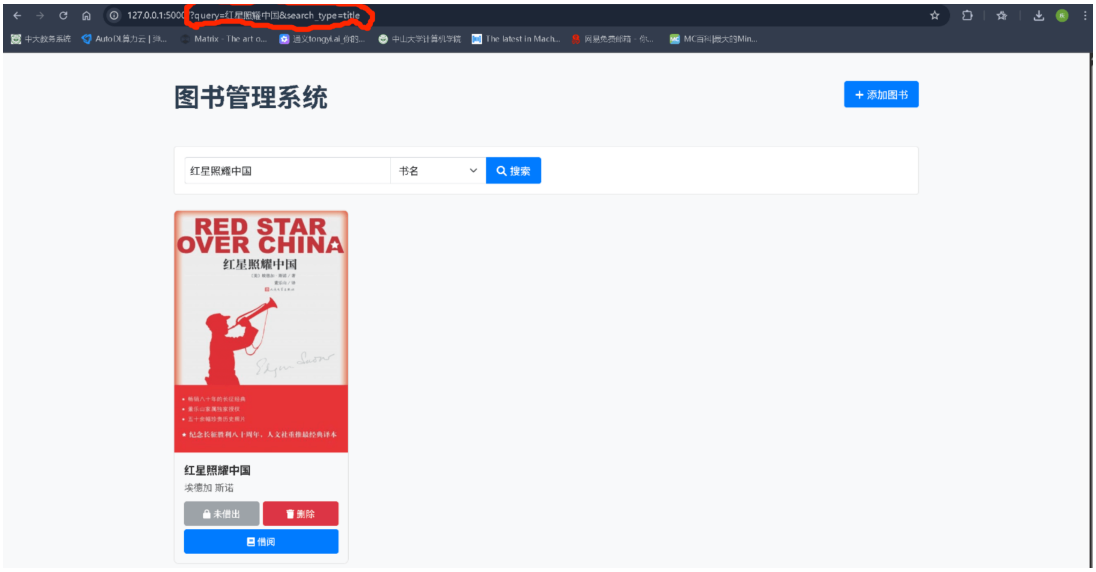

图 5: 按照书名查找图书

单击图书左下角的” 借阅” 或” 归还”，实现借阅或者归还图书。当借阅图书时，可以使用” 下载” 功能模拟借书。

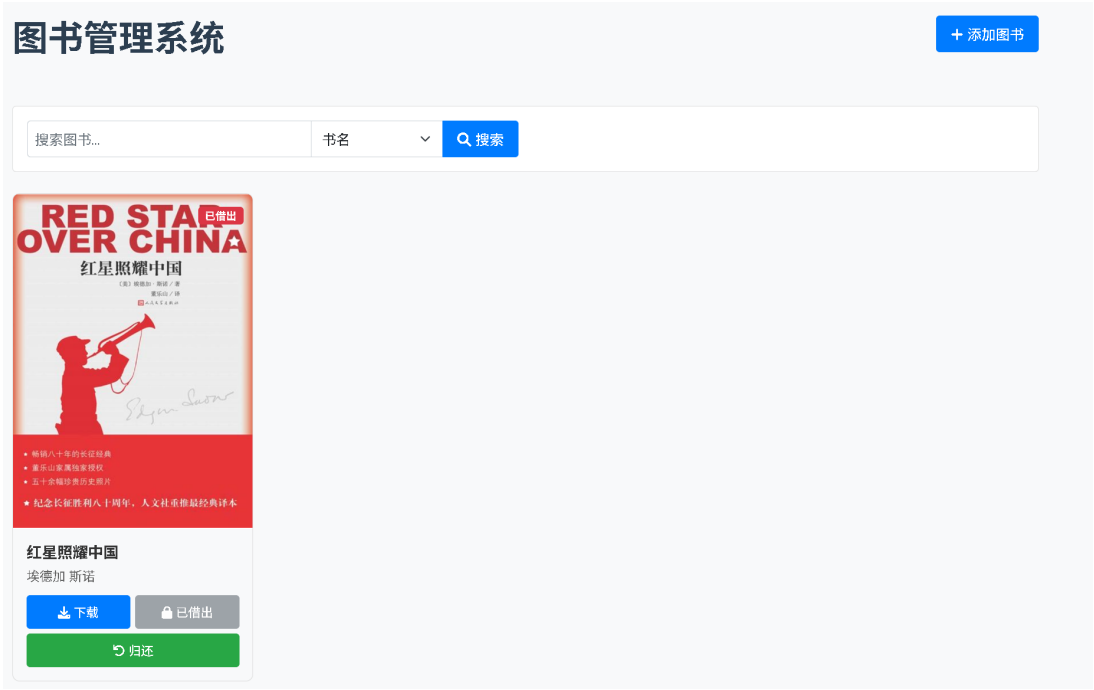

图 6: 借阅图书

在归还图书后，可以单击图书下方的” 删除” 按钮实现删除图书。

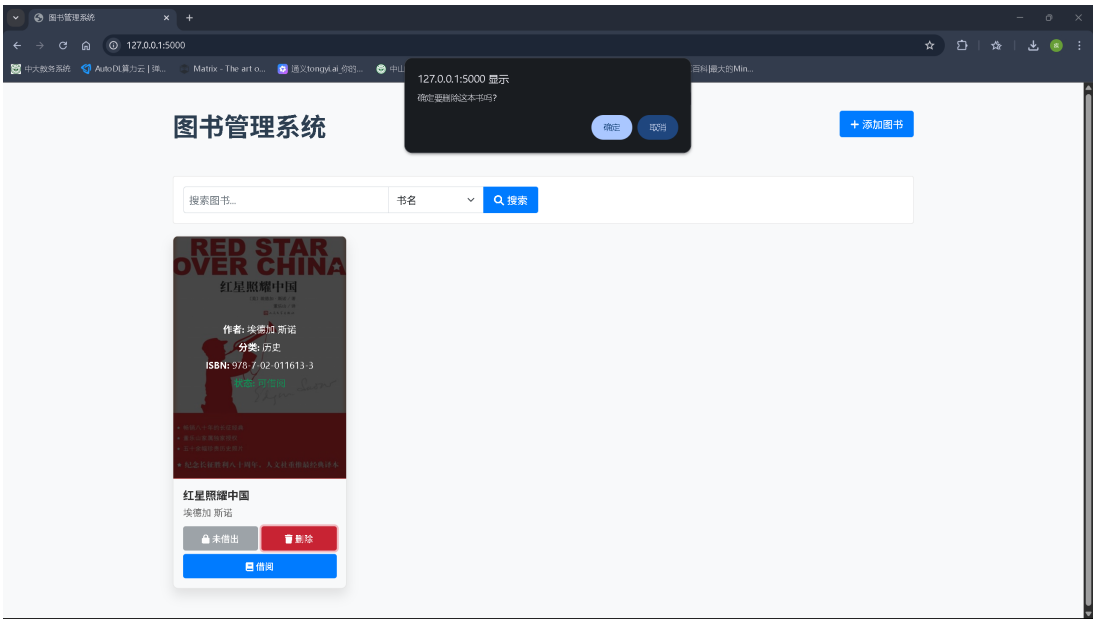

图 7: 删除图书

五　总结

本软件基于 ai 辅助开发，实现了一个简单的图书管理系统的增、删、查、改功能，体验了人工智能时代软件开发的便利性，为后续开发更加复杂的软件和学习打下了基础。