团队名称： 3+1>4

指导教师： 代祖华

完成时间： 2020年6月3日

团队成员： 王嫄 牛莉梅 祁甜 王爽

**[西师爱阅（微信小程序）]**

**《软件设计说明书》**

[一、引言 1](#_Toc24616)

[1.1编写目的 1](#_Toc4247)

[1.2背景 1](#_Toc24600)

[1.3参考资料 1](#_Toc8538)

[二、 总体设计 1](#_Toc6073)

[2.1需求规定 1](#_Toc27100)

[2.2运行环境 1](#_Toc26784)

[2.3基本设计概念和处理流程 1](#_Toc3576)

[2.5人工处理过程 3](#_Toc30067)

[三、 系统数据库逻辑结构设计 4](#_Toc17196)

[3.1 E-R图 4](#_Toc28027)

[3.2转换为关系模式 4](#_Toc5199)

[四、 软件重用方案 4](#_Toc19966)

[五、 关键类的重点服务 5](#_Toc30746)

[六、 接口设计 7](#_Toc3897)

[6.1用户接口 7](#_Toc30902)

[6.2外部接口 7](#_Toc8737)

[6.3内部接口 7](#_Toc23786)

[七、 运行设计 8](#_Toc3427)

[7.1运行模块组合 8](#_Toc1960)

[7.2运行控制 8](#_Toc11069)

[7.3运行时间 8](#_Toc10498)

[八、 系统出错处理设计 8](#_Toc22297)

[8.1出错信息 8](#_Toc29650)

[8.2补救措施 8](#_Toc4378)

[8.3系统维护设计 9](#_Toc26011)

**一、引言**

**1.1编写目的**

软件需求说明书是需求分析阶段的一个文档，是对软件目标及范围的求精和细化，深入描述软件功能和性能以及软件的约束范围，使用户和软件开发者对该软件的初始的规定有个大概的了解，有利于对项目的回溯和指导后续的开发和维护。

**1.2背景**

1、待开发的软件系统的名称：西师爱阅（微信小程序）；

2、本项目的任务提出者：代老师（代祖华）；

3、开发团队：3+1>4（王嫄，牛莉梅，祁甜，王爽）；

4、用户：西北师范大学所有在校教职工、学生；

**1.3参考资料**

《软件工程导论》第六版张海藩 牟允梅著

《构建之法 现代软件工程》第三版邹欣著

1. **总体设计**

**2.1需求规定**

本系统软件主要实现西师爱阅图书馆小程序，主要包括：

1、信息获取模块：获取用户的位置信息和个人信息

2、笔记秀模块：为用户提供一个笔记分享的功能

3、借书模块：用户登录小程序，实现借书、还书的功能

4、管理员模块：管理员模块又包括管理模块、收藏管理模块、借书管理模块、书架模块实现对整个系统的管理

**2.2运行环境**

微信开发者工具、Eclipse、Tomcat7.0

**2.3基本设计概念和处理流程**

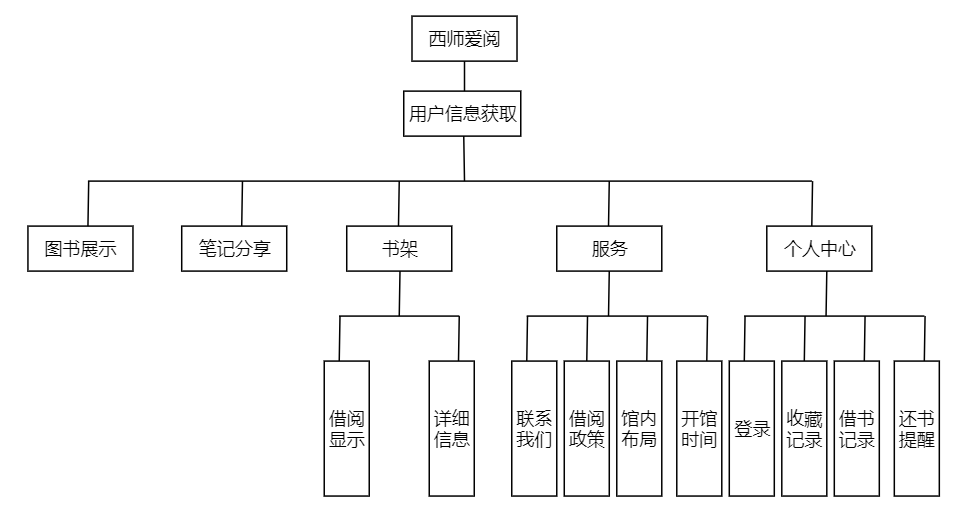


图2.1 基本设计概念

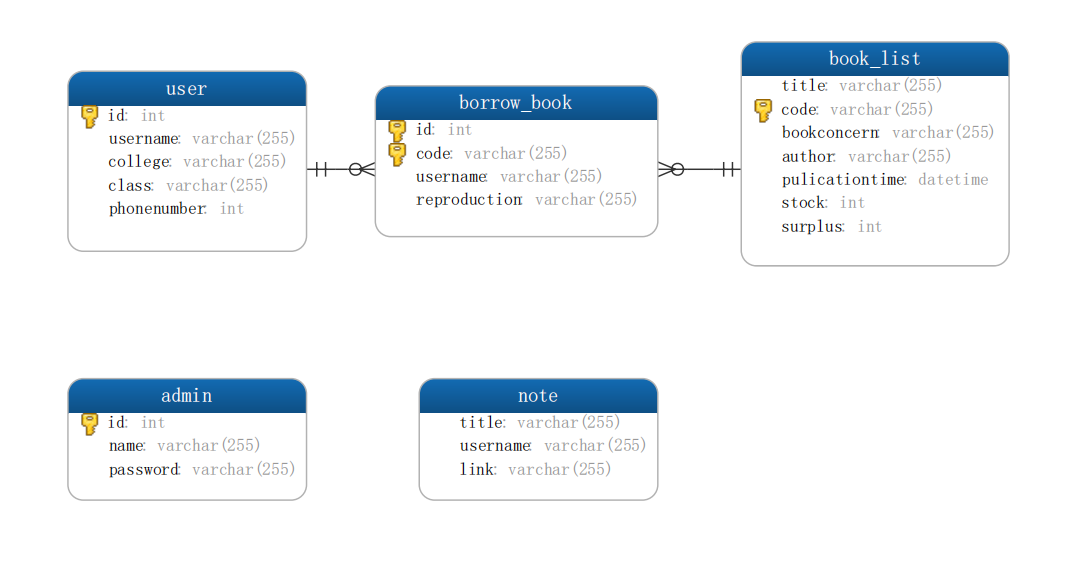


图2.2 系统数据处理流程

**2.4结构**

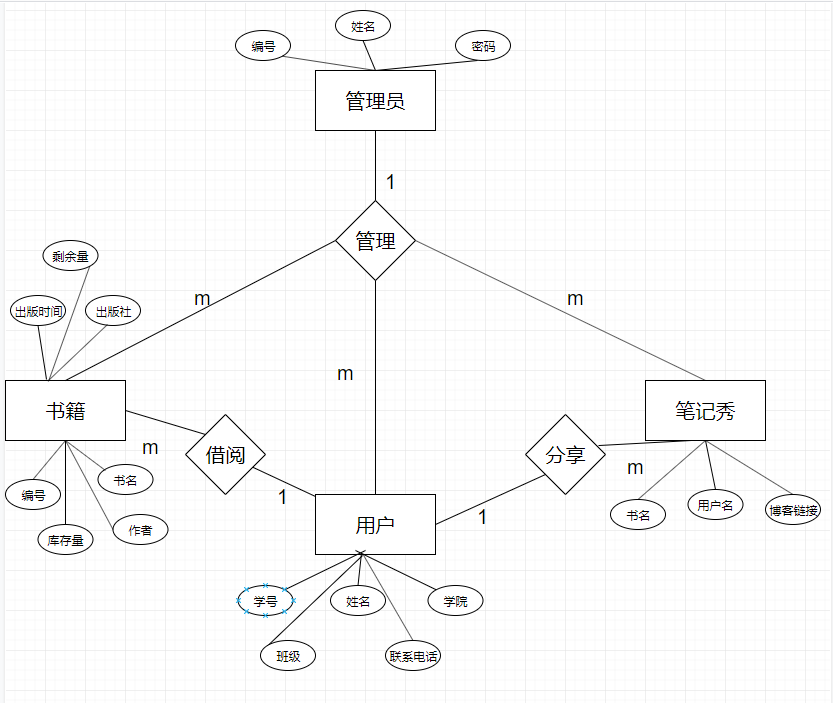
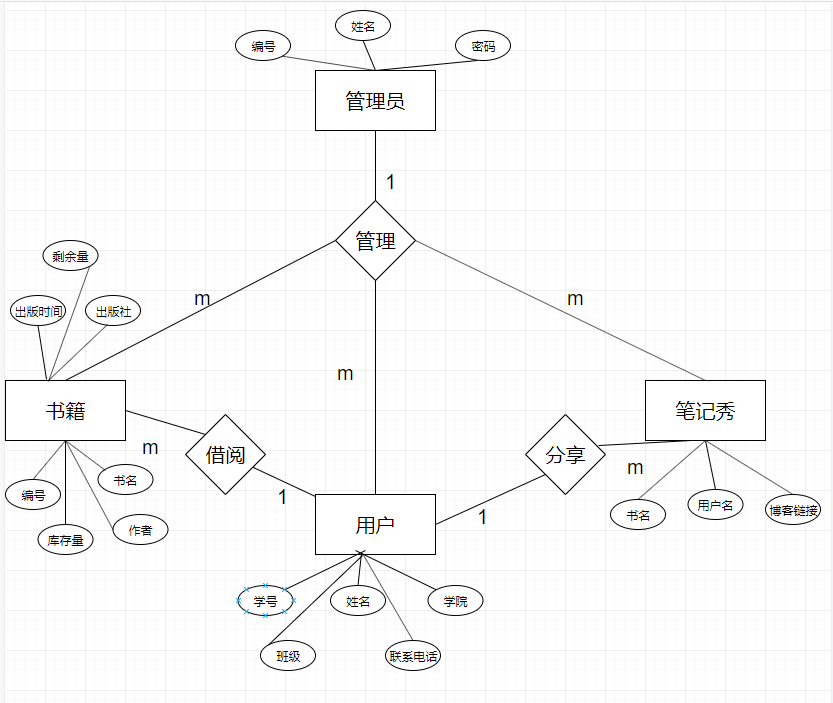


图2.3 西师爱阅E-R图

**2.5人工处理过程**

1. 用户详细信息和学校认证需要用户手动输入；
2. 书单的更新需要管理员手动修改；
3. 借书完成后需要用户自己去图书馆取书、还书。
4. **系统数据库逻辑结构设计**

**3.1 E-R图**



**3.2转换为关系模式**

管理员（编号，姓名，密码）；

用户（学号，姓名，学院，学号，班级，联系电话）；

书单信息（编号，书名，作者，出版社，出版时间，库存量，剩余量）；

笔记秀（书名，用户名，博客链接）；

借书（学号，书籍编号，姓名，还书）；

1. **软件重用方案**

1、知识重用方案

我们之前所学到的很多知识，包括数据库的知识、编程知识、软件工程导论知识等，都可以运用在我们所做的项目中。比如前期调研的需求分析，项目背景等，就可以应用在后期的详细设计中，在设计数据库逻辑结构的时候，就要用到之前在数据库中学到的E-R图知识，在建立问题域对象模型，要用到 UML模型知识。

2、方法和标准的重用方案

对于方法的重用，我们这里采用面向对象的方法，对于标准的重用，我们采用国家规定的软件开发规范。

3、软件成分的重用方案

在编码的过程中，我们考虑调用库函数，实现代码重用，这样可以有效提高编程效率。之前编写的项目计划、成本估计等我们都没有重用，都是结合此次项目重新设计的，但是之后要编写的源代码、用户界面的设计、数据等都可以重用，比如源代码的编写中，所要用到的框架，我们就可以结合现有的框架进行编码，这样就减少了我们的编码工作量，有助于加快项目进度。在用户界面设计中，我们可以重用之前的原型设计中设计好的界面，对应进行设计，因为之前的原型已经经过用户体验了，所以重用原型设计来设计界面，可以保证我们最后研发出来的产品是满足用户审美的，是符合用户需求的。对于数据重用，比如数据库中表的设计，就可以参考现有的小程序进行设计。

4、类构件实现软件重用方案设计

利用面向对象技术,可以更方便、更有效地实现软件重用。面向对象技术中的“类”,是比较理想的可重用软构件,称之为类构件。类构件有3种重用方式,分别是实例重用、继承重用和多态重用，这里我们使用继承重用方案。

对于面向对象语言来说，必须要保证它的可封装性，就要用到继承机制来达到代码复用的目的。继承是所有OOP语言和Java语言不可缺少的组成部分，当创建一个类时，总是在继承，这样就不需要重新粘贴复制这个方法了，减少了代码的出错率。对于接口重用，我们考虑用类指针调用派生类的方法来实现接口重用。为了提高重用的效果,关键是设计一个合理的、具有一定深度的类构件继承层次结构，这样每个子类在继承父类的属性和服务的基础上,只需要加入少量新属性和新服务,这不仅降低了每个类构件的接口复杂度,表现出一个清晰的进化过程,提高了每个子类的可理解性,面且为软件开发人员提供了更多可重用的类构件，比如在后期的编码中，借阅类可以继承书籍类。

1. **关键类的重点服务**

确定类中应有的服务就必须要综合考虑对象模型、动态模型和功能模型，才能正确确定类中应有的服务。对象模型是进行对象设计的基本框架。但是，面向对象分析得出的对象模型，通常只在每个类中列出很少几个最核心的服务。设计者必须把动态模型中动态模型中对象的行为以及功能模型中的数据处理，转换成由适当类所提供的服务。

一张状态图描绘了一类对象的生命周期，图中的状态转换是执行对象服务的结果。本项目的状态转换图见图5-1所示。功能模型指明了系统必须提供的服务。状态图中状态转换所触发的动作，在功能模型中有时可能扩展成一张数据流图。数据流图中的某些处理可能与对象提供的服务相对应。该项目的数据流图见图5-2所示。

根据状态图和数据流图可筛选出本项目的关键类及相应重点服务如下：

书籍：可被借阅、可以电子形式读书等；

图书管理员：根据用户提交相关信息更新库存数据、管理书籍相关信息等；

用户：可以扫码形式借阅书籍、可在该平台上分享自己的读书笔记、身份必须为本校师生及工作人员。

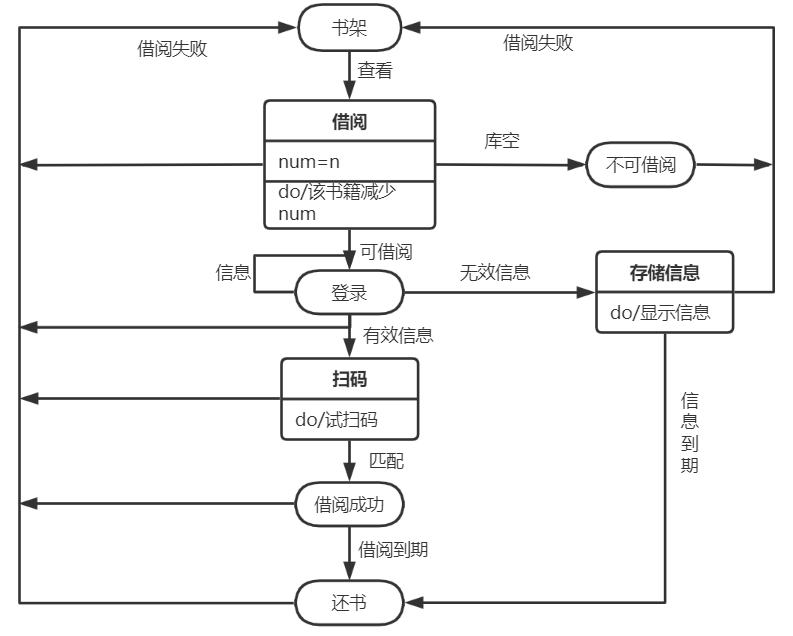


图5.1 西师爱阅状态转换图

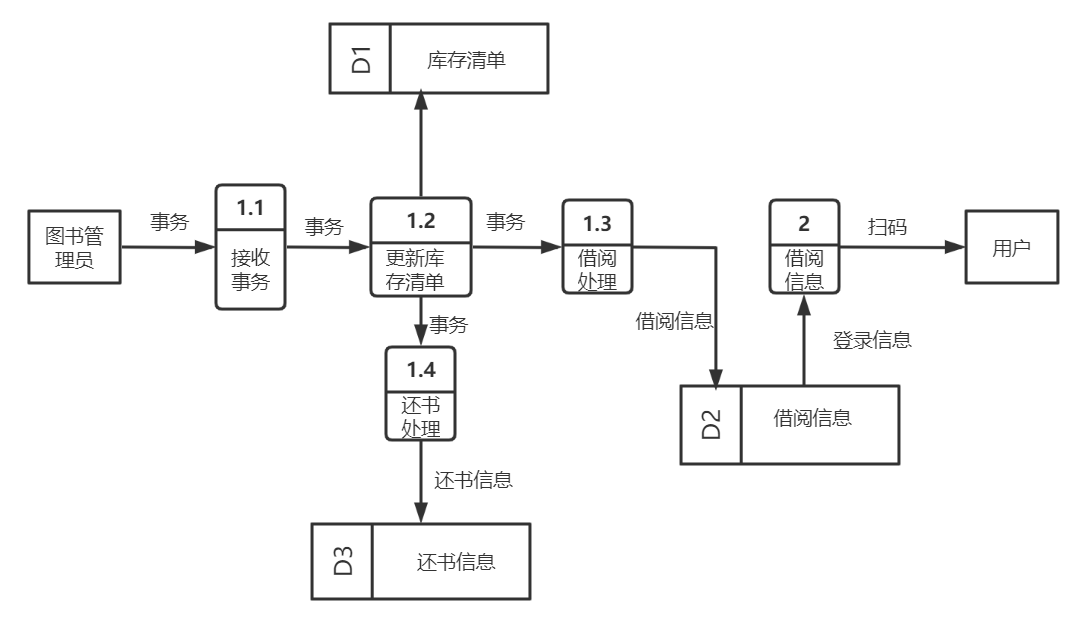


图5.2 西师爱阅数据流图

1. **接口设计**

**6.1用户接口**

管理员运行西师爱阅管理系统的主应用程序，管理人员登录用户名为工号，密码自设，输入用户名和密码，只有输入正确才能进入系统。

用户通过搜索微信小程序西师爱阅进入小程序，进入个人中心进行学校认证，输入学号、学院、班级等信息，即可进行书籍借阅。

**6.2外部接口**

通过JDBC-ODBC桥接口与My SQL数据库连接

**6.3内部接口**

1.系统内部与数据库接口为SQL链接

2.客户端通过配置数据源与服务器建立连接

3.程序内的各个模块之间采用函数调用、参数传递、返回值得方式进行信息传递。

1. **运行设计**

**7.1运行模块组合**

1、信息获取模块：获取用户的位置信息和个人信息

2、笔记秀模块：为用户提供一个笔记分享的功能，用户提供想要分享的读书笔记链接进行分享

3、借书模块：实现借书、还书的功能

4、管理员模块包括：管理模块、收藏管理模块、借书管理模块、书架模块实现对整个系统的管理

**7.2运行控制**

1. 打开微信后向下拉进入搜索小程序，搜索“西师爱阅”进入小程序；
2. 借书、还书必须保证准确，不出现错借等情况。

**7.3运行时间**

1、响应时间小于2s

2、数据的转换和传送时间小于2秒钟

1. **系统出错处理设计**

**8.1出错信息**

1、用户的登录账号或密码输入错误。

2、输入数据精度不符合要求

3、用户的地理位置获取失败。

4、硬件故障对系统的影响。当断开网络时，该系统不能正常打开。

解决办法：检查网络配置，系统是否联网。

**8.2补救措施**

1、 用户的登录账号或密码输入错误这一问题，可以点击忘记密码，用手机号验证登陆。

2、输入数据精度不符合要求这一问题，可以提示系统输出“输入格式不正确，请检查后重新输入！”。

3、对于用户无法获取地理位置这一问题，可以提示用户打开GPS定位，然后重新进入小程序即可；

4、对于硬件故障对系统影响这一问题，可以检查网络配置查看系统是否联网。

**8.3系统维护设计**

1、数据库的备份和恢复功能，

2、采用模块化的设计方法，使模块间具有较高的相互独立性，便于检查、维护和修改。