**图书馆管理系统需求分析说明书**

[一． 引言 1](#_Toc316041091)

[1. 编写目的 1](#_Toc316041092)

[2. 背景 1](#_Toc316041093)

[3. 定义 2](#_Toc316041094)

[二． 任务概述 2](#_Toc316041095)

[1．目标 2](#_Toc316041096)

[2. 用户的特点 2](#_Toc316041097)

[3. 系统功能示例 2](#_Toc316041098)

[三． 需求细则 2](#_Toc316041099)

[1. 对功能的规定 2](#_Toc316041100)

[2. 对性能的规定 5](#_Toc316041101)

[3. 对排版的规定 5](#_Toc316041102)

[4. 对可维护性的规定 5](#_Toc316041103)

[5. 对个性的规定 6](#_Toc316041104)

[6. 对项目过程的规定 6](#_Toc316041105)

# 引言

## 1.编写目的

通过与多位软件使用者进行全面深入地探讨和分析，并完成《图书馆管理系统》市场的前期调查后，提出了这份软件需求分析说明书。

此需求分析说明书对《图书馆管理系统》软件做了全面细致的用户需求分析，明确所要开发的系统应具有的功能、性能与界面，使系统分析人员及软件开发人员能清楚地了解用户的需求，并在此基础上进一步提出概要设计说明书和完成后续设计与开发工作。

本说明书的预期读者为客户、业务或需求分析人员、测试人员、用户文档编写者、项目管理人员。

## 2. 背景

## 3. 定义

# 任务概述

## 1．目标

## 2. 用户的特点

## 3. 系统功能示例

**需求：**

1. 图书借还管理

对于图书进行增、删、查、改

生成相应的借书，还书记录

1. 用户的管理

用户积分、信息、借书次数等的增、改。

1. 使用IO保存用户信息、书本信息、预约信息

借书还书记录信息。

**技术要求：**

1. 用户类
2. 书本类
3. 预约类
4. 借还记录类

借书还书流程：

1. 用户注册登录，浏览图书列表，进入借书界面，开始借书，之后进行还书操作。
2. 浏览图书的界面应当包括图书的列表，图书相关的文字介绍，所有图书的在库状态。
3. 借书界面应当提供对图书的搜索工具，搜索完成后显示相关的图书，通过选择相关的图书来完成此次借书操作。
4. 还书界面应当显示当前用户的借书记录，如果所借书籍已经归还，则不能重复归还，如果没有则显示文本框让用户归还。
5. 若用户所要借的书在库没有库存，则可以预约，预约后，当有人归还此书时会通知相应的预约用户来取书。
6. 用户还可以对非预约过的书籍进行续借操作。借书、预约、续借的次数都应作相应增长

# 需求细则

## 对功能的规定

需求是整个工程的开始，对需求的深刻了解可以提高开发过程中的设计与理解，此需求要求侧重点为：功能模块、设计及代码规范、产品优化及健壮性。分基本功能、主要功能和提高功能，其中，基本功能为必须完成功能，如缺失则不能参加答辩；主要功能指难度适中，适合所有学员的功能，要求全部实现；提高功能适于能力较强的学员完成，通过这类功能，可以对自己的技术能力和综合水平有显著提高。

1. 实体类基本功能

属性：基本

* 1. 用户实体类

描述：包含用户相关内容的相应多种类，应均有各种属性及基本方法

* 1. 管理员实体类

描述：包含管理员相关内容的相应多种类，应均有各种属性及基本方法

* 1. 书本信息实体类

描述：包含书本信息相关内容的相应多种类，应均有各种属性及基本方法

1.4 借还记录实体类

描述：包含借还信息相关内容的相应多种类，应均有各种属性及基本

方法

1.5 预约记录实体类

描述：包含预约信息的相应多种类，应均有各种属性及基本方法

1. IO

属性：基本

描述：使用IO来保存数据，将对应的实体类的信息保存到对应的文件中，比如User实体类就保存到名为User.txt的文件内。

1. MVC模式 或 三层架构

属性：主要

描述：使用MVC模式对C控制、V视图、M模型层进行封装和处理

或使用三层架构，对数据访问层，业务逻辑层，表现层进行封装和处理

1. 管理员主体功能

属性：主要

描述：管理员可以在登录后对书本进行增删改查等操作，可以对借阅次数做出相应的统计。

1. 用户主体功能

属性：主要

描述：用户可以在注册登录后，可以查看个人信息、浏览书本信息，并查找书本进行借书、预约、续借等操作，借书之后可以进行还书。

当书本没有库存的时候，用户可以对书籍进行相应的预约操作，预约后当书本有存库时应当给出相应的提示信息、当书本被预约后，对此书无法进行续借。

1. Swing

属性：提高

描述：使用Swing制作用户交互界面，实现购票系统的窗口化管理

1. 推荐及自选个性功能

属性：提升

描述：完成推荐功能或自定义个性功能，为应用增添个性功能以提高应用的趣味性、安全性等。

推荐：评论、VIP用户等

1. **有服务端和客户端，通过Socket通讯**
2. **支持多客户端并发登录（多线程）**

## 对性能的规定

本软件在设计方面本着方便、实用及娱乐性高的宗旨，性能方面应遵循如下原则：

* 流畅性： 软件运行应该流畅，不会出现卡机；
* 美观性： 软件界面应力求美观，赏心悦目；
* 稳定性： 软件运行过程中不会出现BUG；
* 边界条件：主界面应该全屏；
* 资源损耗：设计尽量节约资源；不用的资源应及时回收（如服务结束，线程应退出）；

## 3. 对可维护性的规定

对可维护性的最终要求：别人能够轻松上手你的代码。

* 结构清晰：封装良好；继承关系清晰；代码块清晰；
* 变量命名规范，变量名应该具有自明性；
* 注释充分：变量、方法、代码功能块、一些复杂算法……等都需要清晰明了地说明；

## 4. 对个性的规定

把项目做出个性出来。下列各项中有和比人不同之处、或很有创意，即可认为有个性。独立设计的软件，一般都会出现一些个性。参考、抄袭不会出现个性。

* 设计思路：包括软件的整体架构、功能块的设计思路等等；
* 功能实现：从用户的角度，使用上发现与众不同的地方；
* 其它方面；

## 5. 对项目过程的规定

本着紧张但不急躁、不参考、不拷贝的原则进行。

* 紧张但不慌张

项目周期只有两周，这还包括项目答辩时间。所以项目时间比较紧张，但不能慌张。要有自己明确的设计思路，一步步沿着思路走下去，以此来巩固自己所学，锻炼自己的独立工作能力。

* 能自己做，绝不参考别人

自己还没有做，还没有想，就去看比人的，这样尽管功能做出来了，但却没有什么意义，真正面试时还是不会。作者和读者，天壤之别。

如果自己实在无法搞定，一个问题卡了快一天了，则可以咨询别人一下想法，再行编码；尽量不直接看别人代码。

* 不拷贝

**一旦发现拷贝，取消答辩资格。答辩时发现，答辩成绩减半。**

copy别人的代码，甚至直接运行别人的代码，以此作为自己的项目进展，这是严禁的。**严禁运行效果出来了，却不知道是哪些代码造成的，严禁明明是自己写的代码，但却不知道为什么这么写。**

# 四．运行环境

## 1. 设备

PC

## 2.支持软件

Eclipse Intellji

用户类字段参考如下：

用户ID、用户的姓名、用户的年龄、用户的性别、用户的等级（0普通用户，1管理员）

书本类字段参考如下：

书本ID、书本的名字、书本的类别、书本的作者、书本的库存量、。。。。。

借书记录类：

借书记录ID、书本ID、借书数量、用户ID、。。。。

预约类：

预约记录ID、书本ID、预约数量、用户ID、。。。。