**项目开发总结报告**

# 1引言

## 1.1编写目的

此说明书对《图书馆管理系统》软件开发总结做了全面细致的说明和介绍，明确该系统整体结构以及相应的功能及使用接口，帮助维护人员和使用方能清楚地了解项目的功能和运行机制，共同维护此系统的良好运行。

本说明书的预期读者为文档编写者、项目管理人员、维护人员、使用方。

本说明书仅适用于图书馆管理系统，任何使用此文档的人员，必须严格按照文档规定及说明及相关法律条款规定，不能擅自修改及非法使用。本开发团队拥有对此文档的最终解释权。

## 1.2背景

说明：

项目开发完毕之后，团队对整个项目进行一个总结分析，旨在为后续维护和管理提供参考。

1. 待开发的数据库名称：LibraryManagementDatabase

使用此数据库的软件系统名称：图书馆管理系统

1. 软件系统开发项目的任务提出者：西南大学商贸学院

用户：西南大学全体学校学生、管理人员

安装该软件和这个数据库的计算站（中心）：西南大学商贸学院机房

## 1.3定义

登录/注册：用户端进行的身份准入操作，通过即进入系统，失败则重试或退出

借阅/归还：书籍的操作，线上完成数据的变更后，管理员需及时线下一致化。

预约/出让：图书馆资源的使用，具体同上一条。

智能推送：向用户推荐喜爱度高的书籍。

公告管理：管理员发布图书馆公告通知。

奖惩计分：将图书馆规则条例纳入系统。

UserTable：管理用户表

BookGRTable：管理书籍表

ResourceTable：资源管理表

BookTable：书籍信息表

ResourceGRTable：资源信息表

InformTable：公告管理表

## 1.4参考资料

有关的参考文件，如下：

1. 软件开发计划书V1.0；
2. 《图书馆管理系统-需求规格说明书》；《图书馆管理系统-概要设计说明书》等
3. 【1】 GB-T 8567-2006, 《计算机软件文档编制规范》[S].

【2】张海藩，牟永敏《软件工程导论》[M]，北京：清华大学出版社，2013.8

【3】王少锋，《UML面向物件教程》[M]，北京：清华大学出版社，2004.2

【4】王珊，萨师煊，《数据库系统概论》[M]，北京：高等教育出版社，2006.5

# 2实际开发结果

## 2.1产品

本系统最终成型情况如下：

1. 程序系统中各个程序的名字，它们之间的层次关系，以千字节为单位的各个程序的程序量、存储媒体的形式和数量：

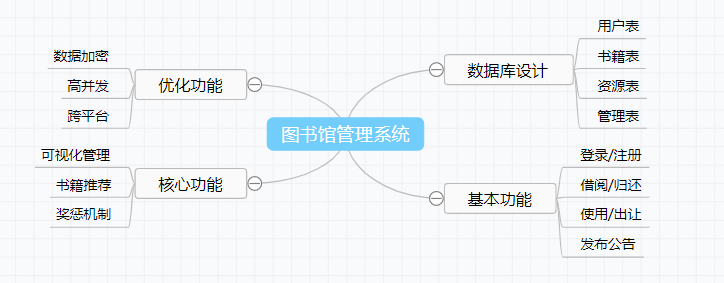


图2-1 程序系统抽象

表2-1 各程序基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 基本信息 | | 存储媒体 | |
| 序号 | 程序名称 | 程序量 | 形式 | 数量 |
| 1 | 登录注册 | 0.6k | 服务器 | 1~5 |
| 2 | 管理 | 3k | 服务器 | 1~5 |
| 3 | 发布 | 0.4k | 服务器 | 1~5 |
| 4 | 可视化 | 1k | 服务器 | 1~5 |
| 5 | 推荐 | 3k | 端系统 | n |
| 6 | 优化 | 2k | 端系统 | n |

1. 程序系统版本情况，各自的版本号及它们之间的区别；

暂行系统程序共有两个版本，一个是测试版本，一个是发行版本。

表2-2 版本迭代表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 版本名 | 版本号 | 介绍 |
| 1 | 测试 | 1.0 | 进行调试 |
| 2 | 发行 | 1.0 | 发布使用 |

1. 每个文件的名称；

本程序的主体开发代码为Java，应用面向对象的方式进行编程，包括有不同的模块和文件，列举主要部分如下：

表2-3 程序文件表

|  |  |  |
| --- | --- | --- |
| 序号 | 文件名 | 作用 |
| 1 | UserController | 用户相关操作逻辑 |
| 2 | RecommendController | 推荐算法相关逻辑 |
| 3 | BookController | 书籍管理相关逻辑 |
| 4 | xxxService | 提供服务类 |
| 5 | xxxImpl | 服务接口类 |

1. 所建立的每个数据库：

数据库的建立在《图书管理系统——数据库设计说明书》中已经详细阐述，这里作简要罗列：

表2-4 数据库信息

|  |  |  |
| --- | --- | --- |
| 数  据  库 | 用户表 | data1 |
| 书籍表 | data2 |
| 资源表 | data3 |
| …… | data4 |

## 2.2主要功能和性能

逐项列出本软件系统产品所实际具有的主要功能和性能，对照可行性研究报告、项目开发计划、功能需求说明书的有关内容后，说明原定的开发目标是达到了、未完全达到、或超过了的情况如下：

表2-5 功能与性能实现分析表

|  |  |  |
| --- | --- | --- |
| 序号 | 功能 | 性能 |
| 1 | 登录注册（达到了） | 高并发（达到了） |
| 2 | 管理书籍（达到了） | 数据保密（超过了） |
| 3 | 管理资源（达到了） | 跨平台（达到了） |
| 4 | 奖惩机制（达到了） | 低延迟（达到了） |
| 5 | 推荐算法（达到了） | 健壮性（达到了） |
| 6 | 发布公告（达到了） | 稳定性（达到了） |

## 2.3基本流程

本程序系统的实际的基本的处理流程：

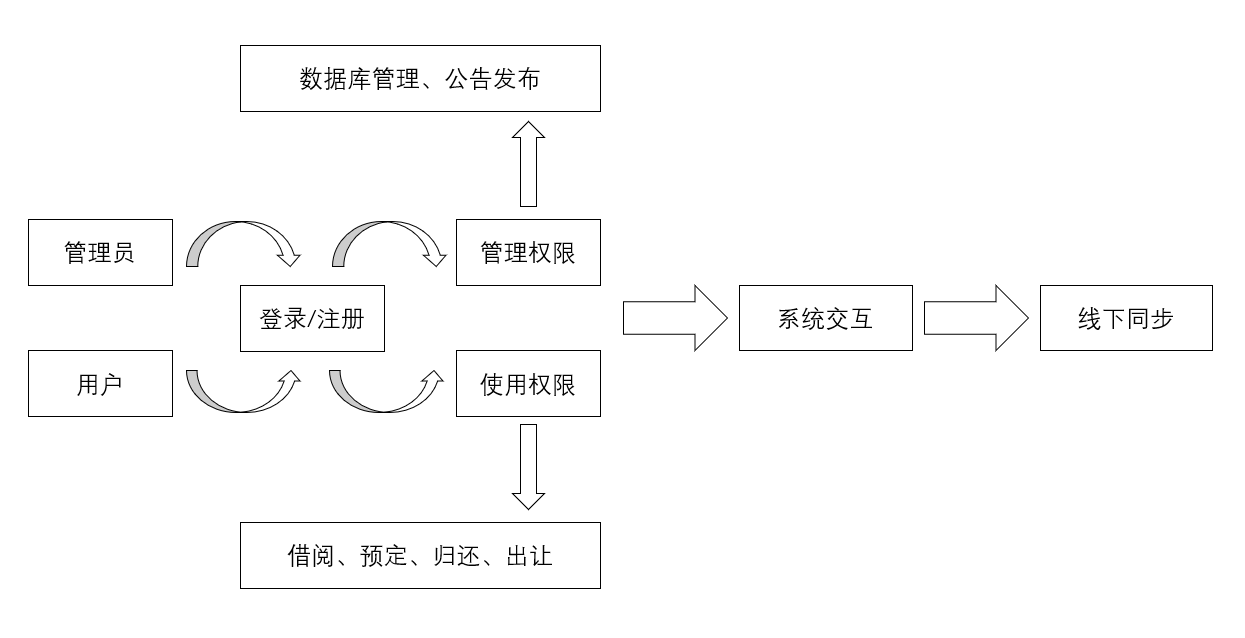


图2-2 程序基本处理流程

## 2.4进度

列出如下进度管理甘特图，可以很清晰的看出，部分原定计划进度延迟了，实际进度整体上是偏后。分析之后，发现原因在于在完成某阶段功能时估计不足，导致进度没有跟上。然后出现了赶工的情况，故影响的整体的进度。

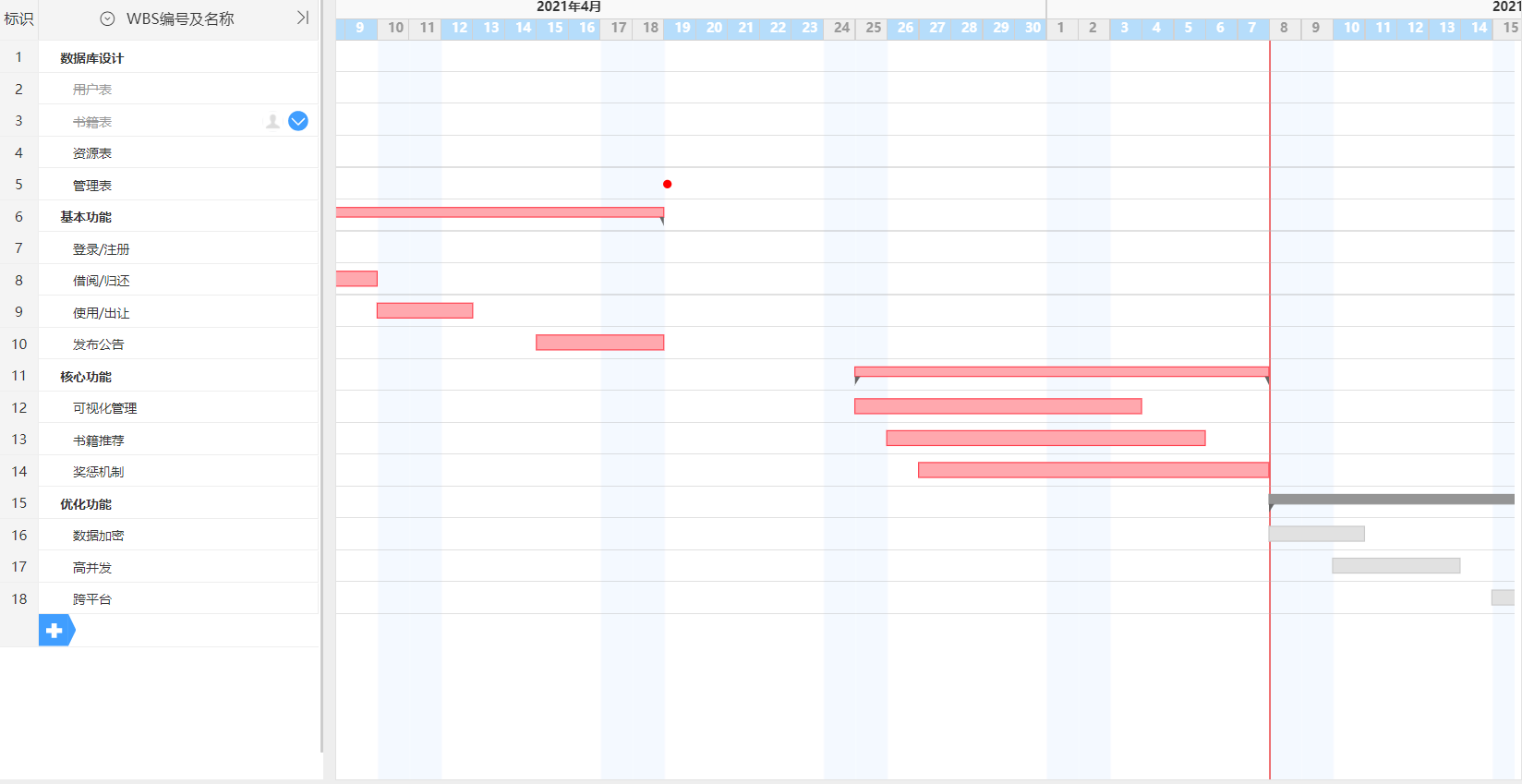


图2-3 项目甘特图

## 2.5费用

原定计划费用与实际支出费用的对比：

表2-6 工时对比表

|  |  |  |  |
| --- | --- | --- | --- |
| 级别 | 职务 | 计划工时 | 实际工时 |
| 1 | 管理人员 | 1/人月 | 1.5/人月 |
| 2 | 开发人员 | 3/人月 | 5/人月 |
| 3 | 测试人员 | 1/人月 | 2/人月 |

表2-7 资源使用表

|  |  |  |  |
| --- | --- | --- | --- |
| 资源名 | CUP | 存储器 | 外设 |
| 时间使用 | 30% | 50% | 20% |

表2-8 项目支出表

|  |  |  |  |
| --- | --- | --- | --- |
| 项目名 | 消耗用度 | 说明 | 原因 |
| 差旅费 | 10% | 结余了 | 线上平台的充分使用 |
| 食宿费 | 40% | 可控 | 合理膳食 |
| 设备费 | 35% | 超出了 | 设备用度过大 |
| 场地费 | 15% | 可控 | 利用了免费场地 |

综上，分析原因

1. 工时方面，实际工时均超出计划，主要是因为前期估计不足，而且流程安排系统性不强。
2. 计算机资源使用上，安排比较合理，充分利用了资源达到了预期效果。
3. 物料消耗上，设备费的超出是由于后期添加了相关设备用度大的功能。

# 3开发工作评价

## 3.1对生产效率的评价

实际生产效率，包括：

1. 程序的平均生产效率，即每人月生产的行数：1000+
2. 文件的平均生产效率，即每人月生产的千字数：10000+

并列出原订计划数作为对比。

表2-9 生产效率评价表

|  |  |  |
| --- | --- | --- |
| 平均生产效率 | 计划 | 实际 |
| 程序 | 500（适中） | 1200（较强度） |
| 文件 | 3500（适中） | 10000（高负荷） |

## 3.2对产品质量的评价

说明在测试中检查出来的程序编制中的错误发生率，即每干条指令（或语句）中的错误指令数（或语句数）。如果开发中制订过质量保证计划或配置管理计划，要同这些计划相比较。

系统开发出来之后，对其进行测试是十分必要的。在测试过程中，难免会发现一些错误，这时候只要错误率在一个可控的范围内，就可以接受，认为此系统是较为可靠的。

系统千行代码错误率公式：

是错误率， 为bug数，为代码行数。

查阅资料发现，系统开发的错误率范围CMMI级别中做出了相关的指标规定，千行代码错误率的标准如下：

CMM1级 11.95‰

CMM2级 5.52‰

CMM3级 2.39‰

CMM4级 0.92‰

CMM5级 0.32‰

很显然，出错率越低，在一定程度上反映代码的质量越优，产品质量相应越好。

## 3.3对技术方法的评价

本系统开发主要采用的是Java框架以及数据库技术，另外辅以相关AI算法。团队成员为了提高效率，采用前后端分离的开发方式，用Github管理项目版本。工具的使用主要是IDEA-java平台以及Navicat数据库管理程序。

逐一分析来看，Java是主干操作，因此对其框架的运用是关键，团队在这方面作的较为不错。此外，一些辅助技术由于经验不足，仅仅是完成了效果，而未能进行优化处理。

## 3.4出错原因的分析

开发中出现的错误主要分为以下三个部分，一是数据库操作部分，一些语句使用有误；二是Java逻辑部分，消息的转递出现了某些问题；三是AI推荐算法的公式推导错误。

针对上述三类错误，团队分析原因如下：其一，数据库操作比较庞杂，出现的均是一些失误，对整体影响不大，易于修改；其二，Java逻辑出错主要是设计之初没有完善好系统，导致后期开发是流程紊乱，出现错误，这导致调试过程变得复杂，难度加大；其三，算法数学公式的推导对程序没有造成很大影响，查阅相关资料完成正确推导后，便解决了问题。

# 4经验与教训

项目整体完成之后，团队深入分析了在这个过程中出现了一些现象和问题，并针对性的进行了讨论，得出了相应的应对措施，以期在将来的开发中从容处理。

首先，团队在制定完成总体规划设计后，多次在开发过程中对其进行修改，导致后期主干思维混乱。虽然未造成严重影响，但是降低了开发效率，浪费了资源。

其次，对时间的把握稍有失当。在某阶段工期交付时，团队发现无法完工，便动员队员开始赶工，尽管在两天熬夜的情况下完成了既定目标，但是这种做法十分不提倡。

最后，团队成员毕竟是学生，缺少实际社会经验，对外来访客侵害性估计不足。团队在开发过程中曾两度数据库被黑客攻破，项目进度陷入瘫痪。值得一提的是，团队保持了一定的风险意识，对数据库文件进行了定时转储，故恢复了较近时间的快照，没有酿成更大的后果。

总的来说，今后在开发项目时，必须思路清晰，计划明确，严格按照既定目标进行，即使存在修改也应详细记录，备查。此外，要格外注意数据库的数据安全。