



图书管理系统 开发计划

项目组成员信息		
小组名称	软件工程 Team20 小组	
学号	姓名	本文档中主要承担的工作内容
14061046	曾宇涛	负责文档中关于系统介绍、项目计划、风险管理、工作要求等部分的撰写和计划书前期排版、审阅
14061054	李子硕	负责文档中关于项目组织和资源、成本估算等部分的撰写
14061033	李梓萌	负责对文档进行大致的审阅
14061045	王晓松	负责对文档进行大致的审阅

2016-10

版本变更历史

表格 1 版本变更历史

版本	提交日期	主要编制人	审核人	版本说明
0.4	2016.10.08	曾宇涛	曾宇涛	初步版本
0.6	2016.10.08	曾宇涛 李子硕	曾宇涛	加上了一些说明
0.8	2016.10.09	曾宇涛 李子硕	曾宇涛	增加了一些章节
1.0	2016.10.09	曾宇涛 李子硕	李梓萌	最终版本

目录

1 范围.....	1
1.1 标识.....	1
1.2 项目概述.....	1
1.3 文档概述.....	1
2 引用文档.....	2
3 交付产品.....	2
3.1 程序.....	2
3.2 文档.....	2
3.3 服务.....	3
3.4 非移交产品.....	3
3.5 验收标准.....	3
3.6 最迟交付期限.....	3
4 所需工作概述.....	4
4.1 所开发系统的需求和约束.....	4
4.2 对文档编制的需求和约束.....	4
4.3 项目进度和资源需求.....	4
4.4 其他需求和约束.....	5
4.4.1 对系统设计原则的约束.....	5
4.4.2 对系统设计/功能的其他约束.....	5
5 实施整个软件开发活动的计划.....	6
5.1 软件开发过程.....	6
5.2 软件开发总体计划.....	9
5.2.1 软件开发方法.....	9
5.2.2 处理关键性需求.....	9
5.2.3 计算机硬件资源利用.....	9
5.2.4 需方评审途径.....	10
6 实施详细软件开发活动的计划.....	10
6.1 项目计划和监督.....	10
6.1.1 软件开发计划:.....	10
6.1.2 CSCI 测试计划.....	11
6.1.3 系统测试计划.....	12
6.1.4 文档移交计划.....	12
6.1.5 跟踪和更新计划.....	13
6.2 建立软件开发环境.....	13
6.2.1 软件工程环境.....	13
6.2.2 软件测试环境.....	14
6.2.3 软件开发文档.....	14
6.3 系统需求分析.....	14
6.3.1 用户输入分析.....	14
6.3.2 运行概念.....	15
6.3.3 系统需求.....	15
6.4 系统体系结构设计.....	15

6.4.1	系统各对象关系示意图	15
6.4.2	系统与用户之间的关系	16
6.4.3	各用户使用用例模型	16
6.4.4	系统程序界面设计	17
7	进度表和活动网络图	18
7.1	进度表	18
7.2	活动网络图	19
8	项目组织和资源	19
8.1	项目组织	19
8.2	项目资源	20
9	培训	20
9.1	项目的技术要求	20
9.1.1	管理技术分析	20
9.1.2	开发技术分析	20
9.2	培训计划	21
10	项目估算	21
10.1	规模估算	21
10.1.1	未调整的功能点计数（UFC）计算	21
10.1.2	加权因子（TCF）计算	22
10.1.3	功能点（FP）计算	22
10.2	工作量估算	23
10.3	成本估算	23
11	风险管理	24
11.1	需求风险	24
11.1.1	可能出现的问题	24
11.1.2	解决方案	24
11.2	计划编制风险	24
11.2.1	可能出现的问题	24
11.2.2	相应解决方案	25
11.2	组织和管理风险	25
11.2.1	可能出现的问题	25
11.2.2	相应的解决方案	25

1 范围

1.1 标识

文档标识号：A2016-10-05-00-20

文档标题：软件设计说明

项目/产品中文全称：图书管理系统

项目/产品英文全称：Library Management System

项目/产品英文简称：LMS_1.0

项目/产品编码：201610050020

1.2 项目概述

本文档是对图书管理系统项目的相关计划书，该系统最为基本的功能是满足教师及学生自行注册，信息查询，图书借还、预约、延期等等相关操作并且为数据库系统管理员提供了对图书信息录入、编排、修改、删除等操作的接口。为了符合实际需求，并使系统更加人性化，我们将系统使用者分为了两种：读者和系统管理员。读者又分为教师和学生，这个两个角色有着各自的使用操作和不同权限，例如，教师可以申请借阅一些保护类书籍（借阅范围更大）、借阅时间也相对于学生有所延长（借阅允许时间延长），一次性在借书籍数量上限也有所放宽（借阅上限增大）；读者可以在系统中进行注册，能够修改查询自己的个人基本信息、当前借阅、借阅历史等，另外，读者可以查询图书馆馆藏书籍的各类信息（包括书籍的分类、版本、出版时间、作者、出版社、ISBN 号、借阅情况、当前可借量）。系统管理员有系统最高权限，可以新增、删除数据库中书籍，修改，查询书籍信息，并且可以对读者进行相应管理。

本系统开发者：计算机学院软件工程课程小组 20 组。支持机构：北京航空航天大学计算机学院软件工程课程。

1.3 文档概述

为了保证项目团队按时保质保量地完成项目开发中的各项目标，便于项目团队成员更好地了解项目情况，使项目工作开展的各个过程合理有序，故以文件的

形式来对项目中的各项问题、安排加以描述，作为项目团队成员以及项目干系人之间的共识与约定，项目生命周期内的所有项目活动的行动基础，项目团队开展和检查项目工作的依据

本文档编写于系统需求分析之前，是基于软件工程——软件项目管理课程部分出发，只从设计的角度来描述该系统的总体架构和基本设计思路、扩展，没有涉及到具体的设计内容，是图书管理系统软件开发设计的重要参考。在此基础上，协同完成项目的开发，确保系统开发的进度与质量。

文档包括的部分有：系统概述、产品交付、工作概述、项目开发计划、项目开发具体细节、进度管理、项目组织和资源、项目预算管理、风险管理等等。

本文档仅限开发小组成员及指导老师间共享。

2 引用文档

由于软件开发计划书是第一个文档，所以没有其他的引用文档。

3 交付产品

3.1 程序

软件名称：图书管理系统

编程语言：Java

开发工具：Eclipse Mars

数据库软件：MySQL

环境支持：Windows 7/8/8.1/10 或 Windows XP

功能：基本功能包括读者的注册、查阅修改个人信息、查询书籍信息、借阅/续借/归还书籍；系统管理员增加/删除/修改/查询书籍信息，增加/删除/修改/查询读者信息。读者信息包括：姓名、学号/工号、类型、可借图书上限、单本借阅时长上限、当前借阅、借阅历史。书籍信息包括：书名、版本、作者、出版日期、出版社、ISBN 编号、书籍数量、书籍当前余量。

3.2 文档

- 《软件开发计划书》——项目计划书

- 《软件需求规格说明书》——项目需求分析
- 《软件设计说明书》——软件设计框架说明
- 《软件测试说明书》——关于系统测试的设计说明书

3.3 服务

为保证用户能够正确而方便地对系统进行使用，我们将会推出以下服务：

视频教程：以视频的形式将使用系统的具体操作方法和流程放在互联网上供人们下载观看教程，具体地了解操作的各个步骤。

技术咨询：在工作时间内可以向技术人员询问系统相关问题，并由其给出相应解答。

远程协助：当客户遇到难以处理且描述的问题，可以向技术人员申请远程支持，协助解决遇到的问题。

软件维护：用户使用过程中遇到 Bug 和系统缺陷，向技术人员反馈之后，将对系统进行相应的修复，并且为使用方提供系统补丁。

软件升级：当系统开发版本升级时，将向用户提供升级程序

3.4 非移交产品

- 系统源代码文件
- 系统相应配置文件
- 数据库配置文件
- 项目内部参考文档（仅限于项目成员之间的交流）

3.5 验收标准

客户根据软件需求分析说明书中各项需求对系统功能逐项进行测试并且检查通过即认定为项目通过验收。

3.6 最迟交付期限

项目最迟交付期限是 2016 年 12 月 20 号。

4 所需工作概述

4.1 所开发系统的需求和约束

系统需要满足包括但不限于图书馆日常基本场景功能的实现。其中包括读者和系统管理员两个角度的各项功能：读者的注册、查阅修改个人信息、查询书籍信息、借阅/续借/归还书籍功能；系统管理员的增加/删除/修改/查询书籍信息、管理读者信息等功能。为使得系统管理信息更加清晰，规定书籍信息至少包括：书名、版本、作者、出版日期、出版社、ISBN 编号、书籍数量、书籍当前余量等，并且可以提供书籍信息扩展的功能。

4.2 对文档编制的需求和约束

项目文档作为对项目的指导或总结（包括但不限于项目开发计划书），需要能够完整而准确地传达相应的信息，达到一定的质量要求，所以需要做到一些几点要求：

针对性：需要分清楚文档使用对象，针对不同类型、不同层次的读者，需要从他们的角度出发，编写适合其阅读和理解的文档，对文档的形式也需要针对不同读者的阅读习惯进行相应的修改。

准确性：文档中对于项目的功能、架构等等的描述应该做到准确无歧义，应该保证用户在使用时不会因此造成阅读障碍。

完整性：对系统的描述应该保证完整，能够让读者在阅读之后对系统有总体上完整清楚的了解。

直观性：可以采用多种形式来对文档进行编写，例如，采用概念图、E-R 图、表格等等，使得文档更加直观易懂，降低读者的阅读门槛，并提高阅读文档的效率。

灵活性：针对项目的种类和项目的不同部分，文档的规模和复杂度都应该有所不同，在编写文档时，需要根据相应的实际情况来对文档进行适当的修改。

4.3 项目进度和资源需求

项目针对实际场景和系统的基本功能实现方式划分不同模块进行开发，同时

根据不同的模块的任务量和资源需求进行合理安排模块开发的时限和资源分配，保证能够使得各模块协调有序开展，同时能够满足项目中对于功能质量、进度安排、资源分配的要求。

4.4 其他需求和约束

4.4.1 对系统设计原则的约束

为保证系统设计的合理性，在设计时应该满足以下原则：

先逻辑后物理的原则：图书管理系统是被严格的划分，采用结构化的方法来进行研究，在综合的基础上进行论证和分析，从而探索和实践的系统逻辑方案，是最终能够为用户解决问题的系统。系统设计应该首先关注需要“做什么”，然后根据系统的分析结果和逻辑程序，进入实施阶段的设计和物理系统。

用户参与原则：图书管理系统是用户直接参与管理的信息系统，是人机交互频繁的系统，会直接参与到辅助和协调工作的任务中去，并最终将成为各阶段建设系统工作的一部分。该系统的用户是学校师生及相关部分图书管理工作，本系统建设的直接目的就是满足他们在图书管理活动中的信息需求与功能需求。在本系统的开发建设中，要把用户的特点和使用上的习惯与要求放在考虑范畴之内，以调节人机关系和提高人机系统的一体化水平为衡量标准。

“自顶向下”的原则：“自顶向下”的原则需要始终在本系统的开发过程中被贯彻执行，在系统分析、系统设计与系统实施等各个阶段中都应该能充分体现。一整套的系统分析、设计、开发的完成是建立在以系统的总体目标和功能作为标准，逐级分解，逐级细化的基础上的。这样建立起来的系统，保证了三方面的平衡：结构的合理化，总体与各功能模块能相互协调，总体目标和总体功能的实现具有保障。

4.4.2 对系统设计/功能的其他约束

可扩展性

可扩展性是指系统确保可持续的增长来满足复杂的业务需求和兼容需求，系统模型中的动态变化刚开始通常表现只是很小的性能/时间开销，但是随着指数增长，对系统性能的需求也相应的增长。这种增长的速度非常之快，它不仅表现

在支持用户的数量上，而且也表现在性能和用户的复杂性与整合服务上。提高图书管理系统的功能，是一个渐进的过程，用户的要求和图书管理技术的发展是不断完善的。因此，可扩展性在这里是特别重要的。

可靠性

在规定的时间内系统对用户的请求做出响应，并且对响应的质量有所保证，同时保证任意合法用户能够在任意合法时间通过校园网或者是任意合法局域网对系统进行访问获取或修改对应信息，这是对系统可靠性的一些重要标准。因此系统的设计必须与硬件、软件支持环境协调，以确保系统的稳定运行。

易于使用

由于图书管理系统是人机交互十分频繁的系统，因此保证系统简单、清晰、易懂是对系统质量的一个重要考核部分。用户界面友好、使用简单、操作方便，系统才能更好地协助图书馆进行相关工作的开展。

可维护性

系统在保证可扩展性的同时应该能够保证系统的可维护性。系统设计应采取充分考虑了系统的硬件和软件的可维护性，采用可扩展、易维护的硬件系统，同时应用软件系统应该采用模块化设计，系统所提供的服务模块/功能应该清晰、明确，满足在业务量增长和硬件、软件升级换代的情况下系统仍旧能够合理有效地工作，维护人员仍旧能够简单便捷地进行日常维护。

安全性

对系统管理员的授权、认证应该保证安全，认证方式应该尽可能多样，出现权限泄露迹象时应该及时预警，如管理员账号异地登录提醒等，保证管理员权限的安全此外，系统必须留下记录或跟踪记录每一个重要的操作和处理，以便有效地管理相关的权利和责任。同时在设计之时应该保证普通用户和管理员用户权限没有重叠，不同权限用户之间的权限分明，保证不出现普通用户因为系统漏洞而获得高权限的情况。

5 实施整个软件开发活动的计划

5.1 软件开发过程

以下为软件开发各阶段相应安排：

阶段	子阶段	任务	产出	验证要求
项目启动阶段	了解项目	1、确定项目 2、了解项目相关知识		
	设定计划	1、对项目内容进行探讨 2、编写项目计划书	《软件开发计划书》	评审
需求分析阶段	需求获取	1、目标人群问卷调查 2、参照同类系统需求描述	系统需求初步调研	
	需求分析	1、需求识别与建模 2、需求分析与综合 3、完成 SRS 文档(系统需求规格文档)撰写 4、对需求模型和系统需求规格文档进行评审和验证	1、系统需求模型 2、SRS 文档	评审
系统设计阶段	概要设计	1、对需求模型进行分析并给出设计思路 2、完成概要设计文档	概要设计文档	
	数据设计	1、分析数据特点，归纳数据信息，对接底层数据库 2、归纳得到数据字典 3、数据库底层设计	数据字典	
	详细设计	1、软硬件结合完成系统细节设计 2、完成软件结构设计说明书	软件结构设计说明书	评审

编程实现阶段	各模块实现	1、实现读者模块的各项功能 2、实现系统管理员模块各项功能 3、进行相关测试	各模块功能源代码	
	界面 UI 设计	1、编写前台用户界面 2、编写系统后台管理员界面 3、进行相关测试	界面源代码	
	软件集成	1、将两大功能模块与数据库集成 2、集成界面、数据库、模块 3、进行相关测试	系统 Alpha 版本	集成测试
系统交付阶段	系统测试	1、系统安装配置 2、按照《软件测试文件》进行相关测试 3、进行相关调整	《软件测试说明文档》	评审
	正式交付	1、提交项目源代码 2、提交相关文档 3、提交用户手册	1、源代码 2、文档 3、用户手册	
项目关闭阶段	总结	1、完成项目总结分析 2、撰写项目总结	项目总结报告	
	维护	1、为用户提供技术支持 2、准备后续的系统升级		

表 5-1 软件开发阶段安排

5.2 软件开发总体计划

5.2.1 软件开发方法

系统的开发过程中将使用面向对象的编程方法，开发模式将采用瀑布流的方式。严格按照这种方式保证软件能够准时保质保量地顺利开发完成。

系统的开发将分为制定计划、需求分析、系统设计、编写程序、软件测试和运行维护等部分，在《项目计划书》、《需求规格说明书》、《软件设计说明书》、《软件测试说明书》等文档的基础上进行系统的开发，并且按照产品开发基线逐步有序进行。

5.2.2 处理关键性需求

5.2.2.1 安全性保证

安全性保证建立在系统设计的正确性和完备性上。为保证正确性上，需要系统可能存在的一些漏洞进行严格的审核（比如是否存在书籍信息/个人信息泄露、用户之间权限是否分明，是否能够保证权限隔离、权限不发生泄露，是否存在通过系统漏洞人为提高权限的可能性）。在此基础之上，才能确保系统能够正确地用于日常使用。

5.2.2.2 保密性保证

在不同环境中对于系统的保密性要求不同，由于学校图书管理系统常常会被整合到学校系统中，故对于设计保密性较强的部分应该进行特别处理，从而适应相关环境中的使用。但是由于我们的图书管理系统对于保密性的要求不是很高，故一般性的约束即可。

5.2.2.3 私密性保证

私密性需求是建立在不同用户之间信息相对隔绝的基础之上的。为了满足不同用户之间的隔绝，就需要保证用户登录时验证过的完整性和不可越过。系统管理员因为可以进行相应信息的管理，但是也需要做到不能将信息泄露。

5.2.3 计算机硬件资源利用

在预算范围内使用计算机硬件资源如服务器也需要保证能够做到不浪费，尽量提高硬件资源的利用率。在项目不同部分需要根据实际情况对其进行合理的分配，不能因此而引起项目开发中的矛盾。

5.2.4 需方评审途径

在软件开发过程中，需方可以进行项目进度的督促，也能够了解相应开发进度和开发内容；但是对于需方也有一定的限制，需要保证开发程序源代码不能被带出，不能泄露开发实现核心细节。不够由于本项目没有很严的要求，所以此点可以适当放宽。

6 实施详细软件开发活动的计划

6.1 项目计划和监督

6.1.1 软件开发计划：

任务名称	任务内容	任务起止时间
项目开发计划书撰写 (项目启动)	描述项目系统的总体开发计划，保证项目团队按时保质保量地完成项目开发中的各项目标；对项目中的各项问题、安排加以描述，作为项目团队成员以及项目干系人之间的共识与约定；作为项目生命周期内的所有项目活动的行动基础，项目团队开展和检查项目工作的依据。	2016.09.22- 2016.10.10
需求规格说明书撰写 (需求分析)	分析使用者对系统的使用要求及对应的功能具体划分，对系统功能需求、性能需求和其它非功能性需求进行详细的描述，并以可验证的方式做出完全、精确陈述，为之后的系统设计提供基础，同时也为后面编写测试样例和系统测试提供依据。	2016.10.11- 2016.10.25
结构设计说明书撰写 (系统设计)	确定系统的总体设计、总体结构、模块设计、接口设计，并给出详细设计以供编程人员参考，用于指导各构件的实施、集成及测试；通过使用多种视图来从不同角度描述本系统的各个主要方面，以满足图书杂志采购和借阅系统的相关涉众对本系统不同的关注焦点和需求；记录并表述了系统架构的设计人员对系统构架方面做出的重要决策；将本	2016.10.26- 2016.11.10

	文档中所涉及的所有术语、缩略语进行详细的定义（数据字典）。	
软件测试说明书撰写 （测试计划）	<p>为了在软件投入生产性运行之前，尽可能多地发现软件的错误。文档将围绕这几个方面出发，设计相关的测试方法/样例，并且记录预期结果供开发人员参考：</p> <p>依据需求分析中的各项需求逐项分析是否完成相应功能的开发；</p> <p>测试系统中的各个功能模块是否满足用户要求，并测试是否存在 bug，是否达到预期；</p> <p>测试系统是否能够使系统进行快速的改进和升级；</p> <p>测试系统是否满足性能和效率的要求；</p> <p>重复测试验证系统的可靠性和对软硬件环境的适应性。</p>	2016.11.11- 2016.11.25
程序编写、调试和系统测试（系统实现）	以项目计划书、需求分析规格说明书、结构设计说明书、软件测试说明书这些文档为基础，设计编写相应程序和调试工作，采用模块化编程和边开发边测试的基本模式，对每个模块都安排专人进行单独测试，系统联调及系统测试，对系统处理逻辑、例外处理能力、容错能力等进行大规模的测试，对发现的问题进行彻底纠正。	2016.11.26- 2016.12.20
项目交付及总结 （系统交付）	对系统源代码及相关文档进行交付并做出相关总结	2016.12.20- 2016.12.20

表 6-1 软件开发计划安排

6.1.2 CSCI 测试计划

CSCI 是计算机软件配置项(Computer Software Configuration Item)，对于软件配置项的测试主要针对系统在不同软件环境下安装配置时可能出现的问题。

系统	MySQL	是否能够正常运行
Win7	5.6	
Win8	5.6	
Win8.1	5.6	
Win10	5.6	
Win7	5.7	
Win8	5.7	
Win8.1	5.7	
Win10	5.7	

表 6-2 CSCI 测试计划表

6.1.3 系统测试计划

系统测试分以下四步进行：单元测试、集成测试、系统测试、验收测试。由于验收测试是需方共同参与下进行，故在此仅仅进行前三项的测试计划。但是要注意这几项测试的区别：

单元测试是对软件中的基本组成单位进行的测试，如一个模块、一个过程等等。它是软件动态测试的最基本的部分，也是最重要的部分之一，其目的是检验软件基本组成单位的正确性。因为单元测试需要知道内部程序设计和编码的细节知识，一般应由程序员而非测试员来完成，往往需要开发测试驱动模块和桩模块来辅助完成单元测试。

集成测试是在软件系统集成过程中所进行的测试，其主要目的是检查软件单位之间的接口是否正确。根据集成测试计划，一边将模块或其他软件单位组合成越来越大的系统，一边运行该系统，以分析所组成的系统是否正确，各组成部分是否合拍。

系统测试是对已经集成好的软件系统进行彻底的测试，以验证软件系统的正确性和性能等满足其规约所指定的要求。系统测试应该按照测试计划进行，其输入、输出和其他动态运行行为应该与软件规约进行对比。

序号	测试类型	内容	负责人	时间
1	单元	读者模块测试		
2	单元	系统管理员模块测试		
3	单元	网页界面测试		
4	集成	读者/管理员集成测试		
5	集成	数据库/网页集成测试		
6	系统	系统完整测试		

表 6-3 系统测试计划表

6.1.4 文档移交计划

根据项目的进度会不断向需方移交软件相关文档，使得能够充分交流。

交付部分	预定的交付日期
《项目计划书》	2016.10.10
《软件需求规格说明书》	2016.10.25
《软件测试说明计划书》	2016.11.10
《软件设计说明书》	2016.11.25
《软件测试文档》	2016.12.20
系统源代码	2016.12.20

表 6-4 文档移交计划表

6.1.5 跟踪和更新计划

由于软件有后续的系统更新计划，所以对于项目依旧应该保持追踪，且在后续的维护过程中不断修复新发现的系统漏洞，并且尝试将项目的可使用范围扩大，兼容更多系统（比如增加 C/S 架构，提供 Android 和 iOS 的应用）。

阶段	内容	时间	负责
维护	技术支持/修复产品漏洞	2016.12.21 - 2017.03.01	
维护/开发	技术支持/v2 版本项目启动	2017.03.01 - 2017.03.15	
	技术支持/v2 版本需求分析	2017.03.15 - 2017.04.01	
	技术支持/v2 版本兼容设计	2017.04.02 - 2017.04.16	
	技术支持/v2 版本编程实现	2017.04.17 - 2017.05.20	
	技术支持/v2 版本系统测试	2017.05.21 - 2017.06.01	
	技术支持/v2 版本发布	2017.06.01	

表 6-5 软件更新计划安排表

6.2 建立软件开发环境

6.2.1 软件工程环境

硬件：

4 台 PC 机：

内存：4G 及以上

硬盘：500G 以上；

CD-ROM：32 倍速以上；

一台服务器：

内存：64G

硬盘：2T

系统：Linux（ubuntu）

网络：

使用 TCP/IP 协议的局域网

软件：

操作系统	Windows 10
开发工具	Eclipse Mars
编程语言	Java
JDK 版本	JDK1.8
数据库	MySQL

表 6-6 软件环境记录表

6.2.2 软件测试环境

硬件：

内存：4G 及以上

硬盘：500G 以上

网络：校园局域网/一台服务器

软件：

系统：Win10

软件：MySQL 5.7 JDK：1.8.0

6.2.3 软件开发文档

文档的书写应该准确规范，规格应该清晰明确。对于软件的一些常用的接口或者是模块设计应该做出简要的描述以确保系统使用者能够对系统有一定的了解和排除错误的能力，以此来减小技术支持人员的工作负担，并为系统使用人员带来一定程度的方便。

6.3 系统需求分析

6.3.1 用户输入分析

对于读者用户来说，其拥有注册、登录、查询个人/书籍信息、借/还/续借书等操作，此外，读者还具有根据不同身份实现差别化操作的功能；

对于系统管理员来说，其拥有登录、增加/删除/修改/查询书籍信息，管理读

者信息等等操作。

6.3.2 运行概念

对于读者的操作而言，注册相当于在数据库的读者信息表中增添一项，而其登录即从读者信息表中查询是否有匹配的读者项，查询个人信息即从数据库中申请查询个人的信息项，查询书籍信息即是先向数据库中申请查询书籍信息表中的对应项；借还书操作相当于向系统提出申请，系统判断申请是否合理，如果合理则系统修改书籍的余量。

此外读者中由于分成了两类，所以在用户的信息表中必须有用户的类别之分，此外系统在辅助读者借阅书籍的操作中应该根据读者的身份来判断操作是否合法，如果不合法应该给出相应的警告信息。

对于系统管理员来说，登录即从管理员信息表中查询是否有对应项；增删改查书籍信息即拥有管理书籍信息表的权限，此外系统管理员拥有的对读者的管理操作即管理员能够对读者信息表进行管理，比如能够在读者信息表中增加读者、删除读者信息，同时更新读者信息，查询读者的信息。

6.3.3 系统需求

需要能够满足上面两类用户的相应操作，同时能够无差错、安全地维护读者信息表、管理员信息表、图书信息表、当前借阅情况表等几项基础的数据表。并且在此基础上应该保证系统尽可能地高效，并且兼容性强，不会出现因为软硬件环境的些许差异而导致系统无法正常运作。上文提及的保密性要求、私密性要求、安全性要求虽然因为项目相对简单，所以对开发的约束力较弱，但是在软件设计和开发时都应该有所体现。

6.4 系统体系结构设计

6.4.1 系统各对象关系示意图

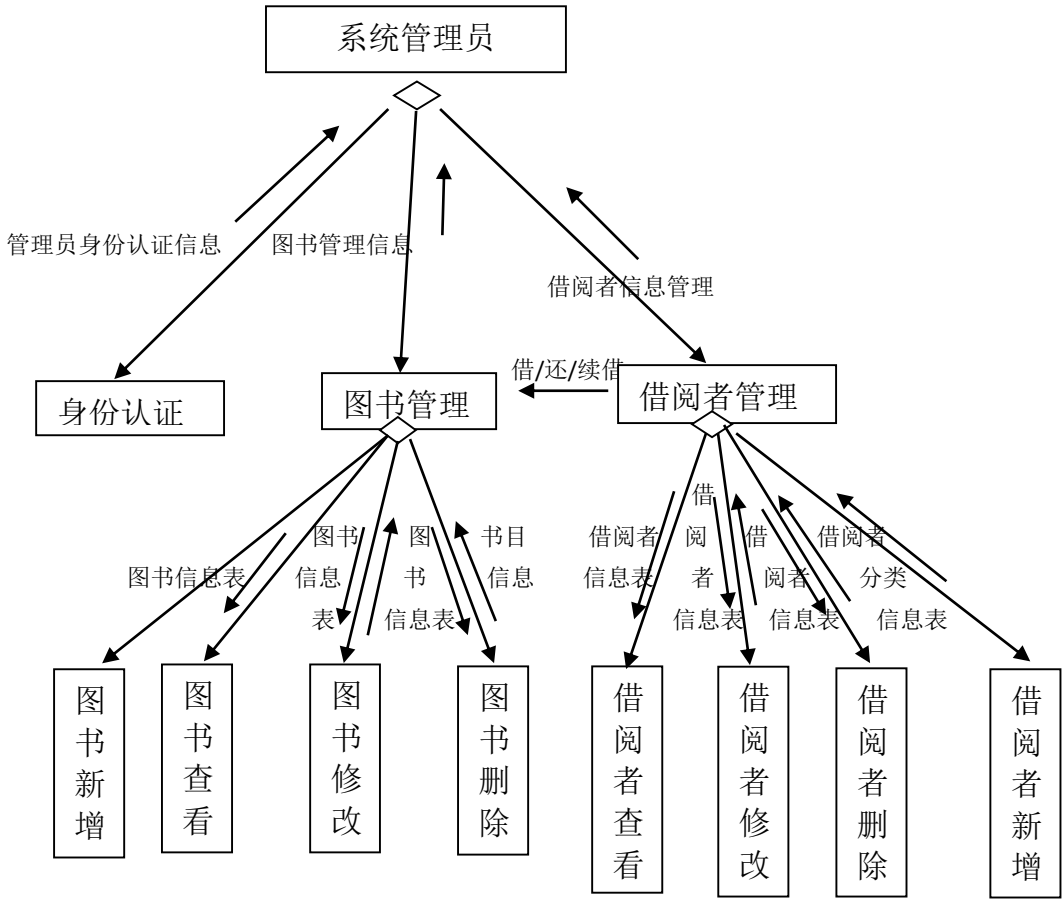
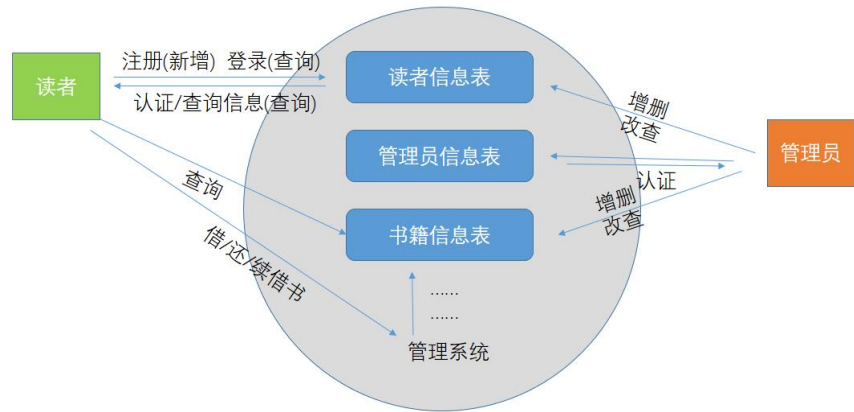


图 6-1 系统各对象关系示意图

6.4.2 系统与用户之间的关系



6.4.3 各用户使用用例模型

系统管理员用户使用基本功能用例：

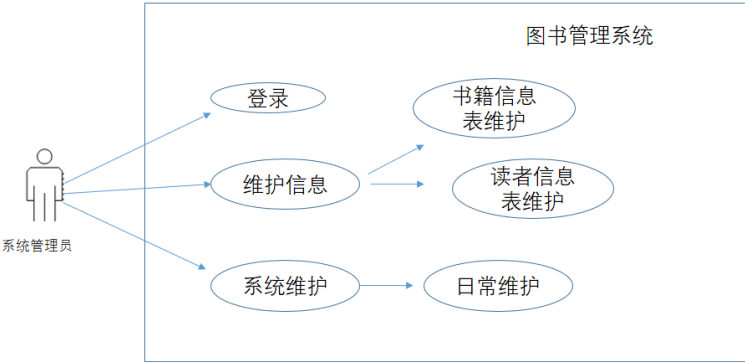


图 6-3 管理员用例模型

读者用户使用基本功能用例：

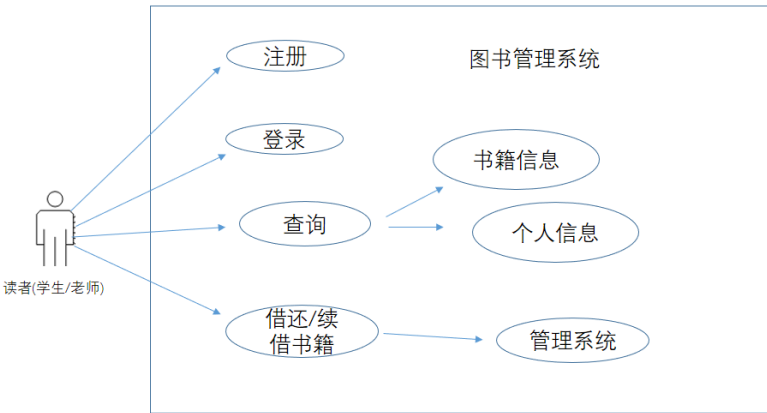


图 6-4 读者用例模型

6.4.4 系统程序界面设计

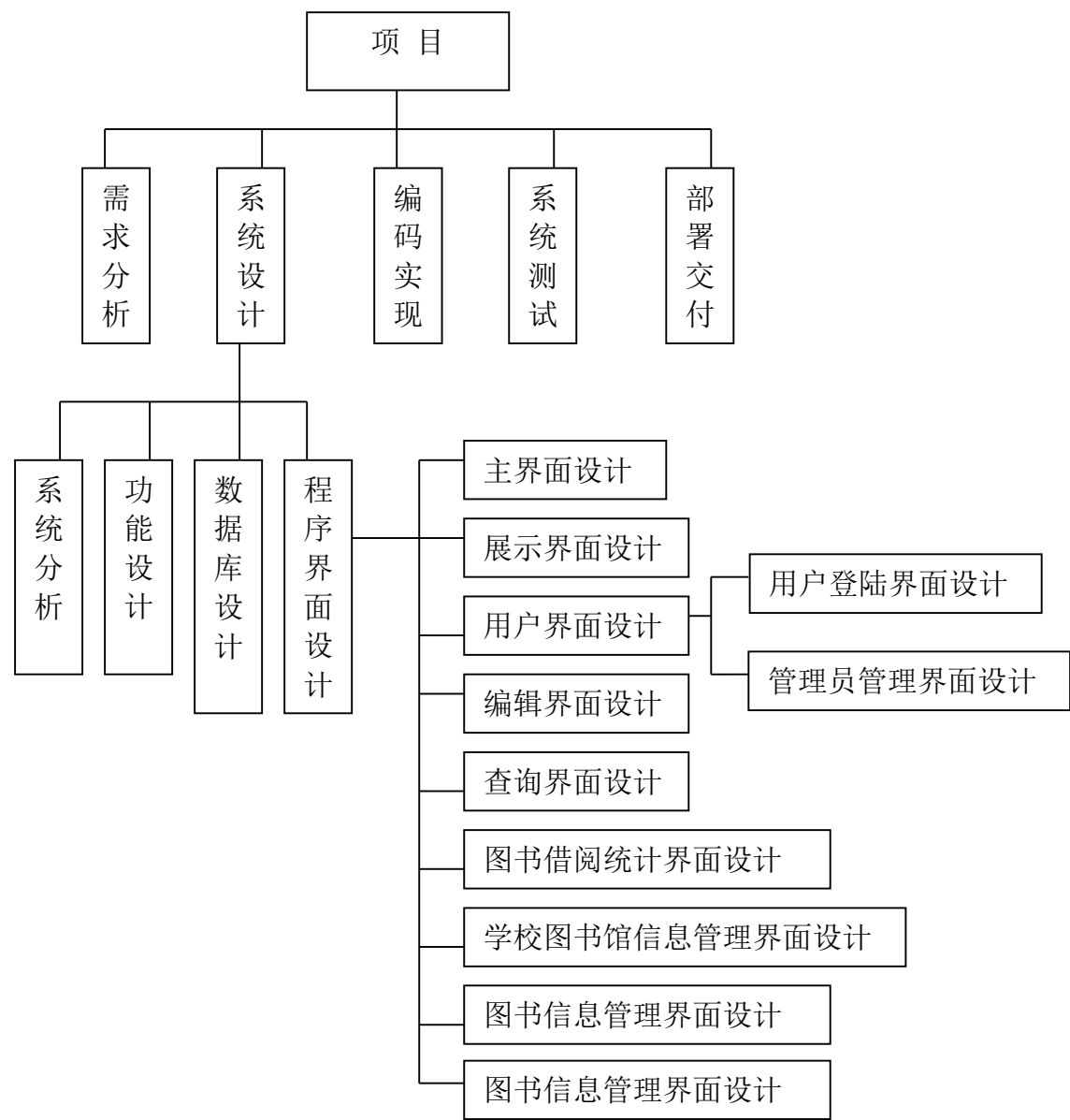


图 6-5 软件界面内容示意图

7 进度表和活动网络图

7.1 进度表

任务开始时间	任务结束时间	任务内容
2016.10.1	2016.10.10	完成软件开发计划
2016.10.11	2016.10.20	完成软件需求规格说明
2016.10.21	2016.10.31	完成对软件各个部分的设计

2016.11.1	2016.11.31	完成对软件的编程，实现基本功能
2016.12.1	2016.12.10	对软件进行相应的优化
2016.12.11	2016.12.20	对软件进行相关测试和修复
2016.12.21	2016.12.31	将软件交付给客户并完成后续工作

表 7-1 项目进度表

7.2 活动网络图

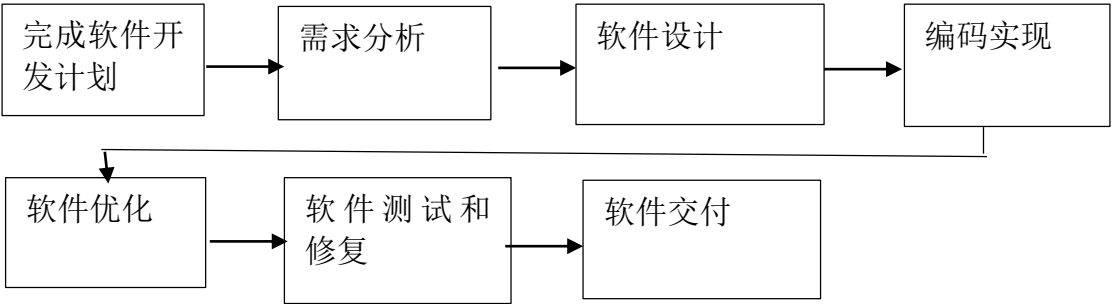


图 7-1 项目活动网络图

8 项目组织和资源

8.1 项目组织

项目组长：曾宇涛

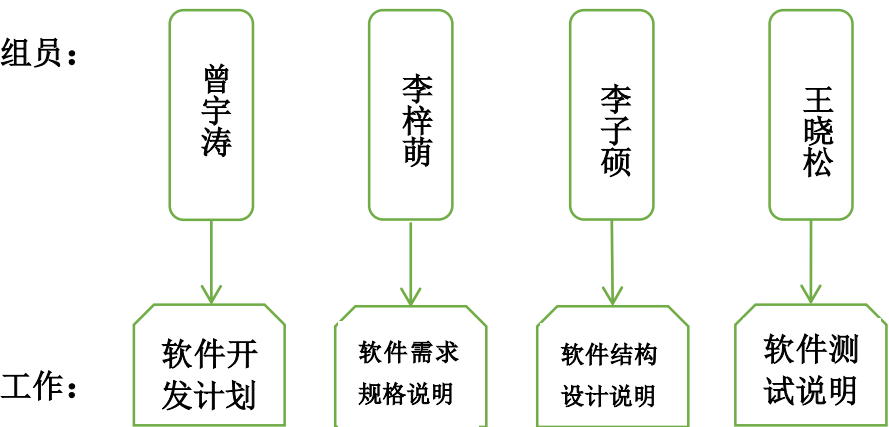


图 8-1 项目组织图

8.2 项目资源

人力资源：

姓名	性别	技能
曾宇涛	男	JAVA, HTML, CSS, JS, SQL
李子硕	男	JAVA, HTML, CSS, JS, SQL
李梓萌	男	JAVA, HTML, CSS, JS, SQL
王晓松	男	JAVA, HTML, CSS, JS, SQL

表 8-1 人力资源图

设备资源

地点：宿舍、空教室或机房

使用设备：私人电脑 4 台，服务器 1 台

9 培训

9.1 项目的技术要求

9.1.1 管理技术分析

根据图书管理系统部分基本功能（图书类别管理，图书入库管理，图书信息修改管理，图书借还管理，图书查询管理，图书超期管理），开发一款图书管理系统所需的管理技术为：文献资源采购技术，对已购文献资源进行分类、著录、编目、校审及加工的技术。因此在项目初期负责人应该主动收集相应的信息进行学习，并且利用集中的时间将相关知识向项目组其他成员进行传播，避免在软件项目开发的时候出现因为业务知识不熟练而造成系统出现严重设计漏洞，大大降低软件开发的效率及质量。

9.1.2 开发技术分析

本图书管理系统采用 B/S 架构（之后随着项目的逐渐推进，将支持客户端的访问，增加系统的使用便捷性），故需要掌握两方面的知识。对于网页部分，开发人员应当具备 Web 开发能力、服务器后端开发能力；除此之外，对于服务器

部分，开发人员应当能熟练使用数据库相关技术。具体如下：

对应于图书管理系统的权限管理，读者管理，数据管理功能，所需开发技术为 DBMS 相关应用技术；为满足客户网上查询、预定、续期的需求，后台人员需具备 Web 开发能力。

9.2 培训计划

为保证项目团队之间沟通顺畅，不会因为知识上的偏重而无法进行相应的交流，故应组织一定的培训来确保项目每个成员都对系统有一个完整地认识 and 了解。培训内容如下：

DBMS 技术培训：每周两次，由组长组织

Web 开发培训：每周一次，由组长组织

10 项目估算

10.1 规模估算

功能点是基于应用软件的外部、内部特性以及软件性能的，一种间接的软件规模测量。功能点可以作为经验统计参数化软件成本估计公式和模型的输入，以对软件的成本进行估计。鉴于功能点法在软件规模测量领域的广泛影响力，该项目规模估计将采用功能点法。功能点法由三个逻辑部分组成：未调整的功能点计数、加权因子和功能点。

10.1.1 未调整的功能点计数（UFC）计算

$$\text{UFC} = \text{外部输入} + \text{外部输出} + \text{外部查询} + \text{内部逻辑文件} + \text{外部接口文件}$$

功能项	权 重		
	简单	一般	复杂
输入	3	4	6
输出	4	5	7
查询	3	4	6
外部文件	7	10	15
内部文件	5	7	10

表 10-1 功能点权重图

查上表，全部权重取最轻

$$UFC = 3+4+3+7+5 = 22$$

10.1.2 加权因子（TCF）计算

$$TCF = 0.65 + 0.1 * \sum_{i=1}^{14} k_i$$

上式 $\sum_{i=1}^{14} k_i$ 表示 14 项复杂度调整项之和， k_i 取值从 0 至 5 共六个等级，

这里所有等级取最高

$$TCF = 0.65 + 0.1 * (14 * 5) = 1.35$$

10.1.3 功能点（FP）计算

$$FP = UFC * TCF = 29.7$$

程序设计语言	级别	每个功能点代码行数
ACCEL	17.00	19
Access	8.50	38
EXCEL 1-2	51.00	6
EXCEL 3-4	55.00	6
EXCEL 5	57.00	6
JAVA	6.00	53
ORACLE	8.00	40
Oracle Developer/2000	14.00	23
C	2.50	128
C++	6.00	53
dBase III	8.00	40
dBase IV	9.00	36
DELPHI	11.00	29
SQL	25.00	13
SQL-Windows	27.00	12
Visible C	6.50	49
Visible COBOL	8.00	40
Visual 1	35.00	9
Visual 4.0	11.00	29
Visual Basic 1	7.00	46
Visual Basic 2	7.50	43

表 10-2 功能点估算图

若该项目用 Java 或 C++编写，代码量估算为：

$$\text{LOC} = 29.7 * 53 = 1574.1 \text{ 行}$$

10.2 工作量估算

每人平均 1 篇文档与 394 行代码

10.3 成本估算

杂费如水电费不计

假设每千行代码（KLOC）编写成本为 1000 元，维护成本为 3000 元，总费用：

$$1574.1 / 1000 * (1000 + 4000) = 6296 \text{ 元}$$

11 风险管理

11.1 需求风险

11.1.1 可能出现的问题

需求已经成为项目基准写入了项目计划书，但需求还在继续变化，甚至在开发过程中开发人员或者是需求方修改需求；

需求定义欠佳，简化了问题，但项目开发已经进行到一定阶段，进一步定义会扩展项目范畴，增加时间成本和项目预算提高，造成项目延期或者是无法正常交付；

产品定义部分过于含混，而在需求分析时客户参与度不够，导致对需求理解不够，最终造成系统的可扩展性大大下降，项目更新和再开发受阻。

11.1.2 解决方案

保证需求方充分表达自己的意见，并且在项目计划书的撰写阶段不断征求其意见，对需求做出及时的更新和修改，并且设计时针对用户的使用习惯和之后的扩展要求对系统的总体设计进行改进，确保需求准确，同时在项目计划书落定之后需求不再更改。

11.2 计划编制风险

11.2.1 可能出现的问题

计划、资源和产品定义仅仅由项目计划书编写人员自定义，项目组内成员对项目理解并不完全一致；

计划过于优化，是“最佳状态”，没有考虑到现实中出现的不定因素，导致计划不现实，后期编程实现过程与计划完全脱节；

计划基于使用特定的小组成员，而该成员对该部分的知识不足，无法完成相应的部分，导致项目搁浅，最终无法交付；

产品规模（代码行数、功能点、与前一产品规模的百分比）比估计要大许多，导致项目最终无法完成或者严重延期；

完成目标日期提前，但没有相应地调整产品范围或可用资源，导致资源浪费，项目开发节奏时快时慢。

11.2.2 相应解决方案

计划编制阶段应该充分考虑各种因素，对于一些不定因素或者是可能出现的突发事件，要及早提出后备方案；

同时计划的编制应该建立在完全考察小组成员的基础之上，针对各人能力差异合理安排任务和进度，保证项目能够稳步推进；

任务量的考察应该参照同类项目的任务量，以避免出现项目量估计过小而造成项目最终无法交付；

对于可能出现的项目提前完成的情况，需要建立相应的协调预案，保证能够充分利用人力资源和硬件资源。

11.2 组织和管理风险

11.2.1 可能出现的问题

仅仅由项目管理人员进行技术决策，导致决策任务堆积，浪费了开发者的开发时间，导致项目推进缓慢，计划时间延长；

项目组织结构混乱，大大降低了生产率；

预算不足导致项目各部分发生竞争，打乱项目计划；

缺乏必要的规范，导致工作失误与重复工作；

非技术的第三方的工作（预算批准、设备采购批准、法律方面的审查、安全保证等）时间比预期的延长。

11.2.2 相应的解决方案

项目技术决策应当由相关开发人员提出解决预案，然后由管理人员做出相应决策，同时对于较小的部分内容，管理人员应当主动放权；

项目组织应该清晰明确，各个成员有着自己独立的分工，不应出现因为分工不明确而造成责任相互推卸的情况；

对于硬件资源的使用应该分时分阶段，尽量将使用任务错开，避免因为资源

竞争使用而拖累项目进度；

对第三方工作需要在项目开始阶段尽可能准备完善，减少之后在此方面花费的时间，加快项目进度。