**软件开发计划**

北航计算机学院软工三组

**1 引言**

**1.1标识**

项目/产品中文全称：图书查询借阅管理系统。

项目/产品英文全称：Library Query Checking Management System

项目/产品英文全称：LQCMS\_1.0

**1.2系统概述**

该软件旨在为高校提供一个高效的、方便使用的、能够满足师生借阅查询需求的图书借阅管理平台。使用本软件，用户注册之后，可以查询书籍的名称，类别等信息，并且根据所得信息进行借阅、预借、归还书籍的操作，并且用户分为教师和学生两类群体，师生之别体现在借阅图书的权限等方面；本系统的管理员还可以进行书目的编排和录入操作。

该项目的需方为北京航空航天大学图书馆，面向的用户群体学校师生。该项目由计算机学院软件工程第三小组负责开发，由北航图书馆、北航计算机学院提供相应支持。

**1.3文档概述**

本文档详细叙述了软件开发设计过程中涉及的一切程序、引用、文档等，为软件开发的每一个步骤诸如日程安排，实现方法做出详尽的计划。需方可以通过本文档掌握软件开发的大致过程，开发者则根据计划来安排生产活动。

本文档的主要内容有项目总体和详细计划软件开发具体计划，项目进度表，项目组织，资源管理，风险评估管理等几大方面。

该文档版权归属于北航计算机学院软工三组，仅在北京航空航天大学内部参考使用。他人出于盈利目的使用本文档必先予以告知，否则当侵权论处。

**1.4与其他计划之间的关系**

本项目是独立开发的项目，与其它项目均无关联，所以本项目计划与其它项目之间没有必然联系。

**1.5基线**

需求基线：详见本文档系统概述、所需工作工作概述两节，以及《软件规格需求说明》。

设计基线：详见于《软件结构设计说明》。

测试基线：该基线包含所有交给测试人员之前所有成果的标识。开发者可以提供的有《软件开发计划》《软件规格需求说明》《软件结构设计说明》，在测试开始之前，开发人员会对测试人员进行必要的讲解和培训，包括数据库的结构和软件的基本使用方式，以至于代码风格等要素。

发布基线：软件发布后有关于指导用户如何使用的内容，可见于《软件用户手册》。

**2引用文件**

本项目开发过程中所使用的文档资料如下。

a. 《软件工程-实践者的研究方法》 （美）Roger S. Pressman 著 郑人杰 马素霞等译

b. 《中华人民共和国国家标准GB/T 8567-2006 计算机软件文档编制规范》

c. 《软件工程国家标准规范——软件文档编制规范》

d. 本项目经核准的计划任务书、合同和上级机关的批文。

**3交付产品**

**3.1 程序**

项目名称：图书查询借阅管理系统

编程语言：PHP,HTML,JAVASCRIPT,CSS

功能描述：交付的产品中，软件部分是一个WEB工程文件，使用PHP,HTML,JAVASCRIPT,CSS编写。图书管理员可以在权限允许的范围内修改书籍信息和读者信息；读者（包括师生两大群体）可以根据需要查询想要的书籍，既可以查询某一本书籍，也可以根据所提供的信息寻找需要的同类型书籍。读者可以根据查询结果和自身需求来索取图书的使用权，也可以通过本系统提供的渠道来归还已借阅的书籍。

**3.2文档**

本工程所包含的其它文档包括如下。

软件规格需求说明：详尽叙述了本项目对规格的具体要求。

软件结构设计说明：描述了与本项目有关的软件结构的设计。

软件测试说明：对测试的计划和方法等与测试相关的具体细节做出了规定

软件用户手册：通过用户手册可以指导用户（包括管理员和师生）如何使用、维护和管理软件，提供与之相关的一切细节。

**3.3服务**

本项目计划提供的服务如下：

管理人员培训：告知管理人员他们所要承担的责任，指导他们熟练掌握相应的图书管理的技能。

咨询服务：管理人员和用户可以询问有关该软件的问题并获得解答。

意见反馈：任何用户均能提供使本软件得到改进的意见和建议，或者是反馈本软件存在的缺陷。

补丁发布：发布并提示用户安装问题修正补丁。

技术支持：某些情况下，采取上门指导的方式。

软件升级：及时通知使用者有关软件升级的事宜，并且在用户授意后进行软件升级。

**3.4非移交产品**

可行性分析报告：综合考虑社会、经济、技术以及其它现实状况，对该软件实现的可能性进行论证，给出可能的实现方法，并同时对其可行性予以论证。

项目开发计划：具体地计划了本项目的开发过程，详细描述了本项目的进度安排，资源状况，人员分工等一切项目开发的指导性工作。

软件规格需求说明：对所开发软件的功能、性能、用户界面及运行环境等做出详细的说明。

概要设计说明书：大体上对程序实现做出规划，为详细设计提供基础参照。

详细设计说明书：描述每一模块的实现方式的细节。

代码编程规范说明：对代码的编程规范做出规约，对特定的方法和类的采用做出规定。

测试计划：对测试做出安排。包括何人，何时，用什么数据对哪一个模块做出测试等。

测试结果分析报告：对测试的结果做出分析，包括问题的分析，经验的总结，以及可能的改进措施等。

开发进度表：对项目的进度进行跟踪，定时填写，定时上交。

项目开发总结报告：软件项目开发完成以后，应与项目实施计划对照，总结实际执行的情况，还需对开发工作做出评价，总结经验教训。

软件问题报告：在软件开发过程中发现的问题及其相关人员、解决方案等状态的记录文件。

软件修改报告：软件投入运行后，如果发现了问题，就对其进行详细描述，记录分析总结的结果。

源程序：软件开发时编写的全部代码以及注释。

**3.5验收标准**

a. 代码的验收：

小组内评审；期末总体评审；代码编写符合HB6465标准；与文档说明保持一致；代码书写风格统一；基本没有下列错误：由于软件缺陷造成丢失数据或者是软件崩溃，设计的功能没有最终实现，使用得出的结果出现明显的错误，响应时间太长无法接受。

b. 文档验收：

小组内评审；期末总体评审；文档格式和内容符合HB6465标准；功能符合与客户的合同要求；清晰易读，没有语病与歧义。

c. 服务验收：

服务硬件达到文档说明的要求；人员能够通过技术考核。

**3.6最后交付期限**

项目开发计划：2016年10月12日

软件规格需求说明：2016年10月13日

软件结构设计说明：2016年10月16日

软件测试说明：2016年10月29日

软件用户手册：2016年11月1日

软件产品：2016年11月1日

**4所需工作概述**

1. 对所要开发系统、软件的需求和约束：该软件系统适用于学校；作为用户，在查询借阅等过程中，师生有别；每一个界面都应该提供该功能界面所应该显示的、足够的信息；数据库不能被随意地改动；人员验证机制必须完善。
2. 对项目文档编制的需求和约束：文档种类全面；内容详实简洁。
3. 所选用的计划/采购策略：购入或租用合适的数据库系统、开发环境；使用符合性能要求的设备。
4. 项目进度安排及资源的需求和约束：无故不得有违开发所计划所做出的日程安排；每日的安排若无不可抗拒的原因必须严格完成；项目开发所涉及的硬件设备以及其它财产非专人不得擅用。

**5实施整个软件开发活动的计划**

本章分为软件开发过程，软件开发总体计划

**5.1软件开发过程**

软件开发过程可分解为需求分析，系统设计，编码，软件测试，用户文档编写，项目部署这六个阶段。

**5.2软件开发总体计划**

**5.2.1软件开发方法**

我们主要使用面向对象的软件开发方法，这是一种自底向上和自顶向下相结合的方法，而且它以对象建模为基础，从而不仅考虑了输入、输出数据结构，实际上也包含了所有对象的数据结构。我们预计使用PHPStorm作为开发工具，使用git作为项目管理工具，bugzilla作为Bug管理系统。在未来项目规模特别大，对性能的要求又比较高的时候可以考虑使用函数式编程，使用hadoop作为数据库管理系统。

从图书借阅管理系统的需求入手，首先确定对象模型，包括类的属性，与子类、父类的继承关系，以及类之间的关联进行自底向上的归纳过程。在自底向上的归纳过程中，为使子类能更合理地继承父类的属性和行为，可能需要自顶向下的修改，从而使整个类体系更加合理。由于这种类体系的构造是从具体到抽象，再从抽象到具体。

**5.2.2软件产品标准**

**5.2.2.1 格式标准**

每一行的代码必须使用两个空格缩进，代码必须缩进整齐，即不同作用域的缩进应该不同，大括号必须要换行。

**5.2.2.2 部首注释标准**

测试时每一行代码的修改都需要添加注释，说明修改的历史，修改的结果。对于功能较为复杂的代码需要在旁边说明功能，使用的算法，数据结构。所有的方法都要写明输入，输出，副作用，方法的规格一定要说明清楚。注释的总字节数应该大于编码的总字节数，注释中不得出现和代码无关的内容。

**5.2.2.3命名规范**

项目中所有的变量命名都必须使用驼峰命名法，严禁使用匈牙利命名法等其它命名规范。变量的命名不得少于5个字符，必须保证变量含义的可读性，易记性。对于函数，文件的命名也同样适用。

**5.2.3可重用的软件产品**

本条分为以下若干条。

**5.2.3.1吸纳可重用的软件产品**

本次项目在未经允许的情况下不得任何使用非开源的软件包。对于开源的软件，如果需要大规模吸纳重用，需要经过严格的评估方可使用。

**5.2.3.2开发可重用的软件产品**

对于数据库底层的操作，可以开发一套框架，专门用于图书借阅系统的开发使用，以便在未来方便项目的升级以及扩展。对于前端，可以基于bootstrap开发一套适用于图书借阅系统的前端框架，方便以后项目界面的升级以及同类系统的开发。

**5.2.4处理关键性需求**

本条应分以下若干条描述为处理指定关键性需求应遵循的方法。描述应覆盖合同中论及它的所有条款。

**5.2.4.1安全性保证**

评价软件安全性需求开发质量的三个常用指标是正确性、完整性和一致性。对于这三个指标，尤其需要保证数据库查询语句的正确实现以及在数据量较大，使用者流量巨大的情况下系统的正确运行。对于外界的恶意攻击，首先要保证的是反SQL注入，没有反SQL注入功能的数据库管理系统是极其危险的，用户的数据将会遭受到巨大的威胁。除此之外， CRSF攻击，XSS攻击也是需要防范的。

**5.2.4.2保密性保证**

数据库的连接权限必须严格管控，不应该存在多个可以访问整个数据库的超级用户。尽可能减少与外部互联网的连接。

**5.2.4.3私密性保证**

用户的个人信息需要进行额外的加密，以免遭到泄露。所有账户的权限都应该受到严格的限制。

**5.2.5计算机硬件资源利用**

预计将搭建一个由3台服务器组成的服务器集群，由数据库管理人员监控其运行，3台服务器中的数据都要进行容灾备份。

**5.2.7需方评审途径**

11月2日需方开始评审软件，线上和线下的评审可以同时进行，线上的评审包括系统的在线使用，线下的评审包括需方的系统管理人员的后台试用。

**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需求分析**

需求分析是整个设计中最重要的一环。

首先需要明确产品定位:

产品定位是产品设计的方向，也是需求文档和设计产出的判断标准。此外，产品定位也是团队成员形成统一的目标和对产品的认识，提高团队的凝聚力和工作效率，可以这么说，产品定位是需求中的需求。

其次就是明确用户需求:

用户需求就是在具体场景中，目标用户的目标事件。根据一定方法（头脑风暴、调研访谈等），可以得出用户需求示意列表。值得注意的是，用户类型不是绝对的互斥和独立，更多是以主要取向为区分标准，这不影响后面的筛选和匹配。

其中目标用户是最为关键的，明确目标人群可以使你更专注于服务某一类特定人群，这样更容易提升这类人群的满意度，也更容易使产品做到差异化和成功。

最后就是确定需求来源:

确定产品定位之后，然后通过不同的方式来收集大量的需求，然后根据这些需求的有效性和真实性、产品定位和项目资源情况进行筛选和匹配，提炼出产品需求，定义出优先级。从产品定位到需求优先级，整个过程不仅涉及对用户的分析和理解，还包括了对产品定位、项目资源的考虑。

以上分析完成之后需要得出一份需求文档，其内容包括:

① 文档备案：包括文档日期、版本号、修改人、修改内容和审核人等信息，一般以表格形式位于文档开头。

② 目录：方便阅读

③ 背景描述：为什么要做这个产品/模块，市场行情，业务目标，产品定位等

④ 用户类型：简单地描述目标用户的情况

⑤ 项目时间安排：启动、结束等时间节点

⑥ 信息结构：简单理解为内容和页面的层级

⑦ 业务流程说明：以流程图形式说明业务各个状态间的切换逻辑（例如：游戏服务器满人时候需要切换到排队登录状态）

⑧ 需求详细说明：每一条需求的详细说明（包括：使用场景、UI描述、功能描述、优先级、输入/输出条件、处理流程、补充说明等）

⑨ 涉及关联业务部门的支持，还需要特别备忘。

需求文档的编写从2016年10月9号开始，2016年10月12号完成整个文档的编写。

**6.1.2系统设计**

从2016年10月12号开始到2016年10月15号，完成对系统的概念设计，逻辑设计，物理设计，验证设计，存取权限等进行了定义，对系统功能各模块进行了详细设计，定义了数据库总体结构、编码命名规范。

1) 概念设计：

对用户要求的数据库系统，通过对其中诸处的分类、聚集和概括，建立抽象的概念数据模型。这个概念模型应反映现实世界各部门的信息结构、信息流动情况、信息间的互相制约关系以及各部门对信息储存、查询和加工的要求等。所建立的模型应避开数据库在计算机上的具体实现细节，用一种抽象的形式表示出来。以扩充的实体—（E-R模型）联系模型方法为例，第一步先明确现实世界各部门所含的各种实体及其属性、实体间的联系以及对信息的制约条件等，从而给出各部门内所用信息的局部描述(在数据库中称为用户的局部视图)。第二步再将前面得到的多个用户的局部视图集成为一个全局视图，即用户要描述的现实世界的概念数据模型。

2) 逻辑设计

主要工作是将现实世界的概念数据模型设计成数据库的一种逻辑模式，即适应于某种特定数据库管理系统所支持的逻辑数据模式。与此同时，可能还需为各种数据处理应用领域产生相应的逻辑子模式。这一步设计的结果就是所谓“逻辑数据库”。

3) 物理设计

根据特定数据库管理系统所提供的多种存储结构和存取方法等依赖于具体计算机结构的各项物理设计措施，对具体的应用任务选定最合适的物理存储结构(包括文件类型、索引结构和数据的存放次序与位逻辑等)、存取方法和存取路径等。这一步设计的结果就是所谓“物理数据库”。

4) 验证设计

在上述设计的基础上，收集数据并具体建立一个数据库，运行一些典型的应用任务来验证数据库设计的正确性和合理性。一般，一个大型数据库的设计过程往往需要经过多次循环反复。当设计的某步发现问题时，可能就需要返回到前面去进行修改。因此，在做上述数据库设计时就应考虑到今后修改设计的可能性和方便性。

**6.1.3编码**

1) 编码风格：

即书写源程序的习惯、程序代码的逻辑结构与习惯的编程技术。一个比较大的软件在编写的时候是合作完成的。那么许多人在编写的时候不能自己用自己的风格，他们必须要有一个统一的标准。这就要求编写时代码有良好的可读性。良好的代码可读性是合作的前提，也是后期软件的维护的前提。

2) 编码原则：

高效率原则包括提高运行效率，提高储存效率，提高输入输出效率。所有I/O操作应该采用缓冲方式，以减少用于通信的额外开销。

从2016年10月15号开始到2016年10月25号完成程序设计，需要首先写好编码规范手册，统一变量命名规范。丙负责前端开发，丁和戊负责后端，前后端人员需要及时沟通，保证前后端的良好对接。

**6.1.4软件测试**

单元测试：

单元测试又称模块测试，是针对软件设计的最小单位 ─ 程序模块，进行正确性检验的测试工作。其目的在于发现各模块内部可能存在的各种差错。

单元测试需要从程序的内部结构出发设计测试用例。多个模块可以平行地独立进行单元测试。

集成测试：

集成测试也称联合测试，将程序模块采用适当的集成策略组装起来，对系统的接口及集成后的功能进行正确性检测的测试工作。其主要目的是检查软件单位之间的接口是否正确，集成测试的对象是已经经过单元测试的模块。

系统测试：

主要包括功能测试、界面测试、可靠性测试、易用性测试、性能测试。 功能测试主要针对包括功能可用性、功能实现程度（功能流程&业务流程、数据处理&业务数据处理）方面测试。

回归测试：

回归测试指在软件维护阶段，为了检测代码修改而引入的错误所进行的测试活动。回归测试是软件维护阶段的重要工作，有研究表明，回归测试带来的耗费占软件生命周期的1/3总费用以上。

从2016年10月26号开始至2016年10月29号。由丙，丁，戊三人独自完成单元测试之后，再一同完成集成测试，系统测试，回归测试。

**6.1.5用户文档编写**

从2016年10月30号开始至2016年10月31号，完成各类文档的编写，系统准备投入运行阶段。

**6.1.6 项目部署**

2016年11月1日，由戊进行项目交接，为接收方部署项目。

**6.1.7项目总结**

项目结束后用一周左右时间，对项目研发、部署等开发过程中的问题、经验教训总结备案，以利于项目经验的积累和开发进度的缩短。

**7 进度表及活动网络图**

**（注：红色字体为重点事件，有严格时间限制）**

**7.1进度表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 任务名称 | 工作时间 | 开始时间 | 结束时间 |
| 1 | 开发前动员，以及项目介绍、大致流程、时间安排的宣讲 | 1工作日 | 2016年10月9日 | 2016年10月9日 |
| 2 | 查找有关开发需求的资料 | 0．5工作日 | 2016年10月10日 | 2016年10月10日 |
| 3 | 通过调研得出用户需求资料，得到相关数据 | 1工作日 | 2016年10月10日 | 2016年10月11日 |
| 4 | 根据数据，讨论分析用户需求，提炼出产品需求 | 0.5工作日 | 2016年10月11日 | 2016年10月11日 |
| 5 | 整理并书写需求分析，得到需求文档 | 1工作日 | 2016年10月12日 | 2016年10月12日 |
| 6 | 业务流程分析、数据流程分析 | 1 工作日 | 2016年10月13日 | 2016年10月13日 |
| 7 | 概要设计文档 | 1工作日 | 2016年10月14日 | 2016年10月14日 |
| 8 | 软件构架文档 | 1工作日 | 2016年10月15日 | 2016年10月15日 |
| 9 | 详细设计文档 | 1工作日 | 2016年10月16日 | 2016年10月16日 |
| 10 | 数据库设计及分析 | 1 工作日 | 2016年10月17日 | 2016年10月17日 |
| 11 | 代码编程规范 | 1 工作日 | 2016年10月18日 | 2016年10月18日 |
| 12 | 编码 | 6 工作日 | 2016年10月19日 | 2016年10月25日 |
| 13 | 单元测试 | 1 工作日 | 2016年10月25日 | 2016年10月25日 |
| 14 | 集成测试 | 0.5工作日 | 2016年10月26日 | 2016年10月26日 |
| 15 | 系统测试 | 0.5工作日 | 2016年10月26日 | 2016年10月26日 |
| 16 | 回归测试 | 2工作日 | 2016年10月27日 | 2016年10月28日 |
| 17 | 用户文档编写 | 1 工作日 | 2016年10月29日 | 2016年10月29日 |
| 18 | 项目部署、项目交接 | 3 工作日 | 2016年10月30日 | 2016年11月1日 |
| 19 | 项目总结 | 2 工作日 | 2016年11月2日 | 2016年11月4日 |

**7.2活动图**

部署

代码编程规范

数据库设计及分析

测试

编码

详细设计文档

业务流程分析

数据流程分析

撰写需求文档

讨论需求

查询用户需求资料

调研需求

开发前动员

**8项目组织和资源**

**8.1人力资源：**

此项目预计投入人员5人次，预计投入时间5\*21天，其中：

管理：1人

软件编码：3人

软件配置管理：1人

产品评估：1人

软件文档编制： 1人

软件测试：3人

**8.2项目组织：采取封闭式范型**

高级工程师（主程序员）： 1人，负责计划、协调和评审团队所有技术活动。

技术人员：2人，负责分析和开发活动。

后备工程师：1人，参与开发活动，但需要支持协助高级工程师的活动，在项目进行过程中可以以最小代价接替高级工程师的活动。

软件资料员：1人，负责软件文档撰写及资料获取。

**9培训**

**9.1项目的技术要求**

系统面向的人员有教师、学生以及系统管理员。对于这些我们需要提供一个平台即一个图书馆网站或者客户具有用户注册、信息查询、书目编排录入、图书借还功能等。我们计划的平台是一个网站。

管理人员需要熟悉数据库的操作，熟悉图书管理。

而对开发人员的要求则是熟悉网站的开发，首先有熟悉前端开发的人员，通晓HTML\CSS以及javascript，其次则有熟悉后端的人员，精通数据库的相关操作并熟悉php、java或者python等后端脚本语言。

**9.2培训计划**

对于目前开发人员的技术水平还是有一定培训的需求的。

对于管理人员的培训，则计划在系统开发完成之后培训管理人员对程序的相关操作能力。

对于开发人员，则在项目开始阶段进行网站开发培训，并针对具体的人员进行前端的培训以及后端的培训。

**10项目估算**

**10.1规模估算**

对于项目规模的估算，

外部输入有用户有用户注册，用户登录，管理员有书籍编排录入。

外部查询有图书信息查询，图书借还及预借，个人信息查询，管理员则有查询用户信息。

内部逻辑文件有用户信息数据库文件，图书信息数据库文件以及系统配置文件。

外部接口文件则需要一个提供给外部访问的网络接口文件。

每一个估算他们的加权因子，外部输入标记为简单，加权因子设为3，外部查询图书信息查询个人信息查询查询用户信息标记为简单，加权因子设为2，图书借还及预借标记为中等，加权因子设为5，内部逻辑文件标记为简单，加权因子设为3，外部接口文件标记为中等，加权因子设为5。

然后计算总量：3\*3 + 3\*2 + 1\*5 + 3\*3 + 1\*5 = 31

假设每一份权重100行代码，则总计代码量为3100行。

**10.2工作量估算**

项目准备： 5人\*日

功能及需求分析：2人\*日

总体设计：5人\*日

详细设计：15人\*日

程序开发之前端界面编写及美化：10人\*日

数据库设计以及架构：15人\*日

后端代码逻辑操作：30人\*日

系统测试：15人\*日

试运行：15人\*日

**10.3成本估算**

首先人力成本，按照100元/人\*日计算，总共为100\*（5+10+5+15+10+15+30+15+15）= 12000元，然后租借服务器等估计为100元。

**10.4关键计算机资源估算**

服务器，存放网页与数据库等信息。

**11风险管理**

**11.1 风险管理计划编制**

在项目初期，邀请所有的开发人员参加风险管理计划会议，讨论项目章程、建设目标、项目范围、工程投资、工程进度以及项目面临的风险和应用措施。根据会议讨论的结论，再结合风险管理策略，编制本项目的风险管理计划。

**11.2 风险识别**

由于项目本身的独创性，决定了其风险的多样性和复杂性。在编制图书查询借阅管理系统项目风险管理计划时，针对该项目可能会出现的风险进行讨论，经过分析筛选，最后确定了该项目的重点风险归纳为以下几类：

技术风险。对系统架构的不熟悉；对项目的所使用的开发工具不熟悉；技术人员没有正确理解项目的业务；客户自身发展过程中业务流程不断的发生变化。

团队风险。关键技术人员中途离职；技术人员和管理层之间关系不佳；项目组成员之间发生冲突，导致沟通不畅。

外部风险。上层或行业政策发生调整；资金产生临时性空缺。

**11.3 定性风险分析**

根据风险管理计划中的定义和项目业务逻辑的特点，确定每一个风险的发生可能性，并记录。首先分析风险发生的可能性，其次则是分析风险对项目的影响，包括对时间、成本、范围等各方面的影响。

**11.4 定量风险分析**

对已知风险进行定性分析后，就需要转化为定量分析，分析各风险对项目目标的影响。定量风险分析需要从以下三个角度进行估计：乐观、中性和悲观。同时，利用历史项目数据来辅助评估。

**11.5 风险应对措施**

1. 加强项目沟通管理。

该项目的风险很多是由于开发和使用方在理解上存在偏差、沟通不充分造成的。因此，在向系统使用方做需求调研时，技术人员应与使用方进行反复、深入的交流，了解系统使用方的真正意图。同时，区分不同层次人员的需求特点，特别是决策层的意图犹为重要，还可以建立系统模型来展示未来的目标系统让使用方有一个更直观的了解。

1. 选择合适的开发技术。

在开发这个图书查询借阅管理系统时，项目组应当采用的是较为熟悉、有类似项目成功案例可以参考的技术。同时，在项目初期，对技术人员进行培训，使他们充分了解该项目的整体业务流程并且对开发工具的熟悉。

1. 加强项目人力资源管理。

首先，围绕这个项目的计划，组建项目团队，配备高素质高水平的项目管理人员和技术人员。其次，加强项目团队建设，促进项目组成员关系团结和谐，并能灵活使用软件开发工具和技术。第三，制定一套有效可行的激励机制，对在这个项目开发过程中表现突出的技术人员实行物质奖励，宣传实现该项目的价值。

**12支持条件**

12.1计算机系统支持。

计算机系统应当配备相应的开发软件，开发环境。

12.2需要需方承担的工作和提供的条件。

需方的工作：需要和技术人员做好项目细节的沟通，并随时反馈。

提供的条件：将项目目标，细节等开发目的供给技术人员。

12.3需要分包商承担的工作和提供的条件。

分包商的工作：从项目管理人员处取得开发计划并进一步部署下一步计划。

分包商的条件：随时向项目管理人员汇报项目的开发进展并等待反馈。

**13注解**

本文档使用的原理：

1. 用分阶段的生命周期计划严格管理
2. 坚持进行阶段评审
3. 实行严格的产品控制
4. 采用现代程序设计技术
5. 结果应能清楚的审查
6. 开发小组的人员应该少而精