基于 MVC 设计模式的在线图书馆管理系统软件开发计划

Online Library Management System Based On MVC Design Pattern Software Project Planning

2018-12-18 发布

目录

1 范围

1.1 标识

本文档标识号：OLMS01/SPP

本文档名称：Online Library Management System Based On MVC Design Pattern Software Project Planning

缩略名：OLMS

版本号：1.0.0, 1.1.0

发布号：20181219100

1.2 系统概述

基于 MVC 设计模式的在线图书馆管理系统(Online Library Management System Based On MVC Design Pattern)适用于各版本的 Windows 系统，本软件用于对图书馆进行管理，大致包括：图书管理员管理，读者管理和书籍管理三个方面。具有一定的并发性，支持多人同时进行操作，功能较为完备，系统可用性、可靠性高，易于维护，具有较高的效率。软件配置管理时应参照《软件配置管理计划》，对于软件质量保证，需参照《软件质量保证计划》。

1.3 文档概述

本文档对基于 MVC 设计模式的在线图书馆管理系统(Online Library Management System Based On MVC Design Pattern)中的策划背景、软件开发活动的总体实施、实施计划等做了详细介绍。

使用中，管理员接口应对读者用户保密。

1.4 与其他计划之间的关系

软件配置管理时应参照《软件配置管理计划》，对于软件质量保证，需参照《软件质量保证计划》。

2 引用文档

文档格式要求按照我国 GJB/438B-2009 国家标准和 IEEE/ANSI830-1993 标准规范要求进行。包括以下文件：

《软件工程项目开发文档范例》

《软件工程国家标准文档》

《软件需求说明书编写规范》

3 策划背景概述

a) 需在 Windows 环境下运行该系统、管理员需要一定的操作基础。

b) 标题中所有中文字体采用宋体，西文字体采用 Times New Roman，其中标题用小四加黑。正文中所有中文字体采用宋体，西文字体采用 Times New Roman，字体为小四字号，采用 1.25 倍行距。表格与图片命名的格式为中文字体采用宋体，西文字体采用 Times New Roman，字体为五号字体。其中表格的外边框笔画粗细为 1.5 磅，表格内部边框笔画粗细为 0.5 磅，表格宽度为 14.50cm。

c) 项目处于系统生命周期中立项、开发、运维的位置。

d) 见 6.进度表和活动网络图。

e) 对图书馆计算机硬件有所要求。

4 软件开发活动的总体实施计划

4.1 软件开发过程

本系统采用自上而下，相互衔接的固定的次序进行软件开发。首先根据客户的需求得到调研报告，进行需求分析与定义，编制系统规格说明；然后根据可行性分析报告和项目开发计划及软件需求规格说明书，进行系统与软件设计，建立整个系统的体系结构；在实现和单元测试阶段，编写程序代码和进行单元测试；然后进行集成与系统测试，将程序按一定顺序集成起来，做成一个完整的系统进行测试；最后在运行和维护阶段，将系统投入使用，根据新的系统需求改进系统单元。最终实现整个系统的开发和设计过程。

4.2 软件开发总体计划

4.2.1 软件开发方法

对于本项目，我们采用面向对象的软件开发方法。面向对象的软件开发方法需要用到以数据流图为核心，还使用 UML(Unified Modeling Language，统一建模语言)进行建模，还使用用例图，类/对象图，对象关系图，实体-关系图(E-R)，时序图和状态转换图(STD)进行分析建模。

4.2.2 软件产品标准

GB/T 13702-1992 计算机软件分类与代码

GB/T 20918-2007 信息技术 软件生存周期过程 风险管理

GB/T 19003-2008 软件工程 GB/T19001-2000

GB/T 15538-1995 软件工程标准分类法

GB/T 9386-2008 计算机软件测试文档编制规范

GB/T 9385-2008 计算机软件需求规格说明规范

GB/T 15532-2008 计算机软件测试规范

GB/T 18221-2000 信息技术 程序设计语言 环境与系统软件接口 独立于语

言的数据类型

GB/T 11457-2006 信息技术 软件工程术语

GB 8567-2006 计算机软件文档编制规范

4.2.3 可重用的软件产品

4.2.3.1 采用可重用软件产品

1)基于软件复用库的软件重用：

它是一种传统的软件重用技术。这类软件开发方法要求提供软件可重用成份的模式分类和检索，且要解决如何有效地组织、标识、描述和引用这些软件成份。

2)与面向对象技术结合：

OO 技术中类的聚集、实例对类的成员函数或操作的引用、子类对父类的继承等使软件的可重用性有了较大的提高。而且这种类型的重用容易实现。所以这种方式的软件重用发展较快。

3)组件连接：

这是目前发展最快的软件重用方式。

4.2.3.2 开发可重用软件产品

软件重用是软件界追求的目标,人们正在努力将基于框架、体系结构和需求的重用变为现实,但是在具体实现技术上还不成熟。为了促进软件的重用,本项目引入了结构化、可重用的软件模式来捕捉并描述成熟的软件知识和经验。利用面向对象的软件开发方法，使用面向对象分析的建模工具进行建模，从而实现开发可重用的软件产品。

4.2.4 关键需求的处理

安全性保证：项目源程序，开发过程中涉及到的相关文档都要保证安全性；开发的项目在安装到平台上后，要保证系统的安全性，要安装相关的防火墙；用户的信息要保证安全；合同的签订应该保证足够的安全，合法，要有第三方进行公证。保密性保证：项目源程序，开发过程中涉及到的相关文档一定要注意保密，防止他人盗用；合同中规定的内容，除了合同签订的双方，其他任何人或组织，都不能够查看。

私密性保证：对于用户信息的一些私密性的东西一定要注意保证安全；对于合同中一些私密性的东西，要主要保护。

其他关键性需求保证：方法、标准、硬件开发和软件开发的相互依赖关系等一定要保证。

4.2.5 计算机硬件资源的利用

服务器：

(1) 处理器(CPU)：Pentium II 300 或更高配置

(2) 内存容量(RAM)：128MB 以上

客户端：

(1) 处理器(CPU)：Pentium 200 或更高配置

(2) 内存容量(RAM)：64MB 以上

4.2.6 决策理由的记录

记录软件架构决策理由的文件，长期来看非常有用，因为架构不会经常变，所以也不用付出过多维护精力。

1 开发前期的文档

2 软件设计文档和代码保证同步

用处：

1.沟通工具

2.移交项目给别人

3.写文件也会迫使自己明确这样决策的理由，有助于确保基础是扎实稳固的。

4.当相关条件变化时，需要重新决策时，这份文档是一个不错的起点。

4.2.7 需方评审所需访问

需方通过开发者提交的程序和文档对项目进行评审，验证是否满足所提出的功能要求。

5 详细的软件开发活动实施计划

5.1 项目策划和监控

基于 MVC 设计模式的在线图书馆管理系统(Online Library Management System Based On MVC Design Pattern)适用于各版本的 Windows 系统，本软件用于对图书馆进行管理，大致包括：图书管理员管理，读者管理和书籍管理三个方面。该项目的主要用户是当地图书馆的工作人员以及在当地图书馆注册过的读者。

该项目的开发活动计划如下：

1) 项目开发资源描述

本项目将有 4 名开发人员与 1 名测试人员共同实现，对设备的要求如下：

服务器：

(1) 处理器(CPU)：Pentium II 300 或更高配置

(2) 内存容量(RAM)：128MB 以上

客户端：

(1) 处理器(CPU)：Pentium 200 或更高配置

(2) 内存容量(RAM)：64MB 以上

2) 开发方法和工具

本项目将使用 MVC 模式进行开发，开发工具为 Intellij IDEA，所用数据库为 MySQL

3) 各部门间的接口与协调及其他

测试先行，在进行编码之前由测试人员先设计好测试用例，将软件测试与软件开发并行进行。

测试计划如下：

1) 各阶段测试目标及要求

本项目将使用 H 模型，将测试与开发并行进行。在最终编码结束后，将进行系统的单元测试、集成测试、系统测试、验收测试等。单元测试要求各模块运行无 bug，不会出现不满足需求或者模块崩溃等情况。集成测试与系统测试将各模块拼接起来，查看是否满足用户需求。验收测试要求用户体验后对系统无更改意见。

2) 各测试阶段资源要求及时间安排

在编码阶段结束后进行单元测试，单元测试通过后进行集成测试，集成测试结束后进行系统测试，最终进行用户验收测试。

3) 测试记录的具体要求

每次测试结束后应记录测试人，测试时间，测试范围，若有 bug 也应对 bug 进行具体描述，以及期待结果。

软件安装策划如下：

在进行软件安装时，应先安装 Intellij IDEA 或 eclipse 等 Java 开发工具，需要安装 MySQL 数据库。之后在数据库中运行项目提供的 sql 文件，之后在 Java 开发工具中运行给定的代码。

计划将使用需求跟踪矩阵，确保各个需求被实现。

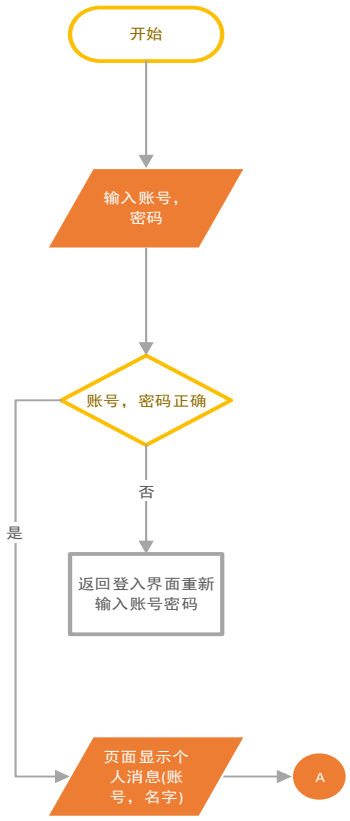
5.2 软件开发环境建立

本软件将在 JDK1.8 的环境下进行开发，将使用 JUnit 进行单元测试，使用 Selenium 进行系统测试与验收测试。

5.3 系统需求分析

5.3.1 Admin

(1) 登录，查询个人信息

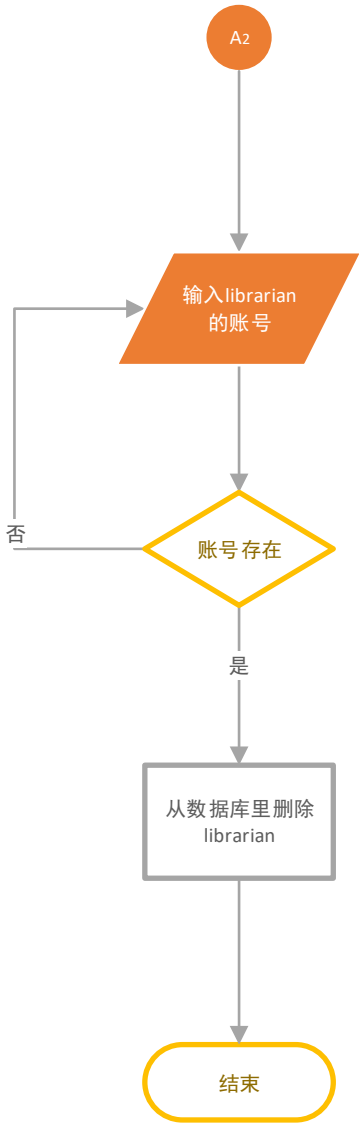


图一. Admin 登录功能演示

(2) 查询 Librarian 信息(包含锁定 Librarian 账号功能)

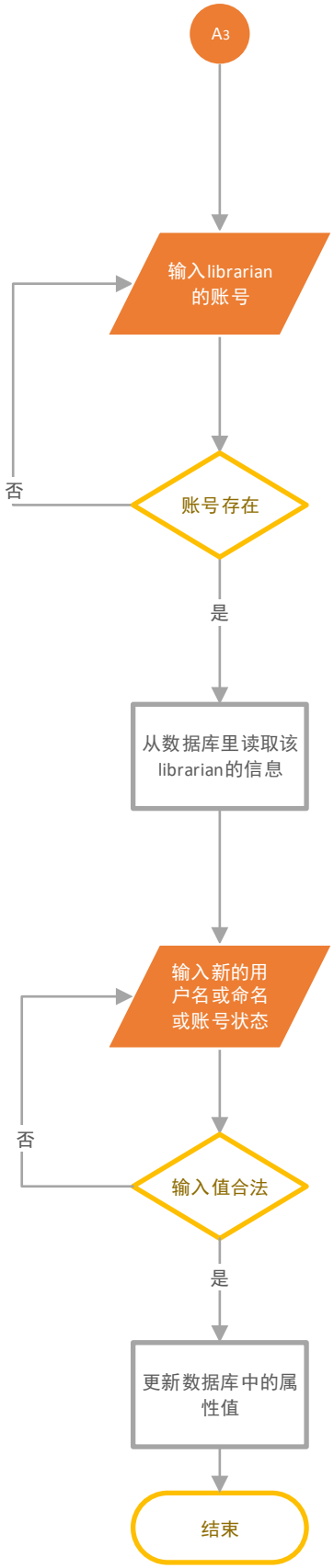
图二. Admin 查询 Librarian 信息功能演示

(3) 删除 Librarian 账号



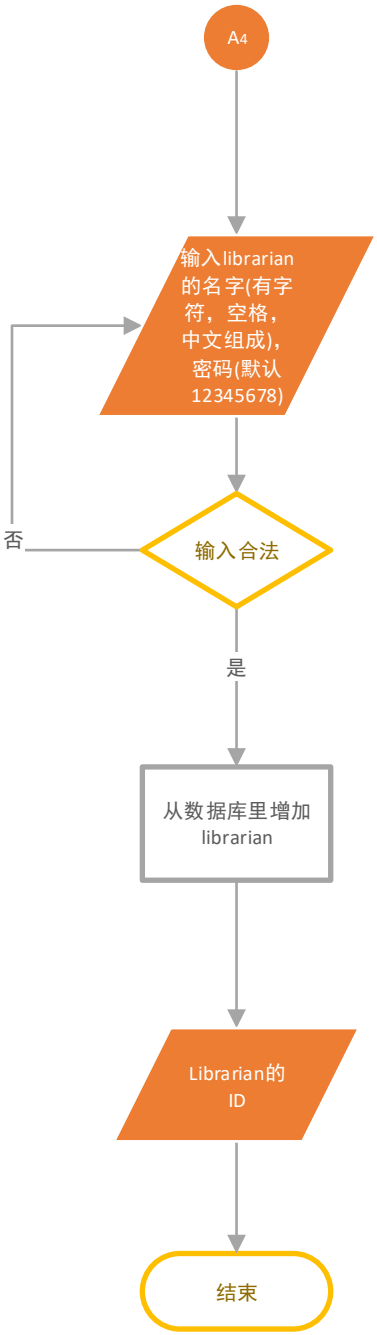
图三.　　Admin 删除 Librarian 账号功能演示

(4) 修改 Librarian 信息



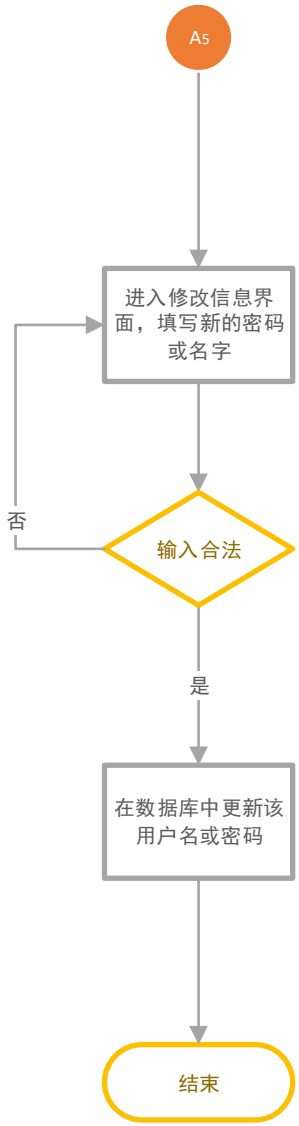
图四.　Admin 修改 Librarian 信息功能演示

(5) 注册 Librarian



图五.　　Admin 注册 Librarian 新账号功能演示

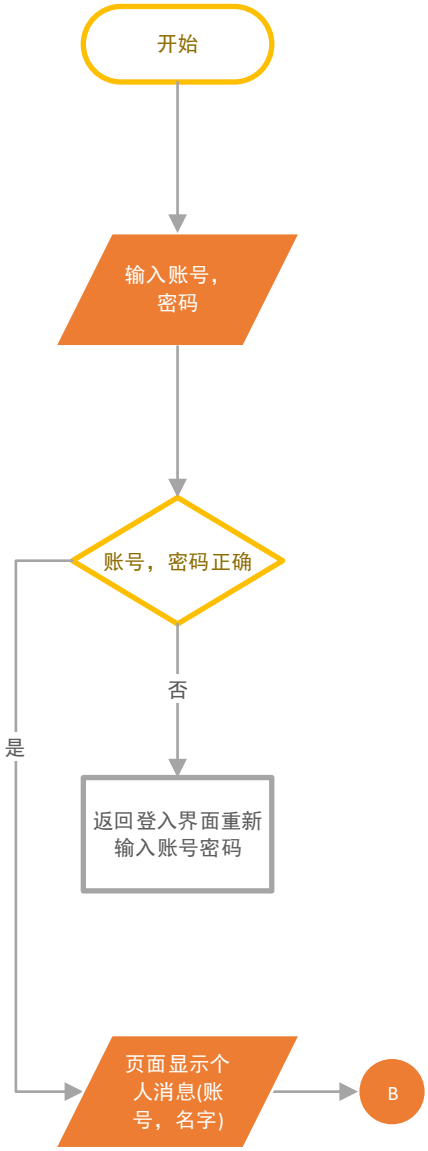
(6) 修改密码，名字



图六.　　Admin 修改自身信息功能演示

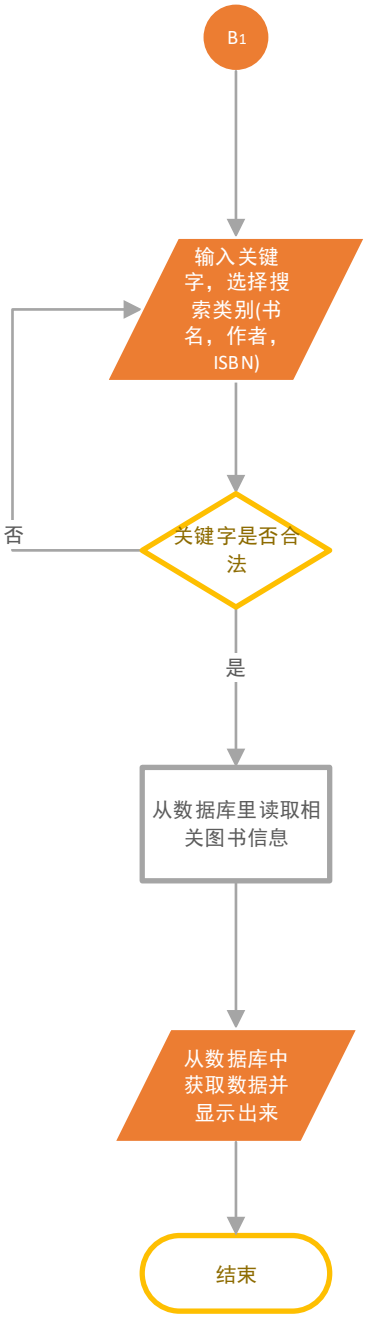
5.3.2 Reader

(1) 登录



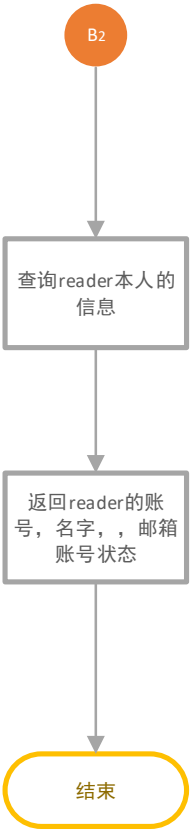
图七. Reader 登陆功能演示

(2) 搜索图书



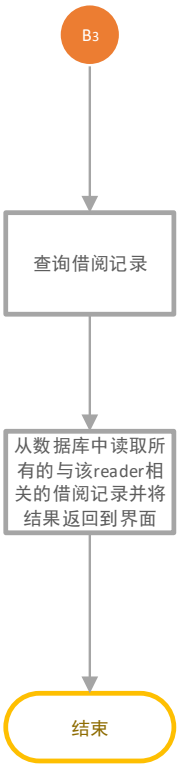
图八.　　Reader 搜索图书功能演示

(3) 查询个人信息



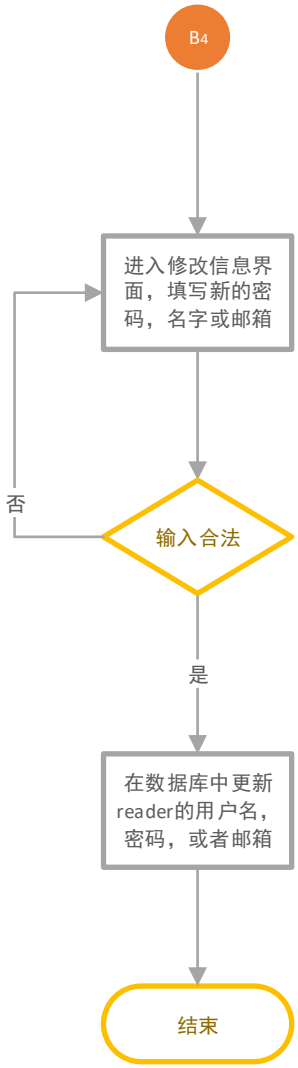
图九.　Reader 查询个人信息功能演示

(4) 借阅历史



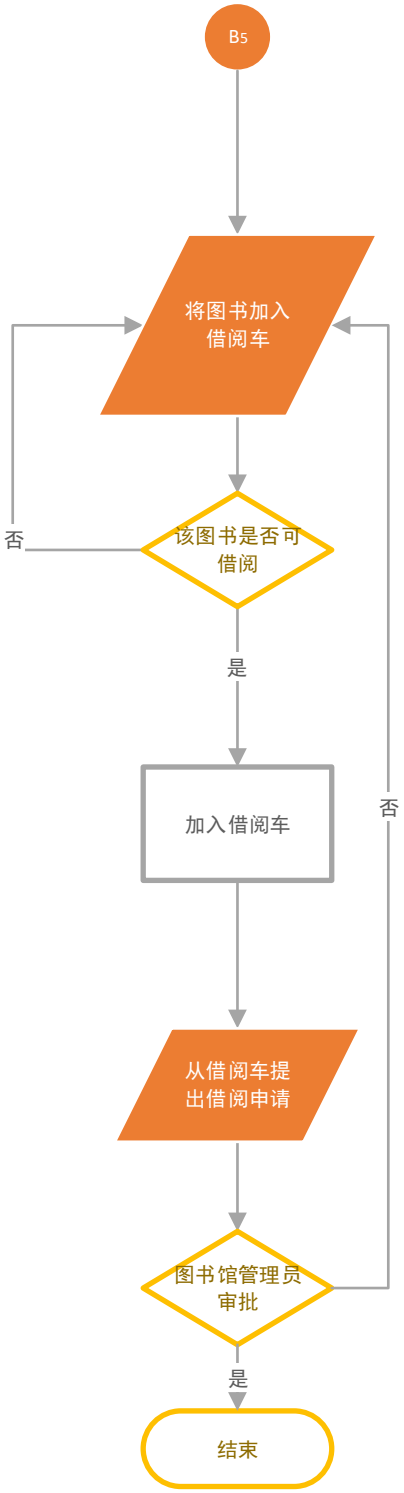
图十.　Reader 查询自身借阅历史功能演示

(5) 修改名字，密码，邮箱



图十一．Reader 修改个人信息功能演示

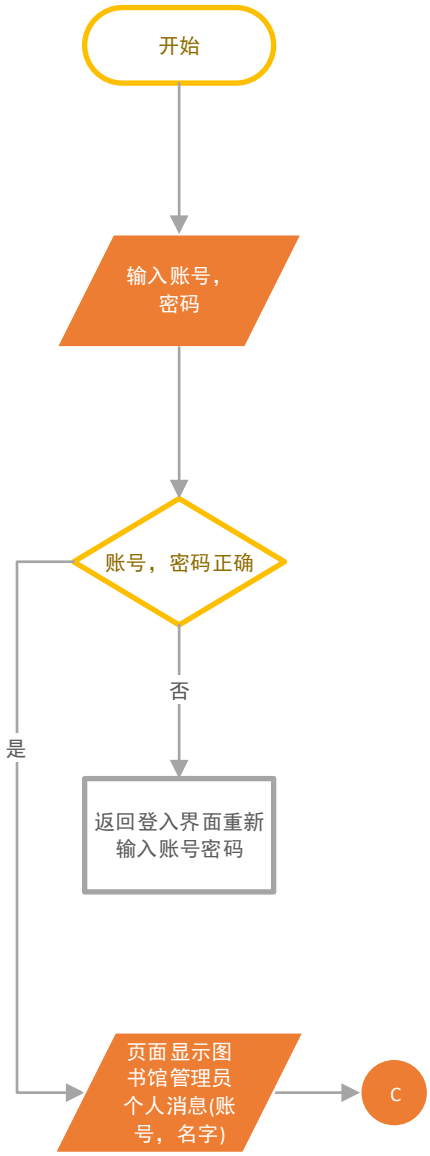
(6) 将可借阅的书添加进借阅车，从借阅车提出借阅申请



图十二. Reader 借阅图书功能演示

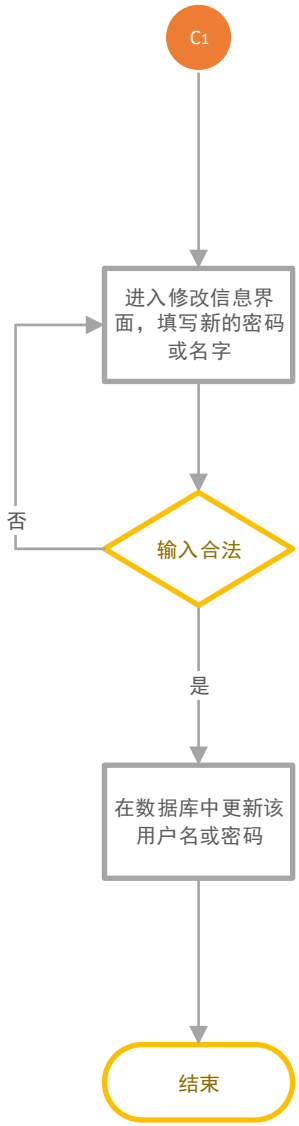
5.3.3 Librarian

(1) 图书管理员登录系统



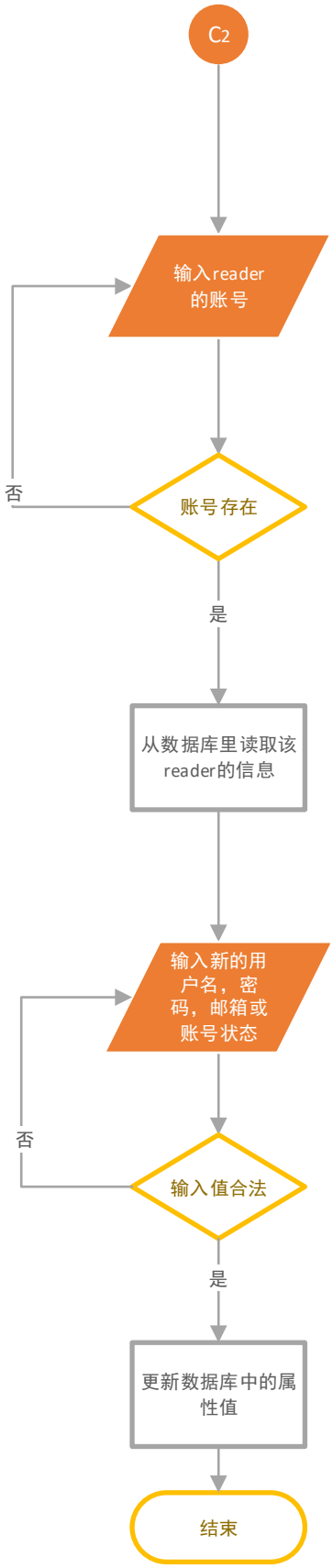
图十三. Librarian 登陆功能演示

(2) 修改密码，用户名



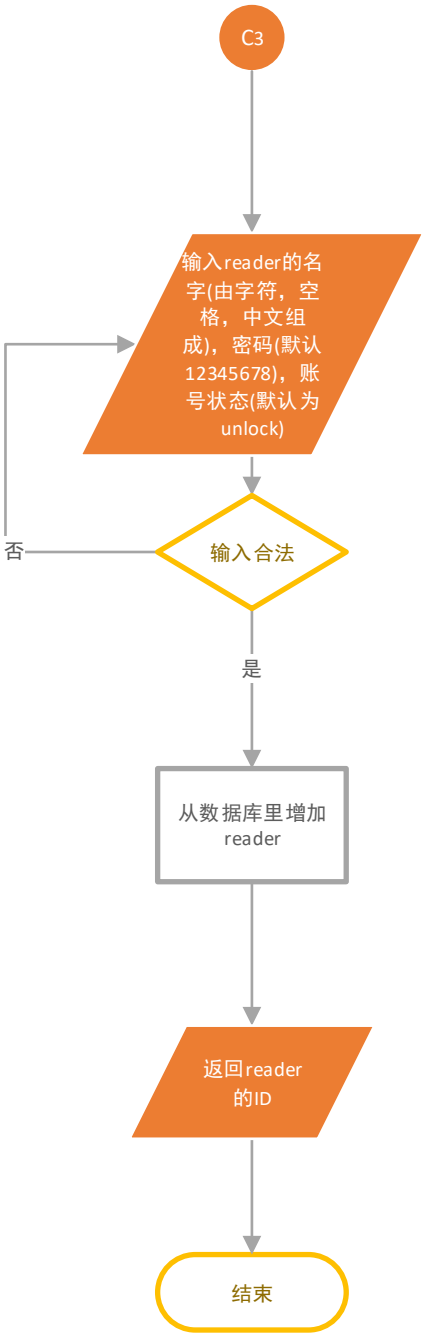
图十四. Librarian 修改个人信息功能演示

(3) 修改 Reader 信息(名字，密码，邮箱，账号状态)



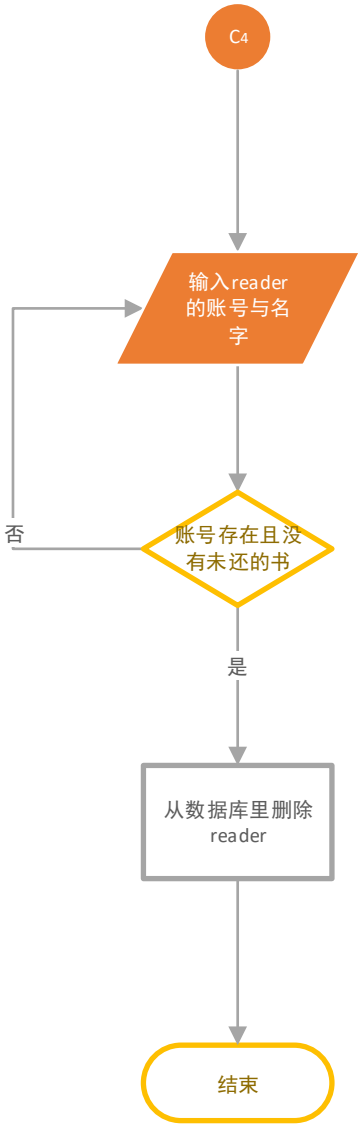
图十五. Librarian 修改 Reader 信息功能演示

(4) 增加一个读者



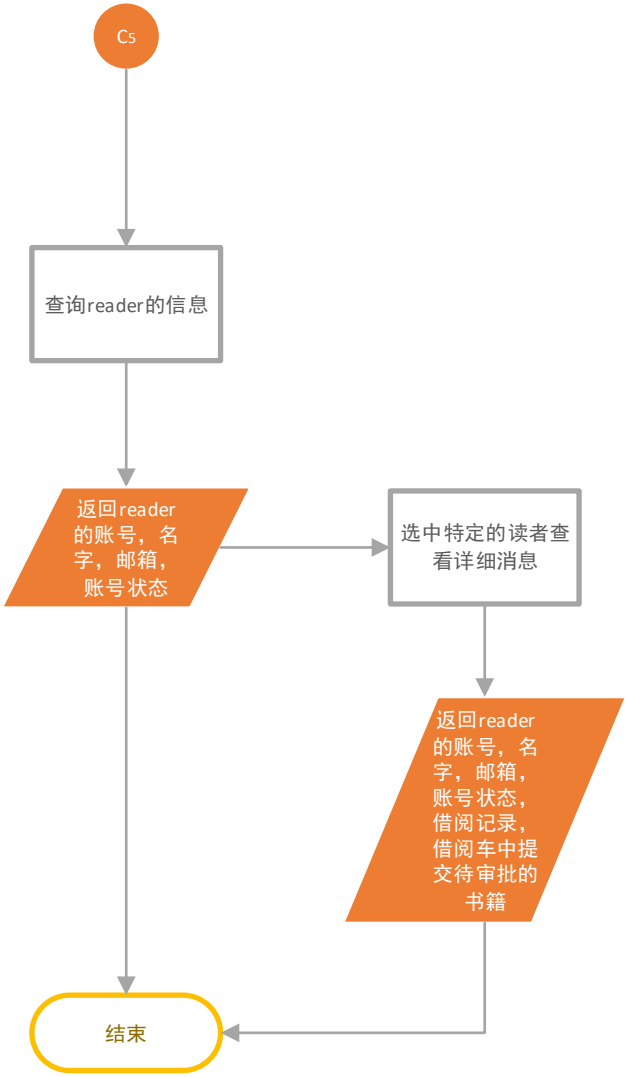
图十六. Librarian 添加新读者功能演示

(5) 删除一个读者



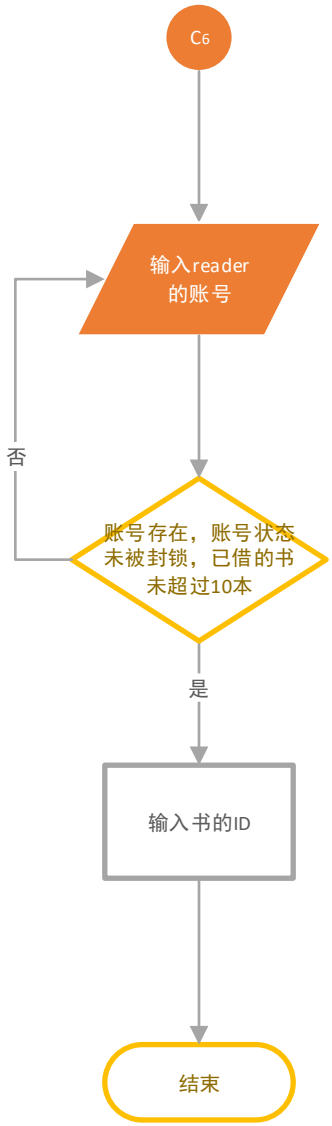
图十七. Librarian 删除 Reader 功能演示

(6) 查询读者信息



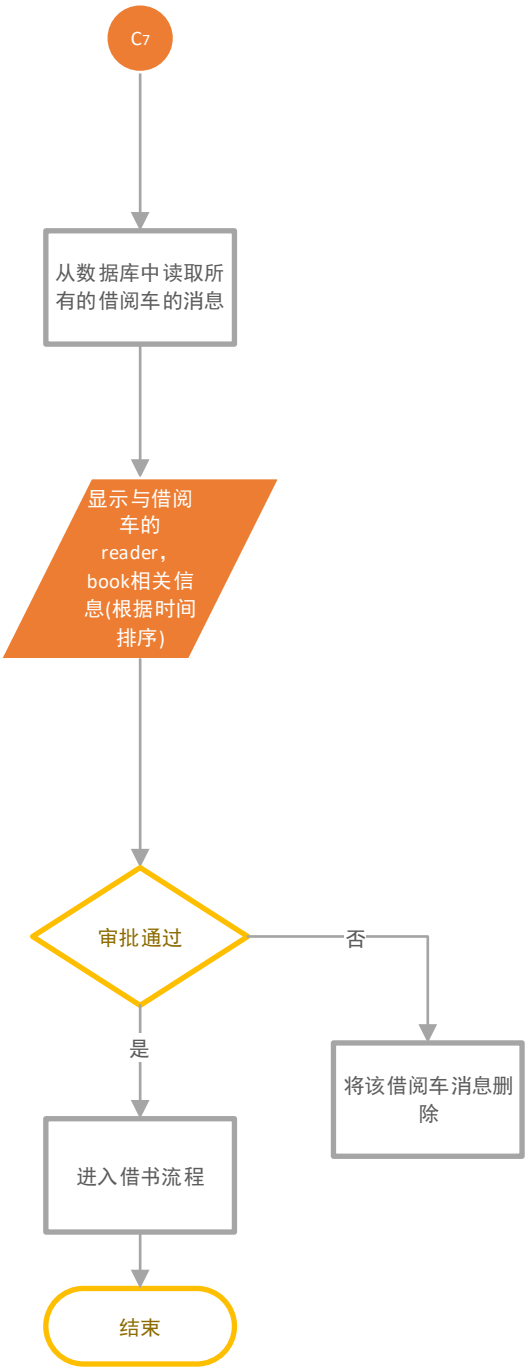
图十八. Librarian 查询 Reader 信息功能演示

(7) 借书操作



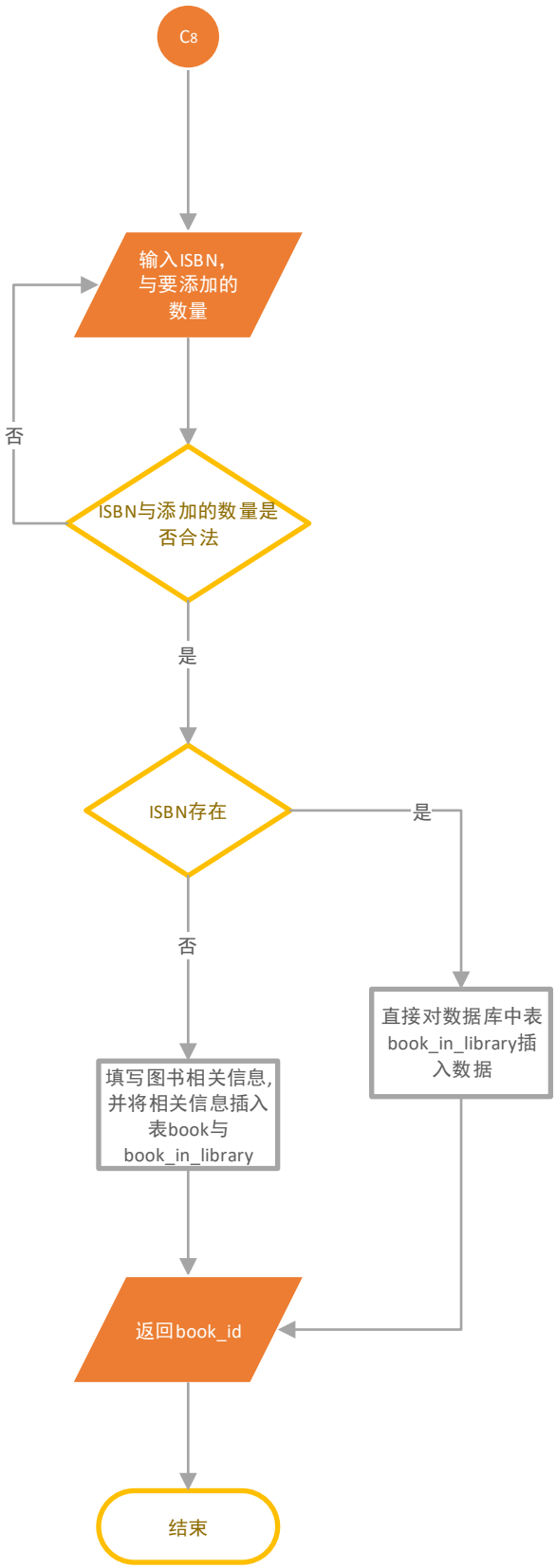
图十九. Librarian 帮助 Reader 借书操作演示

(8) 查看借阅车，审批借阅申请(以申请借阅次序)；



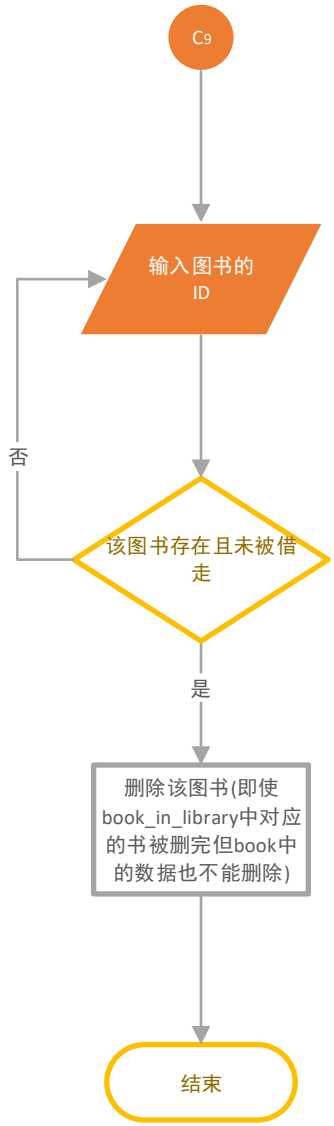
图二十. Librarian 处理 Reader 借阅图书申请操作演示

(9) 增加图书



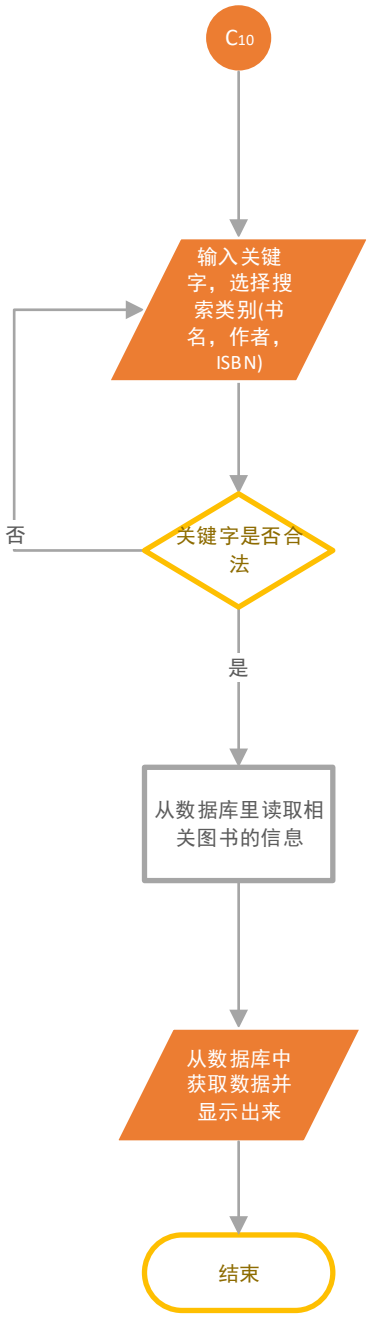
图二十一.　Librarian 添加图书操作演示

(10) 删除图书



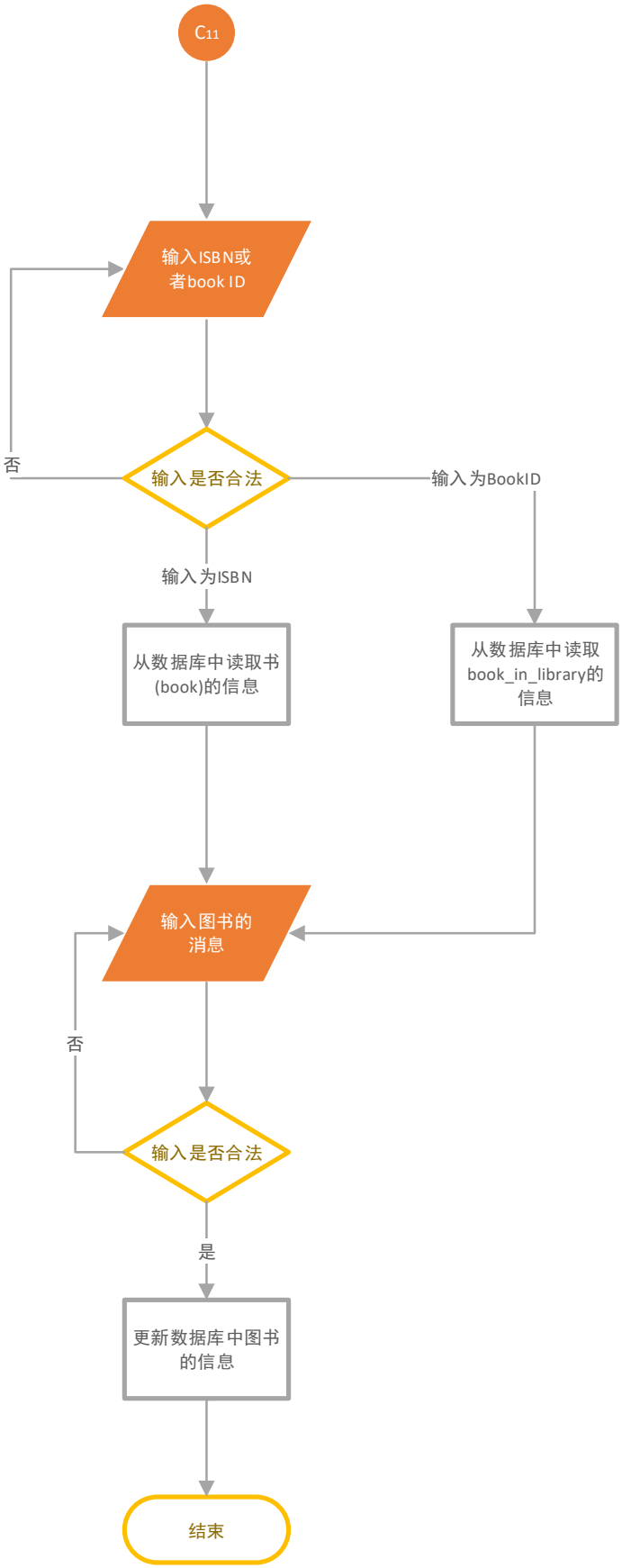
图二十二.　Librarian 删除图书操作演示

(11) 查询图书信息



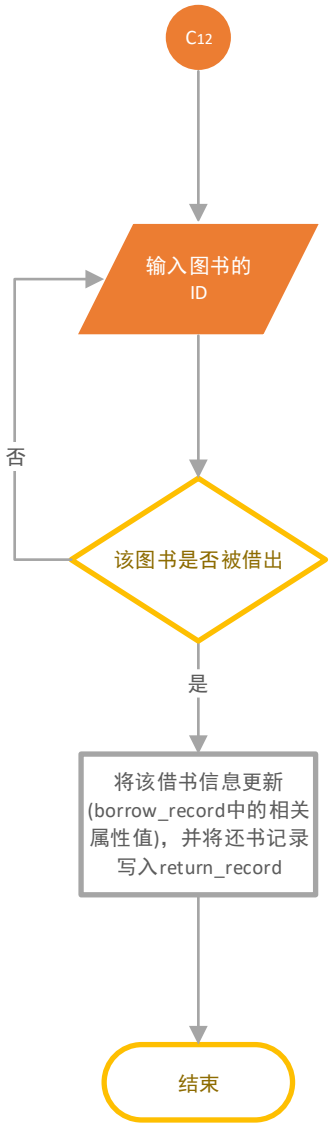
图二十三.　用户查询图书信息操作演示

(12) 修改图书信息



图二十四.　　Librarian 修改图书信息操作演示

(13) 还书操作



图二十五.　Librarian 处理还书请求操作演示

5.4 系统设计

本软件将使用 MVC 模式，将系统分为模型(model)－视图(view)－控制器(controller)三个部分，实现软件架构。

5.5 软件需求分析

同 5.3

5.6 软件设计

本软件用于对图书馆进行管理，大致包括：图书管理员管理，读者管理和书籍管理三个方面，因此将该项目的主要用户定义为当地图书馆的工作人员以及在当地图书馆注册过的读者。

本软件使用 MVC 设计模式，将系统分为模型(model)－视图(view)－控制器

(controller)三个部分，实现软件架构。

5.7 软件实现和单元测试

本软件将使用 Java 进行实现，通过 JDBC 对 MySQL 数据库进行操作。单元测试将在详细设计时进行准备，在编码完成后进行单元测试。本项目将使用 JUnit 进行单元测试，若分支覆盖率，单元覆盖率，方法覆盖率都到达 100%则满足需求。若出现不满足预期结果的情况，则使用需求跟踪矩阵与鱼骨图进行需求确认与查询开发人员，确保 bug 被修复。

开发者应参加分析单元测试的结果，软件相关的分析和测试结果应该记录在相应的软件开发文件中。

5.8 单元集成和测试

集成测试将在概要设计时进行准备，在单元测试的基础上，将所有模块按照设计要求(如根据结构图)组装成为子系统或系统，进行集成测试。若出现不满足预期结果的情况，则使用需求跟踪矩阵与鱼骨图进行需求确认与查询开发人员，确保 bug 被修复。

开发者应参加分析单元集成和测试的结果，软件相关的分析和测试结果应该记录在相应的软件开发文件中。

5.9 CSCI 合格性测试

本产品将在裸机上进行安装与测试，确保用户可以按说明手册正确安装产品后，不会出现 bug。若出现不满足预期结果的情况，则使用需求跟踪矩阵与鱼骨图进行需求确认与查询开发人员，确保 bug 被修复。开发者应参加分析 CSCI 合格性测试的结果，软件相关的分析和测试结果应该记录在相应的软件开发文件中。

5.10CSCI/HWCI 集成和测试

本产品将在各主流 windows 系统上进行安装与测试，若出现不满足预期结果的情况，则使用需求跟踪矩阵与鱼骨图进行需求确认与查询开发人员，确保 bug 被修复。开发者应参加分析 CSCI/HWCI 集成测试的结果，软件相关的分析和测试结果应该记录在相应的软件开发文件中。

5.11系统合格性测试

本产品将在单元及集成测试完成后，进行系统合格性测试。是对整个系统的测试，将硬件、软件、操作人员看作一个整体，检验它是否有不符合系统说明书的地方。若出现不满足预期结果的情况，则使用需求跟踪矩阵与鱼骨图进行需求确认与查询开发人员，确保 bug 被修复。开发者应参加分析系统合格性测试的结

果，软件相关的分析和测试结果应该记录在相应的软件开发文件中。

5.12软件使用准备

本软件在交付给用户前，应先对主要用户进行培训，确保用户能正确的使用软件，实现自身的需求。并且应完成容易理解的用户说明手册，并在交付后现场指导用户进行软件的安装过程。

5.13软件移交准备

移交产品名称	产品描述	质量保证活动	验收标准	交付形式
软件开发计划	基于 MVC 设计模式的在线图书馆管理系统软件开发计划	正规检视及评审	归档	文档
概要设计文档	基于 MVC 设计模式的在线图书馆管理系统总体设计方案	正规检视及评审	归档	文档
详细设计文档	基于 MVC 设计模式的在线图书馆管理系统详细设计方案	正规检视及评审	归档	文档
软件需求规格说明	基于 MVC 设计模式的在线图书馆管理系统软件需求规格说明	正规检视及评审	归档	文档
软件交付说明书	基于 MVC 设计模式的在线图书馆管理系统软件交付说明	正规检视及评审	归档	文档
源代码	基于 MVC 设计模式的在线图书馆管理系统源代码	正规检视及评审	归档	工程文件
可执行文件	在线图书馆管理文件可部署的工程	软件测试	发布	工程

表一.　软件移交准备

5.14软件验收支持

验收范围：针对本项目完整系统进行验收

验收内容：文档审查、功能模块审查、性能审查、用户可用性审查。

验收计划：

任务序号	任务名称	参加人	提交结果
1	审核文档	监理单位	出具审核意见
2	提交验收申请	承建方	文档审核交接
3	组建验收小组	业主方、监理单位	确定小组名单
4	验收小组工作会议	验收小组成员	确定参会人员、职责分工、验收计划、用户使用报告书和专家组成员
5	组织验收	验收小组成员	按照验收计划组织验收
6	验收评审会	验收小组、评审小组、各单位	形成验收意见，签署验收报告
7	文档移交	业主方、监理单位、承建单位	签署文档交接单

表二.　验收流程

5.15软件配置管理

版本号规则：各配置版本标识格式为“s.xy.m”，s 为 1-9 数字，xy 为 0-99 数字，m 为任意长度数字。S 为主版本号，初始版本号为 1，配置项在产品库中变更时，版本号加 1；xy 为受控版本号，初始受控版本号为 00，当体系文件在一次受控库中的变更出库时，受控版本号加 1；m 为开发库版本号，开发库版本号可根据实际情况省略不写。

配置项清单：

序号	配置项名称	配置项描述
1	软件开发计划	基于 MVC 设计模式的在线图书馆管理系统软件开发计划
2	概要设计文档	基于 MVC 设计模式的在线图书馆管理系统总体设计方案
3	详细设计文档	基于 MVC 设计模式的在线图书馆管理系统详细设计方案
4	软件需求规格说明	基于 MVC 设计模式的在线图书馆管理系统软件需求规格说明

5	软件交付说明书	基于 MVC 设计模式的在线图书馆管理系统软件交付说明
6	源代码	基于 MVC 设计模式的在线图书馆管理系统源代码
7	可执行文件	在线图书馆管理文件可部署的工程

表三.　配置项清单

5.16软件产品评价

随着人类社会的不断进步和发展，传统意义上的人工统计和管理图书的方法也变得越来越繁琐、复杂。由于图书和文献的种类和数量不断增长，按照传统模式仅仅依靠图书管理员人工管理图书、管理读者，不仅工作量巨大，而且会不可避免地出现各种错误。为了高效处理图书馆中的各种信息，随时随地管理和掌握图书信息及借阅记录等信息，本文采用 MVC 技术，设计并开发一个基于 B/S 模式的基于 MVC 设计模式的在线图书馆管理系统，使图书的管理更加简洁、高效、方便。

MVC 全名是 Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。基于 MVC 设计模式的在线图书馆管理系统，可以帮助图书馆管理员对图书和读者进行管理，方便图书馆管理员对图书的增加、删除、修改、查找的操作和对读者的借还书的操作。对读者而言，能够在较短的时间内查找到并预览要借阅的图书。同时读者可以对图书进行在线预约，不再担心因时间不充分而无法及时到图书馆借书而导致图书被他人借走的情况。

5.17软件质量保证

工作产品审核：小组内人员交叉评审。

不符合问题的解决：找到的问题按照需求优先级按顺序完成解决。

工具：Junit 测试工具

记录的收集、维护和保存：测试用例和文档统一归档保存。

5.18纠正措施

审查不合格的部分，组织分析，确定问题，相关责任人针对原因制定纠正或预防措施，之后进行跟踪验证，分别汇总不合格/不符合统计结果并将预防措施信息、体系审核结果提交管理评审，查看是否需要修改体系文件和技术文件。

5.19联合评审

联合技术评审：遵循企业级应用软件开发联合技术评审标准。

联合管理评审：遵循企业级应用软件开发联合管理评审标准。

5.20风险管理

项目风险管理表

风险类别	详细描述
合同条款风险	合同条款模糊、合同工期延期罚款
工程技术风险	测试指标验收、配套设备需求
项目资源风险	人力物力资源不足、资源保障不充分
竞争对手风险	项目实施中来自竞争对手的良性或恶性竞争带来的风险
客户风险	环境准备、配套设施准备不足

表四.　项目风险类别

参数	值	定性描述	进度	成本	质量	范围
概率	0.9	非常高				
	0.7	高			√	
	0.5	中	√			
	0.3	低				
	0.1	非常低		√		√
影响	0.8	非常高	进度延期两周	成本超支20%	项目最终成果实际无法使用	每周重大变更3次
	0.4	高	进度延期一周	成本超支15%	项目质量低到客户无法接受	每周重大变更2次
	0.2	中	进度延期5天	成本超支10%	项目质量下降到需要顾客审批同意	每周重大变更1次
	0.1	低	进度延期3天	成本超支5%以内	项目最终成仅有要求极其严格的应用受到影响	每两周重大变更1次
	0.05	非常低	进度延期1天以内	成本超支不明显	项目质量下降不明显	每三周周重大变更1次

表五.　项目风险预测

详情见《项目风险管理计划》。

5.21测量和分析

该项目无需进行数据测量和分析。

5.22保密性

保密性部分仅涉及密码存储和参数传递，遵循企业级应用软件开发规范。

5.23分承制方管理

该项目无需进行分承制方管理。

5.24与软件独立验证和确认(IV&V)机构的联系

通过静态分析、算法分析或代码审查来对软件源代码进行评估。

静态分析是一种对代码的机械性的和程序化的特性分析方法。静态分析一般需进行：控制流分析、数据流分析、接口分析和表达式分析。算法分析包括等式的重新推导或者特定数字技术的适用性的评价。

代码审查是检查代码和设计的一致性、代码执行标准的情况、代码逻辑表达的正确性、代码结构的合理性以及代码的可读性。代码审查应根据所使用的语言和编码规范设计检查单并执行。检查单的设计或采用应经过评审并得到委托方的确认。

使用可追踪性分析从软件需求跟踪到概念文档，再对系统需求进行验证。可跟踪性分析可以用来支持配置管理、测试覆盖率分析、验证和确认结果分析、回归测试、关键性评估以及验证和确认的管理决策。

功能测试是对软件需求规格说明中的功能需求逐项进行的测试，以验证其功能是否满足要求。功能测试一般需进行：用正常值的等价类输入数据值测试；用非正常值的等价类输入数据值测试；进行每个功能的合法边界值和非法边界值输入的测试等。

鲁棒测试是检验软件中已存在的安全性、安全保密性措施是否有效的测试。测试应尽可能在符合实际使用的条件下进行。鲁棒测试一般需进行：对安全性关键的软件部件，必须单独测试安全性需求；在测试中全面检验防止危险状态措施的有效性和每个危险状态下的反应；对设计中用于提高安全性的结构、算法、容错、冗余及中断处理等方案，必须进行针对性测试等。

边界测试是对软件处在边界或端点情况下运行状态的测试。边界测试一般需进行：软件的输入域或输出域的边界或端点的测试；状态转换的边界或端点的测试；功能界限的边界或端点的测试等。

5.25与相关开发方的协调

该项目由开发小组独立完成，无相关开发方，不需要与相关开发方协调。

5.26项目过程的改进

(1)过程分析

考察和理解现有的过程，在一些情况下需要对过程的某些环节进行度量和定量分析，利用取得的数据来表明过程的状况。同时，这些数据可以用来与过程改进后的状况进行对比。

(2)确定改进

利用过程分析的结果，找出原有过程中质量、进度和成本的瓶颈。针对发现的问题，制定过程改进方案，提出需要采用什么规程、方法和工具的建议。

(3)过程变更

实施过程变更，把新的规程、方法和工具安置于合适的过程环节上，并且与其他的软件过程活动集成起来。

(4)培训

没有培训的过程变更在大多数场合注定要失败。有的单位在培训工作不够充分的情况下，强制推行过程变更，这样做不会收到好的效果。

(5)调整过程变更

在初步实施过程变更后，不可能立即收到圆满的效果，在过程修改后还可能会出现一些小的问题，这就需要进行适当的调整。

5.27未提及的其他活动

无。

6 进度表和活动网络图

a) 进度表

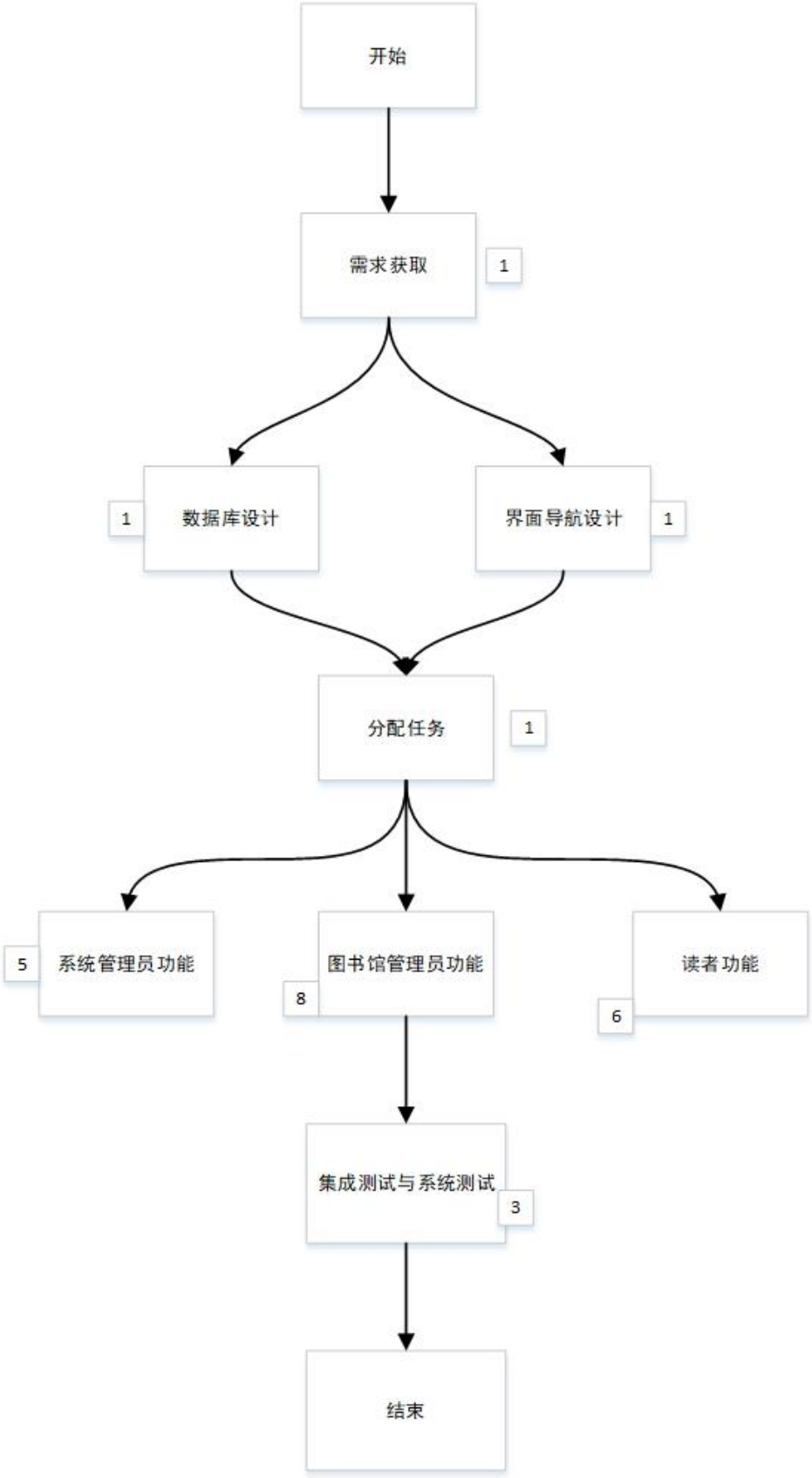
基于MVC设计模式的在线图书馆管理系统	**20 days**	**2018年12月16日**	**2019年1月6日**
项目管理	**3 days**	**2018年12月16日**	**2018年12月18日**
计划安排	**1 day**	**2018年12月16日**	**2018年12月16日**
获取需求	1 day	2018年12月16日	2018年12月16日
分配任务	1 day	2018年12月16日	2018年12月16日
选择服务器软件和第三方库	1 day	2018年12月16日	2018年12月16日
安排时间并创建发布计划	1 day	2018年12月16日	2018年12月16日
初始化	**2 days**	**2018年12月17日**	**2018年12月18日**
设计数据库	1 day	2018年12月18日	2018年12月18日
设计页面之间的导航流程	1 day	2018年12月18日	2018年12月18日
第一版	**4 days**	**2018年12月19日**	**2018年12月23日**
冲刺1	**2 days**	**2018年12月19日**	**2018年12月20日**
系统管理员登入登出+单元测试	1 day	2018年12月19日	2018年12月19日
系统管理员注册图书馆管理员+单元测试	1 day	2018年12月19日	2018年12月19日
系统管理员对图书馆管理员的删改查+单元测试	1 day	2018年12月20日	2018年12月20日
图书馆管理员登入登出+单元测试	1 day	2018年12月20日	2018年12月20日
读者登入登出+单元测试	1 day	2018年12月20日	2018年12月20日
冲刺2	**2 days**	**2018年12月21日**	**2018年12月23日**
图书馆管理员添加书籍+单元测试	1 day	2018年12月21日	2018年12月21日
图书馆管理员注册读者+单元测试	1 day	2018年12月21日	2018年12月21日
图书馆管理员管理读者+单元测试	1 day	2018年12月23日	2018年12月23日
借书功能+单元测试	1 day	2018年12月23日	2018年12月23日
还书功能+单元测试	1 day	2018年12月23日	2018年12月23日
集成测试+系统测试	1 day	2018年12月23日	2018年12月23日
完成第一版[里程牌 1]	0 days	2018年12月23日	2018年12月23日

第二版	**3 days**	**2018年12月24日**	**2018年12月26日**
冲刺3	**1 day**	**2018年12月24日**	**2018年12月24日**
读者浏览借阅记录+单元测试	1 day	2018年12月24日	2018年12月24日
冲刺4	**1 day**	**2018年12月24日**	**2018年12月24日**
读者预约书籍+单元测试	1 day	2018年12月24日	2018年12月24日
冲刺5	**2 days**	**2018年12月24日**	**2018年12月25日**
读者查书+单元测试	2 days	2018年12月24日	2018年12月25日
图书馆管理员查书+单元测试	2 days	2018年12月24日	2018年12月25日
图书馆管理员处理被预约的书籍+单元测试	1 day	2018年12月24日	2018年12月24日
图书馆管理员查看所有图书的借阅记录+单元测试	1 day	2018年12月24日	2018年12月24日
冲刺6	**2 days**	**2018年12月25日**	**2018年12月26日**
系统管理员修改个人信息+单元测试	1 day	2018年12月25日	2018年12月25日
读者修改个人信息+单元测试	2 days	2018年12月25日	2018年12月26日
图书馆管理员修改个人信息+单元测试	2 days	2018年12月25日	2018年12月26日
集成测试+系统测试	1 day	2018年12月25日	2018年12月25日
完成第二版 [里程牌 2]	0 days	2018年12月25日	2018年12月25日
第三版	**5 days**	**2018年12月26日**	**2018年12月31日**
冲刺7	**5 days**	**2018年12月26日**	**2018年12月31日**
图书馆管理员查询读者信息	2 days	2018年12月26日	2018年12月27日
图书馆管理员删除图书	2 days	2018年12月27日	2018年12月28日
图书馆管理员查询图书信息	2 days	2018年12月29日	2018年12月31日
图书馆管理员修改图书信息	3 days	2019年1月29日	2019年1月31日
冲刺8	**5 days**	**2019年1月1日**	**2019年1月5日**
读者在线借阅图书	2 days	2019年1月1日	2019年1月2日
集成测试+系统测试	3 days	2019年1月3日	2019年1月5日
完成第三版 [里程牌3]	0 days	2018年12月26日	2018年12月26日

第三版	**5 days**	**2018年12月26日**	**2018年12月31日**
冲刺7	**5 days**	**2018年12月26日**	**2018年12月31日**
图书馆管理员查询读者信息	2 days	2018年12月26日	2018年12月27日
图书馆管理员删除图书	2 days	2018年12月27日	2018年12月28日
图书馆管理员查询图书信息	2 days	2018年12月29日	2018年12月31日
图书馆管理员修改图书信息	3 days	2019年1月29日	2019年1月31日
冲刺8	**5 days**	**2019年1月1日**	**2019年1月5日**
读者在线借阅图书	2 days	2019年1月1日	2019年1月2日
集成测试+系统测试	3 days	2019年1月3日	2019年1月5日
完成第三版 [里程牌3]	0 days	2018年12月26日	2018年12月26日
监控	**20 days**	**2018年12月16日**	**2019年1月6日**
每日站会	20 days	2018年12月16日	2019年1月6日
冲刺回顾	20 days	2018年12月16日	2019年1月6日
将关键任务重新分配给成员	20 days	2018年12月16日	2019年1月6日
关闭	**1 day**	**2019年1月6日**	**2019年1月6日**
最终项目汇报	1 day	2019年1月6日	2019年1月6日
冲刺回顾	1 day	2019年1月6日	2019年1月6日
项目完成 [里程牌4]	0 days	2019年1月6日	2019年1月6日

图二十六.　进度表

b) 活动网络图



图二十七. 网络活动图

7 项目资源

a) 人力资源

本团队由 4 名高级软件工程师组成，其中胡钰玺同时担任项目经理，朱日勇同时担任软件架构师，刘卓程同时担任软件测试师，姬轶同时担任平面设计师。

b) 为适应合同(或软件研制任务书)中的工作，开发人员工作的地理位置、要使用的设施、保密区域和设施的其他特征。

开发人员均在西安进行系统设计与开发

c) 合同(或软件研制任务书)中工作需要的、且由需方提供的设备、软件、服务、文档、数据及设施，并给出何时需要上述各项的进度表。

见 6 中进度表。

8 注释

MVC：model, view, control 的缩写，分别代表模型、视图、控制器。