

合肥工业大学

《专业实习与实训》

报告

题 目： 图书管理系统

学 号： 2019217192

姓 名： 周布伟

专业班级： 计算机科学与技术 19-2 班

时 间： 2022 年 10 月-12 月

计算机科学与技术专业“专业实习与实训”验收细则			
成绩等级	具体表现	教师评分	
优秀（100-85]	1)能够在规定时间内完成项目，项目具有完整性，交互设计优秀；2)工作量充足；	<input type="checkbox"/>	
良好（85-75]	1)能够在规定时间内完成项目，且项目具有较好的完整性，交互设计较为优秀；2)工作量较为充足；	<input type="checkbox"/>	
中等（75-65]	1)能够在规定时间内完成项目，项目完整性较好，交互设计一般；2)工作量一般；	<input type="checkbox"/>	
及格（65-60]	1)虽完成项目编码但未能在规定时间内完成的；	<input type="checkbox"/>	
不及格（<60）	1)未进行验收的；	<input type="checkbox"/>	

计算机科学与技术专业“专业实习与实训”实习报告评分细则			
成绩等级	具体表现	教师评分	
优秀（100-85]	1)实习报告格式完美，充分采用图表来说明问题，章节设计优秀，工作量饱满；	<input type="checkbox"/>	
良好（85-75]	1)实习报告格式良好，采用了图表来说明问题，章节设计良好，工作量较为饱满；	<input type="checkbox"/>	
中等（75-65]	1)实习报告格式较好，章节设计较好，工作量一般；	<input type="checkbox"/>	
及格（65-60]	1)虽提交报告但未能在规定时间内完成的；	<input type="checkbox"/>	
不及格（<60）	1)未提交报告的；	<input type="checkbox"/>	

教师签名：

1 实训目的及内容简介

此次《专业实习与实训》，我们运用前期的集中学习所得的知识和锻炼的能力，完成了图书管理系统项目需求的确定、项目方案的制定，并在规定的时间内完成了该项目的部署与实现。

1.1 实训目的

掌握基本的项目开发基本技能，锻炼团队开发能力；理解并运用项目开发的基本方法、流程，锻炼独立发现问题并解决问题的能力，实现多思考、多联想、多实操的学习模式；理解 MVC 思想，掌握 Web 应用开发；分析并独立撰写需求文档，完成数据库设计，能运用数据库、计算机网络等专业知识实现需求，并在实现过程中运用创新能力对系统进行个性化的实现。

1.2 实训内容

本次《专业实习与实训》主要分为集中授课学习与项目设计与总结验收两个阶段。集中授课学习阶段邀请校外公司人士徐星远老师为计算机与物联网专业的同学讲授若依框架开发 Web 应用、数据库设计等相关内容，指导同学完成项目的设计，使同学具备一定的项目开发经验。项目设计与总结答辩阶段需要同学结合集中授课学习阶段得到项目开发的知识与技能，以小组的形式完成 Web 网页项目的设计与制作，最终老师以验收、项目报告设计评估等方式对实践成果进行检验。两个阶段依次进行，同学在每周的周一与周五固定的时间进行集中授课学习，在其余时间进行项目的设计与制作，具体实践计划安排如下。

第一阶段——集中授课学习阶段：2022 年 10 月 17 日至 2022 年 11 月 11 日，每周周一和周五

计算机专业 5 个班级和物联网专业 2 个班级的同学按照机房安排于每周周一和周五集中计算机中心进行项目开发相关知识的学习，本班级被安排在第一机房。

2022 年 10 月 17 日 10:10-12:00，第一次授课：项目环境搭建与若依框架介绍。了解课程安排；依托 Eclipse，学习项目搭建的方法，完成若依的部署与调试。

2022 年 10 月 21 日 10:10-12:00，第二次授课：学习 Spring、SpringMVC 与接口编程。了解控制反转（Inversion of Control, IOC）和依赖注入（Dependency

Injection, DI) 核心思想和工作原理, 以及 IoC 容器的两种实现。

2022 年 10 月 24 日 10:10-12:00, 第三次授课: 安全认证与权限管理架构。了解权限管理的基本功能; 学习 Shiro 框架及其基本原理, 理解其内部的架构与配置方法; 掌握通过 Shiro 与 Spring 整合配置的步骤。

2022 年 10 月 28 日 10:10-12:00, 第四次授课: Mybatis 初步。理解使用 JDBC 的缺陷, 包括通过 JDBC 连接数据库的步骤; 掌握 Mybatis 的整体架构和配置, 熟悉使用 Mybatis 进行 CRUD 操作; 学习分层管理的思想。

2022 年 10 月 31 日 10:10-12:00, 第五次授课: 深入学习 Mybatis, 了解模板技术。理解关系型数据库的多种关系类型, 熟悉各种类型的 Mybatis 文件配置; 了解使用动态 SQL; 学习使用 Thymeleaf 模板引擎渲染界面, 掌握其基本语法; 初步熟悉项目开发的基本流程。

2022 年 11 月 04 日 10:10-12:00, 第六次授课: 实现学生档案功能。依托若依框架, 通过数据库建表、新建模块、设置 Mapper 和 Service 等步骤, 完成学生档案功能的实现; 进一步熟悉分层项目开发的流程与步骤。

2022 年 11 月 07 日 10:10-12:00, 第七次授课: 实现列表页面设计与 CRUD 操作。依托若依框架, 完成列表页的界面设计; 分模块实现修改、删除、导出等操作。

2022 年 11 月 11 日 10:10-12:00, 第八次授课: 实现客户模块, 多表操作, 杂项与答疑。通过分层开发实现客户模块, 配置相关文件, 完成多表操作; 学习树表 zTree 和详细按钮。

第二阶段——项目设计与总结验收阶段: 2022 年 11 月 14 日至 2022 年 12 月 09 日

第一阶段后, 根据项目选题构建项目, 利用课内与课外所学, 按照项目开发的基本流程, 对各个页面的前端后端以及数据库等内容进行设计实现。最后以验收、实践报告等形式展现项目成果。本项目选择图书馆管理系统作为项目实践题目, 依托培训中所学的框架和工具进行开发和设计。

2 项目题目简介

本节将给出此次图书管理系统项目的背景和项目实施的目的与意义, 针对目

的，提出本次项目的设计任务与要求。

2.1 项目背景与设计意义

随着计算机以及网络技术的不断发展，图书馆也进入了信息化的时代。图书馆，是文化的传播者，图书的收藏者，精神的守护者，是一座城市的文化地标。然而，当图书馆发展到一定的规模时，读者和书刊是每个图书馆管理者都需要考虑的因素，如何在能够管理好书刊和读者，又能带来一定的收益，是一个需要解决的关键问题。

2.2 设计任务与要求

设计并实现图书馆管理系统，主要对图书进行增删改查操作。主要的设计要求包括：

- (1) 信息要求：管理系统中应包括图书类别、期刊类别、出版社等基本信息管理，还应该管理图书信息、期刊信息、客户信息。
- (2) 功能要求：管理系统应该实现书刊的借阅、归还管理，包括信息的产生、查询等，针对读者的不同需求，设置不同等级的会员，提供不同的优惠策略。书店管理系统还应该实现余额管理，支持押金的收款、余额的充值和退款等功能。

3 项目需求分析与设计

第 2 节的题目要求指出了本项目旨在设计和实现一款支持书刊管理、借阅、归还管理、余额管理等功能的图书管理系统。本节将根据上文提出的设计任务与要求进行需求分析，然后设计数据字典和数据流图，着眼于了解用户的数据需求和事务需求。然后，针对需求分析进行数据库的概念结构和逻辑结构的设计，旨在将需求分析中的用户需求抽象为概念模型，并将概念模型转换为数据库管理系统所支持的数据模型相符合的逻辑结构。

3.1 需求分析

用户需求主要包含数据需求和事务需求，数据需求描述了使用本图书管理系统的用户所需要的数据，事务需求主要反映了使用者的实现某项事务的要求。

- (1) 数据需求

本系统主要管理员信息，其主要管理书刊和读者的数据信息，读者主要提供个人信息，并参与书刊借阅和零售的全过程。

管理员主要指书店的管理员，是系统中权限最大的用户，不但能够对书刊进行添加、修改、删除等管理操作外，还能够对读者客户进行添加、注销、修改信息等管理操作，包括用户名的变更、会员的升级与降级、余额的充值与提现等。对此，其数据需求主要包括书刊信息、读者客户信息、会员信息、借阅信息等，以进行书店书刊和读者客户的一体化管理，提高经营效率和经济效益。

书刊信息，让管理员对书店中的书刊有直接的了解和记录，便于书刊的维护。包括书刊的编号、名称、出版社、价格、总量、在册数量、作者、简介、书刊类别等信息。

读者客户信息，使管理员不仅记录经常光顾书店的读者信息，还能够帮助管理员实现书籍零售借阅的定位和追踪。包括读者的姓名、性别、手机号、会员等级、账号状态、余额、已借阅数量等信息。

会员信息，让管理员能够自主设定借阅的额度、优惠措施和惩罚措施，扩大书店的影响力，增加书店的经济效益。包括押金、出租折扣、零售折扣、可借阅数量、可续借次数、逾期扣费、损坏扣费、丢失扣费等。

借阅信息，为管理员能够快速查询和追踪读者与书籍的历史行为，进一步维护读者和书刊信息。包括客户编号、书刊编号、借阅时间、到期时间、续借次数、借阅状态、归还时间、扣费类型、扣费金额等。

个人信息，主要便于管理员在系统上的登录与注册。包括管理员的用户名、密码和账号状态。

书刊类型信息，主要便于管理员在系统上管理书刊的类型，保证书刊的类型是统一的，不会出现同名异义或异名同义的情况，更便于规范化书店的图书分区和摆放位置。

出版社信息，主要便于管理员了解和联系出版社，用于书刊的交易，该名称应该是统一的，不会出现同名异义或异名同义的情况。

（2） 事务需求

管理员不仅要维护书刊、读者、会员类型等基本信息，还要记录读者的借阅、零售等操作，包括书刊管理、读者管理、会员类型管理、借阅管理等。

书刊管理，图书、期刊等数量、名称、出版社、价格、总量、作者、简介、书刊类别等信息管理；书刊基本信息，如名称、数量、作者、出版社、价格等的添加、删除、修改；查询书店中图书的信息等。

读者管理，主要有读者的添加与注销，读者信息的修改，如姓名、性别、会员等级、账号状态、余额等信息。

会员类型管理，支持不同等级的会员优惠和惩罚措施，涵盖借阅、零售，为客户提供有针对性的服务。

借阅管理，实现读者对书刊的借阅，记录操作的时间、操作的业务对象等，借阅操作中还涉及归还、续借的管理与记录。

3.2 概念结构设计

通过对用户和系统的需求分析，我们可以得到详细的数据描述，对此还可以通过数据流图定义数据项和数据结构，描绘数据信息在系统中流动和处理的情况和逻辑内容。

(1) 系统顶层数据流图

管理系统的使用者主要是图书馆管理人员，服务对象是用户（读者）。如图 1 所示，读者来到图书馆主要进行图书的借阅、归还等事务，管理人员使用该系统为用户完成用户要求的业务，读者接受服务后，离开图书馆，管理员还可以管理书刊信息和会员信息。



图 1 系统顶层数据流图

(2) 系统 0 层数据流图

将系统顶层数据流图进一步细化，可以得到的 0 层数据流图，如图 2 所示，其中涵盖了借阅管理、客户管理、书刊管理、会员管理等事务的数据流与数据存储表示，用户（读者）和管理员作为系统服务的实体对象，分别能够完成相应的操作。

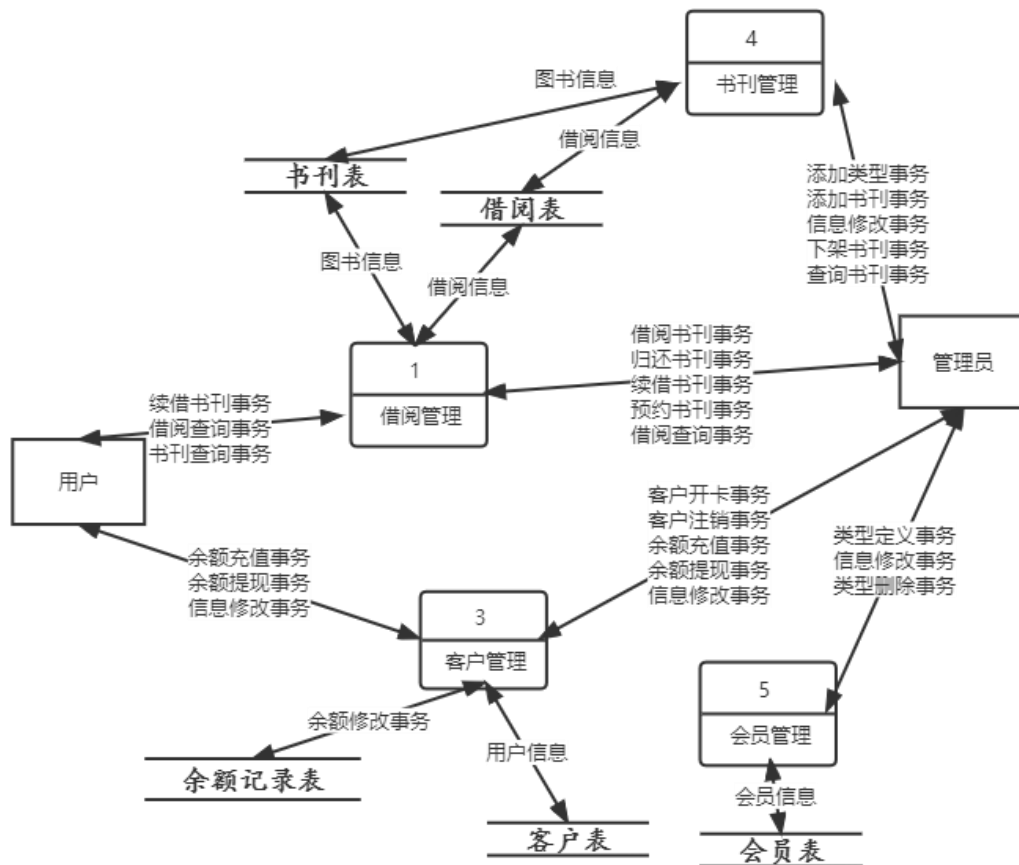


图 2 系统 0 层数据流图

(3) 数据字典

为了进一步说明数据流图中的关键数据流和存储过程的逻辑内容，下面给出数据字典，它是数据库中所有对象及其关系的信息集合，数据字典主要通过数据项和数据结构的定义来说明数据流图中的数据流和数据存储。其具体定义如表 1 所示，其中主要包含了管理人员信息、会员类型、余额记录信息、书刊信息、借阅信息记录等。

表 1 数据字典定义

数据流名	别名	描述	定义
管理人员信息	user	记录每个管理人员信息的信息表格	管理员 ID=管理员 ID+姓名+密码+信息状态
会员类型	vip	记录每个会员类型的信息表格	会员类型=会员类型名+押金+出租折扣+零售折扣+可借阅数量+可续借次数+可借阅天数+逾期扣费+损坏扣费+丢失扣费
余额记录信息	balance	记录每次金额变更的	余额记录信息=操作时间+客户 ID+操

书刊信息	book	信息表格 记录每本书刊具体的 信息表格	作金额+操作类型 书刊信息=ISBN+书刊名称+出版社+ 价格+总量+在册数量+预约数量+单位 +作者+简介+书刊类型+ 书刊类别
借阅信息记录	borrow	记录每次借阅发生的 具体信息的表格	借阅信息=客户 ID + ISBN + 借阅时间+ 到期时间+续借次数+借阅状态+归还 时间+扣费类型+扣费金额

(4) ER 图

为了描述现实世界中的概念模型,进一步转换需求分析中的数据流图和数据字典,我们使用 E-R 图辅助概念结构的设计,如图 3 所示,其中包含了会员、客户、书刊、余额记录、管理员等实体,其相关的属性用椭圆形标识,并与实体连接,实体之间发生多种联系,部分联系包含了相关的属性。

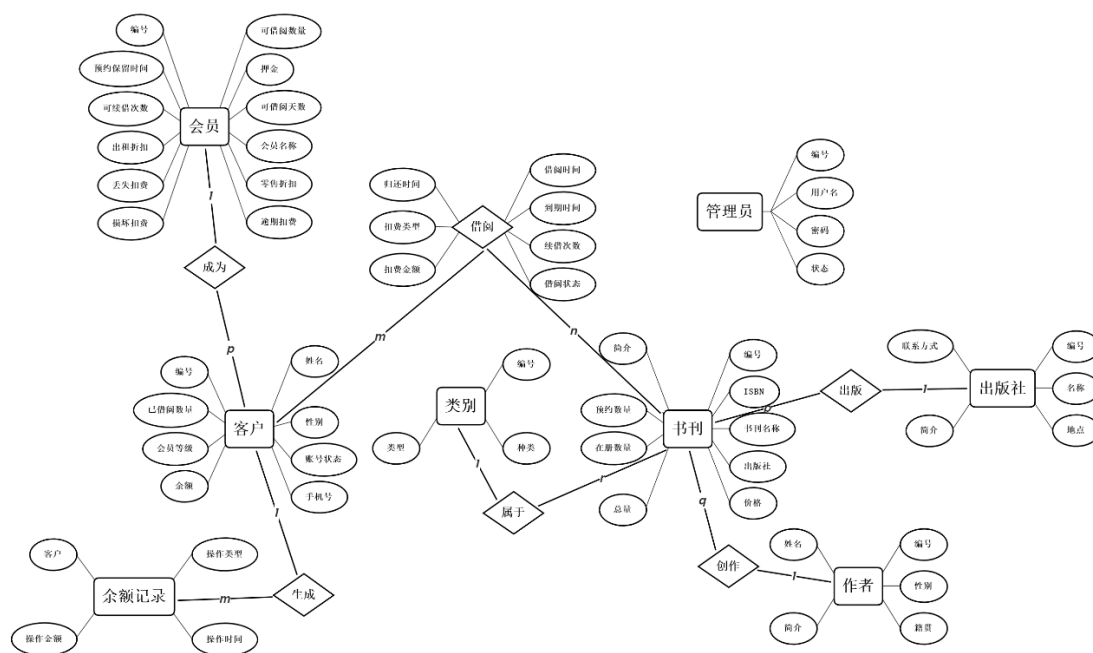


图 3 图书管理系统的 ER 图

由于管理员不直接参与整个书店的操作业务中,所以并未与其他实体发生联系,但是它的基本属性是之后设计数据库表和具体应用的前提,故在此将其单独列出。客户与书刊之间主要发生了借阅联系,都是多对多的联系,即一位读者可以同时借阅多种书,一种书可以被多个读者借阅,其中包含了发生联系之后增加的属性。值得注意的是,此处使用一种书作为基本的实体,而非一本书,故其中还包含了总量、在馆数量等信息。此外,在需求分析中还涉及到会员管理和余额管理等需求,在此专门定义了相关的实体用于后续的详细设计,其中会员包含了

押金、可借阅数量、可借阅天数、出租折扣等基本属性，余额记录包括了操作类型、操作金额等属性，且是一对多的联系。为了便于图书馆管理员进行书刊类型分区，与出版社进行交易，避免出现同名异义或异名同义的情况，E-R 图还应该加入书刊类型和出版社，并将其作为实体，而非仅仅是其他实体的属性。

3.3 逻辑结构设计

(1) 关系模式与基本表

由于概念结构是独立于任何一种数据模型的信息结构，为了便于后期数据库设计和项目开发，我们需要进行逻辑结构的设计，即把概念设计阶段设计好的概念模型 E-R 图转换为关系模型，得到最初如下 9 个关系模式，关系的主码用下划线标出。

客户（客户 ID，姓名，手机号，性别，会员等级，余额，账号状态，已借阅数量）

管理员（管理员 ID，用户名，密码，状态）

余额记录（操作时间，客户 ID，操作金额，操作类型）

会员（会员 ID，会员名称，押金，出租折扣，零售折扣，可借阅数量，可续借次数，可借阅天数，预约保留时间，逾期扣费，损坏扣费，丢失扣费）

书刊（书刊 ID，书刊名称，书刊编号，出版社，价格，总量，在册数量，预约数量，单位，作者，简介，书刊类型，书刊类别）

借阅（客户 ID，书刊 ID，借阅时间，到期时间，续借次数，借阅状态，归还时间，扣费类型，扣费金额）

类型（类型 ID，类型，种类）

作者（作者 ID，姓名，性别，籍贯，简介）

出版社（出版社 ID，名称，联系方式，地点，简介）

将上述各关系模式在 MySQL 数据库和 Navicat 平台上实现，使用 CREATE 语句为其创建相应的数据库表，对应的 9 张基本表结构和满足的规范化程度如表 2 ~ 表 10 所示，可见所有的表都满足较高的规范化程度，几乎不存在数据的操作异常，且具有较低的冗余。

表 2 客户表 Client: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
-----	-----	----	----	------	----	----

客户 ID	clientID	char	11	主码	否	
手机号	clientPhone	char	11		否	
姓名	clientName	varchar	30		否	
性别	clientSex	char	1			check 男/女
会员等级	clientGrade	varchar	8	外码	否	
余额	clientBalance	decimal	10,2		否	
账号状态	clientStatus	varchar	4		否	正常、冻结、注销
已借阅数量	clientBorrowNum	smallint	2		否	

表 3 管理员表 Admin: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
编号	admin_id	varchar	11	主键	否	
用户名	adminName	varchar	30	主码	否	
密码	adminPassword	varchar	20		否	
状态	adminStatue	varchar	2		否	check 正常、注销

表 4 余额记录表 Balance: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
操作时间	opTime	date	6	主码	否	
客户 ID	clientID	varchar	11	主码, 外码	否	
操作金额	opAmount	decimal	10,2		否	
操作类型	opType	varchar	4		否	

表 5 会员表 VIP: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
会员编号	vipId	varchar	8	主	否	
会员名称	vipName	verchar	10	主码	否	
押金	vipDeposit	decimal	10,2		否	
出租折扣	vipRentDiscount	decimal	5,4		否	
零售折扣	vipSellDiscount	decimal	5,4		否	
可借阅数量	vipBorrowNum	smallint	2		否	
可续借次数	vipRenew	smallint	2		否	
可借阅天数	vipBorrowDate	smallint	4		否	
预约保留时间	vipTime	smallint	2		否	
逾期扣费	vipOverdue	decimal	4,2		否	
损坏扣费	vipDamage	decimal	4,2		否	

表 6 书刊表 Book: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
书刊 ID	bookID	char	13	主码	否	
书刊编号	bookNum	char	13		否	
书刊名称	bookName	varchar	30		否	
出版社	bookPublic	varchar	30		否	
价格	bookPrice	decimal	10,2		否	

总量	bookTotalNum	smallint	3	否	
在册数量	bookNowNum	smallint	3	否	
预约数量	bookReserveNum	smallint	3	否	
作者	bookAuthor	varchar	30	否	
简介	bookIntroduction	varchar	500		
书刊类型	bookType	varchar	10	否	check 书籍、期刊
书刊类别	bookCategory	varchar	10	否	

表 7 借阅表 Borrow: 第三范式

属性名	字段名	类型	宽度	主/外码	空值	描述
客户 ID	clientID	char	13	主, 外	否	
ISBN	bookISBN	char	13	主, 外	否	
借阅时间	borrowDate	date	6	主	否	
到期时间	expiryDate	date	6		否	
续借次数	renewNum	smallint	2		否	
借阅状态	borrowStatus	verchar	3		否	check 在借、归还
归还时间	returnDate	date	6			
扣费类型	deductionType	verchar	3			无、逾期、损坏、丢失
扣费金额	deductionAmount	decimal	10,2			

表 8 类型表 Type: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
编号	type_id	varchar	10	主键	否	
类型	type_class	varchar	10		否	杂志/图书/……
种类	type_category	varchar	10		否	科技/人文/……

表 9 作者表 Author: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
编号	author_id	varchar	11	主键	否	
姓名	author_name	varchar	30		否	
性别	author_sex	char	1			check 男/女
籍贯	author_location	varchar	30			
简介	author_introduction	varchar	500			

表 10 出版社表 Publisher: BC 范式

属性名	字段名	类型	宽度	主/外码	空值	描述
编号	publisher_id	varchar	8	主键	否	
名称	publisher_name	varchar	30		否	
联系方式	publisher_phone	char	11			
地点	publisher_location	varchar	30			
简介	publisher_introduction	varchar	500			

(2) 用户子模式与视图

通过逻辑结构设计，我们将概念模型转换为逻辑模型，定义了关系模型和基

本表，然而，该逻辑模型是全局的，对应于数据库三级结构中的模式，还需要根据局部应用需求，结合具体的 MySQL 关系数据库管理系统设计用户的外模式，可以通过定义视图来实现。视图是从一个或几个基本表（或视图）导出的表。它是一个虚表。使用视图不仅能够简化用户对系统的使用，还能够确保数据库在一定程度上的安全和提高数据的独立性。

为了实际应用的需要，本书店管理系统中建立了多张视图，主要包括读者会员信息视图、读者借阅视图、借阅信息视图等，以下将列举本人参与设计和较为典型的视图进行介绍。

由于进行书籍借阅等操作时需要对书籍的出版社、作者等信息进行查询，而图书表中仅仅包含了这些字段的编号，为了展示名称，需要通过视图实现，并简化查询，通过以下语句将图书表、作者表和出版社表进行连接构建视图，使用该视图，在完成图书业务操作执行查询语句时，无需使用复杂的语句，只需查询视图中的内容即可。

```
SELECT
    `book`.`book_id` AS `book_id`,
    `book`.`book_isbn` AS `book_isbn`,
    `book`.`book_name` AS `book_name`,
    `book`.`publisher_id` AS `publisher_id`,
    `book`.`book_price` AS `book_price`,
    `book`.`book_total_num` AS `book_total_num`,
    ( `book`.`book_now_num` - `book`.`book_reserve_num` ) AS `book_left_num`,
    `book`.`book_reserve_num` AS `book_reserve_num`,
    `book`.`book_author_id` AS `book_author_id`,
    `book`.`book_introduction` AS `book_introduction`,
    `book`.`book_type_id` AS `book_type_id`,
    `publisher`.`publisher_name` AS `publisher_name`,
    `author`.`author_name` AS `author_name`
FROM
    (
        ( `book` JOIN `publisher` ON ( ( `book`.`publisher_id` =
        `publisher`.`publisher_id` ) ) )
        JOIN `author` ON ( ( `book`.`book_author_id` = `author`.`author_id` ) ) )
```

读者进行借阅操作时需要对读者的会员信息进行查询，以获得读者拥有的折扣优惠、罚款额度、借阅期限、借阅数量等信息，为了简化查询，通过以下语句将客户表和会员表进行连接构建视图，类似的还有书刊借阅视图等，在此不再赘述。

```

SELECT
    `client`.`client_id` AS `client_id`,
    `client`.`client_phone` AS `client_phone`,
    `client`.`client_name` AS `client_name`,
    `client`.`vip_id` AS `vip_id`,
    `client`.`client_balance` AS `client_balance`,
    `client`.`client_status` AS `client_status`,
    `client`.`client_borrow_num` AS `client_borrow_num`,
    `vip`.`vip_name` AS `vip_name`,
    ( `vip`.`vip_borrow_num` - `client`.`client_borrow_num` ) AS
`client_left_borrow_num`,
    `vip`.`vip_borrow_date` AS `vip_borrow_date`,
    `vip`.`vip_rent_discount` AS `vip_rent_discount`
FROM
    ( `client` JOIN `vip` ON ( ( `client`.`vip_id` = `vip`.`vip_id` ) ) )

```

3.4 物理结构设计

为了加快查询的速度，我们在物理结构上进行优化，即定义若干索引，快速定位到需要查询的内容。索引主要给数据库管理系统使用，数据库管理员无法使用索引，只能定义索引，而索引是把“双刃剑”，它需要占用一定的存储空间，如果数据库表频繁地发生更新，那么索引的维护成本和负担就会增加。本系统除了 MySQL 会默认在各表的主码上建立索引来方便进行实体完整性检验之外，还考虑建立的主要索引如下。

(1) 会员名称索引

当读者进行借阅等业务操作时，系统往往需要查询该读者的会员名称，以确定借阅期限、借阅数量、出租折扣等会员信息，当用户量增加时，对会员名称的查询就有可能成为瓶颈，对此需要建立会员名称索引，以加快查询，而且，当读者开卡后，对读者基本表的操作相对较少，不会因为索引而造成更大的开销。

(2) 借阅和零售的日期索引

随着读者数量的增加和业务的不断发生，借阅等基本表都会大规模的数据量，而实际应用场景中需要通过日期判断读者是否发生逾期，或统计某段时间内书店借阅的数量，由此需要建立借阅的日期索引以提高查询的速度，但是，这类数据表的数据更新频率高，在所有的维护上也需要一定的开销，故还需要视实际应用场景测试后再做决定。

(3) 客户名称和手机号索引

在实际的应用场景中很少直接通过客户的 ID 来查询信息，更多的时候是通

过客户名称或手机号进行查找，为了提高查找速度，可以为其建立索引，且如上文所述，该类数据表中数据的维护并不复杂，不需要为索引造成更大的开销。

3.5 安全性设计

系统主要从数据库和系统本身来考虑安全性，通过数据库设计和系统开发，主要采用如下措施确保系统和数据库的安全性。

采用数据备份与恢复策略，管理员可以随时备份和恢复数据，系统还能够定期保存数据，避免突发事件发生带来的毁灭性影响；

使用 MD5 哈希函数以密文的形式对敏感信息进行保存，如管理员密码等，防止数据库泄露或非法用户入侵对管理员和客户信息造成的损害；

基于 Session 机制，防止非法用户绕过登录功能直接进入系统，加入验证码机制，防止不法分子使用机器人登录账号；

采用权限管理，建立健全规章制度，加强工作人员的安全防范意识，只有管理员才能进入系统，负责系统的客户和书籍信息；

使用视图，不仅简化查询，提高数据独立性，还能保护其他数据，限制业务操作的作用范围。

4 项目实施方案

通过需求分析和数据库设计，本项目已经具有一定的雏形，基于 MyBatis 框架、Vue 框架、Maven 管理工具等，结合 MVC 设计思想，便可以进行项目网页端的应用程序设计与开发。下面将介绍本项目的开发环境、功能模块、界面设计和事务设计，并重点本系统中我所实现的部分。

4.1 开发环境与软件

本次项目的开发环境如表 11 所示，采用前后端分离的思想，前端使用 Vue 框架，后端使用 Servlet 处理，划分 Service 层和 DAO 层。期间使用了 Navicat Premium 12V、MySQL 5.5 和 JDK8.0，引用了 junit、mybatis、fastjson 等依赖，采用了 tomcat、mybatis-generator 等插件。

表 11 开发环境

类型	配置
设备型号	HUAWEI MateBook 13

处理器	Intel(R)Core(TM) i7-8565U CPU @1.80GHz 1.99 GHz
已安装的内存(RAM)	8.00GB
系统类型	Windows 10 家庭中文版, 64 位操作系统, 基于 x64 的处理器
硬盘名称	WDC PC SN720 SDAPNTW-512G-1027
采用软件	IntelliJ IDEA2020.1 旗舰版
采用编程语言	Java

4.2 功能模块

本项目面向图书馆管理员进行设计，主要设计开发管理员端的应用，其中管理员端包括书刊管理、借阅管理、会员管理、客户管理等功能模块

在需求分析中发现目前大部分中小型的图书馆都设有专门的收银台进行书刊的借阅操作，自助操作的实现需要依托还书机器、图书磁性条码感应等硬件设备，这些功能的实现需要读者自行到收银台完成借阅或零售的操作。功能框图如图 4 所示，主要功能包括书刊及其类型的添加、下架、修改和查询，实现会员和客户的管理，并支持书刊的借阅与归还。

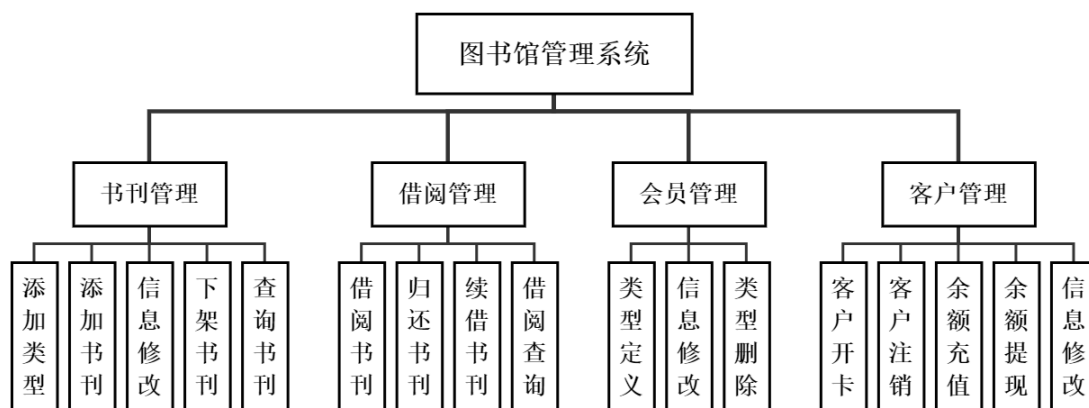


图 4 系统功能框图

书刊管理中，管理员能够为书刊添加相应的类型，如文学类、计算机类等，能够添加书刊信息，并支持信息的修改和查询操作，考虑到书刊在书店借阅和购买的过程中在客户和管理员之间产生流动，形成记录，若直接删除书刊的信息，这些记录也有可能随之消失，无法在必要时进行审查，故在此并未做真正的删除操作，而是将书刊的在馆数量清零，无法进行借阅和购买的操作，因而完成了下架的处理。

借阅管理中，读者能够在管理员端通过管理员人工完成借阅、归还和续借的操作，借阅过程中能够查询读者和书刊的信息，操作过程中将依据客户的会员信息产生一定的费用，其中形成的记录可以查询。

会员管理定义了会员的借阅期限、借阅数量、出租折扣、逾期罚款额度等信息，书店管理员可以依据书店的经营情况进行适当的调整，由于会员信息与余额相联系，所以当管理员配置时需要检查是否会影响客户，如缩短借阅时长后导致部分客户出现逾期的情况等。会员信息的删除并不是真正的删除，会员的记录将会保留，只是不会参与客户的事务中。

客户管理主要面向客户，支持客户的添加和信息修改，支持客户余额的充值和提现，确保完整性的约束，由于书店的客户会形成借阅和购买的记录，不能对其进行删除的操作，在此用注销代替，注销后的客户不再参与事务中。

4.3 界面设计与实现

依据系统的需求设计，本系统设计主要设计了登录界面和主界面，基于此可以满足管理员的相关需求。

(1) 登录界面

管理系统的登录界面如图 5 所示，主要包含封面图、版权信息、账号密码提示框，用户根据界面中的提示输入用户名、密码和验证码，即可登录系统，完成相关的业务操作，若用户的用户名、密码或验证码输入错误，系统将提示错误信息，并让用户返回重新登录。



图 5 登录界面

(2) 用户主界面

管理系统的用户主界面如图 6 所示，系统的主界面整体结构为左右结构，左侧为选项栏，展示了当前用户的所有功能模块，右侧区域主体部分为内容区，用于显示选项栏的项目内容，界面右上角显示了当前用户的用户名称，用户可点击“安全退出”退出系统。后续的功能模块将基于该网页框架进行设计和开发。

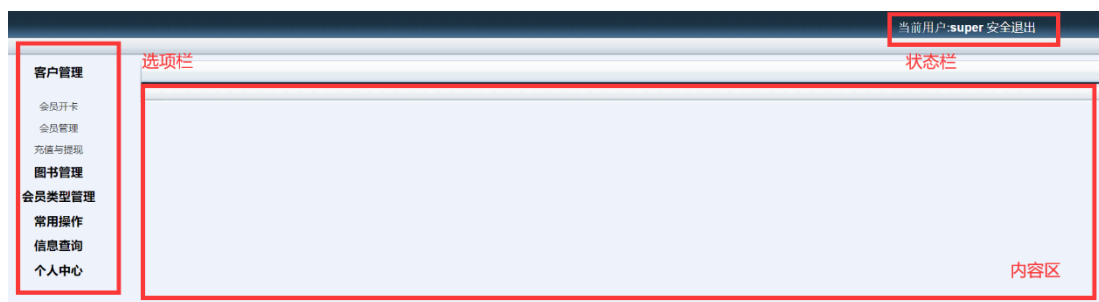


图 6 用户主界面

4.4 事务设计与实现

本系统提供了书刊管理、借阅管理、会员管理、客户管理等事务处理的支持，主要通过 MVC 思想和嵌入式 SQL 编程开发实现，还结合触发器和存储过程，不仅能够实现完成事务的流程，还能给为系统的完整性和安全性提供保障，本节将介绍系统中我所设计运用的触发器和存储过程。

触发器能够在任何用户对表的增、删、改操作时由服务器自动激活，可以实施更为复杂的检查和操作，保证系统的完整性约束。在本系统中，有在馆数量修改（借阅、归还）、余额交易记录等多个触发器，其中，我实现的是余额交易记录的触发器。

该触发器定义在客户表 Client 上，当读者进行余额的充值和提现操作时，触发器就会记录当前操作的客户信息、操作时间和操作的金额，并根据操作的金额判断操作的类型是充值还是提现，其具体的定义实现如下，主要判断新的余额值和旧的余额值之间的大小关系，如果新的值大于旧的值，则为充值，反之为提现，并将结果插入到余额表 Balance 中。

```
IF new.clientBalance > old.clientBalance THEN
    INSERT INTO `bookdb`.`balance`(`opTime`, `clientPhone`, `opAmount`, `opType`)
    VALUES (NOW(), new.clientPhone, new.clientBalance - old.clientBalance, '充值');
ELSEIF new.clientBalance < old.clientBalance THEN
    INSERT INTO `bookdb`.`balance`(`opTime`, `clientPhone`, `opAmount`, `opType`)
    VALUES (CURRENT_DATE, new.clientPhone, old.clientBalance - new.clientBalance, '
    提现');
END IF
```

系统在进行借阅、信息查询的过程中往往到多张数据表的查询和修改操作，后端往往需要编译并执行多条 SQL 语句，与数据库进行多轮的交互才能完成既定的事务，在这过程中往往需要多次连接数据库且容易出现问题，对此本系统定义了借阅、指定时间段查询数量、指定客户查询数量等多个存储过程简化这一过

程, 存储过程将复杂操作存储过程封装起来与数据库提供的事务处理结合在一起使用, 不仅能够提高数据库执行速度, 还能够重复使用, 减少数据库开发人员的工作量, 进一步保障数据库的安全性和稳定性。在本系统中, 我设计并实现了借阅、指定时间段查询数量、指定客户查询数量等多个存储过程。

存储过程 `insertBorrow` 实现读者的借阅业务, 完成借阅操作和读者费用的扣除, 通过传入读者的 ID 和书刊的 ID, 利用视图查询读者的会员信息, 得到可借阅天数和出租折扣, 利用书刊的 ID 查询书刊表 `Book`, 得到书刊的价格, 计算可得此次借阅过程的费用, 然后向借阅表 `Borrow` 中插入借阅信息, 其中根据客户的可借阅天数设定到期时间, 通过 `DATE_ADD(NOW(),INTERVAL @time DAY)` 进行计算, 利用 `CURRENT_DATE` 获取当前时间, 插入信息的同时, 与之相应的触发器会启动, 更改在馆的图书数量。最后更新客户的余额。实现过程如下。

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertBorrow`(IN `clientID` char(11), IN
`bookID` char(13))
BEGIN
    #Routine body goes here...
    set @time = 0;
    set @ratio = 0.00;
    set @price = 0.00;

    select `可借阅天数`,`出租折扣` into @time, @ratio from v_c_vip where `手机号`
    `=`clientID`;
    select bookPrice into @price from booknew where bookISBN = `bookID`;
    set @cost = @ratio * @price;

    INSERT INTO `bookdb`.`borrow`(`clientPhone`, `bookISBN`, `borrowDate`,
    `expiryDate`, `renewNum`, `borrowStatus`, `returnDate`, `deductionType`,
    `deductionAmount`) VALUES (`clientID`, `bookID`, CURRENT_DATE, DATE_ADD(NOW(),INTERVAL
    @time DAY), 0, '在借', NULL, NULL, NULL);
    UPDATE v_c_info SET clientBalance = clientBalance - @cost;
END
```

`getBookBNum` 存储过程分别对应查找规定日期范围内借阅书刊的数量, 该存储过程的实现相对较为简单, 只需要根据输入的读者 ID、起始日期和截止日期进行查询即可, 当输入的起始日期晚于截止日期, 结果将返回空值。实现过程如下。

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `getBookBNum`(IN `begin` date,IN `end`
date)
BEGIN
```

```
SELECT `借阅日期` borrowDate, `图书名称` bookName, count(*) borrowNum from  
v_b_bw_date where `借阅日期` >=begin and `借阅日期` <= end group by `图书编号`, `图书名  
称`;  
END
```

getClientBNum 存储过程分别对应查找指定读者借阅的情况，该存储过程的实现设计到了多表的查询，不仅要查询读者表 Client，还需要查询借阅表 Borrow，为了简化查询操作，系统为此定义了视图，将读者表与其他两个表进行连接，在执行存储过程时只需执行一次对视图的查询即可，实现过程如下。

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `getClientBNum`(IN `clientID` char(11))  
BEGIN  
    select * from v_c_b_bw_count where v_c_b_bw_count.clientPhone=`clientID`;  
END
```

4.5 项目架构设计与开发

本项目基于前后端分离、MVC 的设计思想并进行分层实现，如图 7 所示，项目在层次上自顶向下被划分为 servlet 层、service 层、mapper 层（DAO 层），其中 servlet 层与前端进行交互，service 层为上层提供需要的功能，mapper 层负责对接数据库，实现数据库的相关操作，向上提供经过封装的数据库操作接口。

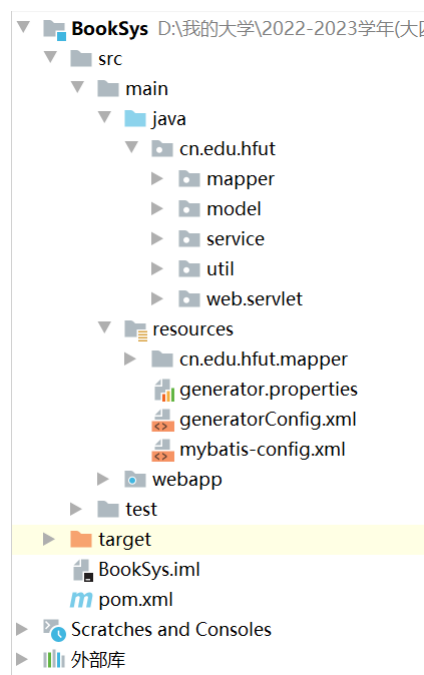


图 7 项目架构

5 项目实例验证与分析

经过前期的需求分析、数据库设计和应用程序设计和开发，本系统实现了设计阶段的功能模块，满足了需求分析阶段读者客户和管理员的使用需求。下面将介绍本系统的测试和运行过程，以此来展示本系统的功能模块的实现情况，着重体现我在项目开发过程中设计和实现的部分。

5.1 借阅管理模块测试与分析

本模块主要划分为借阅书刊、归还书刊和借阅查询，我主要负责设计和实现借阅书刊、借阅查询、部分归还操作与图书管理的功能。

(1) 借阅书刊子模块的测试

通过左侧选项栏点击“常用操作”，如图 8 所示，选择图书借阅即可进入借阅书刊的模块中。该模块主要分为上下两部分，上方用于查询读者信息，下方用于查询书刊信息。

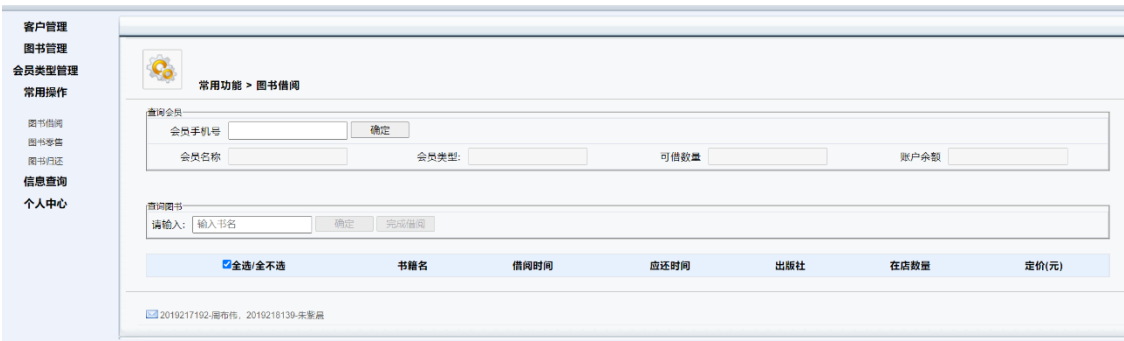


图 8 借阅书刊子模块

在查询会员中输入客户的手机号即可查到客户的信息，执行查询后会通过嵌入式 SQL 语句查询客户表 Client，若不存在则会如图 9 所示，进行提示。

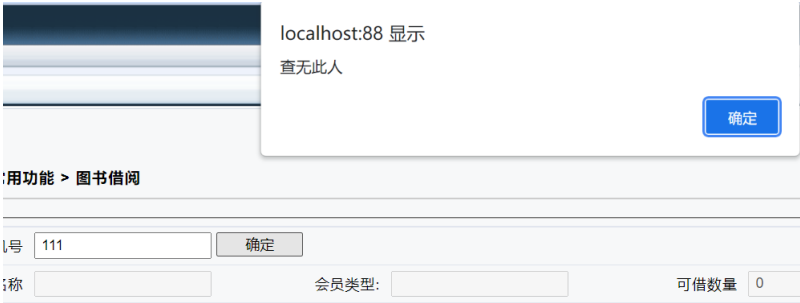


图 9 查询不存在的客户信息

当查询到了客户的会员信息，则会在下方展示，且上方的确认按钮锁定，下

方的按钮解锁，如图 10 所示。

图 10 完成会员查询

此时在下方可以进行书刊的查询，类似于客户信息的查询，若书刊不存在也会给出提示，如图 11 所示，当输入正确的书刊名称，下方便会显示书刊的信息，如图 12 所示，其中包括借阅的时间、应还的时间、在馆的数量等信息，可以进行选择借阅，最后点击“完成借阅”即可实现借阅的操作，系统会给出借阅成功的提示。

图 11 查询不存在的书刊

图 12 显示书刊的信息

(2) 借阅查询子模块的测试

通过左侧选项栏点击“信息查询”，选择借阅历史记录即可进入借阅查询的模块中。该部分支持多种条件的查询，且能实现关键字模糊查询，如图 13 所示。

若不输入关键字，默认查询条件选择“全部”，点击“搜索”后即可获取所有的客户借阅信息，包括联系方式、会员名称、书刊名称、借阅时间、归还时间、到期时间等信息，若图书尚未归还，则归还时间将为空。

查询条件: <input checked="" type="radio"/> 全部 <input type="radio"/> 已归还 <input type="radio"/> 未归还 <input type="radio"/> 最近一周需归还 请输入关键字: <input type="text"/> <input type="button" value="搜索"/>					
联系方式	会员名称	书籍名称	借阅时间	归还时间	到期时间
14790409310	张一2	数据挖掘	2022-06-16	2022-06-16	2022-07-16
18856316698	橙子	数据挖掘	2022-06-16	2022-06-17	2022-07-16
13616644404	super	活着	2022-06-17		2022-08-16
13999999900	ZZ1	活着	2022-06-17	2022-06-17	2022-08-16
13999999955	zzzz	活着	2022-06-17		2022-07-17
14790409310	张一2	活着	2022-06-16	2022-06-16	2022-07-16
18856316698	橙子	活着	2022-06-16	2022-06-17	2022-07-16

2019217192:周布伟, 2019218139:张黎晨

图 13 借阅查询子模块

切换查询条件，不输入任何关键字，点击“搜索”，结果如图 14 所示，展示了所有已经归还书刊的借阅信息，类似的，可以更换条件，查询未归还、最近一周需归还的信息。该功能本质上是在执行 SQL 语句时增加约束条件，即可实现预期的功能。

<input type="radio"/> 全部 <input checked="" type="radio"/> 已归还 <input type="radio"/> 未归还 <input type="radio"/> 最近一周需归还 请输入关键字: <input type="text"/> <input type="button" value="搜索"/>					
联系方式	会员名称	书籍名称	借阅时间	归还时间	到期时间
14790409310	张一2	数据挖掘	2022-06-16	2022-06-16	2022-07-16
18856316698	橙子	数据挖掘	2022-06-16	2022-06-17	2022-07-16
13999999900	ZZ1	活着	2022-06-17	2022-06-17	2022-08-16
14790409310	张一2	活着	2022-06-16	2022-06-16	2022-07-16
18856316698	橙子	活着	2022-06-16	2022-06-17	2022-07-16

图 14 查询已归还的借阅信息

在关键字输入框中输入读者的手机号或名称，可以通过模糊查询的方式找到用户的借阅信息，如图 16 所示，若信息不存在也会给出提示。

<input checked="" type="radio"/> 全部 <input type="radio"/> 已归还 <input type="radio"/> 未归还 <input type="radio"/> 最近一周需归还 请输入关键字: <input type="text" value="14"/> <input type="button" value="搜索"/>					
联系方式	会员名称	书籍名称	借阅时间	归还时间	
14790409310	张一2	数据挖掘	2022-06-16	2022-06-16	
14790409310	张一2	活着	2022-06-16	2022-06-16	

<input checked="" type="radio"/> 全部 <input type="radio"/> 已归还 <input type="radio"/> 未归还 <input type="radio"/> 最近一周需归还 请输入关键字: <input type="text" value="子"/> <input type="button" value="搜索"/>					
联系方式	会员名称	书籍名称	借阅时间	归还时间	
18856316698	橙子	数据挖掘	2022-06-16	2022-06-17	
18856316698	橙子	活着	2022-06-16	2022-06-17	

图 15 通过手机号或会员名称模糊查询借阅信息



图 16 查询的信息不存在

此外，还可以在通过左侧选择借阅时间段信息，输入日期，即可查询到该时间范围内的借阅信息。

请输入起始时间: 2022-06-13 请输入终止时间: 2022-06-17 搜索		
借阅时间	书籍名称	借阅数量
2022-06-16	数据挖掘	2
2022-06-17	活着	5

图 17 查询指定时间范围内的借阅信息

5.2 会员管理模块测试与分析

本模块分为类型定义、信息修改和类型删除。

(1) 类型定义模块的测试

在左侧选项栏中选择“会员类型管理”，点击“添加类型”即可进入会员类型的添加页面，输入相应的信息即可完成会员类型的添加，如图 18 所示，点击提交后，若其中的内容出现出现负数或其他违反完整性约束的情况，则会激活触发器，导致会员类型的添加失败。

会员名称	aa
押金	50.0
借阅折扣	0.05
销售折扣	1.0
借阅数量	3
续借数量	2
借阅日期	30
预约时长	7
逾期罚款额度	0.01
损坏罚款额度	1.0
丢失罚款额度	1.0
提交	

图 18 会员类型的添加

若会员添加成功，如图 19 所示，在客户开卡和信息修改处也会加入该会员名称。

Figure 19 shows a form for adding a new member type. The form includes fields for '用户名' (Username), '性别' (Gender) with radio buttons for '男' (Male) and '女' (Female), '会员类型' (Member Type) with radio buttons for 'aa', '普通会员' (Regular Member), and '黄金会员' (Gold Member), '押金' (Deposit) with a value of 50.0, and '电话号码' (Phone Number). A '提交' (Submit) button is at the bottom. The 'aa' radio button is highlighted with a red box.

图 19 新增定义的会员类型

(2) 信息修改模块的测试

在会员列表中点击修改，系统会确认是否执行修改操作，点击确定后即可进入修改页面，修改后进行提交，若不违反完整性约束，则修改成功，结果如图 20 所示。

会员名称	押金	出租折扣	销售折扣	可借阅数量	可续借次数	可借阅天数	预约保留时间	逾期扣费额度	损坏扣费额度	丢失扣费额度	修改	删除
普通会员	50.0	0.05	1.0	3	2	30	7	0.01	1.0	1.0	修改	删除
aa	100.0	0.59	0.88	7	2	30	10	1.05	1.05	1.05	修改	删除
黄金会员	200.0	0.03	0.76	3	2	60	7	0.01	1.0	1.0	修改	删除

图 20 修改会员信息

(3) 类型删除模块的测试

由于会员信息与其他用户牵连，直接删除会导致用户表的外码为空，影响正常的业务实现，故当客户中有该会员类型，如图 21 所示，则不能删除，反之，会进行删除。



图 21 含客户信息，无法删除会员类型

5.3 客户管理模块测试与分析

本模块主要分为客户开卡、客户注销、余额充值、余额提现和信息修改等子

模块。

(1) 客户开卡模块的测试

在本子模块中输入要求的信息，如图 22 所示，选择会员类型，不同的会员类型会展示不同的押金要求，且押金字段锁定，无法输入，只会随着会员类型的选择而变化，点击提交即可开卡成功，由于在一些查询中往往使用用户的手机号作为查询条件，故在此限制手机号唯一，如重复则创建失败，新添加的客户余额为 0。

用户名:	<input type="text" value="aaa"/>
性别	<input type="radio"/> 男 <input checked="" type="radio"/> 女
会员类型	<input type="radio"/> 普通会员 <input checked="" type="radio"/> 黄金会员
押金	<input type="text" value="200.0"/>
电话号码	<input type="text" value="13616644400"/>
<input type="button" value="提交"/>	

图 22 客户开卡

(2) 客户注销模块的测试

由于客户的信息与借阅、零售等记录牵连，故无法实现真正的删除操作，而是如图 23 所示，改变客户的状态，限制客户的所有操作。若客户含有借阅记录，如图 24 所示，则注销失败。

用户名	联系方式	性别	会员等级	余额	状态	借阅数量	操作
super	13616644404	男	黄金会员	-6.4	正常	1	修改 注销
zbw	13811198220	男	普通会员	-37.8	注销	0	修改 注销
ZZZZ1	13999999333	男	黄金会员	0.0	注销	0	修改 注销
ZZ1	13999999900	男	黄金会员	62.2	正常	1	修改 注销
zzzz	13999999955	男	普通会员	97.75	正常	1	修改 注销
张一2	14790409310	女	普通会员	77.18	正常	2	修改 注销
傻子	18856316698	女	普通会员	59.1	正常	2	修改 注销

图 23 客户列表

localhost:88 显示

此会员有借阅记录,注销失败

确定

图 24 含有借阅记录，注销失败

（3）余额充值与提现模块的测试

本子模块只需要填写客户的手机号和金额即可完成充值和提现，如果金额为正数则为充值，为负数即为提现，操作的过程和结果如图 25 所示。

13999999900	男	黄金会员	62.2
13999999955	男	普通会员	97.75
14790409310	女	普通会员	77.18
18856316698	女	普通会员	59.1

手机号码：

14790409310

金额：

100

提交

13999999900	男	黄金会员	62.2
13999999955	男	普通会员	97.75
14790409310	女	普通会员	177.18
18856316698	女	普通会员	59.1

图 25 充值的实现过程

当提现的金额超过客户的余额时，从实际应用场景来看，应该出错，但是在测试时能够正常提现，故需要重新修改，通过添加触发器限制提现的金额不能超过余额，回归测试后结果如图 26 所示，该操作将不被执行。

localhost:88 显示

操作失败

确定

图 26 提现金额大于余额，操作失败

（4）信息修改模块的测试

在用户列表中点击“修改”，系统会提示是否确认修改，确认即可进入修改界面，界面中可以更改客户的名称、性别和会员类型，提交后即可完成修改。

5.4 其他模块的测试

本节已经将关键模块的测试通过图文的形式予以展示，由于本系统的功能较

多，囿于篇幅，部分简单的测试点在此不再赘述，所有的测试点和测试结果如表 12 所示，测试结果表明，大部分模块能够完成预期的工作，少部分模块经过测试发现了存在的问题，经过改进后也达到了预期的效果，最后通过回归测试，所有的模块都实现了相应的功能，满足了设计要求。

表 12 系统测试结果汇总

主模块	子模块	测试点	结果
书刊管理	借阅书刊	客户查询、书籍查询、借阅操作	能够根据客户和书籍是否存在给出提示信息，实现借阅操作
	借阅查询	条件限制、关键字查找	能够依据条件和关键字实现模糊查找
会员管理	客户开卡	添加客户信息	在完整性约束下添加成功
	客户注销	删除会员信息	能够依据有无余额和借阅记录进行删除
	余额充值和提现	充值和提现余额	当提现金额超过余额时出现问题，修改后测试正确
	信息修改	修改客户信息	在完整性约束下修改成功
	图书借阅	查询客户信息	通过手机号查询，若找到则展示用户信息，若未找到则展示错误提示信息
		查询书籍信息	通过 ISBN 号查询图书信息，若找到则展示图书的信息，若未找到则展示错误信息
借阅管理	图书借阅	计算借阅金额	能根据借阅的书籍和会员信息正确计算借阅的金额
		提交借阅	能够产生借阅记录，并对用户的余额进行相应的扣除
	图书归还	查询客户信息	能够根据输入的手机号查询，查询成功则展示用户的个人信息和借阅信息
		归还操作	能够对需要归还的书籍进行选择，不需要归还的书籍进行取消
		提交归还	能够修改借阅记录中的借阅状态

6 实训总结

通过前期的需求分析、设计和测试，我们完成了图书馆管理系统的设计与开发，在此，我们针对该系统的设计进行总结和反思。

6.1 设计总结

在本次《专业实习与实训》中，我们结合前期授课所学，设计并开发实现了图书馆管理系统，完成了需求分析、数据库设计、应用程序设计和运行测试，并取得了预期的效果。本次设计开发实现的图书馆管理系统能够支持图书馆管理员和客户的多项事务，包括书刊管理、借阅管理、会员管理、客户管理等，能够形成业务闭环，满足图书馆管理员和客户的需求，提高了图书馆管理员和读者客户的业务执行效率。

本次设计的开发全过程使用 git 工具进行代码管理和记录，相关代码也已经开源至 <https://github.com/ZhoBudweiser/Book-Management-System> 中。

6.2 未来的工作

尽管本系统在功能上已经有了一定的雏形，但是还有很大的改进空间。由于项目设计时间有限，我们很难考虑到所有系统测试的用例，故在系统测试上还存在尚未发现的问题。在模块的设计上，系统在未来还可以加入更多的客户行为分析，基于客户产生的借阅与购买历史统计数据，计算统计量，如最大值、最小值、均值、方差等，便于书店管理员对书刊的数量进行有效的决策。

6.3 设计体会与反思

本次项目设计以小组的形式开展，在小组的开发过程中，我作为组长，需要确定开发计划和分工，协调开发进度，在数据库设计中，我负责设计 ER 图，调整关系模式，设计存储过程，定义部分触发器。在应用程序开发中，我负责客户管理、会员类型管理、图书借阅及其查询等模块的设计与实现。

通过本次项目设计，我锻炼了团队开发的能力，积累了团队协作的经验，在数据库的设计开发过程中，我们对数据库的规范化、完整性、安全性等要求有了更深刻的认识和体会，而且能够将知识点融会贯通。希望自己在未来能够设计开发出功能更加完备、各项性能更加优越、用户体验更加友好的软件系统。