## 图书馆后台管理系统分析与设计成果报告

班级\_\_\_\_\_\_\_\_\_\_ 学号\_\_\_\_\_\_\_\_\_\_\_ 姓名 \_\_\_\_\_\_\_\_\_\_\_ 得分 \_\_\_\_\_\_\_\_\_\_

其它组员：学号\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 姓名 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

其它组员：学号\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 姓名 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**主要工作：**

组员1： \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

组员2： \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

组员3： \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### 1选题

1．选题名称

图书馆后台管理系统分析与设计

2．选题描述

图书馆后台管理系统为图书馆的角色管理、用户管理及图书库存管理。用户可以通过此系统查询图书馆藏书，图书管理员通过此系统对馆内的藏书数量、类目进行管理，超级管理员可以拥有所有权限并可以基于角色分配权限。

#### 2用户需求.

2.1 功能需求

1. 用户功能：

个人信息注册，注册用户均为图书馆普通用户，超级管理员可以调整其权限。

修改个人信息。

查询馆内图书数量及类目。

可以查看公告。

1. 图书管理员：

具有普通用户的所有功能。

图书管理：对馆内图书的数量和类目进行增删改查。

借阅记录查询。

1. 用户管理员：

用户管理：删除普通用户。可以重置用户密码（用于找回密码）。

公告管理：可以发布、修改及查看历史公告。

1. 超级管理员：

系统的最高权限者，可以使用系统内所有功能。

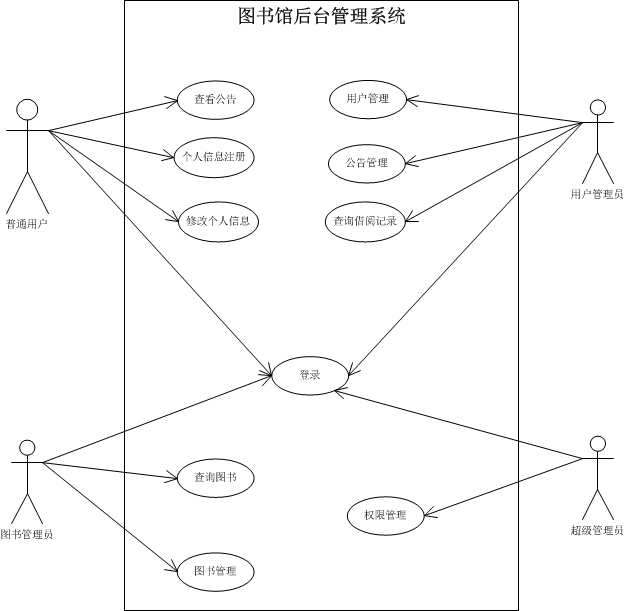
权限管理：实现系统内所有与权限相关的操作（如授权及撤销权限等）。

2.2 用例模型

概述

图书馆后台管理系统中包含的普通用户功能、用户管理员功能、图书管理员功能及超级管理员功能。

2.2.1 用例图



2.2.2用例模型说明

图书馆后台管理系统主要参与者包括普通用户、用户管理员、图书管理员、超级管理员。

主要用例有查看公告、个人信息注册、登陆、修改个人信息、用户管理、公告管理、查询借阅记录、查询图书、图书管理、权限管理等。

普通用户可以登陆、修改个人信息、查看公告并查询图书。

图书管理员除了具有普通用户的功能外，还可以对图书的类目和数量进行管理。

用户管理员除了具有普通用户的功能外，还可以对用户进行管理，删除普通用户、重置用户密码等，并对公告栏进行管理，包括发布公告、修改公告、删除公告、查询历史公告等，超级管理员可以使用系统的全部功能，同时可以进行系统的权限管理，授予普通用户权限使其成为图书管理员或用户管理员。

#### 3数据库设计

3.1 概念结构设计

E-R图及其说明



3.2 逻辑结构设计

关系模式

user(id,username,password,name,phone,regDate)

role(id,identity)

permission(id,name,resource)

user\_role(id,user\_id,role\_id)

role\_permission(id,role\_id,permission\_id)

book(id,name,author,publish,pubdate,ISBN,imagePath,borrowNum,stock,summary)

tag(id,name)

tag\_book(id,tag\_id,book\_id)

borrow(id,book\_id,user\_id,borrowDate,day,isOverdue,isReturn)

notice(id,title,content,releaseTime)

3.3 数据库实现SQL代码

基于MySQL数据库给出全部的SQL脚本

CREATE DATABASE IF NOT EXISTS `bookSystem` DEFAULT CHARACTER SET utf8;

USE `bookSystem`;

DROP TABLE IF EXISTS `book`;

CREATE TABLE `book` (

`id` int(9) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`name` varchar(255) NOT NULL COMMENT '图书名',

`author` varchar(64) DEFAULT NULL COMMENT '作者',

`publisher` varchar(64) DEFAULT NULL COMMENT '出版社',

`pubdate` varchar(32) DEFAULT NULL COMMENT '出版时间',

`isbn` varchar(16) DEFAULT NULL COMMENT '国际标准书号',

`imagePath` varchar(128) DEFAULT NULL COMMENT '封面',

`borrowNum` int(4) NOT NULL DEFAULT '0' COMMENT '借阅次数',

`stock` int(4) NOT NULL DEFAULT '1' COMMENT '库存量',

`summary` text COMMENT '图书介绍',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=30155732 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `borrow`;

CREATE TABLE `borrow` (

`id` int(4) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`book\_id` int(4) NOT NULL COMMENT '图书ID',

`user\_id` int(4) NOT NULL COMMENT '借阅者ID',

`borrowDate` date NOT NULL COMMENT '借阅时间',

`day` int(4) NOT NULL COMMENT '借阅天数',

`isOverdue` tinyint(1) NOT NULL DEFAULT '0' COMMENT '是否超期',

`isReturn` tinyint(1) NOT NULL DEFAULT '0' COMMENT '时候归还',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `category`;

CREATE TABLE `category` (

`id` int(4) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`name` varchar(64) NOT NULL COMMENT '类型名',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `notice`;

CREATE TABLE `notice` (

`id` int(4) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`title` varchar(128) NOT NULL COMMENT '标题',

`content` tinytext NOT NULL COMMENT '内容',

`releaseTime` date NOT NULL COMMENT '发布时间',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=10 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `permission`;

CREATE TABLE `permission` (

`id` int(4) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`name` varchar(32) NOT NULL COMMENT '权限名称',

`resource` varchar(64) NOT NULL COMMENT '权限',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=35 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `role`;

CREATE TABLE `role` (

`id` int(4) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`identity` varchar(32) NOT NULL COMMENT '身份',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `role\_permission`;

CREATE TABLE `role\_permission` (

`id` int(4) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`role\_id` int(4) NOT NULL COMMENT '角色表主键',

`permission\_id` int(4) NOT NULL COMMENT '权限表主键',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=20 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `tag`;

CREATE TABLE `tag` (

`id` int(4) NOT NULL AUTO\_INCREMENT COMMENT '标签主键',

`name` varchar(64) NOT NULL COMMENT '标签名',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=145 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `tagbook`;

CREATE TABLE `tagbook` (

`id` int(4) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`tag\_id` int(4) NOT NULL COMMENT '标签主键',

`book\_id` int(9) NOT NULL COMMENT '书本主键',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=2902 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `user`;

CREATE TABLE `user` (

`id` int(4) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`username` varchar(30) NOT NULL COMMENT '用户账号',

`password` varchar(128) NOT NULL COMMENT '用户密码',

`name` varchar(32) NOT NULL COMMENT '姓名',

`email` varchar(64) NOT NULL COMMENT '邮箱',

`phone` char(11) NOT NULL COMMENT '电话',

`regDate` date NOT NULL COMMENT '注册日期',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=14 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `user\_role`;

CREATE TABLE `user\_role` (

`id` int(4) NOT NULL AUTO\_INCREMENT COMMENT '主键',

`user\_id` int(4) NOT NULL COMMENT '用户表主键',

`role\_id` int(4) NOT NULL COMMENT '角色表主键',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

#### 4类设计

4.1 主要系统类、功能及分类

4.1.1模型层类

永久层主要包括BookMapper，NoticeMapper，PermissionMapper，RoleMapper，TagMapper，UserMapper等接口及映射文件。

业务层主要包括BookService(Impl)，NoticeService(Impl)，PermissionService(Impl)，RoleService(Impl)，TagService(Impl),UserService(Impl)等接口（实现类）。

4.1.2控制层类

主要包括BookController，BorrowController，HomeController，LoginController， NoticeController,PermissionController,RegisterController,RoleController,TagController,UserController等类。这些类的职责是接受View层的请求，与模型层交互完成业务功能后，再将处理结果反馈给View层。

4.1.3视图类（JSP页）

主要包括addBook.jsp,bookDetail.jsp,bookList.jsp,tagList.jsp,borrowList.jsp,

latestNotice.jsp,noticeList.jsp,releaseNotice.jsp,addPermissionToRole.jsp,permissionList.jsp,rolePermisstion.jsp,addUserToRole.jsp,roleList.jsp,roleMember.jsp,home.jsp,nopermission.jsp,welcome.jsp,exception.jsp,login.jsp,register.jsp等。

4.2 类图

1. 用户管理模块



1. 角色管理模块



1. 图书管理模块



1. 公告管理模块



1. 借阅管理模块

#### 

#### 5业务流程顺序图

5.1.1 注册用例

5.1.2 登录用例



5.1.3添加角色用例



5.1.4 历史公告删除用例



5.1.5查看图书详细信息用例



5.1.6修改用户基本信息用例

