

数据库系统实验报告

课程名称: 数据库系统原理 实验项目: 实验五 数据库管理系统设计
学生姓名: 庄毅非 专业: 软件工程 学号: 3200105872
指导老师: 苗晓晔 实验日期: 2022 年 4 月 9 日

一、实验目的和要求

掌握使用 JDBC 或 ODBC 进行数据库管理系统开发的能力

二、实验内容和要求

1. 设计图书管理系统的 ER 模型
2. 设计对应的实体和联系模式
3. 使用 JDBC 或 ODBC 实现一个简单的图书管理系统, 能够对图书、借书证和图书借阅行为进行管理

三、开发环境

1. OS: MacOS Monterey 12.3.1
2. 数据库管理系统: Mysql
3. 实现框架:
 - a) 后端: go/gorm
 - b) 前端: nextjs.mui

四、程序用法和实验步骤

4.1 确定需要建立的数据表

4.1.1: Book 表

图书馆中有许多种类的藏书, 每本书本身的属性有书号, 类别, 书名, 出版社, 年份, 作者, 价格, 总藏书量, 库存。图书管理员需要在图书上执行借阅和归还操作。

4.1.2: Card 表

图书馆使用借书证管理用户的借阅信息, 借书证上的信息包括卡号、姓名、单位、类别, 图书管理员需要能够添加借书证, 以及通过借书证号查询借书证。

4.1.3 Borrow 表

系统使用 borrow 表管理借阅行为, borrow 表的信息包括书号, 借书证号, 借期和换期。图书管理员需要能够执行归还和借阅操作。

4.2 建立关系模型

最终建立的数据概念模型如下

对象名称	属性
书	书编号、类别、标题、出版社、年份、作者、价格、总藏书量、现有库存
借书证	借书证号、持卡者姓名、部门、类别
管理员	用户名、密码
借阅信息	自增编号, 借书证号。书编号、借阅时间、归还时间

4.3 在 ODBC 中建立对应模式

```

1 Book (Bno,Category,Title,Press,Year,Author,Price,Total,Stock)
2 Card (Cno, Name, Department, Type)
3 User (Username,Password)
4 Borrow (ID, CardCno,BookBno,Borrow_date, Return_date)

```

4.4 各模块功能设计

模块名称	功能描述
管理员登陆	输入账号、密码登陆进系统，或者返回登陆失败，并显示失败原因
管理员注册	进行管理员账户的注册
图书入库	1. 单本入库 2. 批量入库
图书查询	要求对书的类别，书名，出版社，年份（区间），作者，价格，价格（区间）进行查询
借书	1. 输入借书证号 显示该借书证所有已经借的书籍 2. 输入书号 如果书籍还有库存，则借阅成功，否则输出书籍没有库存，输出最近一次归还的时间
还书	1. 输入借书证卡号 显示该借书证所有已经借的书籍 2. 输入书号 如果该书籍在已经借阅的书籍列表中，那么还书成功，否则输出出错信息
借书证管理	增加或删除借书证

4.5 搭建数据库

我选用的 ODBC 是 gorm，为了建立数据表，我在 book_backend/model 文件夹中建立了 Book.go, Card.go, User.go, Borrow.go 四个表文件，内容如下

Book.go

```

1 package model
2 type Book struct {
3     Bno      string `json:"bno" gorm:"primaryKey;size:10"`
4     Category string `json:"category" gorm:"size:40"`
5     Title     string `json:"title" gorm:"size:40"`
6     Press     string `json:"press" gorm:"size:40"`
7     Year      int    `json:"year"`
8     Author    string `json:"author" gorm:"size:10"`
9     Price     float64 `json:"price" gorm:"size:10"`
10    Total     int    `json:"total"`
11    Stock     int    `json:"stock"`
12 }

```

Card.go

```

1 package model
2 type Card struct {
3     Cno      string `json:"cno" gorm:"primaryKey;size:7"`
4     Name     string `json:"name" gorm:"size:10"`
5     Department string `json:"department" gorm:"size:40"`
6     Type     string `json:"type" gorm:"size:1;check type in ('T','S')"`
7 }

```

User.go

```

1 package model
2 type User struct {
3     Username string `json:"username" gorm:"primaryKey;column:username"`
4     Password string `json:"password" gorm:"NOT NULL"`
5 }

```

Borrow.go

```
1 package model
2
3 import "time"
4
5 type Borrow struct {
6     ID          int    `json:"id" gorm:"primaryKey;autoIncrement"`
7     CardCno      string `json:"cno" gorm:"references:Cno;column:cno;primaryKey;size:7"`
8     BookBno      string `json:"bno" gorm:"references:Bno;column:bno;primaryKey;size:8"`
9     Borrow_date string `json:"borrow_date"`
10    Return_date string `json:"return_date"`
11    Book         Book   // 外键约束
12    Card         Card   //外键约束
13 }
14
```

4.6 后端启动

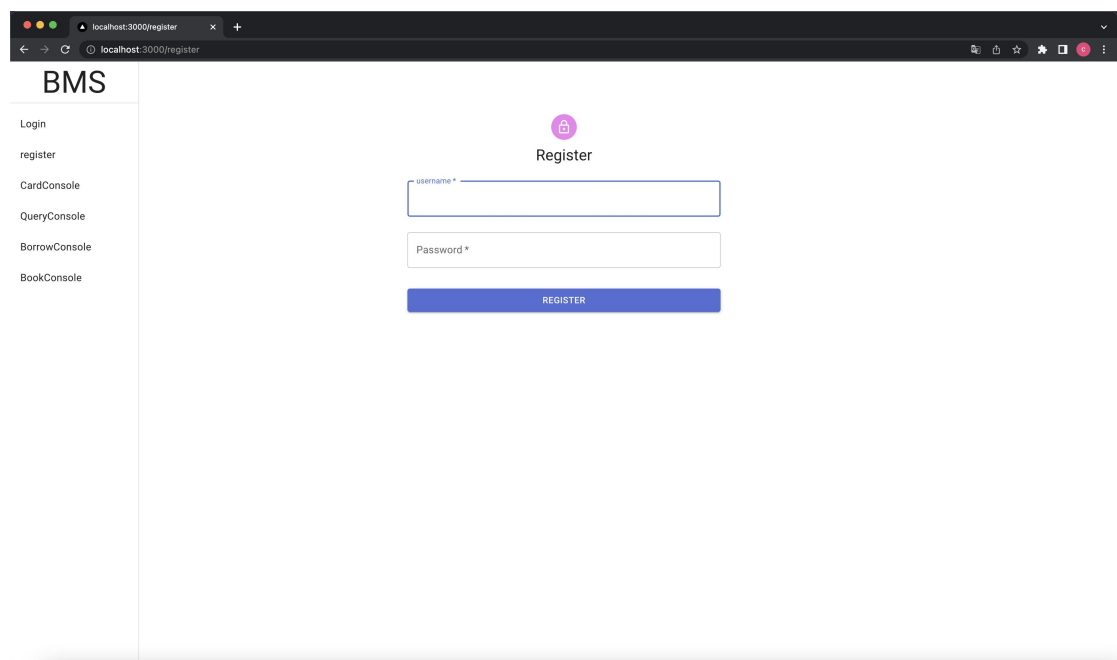
我选用的是 gin 网络框架搭建后端，这是一个 golang 的网络微框架，具有速度快，搭建路由简单等特性。在项目中，我首先通过 gorm 连接到数据库，之后在 gin 引擎添加各表功能对应的 api，并监听 9000 端口，前端只需要访问 9000 端口即可获取所需信息。

4.7 前端启动

我选用的是 nextjs 进行前端搭建，这是一个基于 React 和 typescript 的前端框架，能够智能读取页信息，快速进行网页搭建，具有灵活性高，性能强大的特点。

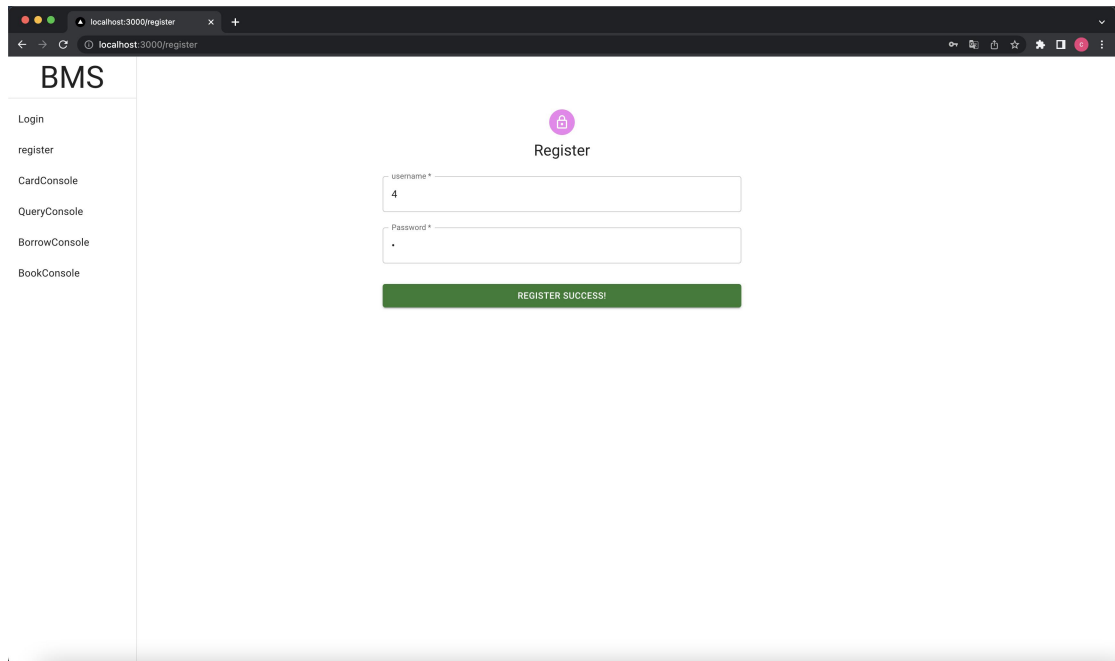
五、实验结果和分析

5.1 用户注册

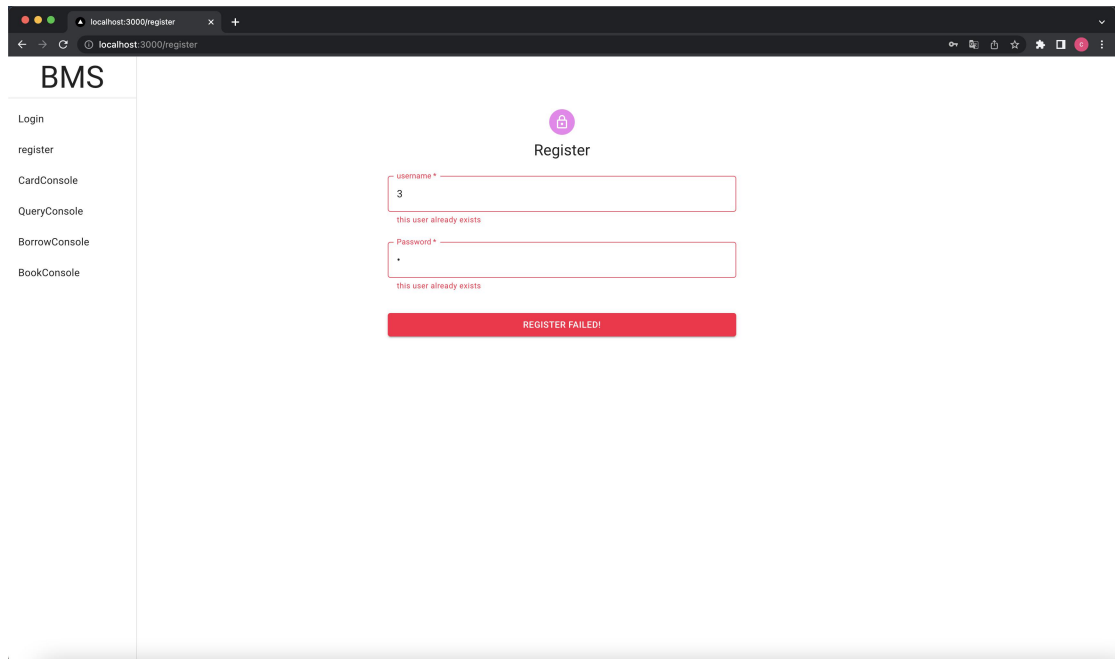


用户在输入账号和密码之后，点击 Register 按钮就可以进行注册，如果注册成功和失败都会回显信息。

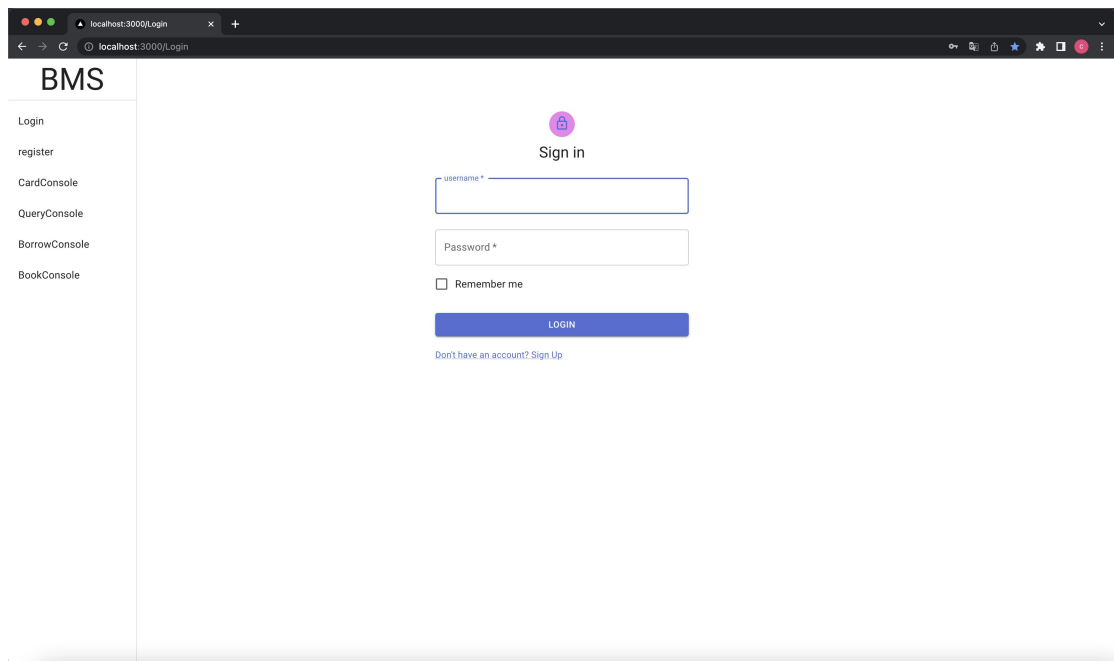
注册成功：



注册失败:



5.2 用户登录



和注册一样，无论成功失败都会回显对应信息，这里限于篇幅不再展示。

值得注意的是，为了确保安全性，在登陆成功之后，后端会给前端返回一个 jwt 协议生成的 token，添加到 cookie 中，用户只有持有上述 token 才能进行借书证管理，借阅信息管理等操作，token 有效期为 10 分钟。

5.3 图书管理

5.3.1 查询所有书籍的信息

Information of all books

bno	Author	Category	Title	Press	Total	Stock	Price	Year
456	jfk	history	UK	national press	123	111	111.11	2001
999	zyf	math	newTitle	press	0	0	0	2010
1010	zyf	math	newTitle	press	123	124	2001	2021
7777	newtown	physics	principle	UK national press	100	99	1000	1600
7778	ntown	physics	principle	UK national press	100	99	1000	1600
7779	newtown	physics	principle	UK national press	100	99	1000	1600
9999	zyf	physics	abook	press	0	0	0	0
book_no_1	xxx	computer_science	computer architechure	xxx	2	2	90	2003
book_no_2	xxx	physics_science	physics theroy	xxx	50	50	190	2010
book_no_3	xxx	unknown	physics theroy	xxx	50	50	190	2010

5.3.2 更新书籍信息或加入新的书籍（图书入库）

如果书籍是已经存在的，那么将其更新，否则视为插入新的书籍。

Adding/update your book here

bno *
12345

author *
2

category *
3

title *
3

press *
123

total *
3

stock *
3

year *
3

price *
3

ADDING SUCCESS!

5.3.3 图书查询

相等查询，这里以上一步插入的数据为例

book query!

constrains

Type	searchConstrain *
bno ▼	12345

select type for querying book

QUERY

bno	Author	Category	Title	Press	Total	Stock	Price	Year
12345	2	3	3	123	3	3	3	3

区间查询，查询 2001~2021 年之间的书籍

book query!

constrains

Type	minYear *	maxYear *
year ▼	2001	2021

select type for querying book

QUERY

bno	Author	Category	Title	Press	Total	Stock	Price	Year
1010	zyf	math	newTitle	press	123	124	2001	2021
456	jfk	history	UK	national press	123	111	111.11	2001
999	zyf	math	newTitle	press	0	0	0	2010
book_no_1	xxx	computer_science	computer architechure	xxx	2	2	90	2003
book_no_2	xxx	physics_science	physics theroy	xxx	50	50	190	2010
book_no_3	xxx	unknown	physics theroy	xxx	50	50	190	2010

5.4 图书借阅

借阅前

query by cno

input cno here *
1214

QUERY BORROWS				
BorrowID	CardID(cno)	BookID(bno)	borrow_date	return_date
1	1214	1010	2001-10-03	2010-10-03
3	1214	999	2001-10-03	2002-10-08
4	1214	1010	2001-10-03	2002-10-08
5	1214	1010	2001-10-03	2002-10-07
6	1214	1010	2001-10-03	2000-10-03
7	1214	1010	2001-10-03	2022-03-27
8	1214	12345	2022-04-09	2022-04-09

借阅新书籍

Input cno and bno to borrow book

cno *

1214

bno *

12345

BORROW BOOK

借阅之后

BorrowID	CardID(cno)	BookID(bno)	borrow_date	return_date
1	1214	1010	2001-10-03	2010-10-03
3	1214	999	2001-10-03	2002-10-08
4	1214	1010	2001-10-03	2002-10-08
5	1214	1010	2001-10-03	2002-10-07
6	1214	1010	2001-10-03	2000-10-03
7	1214	1010	2001-10-03	2022-03-27
8	1214	12345	2022-04-09	2022-04-09
9	1214	12345	2022-04-09	2022-04-09
10	1214	12345	2022-04-09	not returned yet

归还书籍

Input cno and bno to return book

cno *

1214

bno *

12345

RETURN BOOK

归还后

BorrowID	CardID(cno)	BookID(bno)	borrow_date	return_date
1	1214	1010	2001-10-03	2010-10-03
3	1214	999	2001-10-03	2002-10-08
4	1214	1010	2001-10-03	2002-10-08
5	1214	1010	2001-10-03	2002-10-07
6	1214	1010	2001-10-03	2000-10-03
7	1214	1010	2001-10-03	2022-03-27
8	1214	12345	2022-04-09	2022-04-09
9	1214	12345	2022-04-09	2022-04-09
10	1214	12345	2022-04-09	2022-04-09

如果书籍借阅完毕，输出最近一次归还信息

Input cno and bno to borrow book

cno *

12314

bno *

12345

BORROW BOOK

书籍已经被借阅完毕，最近归还信息是

BorrowID	CardID(cno)	BookID(bno)	borrow_date	return_date
8	1214	12345	2022-04-09	2022-04-09

5.5 借书证管理

5.5.1 添加借书证

Adding your card here

cno *	username *	department *	Type
9999	db	dbb	T

select type of card

ADDING SUCCESS!

5.5.2 删除借书证

Input cno to delete card

cno *

9999

DELETE CARD

六、实验心得

这一次大作业对我而言有点难度。在技术选型方面，本来选择的是 C++ 中的 qt 框架进行可视化界面的搭建，但是由于 qt creator 工具下载频繁失败，最终选择使用网页进行 gui 搭建。后端由于之前学习过 golang 语言，所以选择 gin+gorm 框架，前端选择的是 Nextjs in typescript + material ui。网页端开发相对比较简单，安全性方面使用 jwt 进行加强。

这是我第一次搭建一个这样的简易项目，因此在搭建后端框架的时候遇到了一些问题，比如 gorm 框架对于如何实现外键约束，给出的文档比较简略，后来通过 stackoverflow 才终于找到对应的解答。这次项目也促使我更加深入了解 orm 框架内部机理，锻炼了我阅读文档的能力。

在搭建前端的时候，刚开始是通过 React + ReactRouter 进行搭建，但是后来在切换路由的时候网页频繁报错，发现是忘记 push history 的原因，为了避免频繁的书写路由声明，于是切换到了集成化程度更高的 nextjs 进行开发。

在网络方面，一开始因为跨域问题，前端一直拿不到后端的数据，后来在询问同学之后，在后端加入对应的跨域配置中间件才解决了问题。

在实现整个项目之后，我对于数据库的了解有了一定的加深，未来还需要继续学习，编写出性能更好的代码。