

# 二哥的图书馆系统

## LiBrother

张哲恺 冼藏越洋 徐遥 王辰 张嘉恒 朱奕

2015 级 ACM 班

上海交通大学

2015 年 11 月 23 日

## 目录

<b>1 综述</b>	<b>3</b>
<b>2 项目简介</b>	<b>3</b>
2.1 项目名称 . . . . .	3
2.2 项目的设计框架 . . . . .	3
2.3 支持的功能 . . . . .	4
2.4 主要模块 . . . . .	4
2.5 主要功能的图形界面 . . . . .	4
<b>3 模块介绍</b>	<b>4</b>
3.1 common . . . . .	4
3.1.1 简介 . . . . .	4
3.1.2 具体文件和内容 . . . . .	4
3.2 LiBrotherClient . . . . .	5
3.2.1 简介 . . . . .	5
3.2.2 具体的设计思路 . . . . .	5
3.2.3 一些典型的窗口 . . . . .	5
3.3 LiBrotherClientBackend . . . . .	5
3.3.1 简介 . . . . .	5

目录	2
3.3.2 具体的类和实现的功能	6
3.4 LiBrotherServer	6
3.4.1 简介	6
3.4.2 具体功能的实现	6
3.5 liblog	6
3.5.1 简介	6
3.5.2 具体功能	6
<b>4 设计上的亮点</b>	<b>7</b>
4.1 服务端和客户端的分离	7
4.2 良好的扩展性	7
4.3 强大的搜索功能	7
4.4 漂亮的图形界面	7
4.5 可靠的加密算法	7
<b>5 界面化的精心设计和效果</b>	<b>8</b>
5.1 提示信息的弹出	8
5.2 界面转换不会丢失信息	8
5.3 不同用户等级进入不同界面	8
5.4 登录不影响搜索	8
5.5 主页面的展示	9
5.6 用户注册页面的展示	9
5.7 用户主页面的展示	10
5.8 用户借阅图书页面的展示	10
<b>6 组员的分工合作</b>	<b>11</b>
<b>7 还可以改进的地方</b>	<b>11</b>
7.1 工程构建方面	11
7.2 页面设计方面	11
<b>8 小结与感想</b>	<b>11</b>

## 1 综述

随着社会信息量的与日俱增，在计算机日益普及的今天，图书馆管理也需利用计算机作为平台，开发一套行之有效的图书管理系统，这对提高学校图书管理信息化、网络化的水平具有重要的现实意义。

而对于刚刚接触 C++ 程序设计和面向对象编程的我们，图书馆工程无疑是一个很好的锻炼机会，有助于提高我们的代码能力和对面向对象编程的理解。

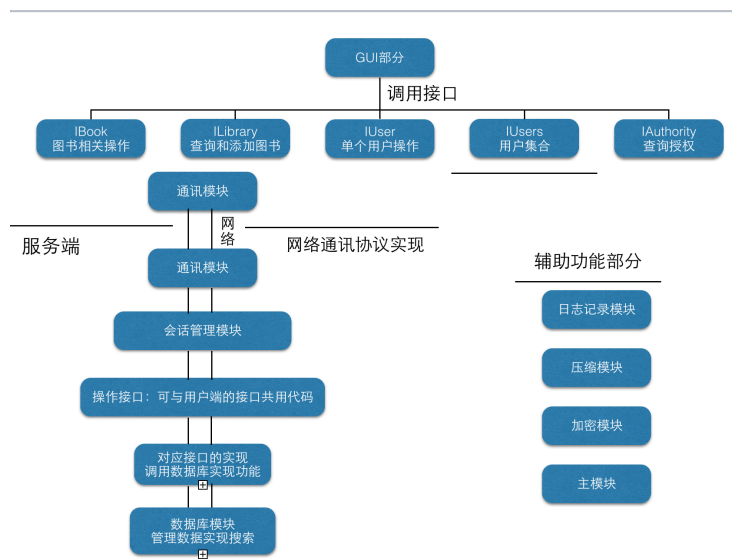
我们组所设计的图书馆系统: 二哥的图书馆就很好地实现了图书馆的绝大部分功能，并且效率较高，管理方便。

## 2 项目简介

### 2.1 项目名称

LiBrother 二哥的图书馆

### 2.2 项目的设计框架



## 2.3 支持的功能

实现了书籍管理，用户管理，保存数据和搜索在内的所有功能。

## 2.4 主要模块

具体分为 common, LiBrotherClient, LiBrotherClientBackend, LiBrotherServer, LiBrother, liblog 等模块。

## 2.5 主要功能的图形界面

利用 Qt 实现了用户，管理员和图书馆的界面化，并且支持大多数功能。

# 3 模块介绍

## 3.1 common

### 3.1.1 简介

包含了前端和后端共有的接口和一些结构体的声明，供其他模块使用。

### 3.1.2 具体文件和内容

#### **common\_type.h**

定义了图书信息的结构体 TBookBasicInfo

定义了用户信息的结构体 TUserBasicInfo

定义了借阅图书的结构体 TBorrowInfo

定义了图书管理员的结构体 TAuthorization

#### **common\_interface.h**

定义了错误的枚举类型 Errtype

定义了错误信息的结构体 TErrInfo

定义了所有接口的基类 IAbstract

定义了一个专用于接口的数组 IFvector

#### **function\_inferface.h**

定义了管理单本图书的图书接口 IBook

定义了管理所有图书的图书馆接口 `ILibrary`

定义了管理单个用户的用户接口 `IUSer`

定义了管理全部用户的管理员接口 `IUerManager`

定义了用于创建其他接口的类工厂 `ILibClassFactory`

## 3.2 LiBrotherClient

### 3.2.1 简介

利用 *Qt* 而实现的图书馆系统的界面化，实现了绝大多数的功能。

### 3.2.2 具体的设计思路

模仿真实的图书馆操作系统，利用窗口、按钮和槽函数，配合整个工程所能实现的功能，设计美观的界面。

### 3.2.3 一些典型的窗口

#### **mainwindow**

整个界面的主窗口，从这里用户或者管理员根据需要进入其他界面。

#### **managermain**

管理员主窗口，只有管理员才有资格进入，并且提供了管理员可能的操作。

#### **usermain**

用户的主窗口，提供了已经登录用户的可能的操作。

## 3.3 LiBrotherClientBackend

### 3.3.1 简介

作为后端，具体实现了前端所调用的函数，链接了客户端和服务端。

### 3.3.2 具体的类和实现的功能

#### CBook

对于单本图书，实现了构建图书基本信息，获得图书信息，删除图书，获取出借情况等函数。

#### CUser

对于用户，实现了获取、修改、设定个人信息，借阅图书和归还图书等函数。

#### CUserManager

对于图书管理员，实现了获取用户信息，插入用户等函数。

#### CLibrary

对于图书馆，实现了利用图书基本信息查找图书，插入图书等函数。

#### CAuthManager

对于当前的用户，实现了登录，注销，修改密码，获取权限等级等函数。

## 3.4 LiBrotherServer

### 3.4.1 简介

主要完成了服务端网络模块，会话管理和数据库的实现。

### 3.4.2 具体功能的实现

主要可以支持对图书的插入、编辑，对用户和管理员的请求给出回应，对用户的密码进行加密，对图书和用户数据进行压缩等具体功能。

## 3.5 liblog

### 3.5.1 简介

主要完成了日志系统，供图书管理员使用来更好地管理图书。

### 3.5.2 具体功能

实现了记录日志，保存日志，对管理员输入的日志信息的判断和处理的函数。

## 4 设计上的亮点

### 4.1 服务端和客户端的分离

在二哥的图书馆系统中，我们组很好地实现了服务器端和用户端的分离。服务端负责设置整个系统所要完成的功能和操作，而客户端就是利用服务器端所提供的功能。利用二者的分离，使我们的图书馆系统更加方便管理和使用。增加或删除功能直接在服务端完成，用户端没有必要知道这个过程是怎样完成的，而只需要学会去使用就可以了，这充分考虑了用户的需求，方便了用户使用，更加接近现实中所使用的真实的图书馆系统。其还有较高的使用价值，能够允许有大量用户同时使用。此外，我们的前端和后端均支持跨平台，效果很不错。

### 4.2 良好的扩展性

我们将整个图书馆系统进行合理的模块划分，不同的模块代表不同的部分，各模块之间可以相互调用，但是不必知道对方具体的实现过程，函数若有改动，只需要所在模块改动就好，不会改动很多代码，方便修改。其次，在各个模块，不同对象的操作被封装在不同的类中，也方便了代码的改动。

### 4.3 强大的搜索功能

在搜索图书时，我们利用的搜索算法非常高效，支持中英文搜索和其他图书特性搜索，甚至在支持模糊搜索。

### 4.4 漂亮的图形界面

我们花费较长的时间在界面设计上，利用窗口的嵌套和槽函数，做出了非常美观的图书馆界面。不仅能够支持大多数功能，而且所有的界面全真模仿了真正的图书馆系统，带给用户亲切感，也方便了用户使用。

### 4.5 可靠的加密算法

我们在通讯上支持 TLS 加密，密码在传输过程中使用了 SHA-256 算法加密，进入数据库通过 bcrypt 加密，保证足够的安全性。

## 5 界面化的精心设计和效果

### 5.1 提示信息的弹出

我们的界面对用户非常友好，大部分的操作调用函数都对返回值是否为 *true* 做了相应的判断，如果 *false* 就说明有用户输入不合法，导致无法完成操作，我们都会通过 *QMessage* 模块，以弹出警告窗口的方式告知用户出错的地方是哪里，以及需要如何去修改。

### 5.2 界面转换不会丢失信息

为了保证在界面与界面之间的衔接能够不丢失所需要的信息，我们添加了 *public* 的函数 *getBookID*，这样就可以将我们 *ListWidget* 中的选中的信息（通常是书本的 ID 号）直接传给之后对该书的操作信息中，从而实现跨界面的信息衔接。值得强调的是，为了实现这一操作，我们精心设计了 *setItemData* 这一函数，使得 *listWidget* 中的每一个 *item* 在 *addItem* 之处就保存了 *data* 信息（也就是书本的 ID 号）。

### 5.3 不同用户等级进入不同界面

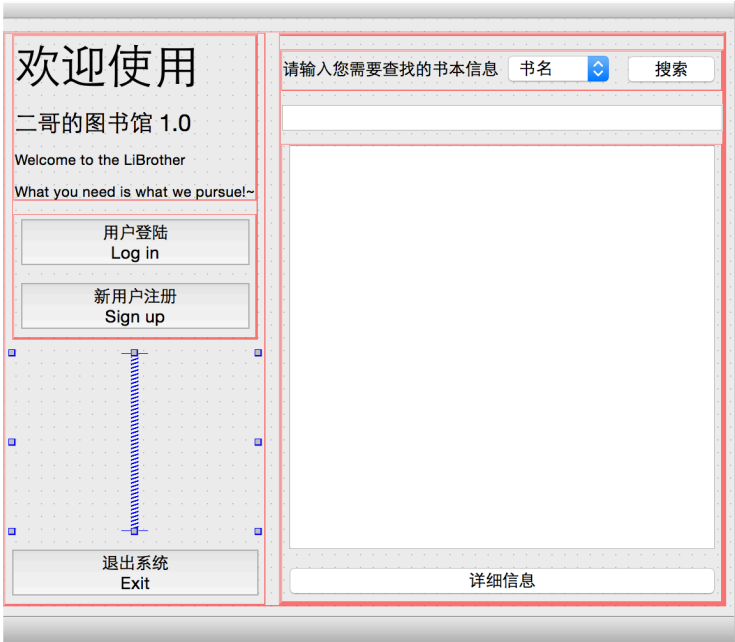
通过对后端用户权限等级的调用，我们实现了不同的等级所能进入的不同界面，其中初等管理员界面和高等管理员界面是决定着他们是否能修改图书信息以及用户信息的关键因素。

### 5.4 登录不影响搜索

书目信息板块也就是 *bookdata.ui* 的单独设立是实现主界面书本搜索和读者登录后搜索。两个界面一个核心，而且是公用的，并且不会出现矛盾，只要通过 *getcurrentuser* 函数的调用，判断返回值真假就能知道有没有用户登录，能否进行借书操作，使用方便简洁。



5.5 主页面的展示



5.6 用户注册页面的展示



5.7 用户主页面的展示

XXXXXXXX		您好!	
用户ID:		123456789	
您已借阅了以下书本:			
		继续借书	
		归还此书	
		管理图书 (初级管理员)	
		管理用户 (高级管理员)	
		刷新	
		修改密码	
		登出 Log out	

5.8 用户借阅图书页面的展示

书名		库存:	
作者:		我要借阅	退回
出版社:			
ISBN号:			
简介:			

## 6 组员的分工合作

具体的分工合作如下：

**张哲恺** 负责整个工程的框架构建，公共接口和结构体的声明，数据库的具体实现和协调各个部分。

**冼藏越洋** 负责操作数据库。

**徐 遥** 负责客户端和服务端的通讯协议。

**王 辰** 负责整个工程的界面设计和链接。以及最后的 presentation。

**张嘉恒** 负责协助王辰完成工程的界面设计和链接，以及最后 PPT 制作和 REPORT 的撰写。

**朱 奕** 负责最后对工程的补充和完善。

## 7 还可以改进的地方

### 7.1 工程构建方面

在工程构架方面，我们的数据库直接调用了现成的数据库。如果有时间的话，我们可以自己写一个数据库，专门用于我们的图书馆系统，这样效率会更高。此外，我们的系统不支持图书的批量读入，之后可以进一步完善。

### 7.2 页面设计方面

在页面设计中，管理员搜索书本并对选中书目进行编辑的搜索界面与用户再借书的搜索界面是同一个，虽然进行了 authLevel 的判断，但是用户搜索的时候会看到管理员的修改按钮，比较奇怪，可以进一步改进。

## 8 小结与感想

在设计这个项目时，我们组从多方面进行考量，模仿真实的图书馆系统的设计和界面。我们将一个大工程进行划分，分成具体的模块，例如客户端和服务端以及 GUI。并且在组内也进行了良好的分工。组长张哲恺作为有工程经验的大腿，为大家构建了图书馆的整体框架和一些公共接口的定义，是整个工程的基础和核心。

在做工程的过程中，我们也遇到了各种各样的困难，包括代码的重复提交，一些 bug 难以修复，以及 Windows 系统和 mac 系统不能兼容等，但是大家始终团结一致，没有放弃，尽最大的努力去完成它，这也让我们体会到了团结的力量。

对于最终的成果，二哥的图书馆 (LiBrother)，大家还是比较满意的。它实现了我们最初设想的绝大多数功能，完成了所有的基础要求并且实现了一些选做功能，界面设计也完全接近真实的图书馆操作系统，美观完整。但是仍然存在值得改进的地方，可以更加简洁，更加人性化。

最后，感谢助教提供了这次机会，能让大家进一步理解 C++ 程序设计面向对象编程的特点，进一步提高代码能力。更重要的是，我们从这次工程中学会了合作与责任，这将对我们之后的人生产生深远的影响！