

# 第七小组报告

## 数据层

结构体部分

书籍部分的结构体，文件存放分别是Book.dat, BookNameIndex.dat, BookIdIndex.dat

```
struct Book
{
    string isbn;    //书籍的ISBN编号
    string name;    //书名
    string author;  //作者
    string press;   //出版社
    string category; //专业类别
    long pubdate;   //出版时间
    int borrowNum;  //已借阅数量
    int num;        //图书总数
    float price;    //价格
};

struct BookNameIndex
{
    char name[avglen]; //书名
    long index; //在book.dat文件中地址
};

struct BookIdIndex
{
    int id; //每一本书的编号
    long index; //在book.dat文件中地址
    bool isBorrowed; //是否被借阅（false代表未被借阅）
};
```

用户部分结构体，文件存放分别是User.dat, UserIdIndex

```
struct User
{
    char id[user_avglen]; //学工号
    char realName[user_avglen]; //真实姓名
    char major[user_avglen]; //专业
    char grade[user_avglen]; //年纪
    char pwd[user_avglen]; //密码
    char phone[user_avglen]; //手机
    char email[user_avglen]; //邮箱
    char sex[1]; //性别
    int age; //年龄
};

struct UserIdIndex
{
    int id; //用户id
    long index; //在user.dat文件中地址
};
```

借阅日志部分，文件存放在borrowInfo.dat

```

struct BorrowInfo
{
    char userId[30]; //借阅人ID
    char userName[30]; //借阅人姓名
    char bookId[30]; //出借书id
    char bookName[30]; //出借书名
    char borrowDate[20]; //借阅日期
    char backDate[20]; //应还日期
    int isback; //判断是否归还, 1表示归还, 0表示未归还
};

void ReadInfo(); //显示最早借阅历史(正序显示)
void ReverseReadInfo(); //显示最新借阅历史(倒序显示)
void SaveInfo(string userId, string userName, string bookId, string bookName); //
//接受参数, 创建新的借阅记录
void ReadInfoById(char* TmpId); //按照用户id查看借阅历史
void ReadEndInfo(); //显示超期图书

```

留言板部分, 文件存放在board.dat

```

const int MAXNameLength = 100; //最大姓名长度
const int MAXIdLength = 100; //最大id长度
const int MAXcontentLength = 256; //最大留言内容长度
const int MAXtimeLength = 8; //时间长度
typedef struct {
    char name[MAXNameLength]; //留言人姓名
    char id[MAXIdLength]; //留言人id
    char content[MAXcontentLength]; //留言内容
    char time[MAXtimeLength]; //留言时间
    int delFlag; //懒惰删除标志, 0表示保留, 1表示删除
}BoardMessage;

bool ReverseReadBoardMessage(); //显示最新留言(倒序显示留言)
bool ReadBoardMessage(); //显示最早留言(正序显示留言)
bool AddBoardMessage(); //添加留言
bool ReadBoardMessageById(char* TmpId); //按照用户id查看发言内容
bool DeleteBoardMessage(); //删除指定留言
bool RealDeleteBoardMessage(); //进行文件扫描 删除标记30%以上时 执行磁盘删除 减少空间占用

```

公告部分, 文件保存在notice.dat

```

const int sizetitle = 20;
const int sizeavglen = 100;
const int sizetimes = 21;
struct Notice
{
    int id; //公告编号
    char title[sizetitle]; //公告标题
    char content[sizeavglen]; //公告内容
    char times[sizetimes]; //公告时间
};

bool ReadNotice(); //公告文件读出
bool ReadNotice(int id); //单条公告文件读出
bool WriteNotice(struct Notice); //公告文件写入
bool RewriteNotice(string site, struct Notice); //公告文件重写
bool WriteNotice(string site, struct Notice); //公告文件写入
bool DeleteNotice(int id); //单条公告删除

```

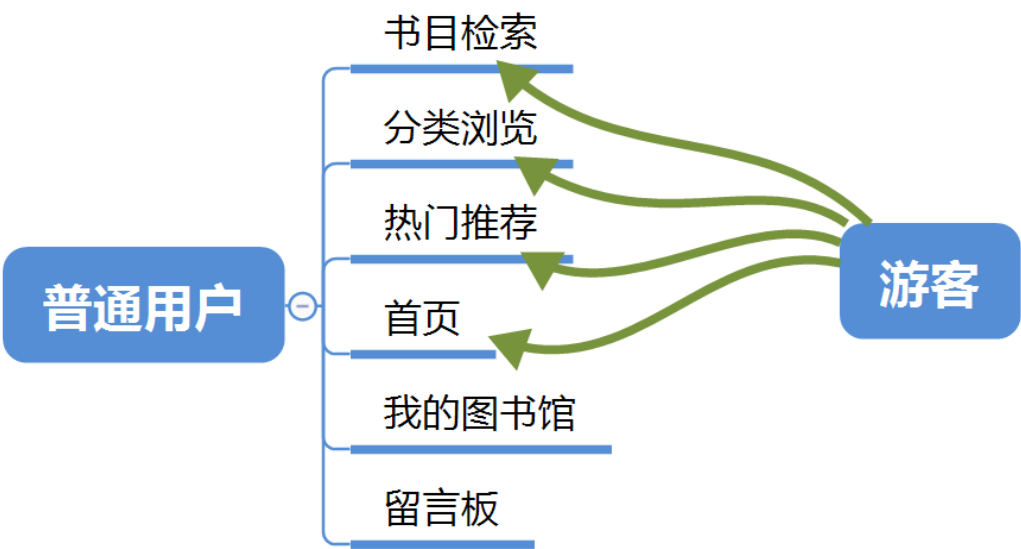
```
bool ModifyNotice(struct Notice); //单条公告修改
bool CopyNotice(); //公告文件备份
```

关于文件存储方式，查询算法

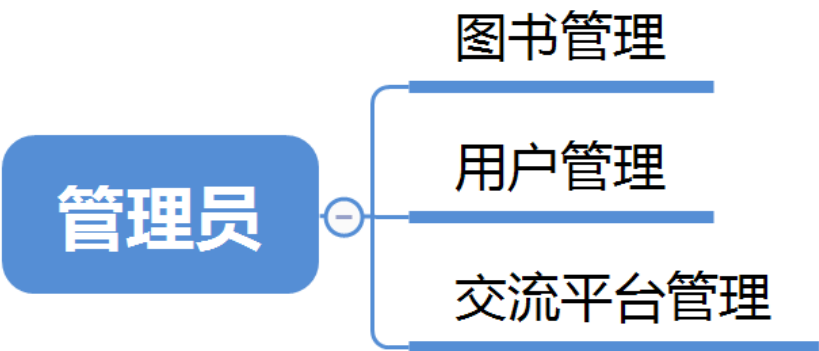
由张根同学讲解

# 功能层

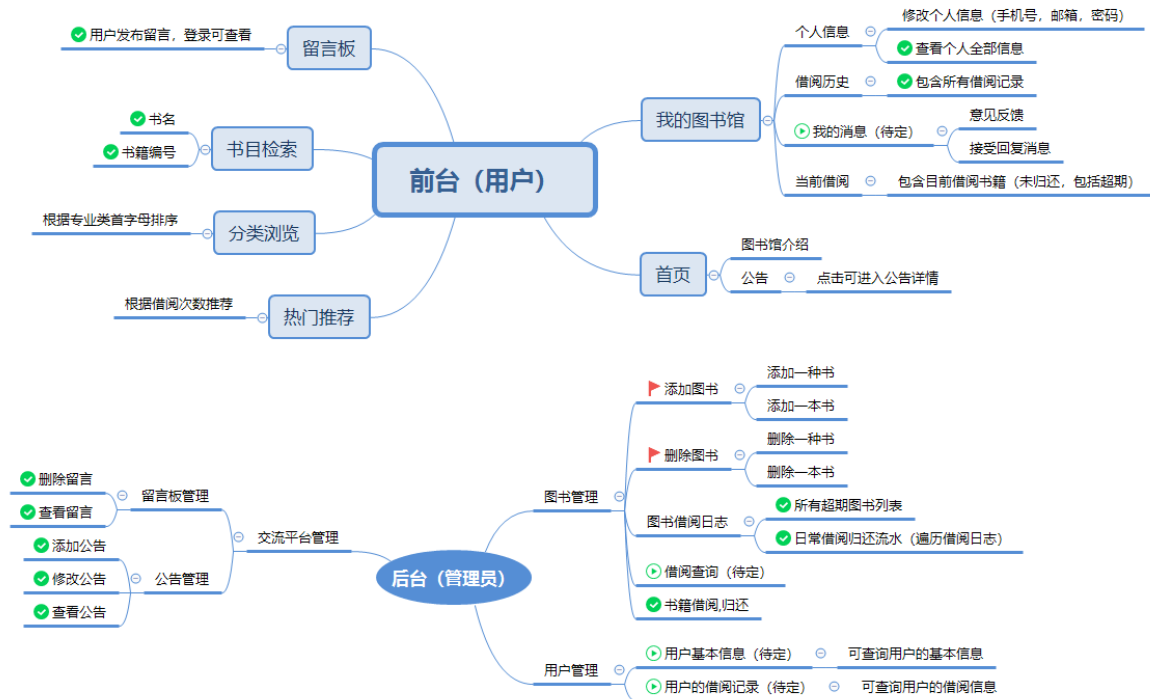
系统前台主用例图



系统后台主用例图

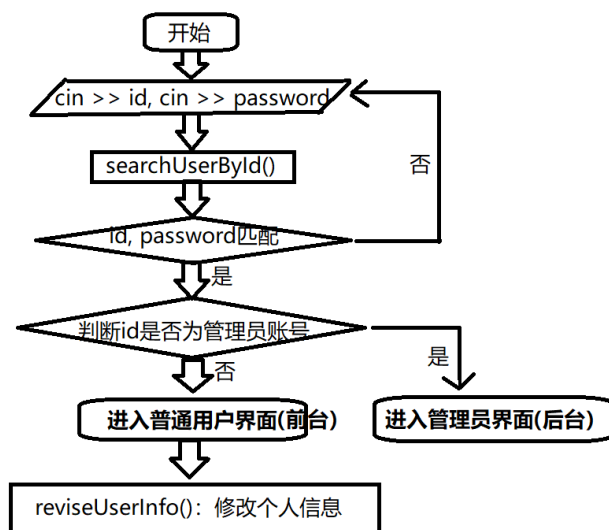


详细功能图

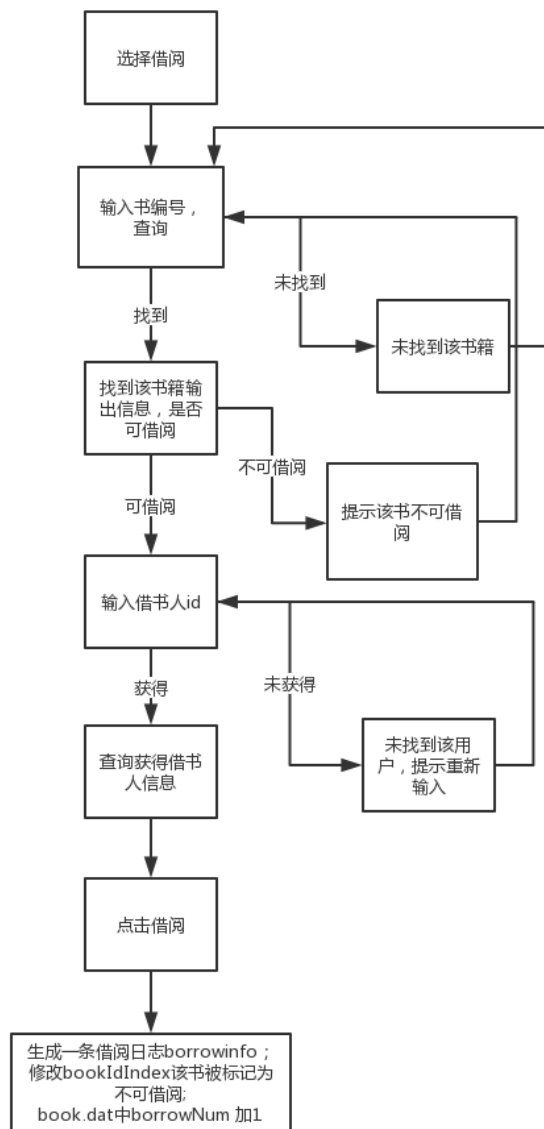


## 部分功能程序框图

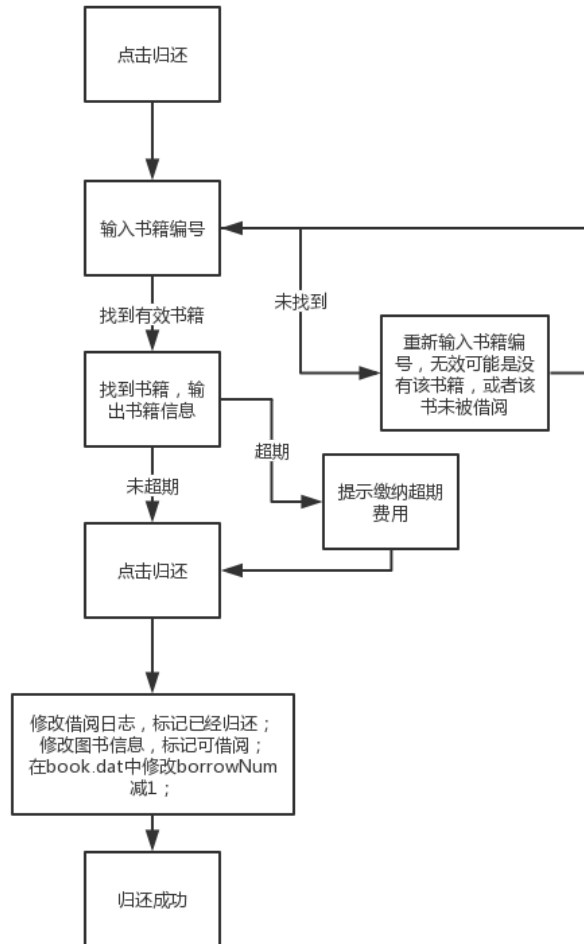
### 登录



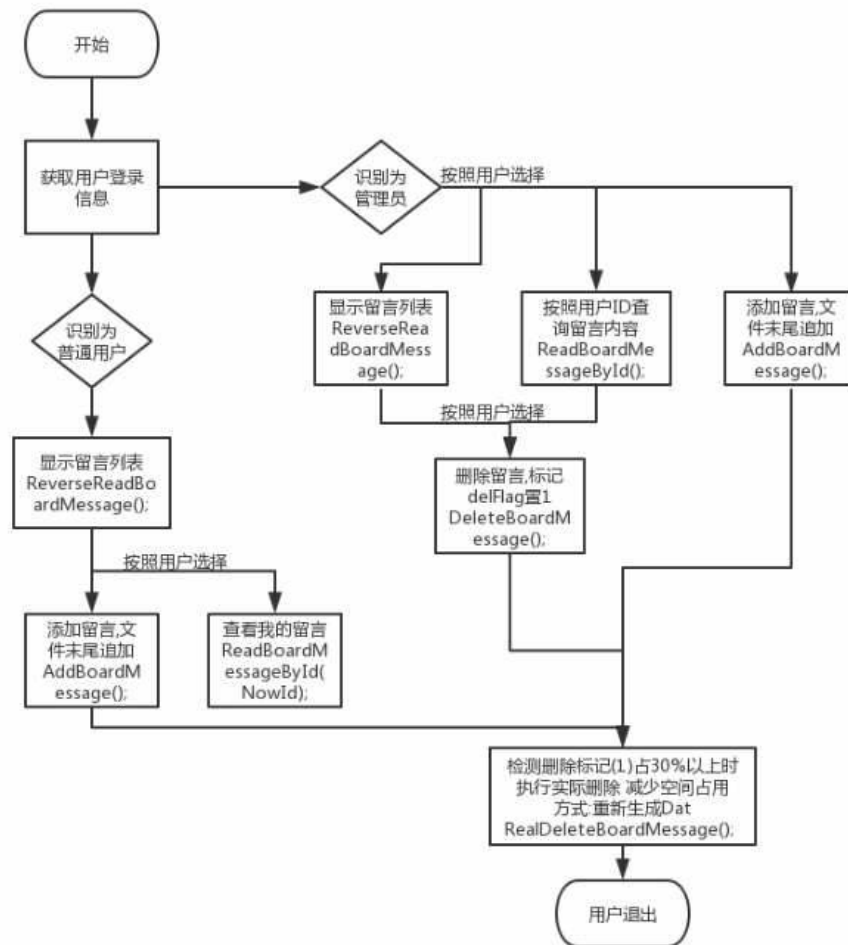
### 借阅



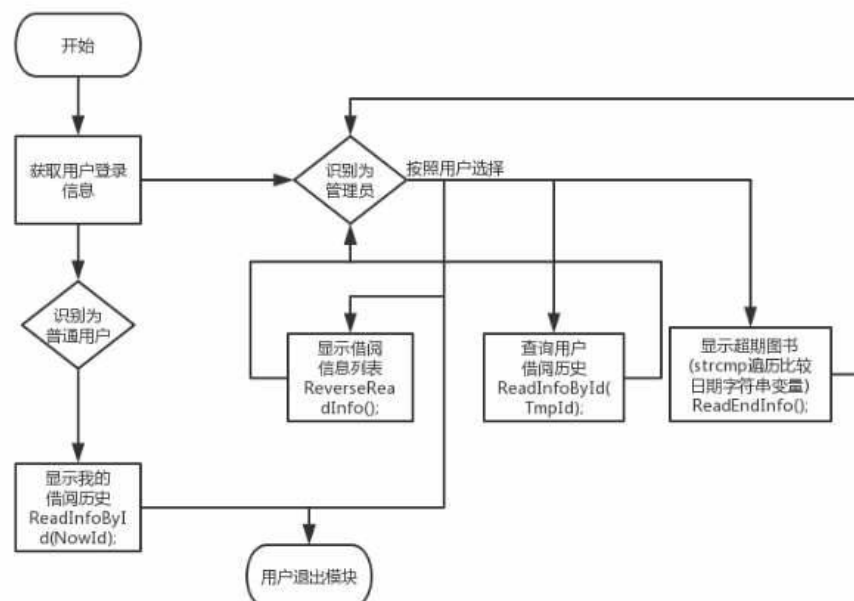
归还



留言板部分



借阅日志部分



# UI层

---

使用qt开发前台界面

详情见文件夹中qt框架图片