

JAVA 高级编程大作业

影院管理

姓名：赵书彬 班级：软测 1802 学号：201873030 成绩：_____

姓名：刘晨欣 班级：网安 1801 学号：201892100 成绩：_____

大连理工大学

Dalian University of Technology

目录

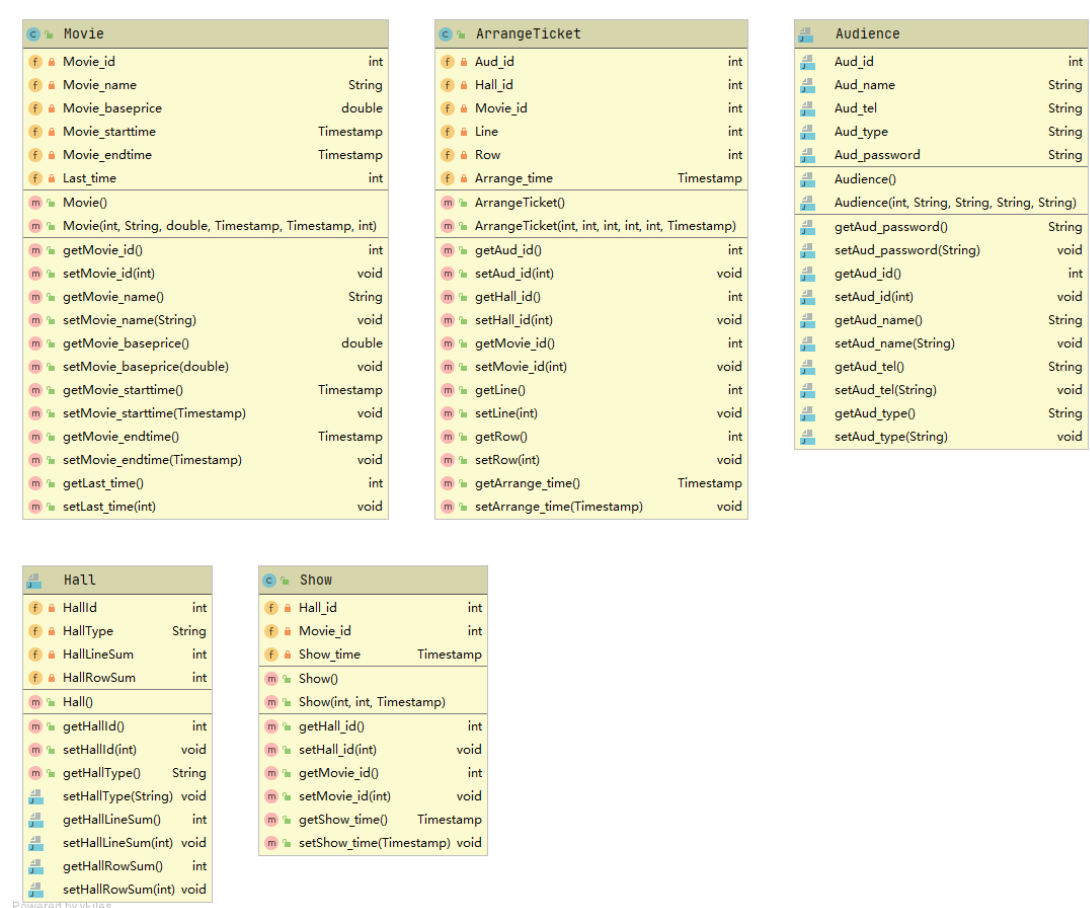
1	系统主要的类图.....	4
1.1	实体类(Entity)类.....	4
1.2	DAO(DBOpeartion)类.....	5
1.3	服务(Service)类.....	6
1.4	用户功能抽象(User)类.....	7
1.5	权限-用户(Authority)类.....	8
1.6	主界面(Menu)类.....	8
2	系统数据库设计.....	9
2.1	各数据库表以及功能.....	9
2.2	电影信息表.....	10
2.3	影厅信息表.....	10
2.4	观众信息表.....	10
2.5	放映信息表.....	11
2.6	订票信息表.....	11
3	主要功能设计.....	12
3.1	功能框图.....	12
3.2	功能说明.....	12
4	主要代码设计.....	13
4.1	总体架构描述.....	13
4.2	数据访问层.....	14
4.3	业务逻辑层.....	16
4.4	控制层.....	17
4.5	表示层.....	19
5	实现界面.....	20
5.1	程序流程图.....	20
5.2	主界面.....	20
5.2.1	登陆.....	21
5.2.2	注册.....	21
5.2.3	忘记密码.....	22
5.3	管理员服务.....	23
5.3.1	查询信息.....	24
5.3.2	影片管理.....	25
5.3.3	放映管理.....	26
5.3.4	修改账户信息.....	27
5.4	普通用户服务.....	28
5.4.1	订票.....	28
5.4.2	退票.....	31
5.4.3	改签.....	32
5.4.4	查询上映电影&个人购票记录.....	32
5.5	退出.....	33
6	对课程的体会.....	33
6.1	赵书彬.....	33

6.1.1	心得.....	33
6.1.2	分工.....	34
6.2	刘晨欣.....	34
6.2.1	心得.....	34
6.2.2	分工.....	34

1 系统主要的类图

1.1 实体类(Entity)类

实体类 UML 图如下：



主要有电影类，影厅类，观众类，影票类，放映类。实体类的属性与数据库各个表的属性严格对应，以形成逻辑上以及数据上的对应关系。同时每个类提供了一些获取和修改数据的基础操作以便后续处理。

1. 电影类（Movie）保存电影信息,包括电影编号，上映时间，下架时间，类型，电影基本价格。

2. 影厅类（Hall）保存影厅信息，包括影厅编号，影厅类型，该放映厅的总

行数和总列数。

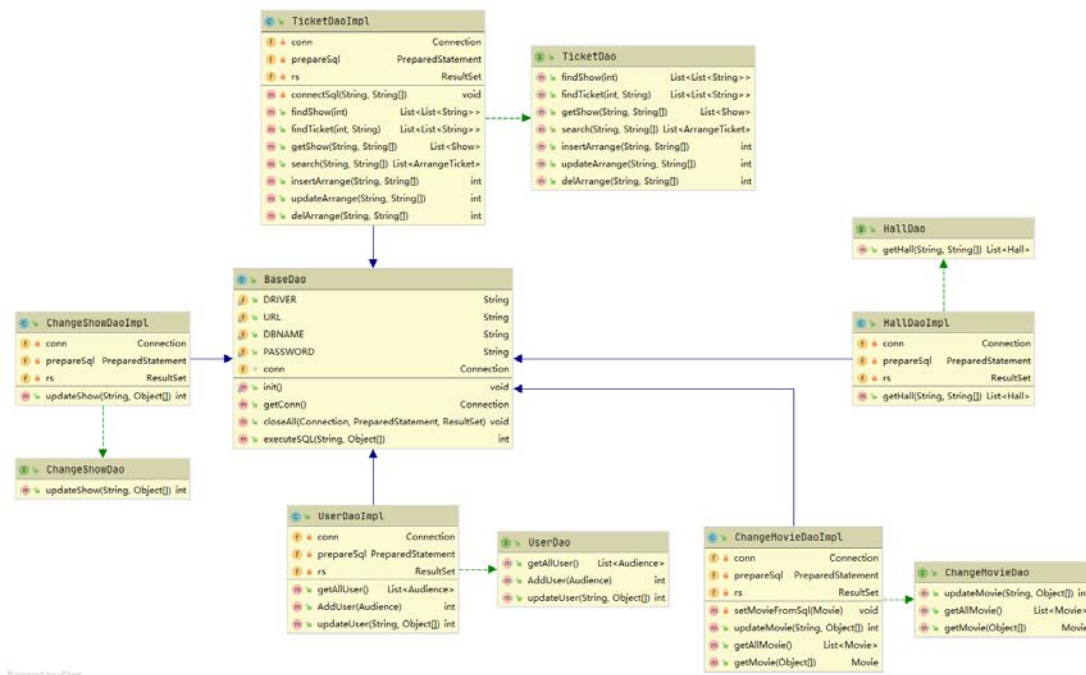
3. 观众类 (Audience) 保存用户信息, 包括用户编号, 用户名, 密码, 电话, 用户类型。

4. 影票类 (ArrangeTicket) 保存订票信息, 包括电影编号, 影厅编号, 放映时间, 座位排数, 座位列数。

5. 放映类 (Show) 保存放映信息, 包括影厅编号, 电影编号, 放映时间。

1.2 DAO(DBOpeartion)类

DAO 类及接口 UML 图如下:



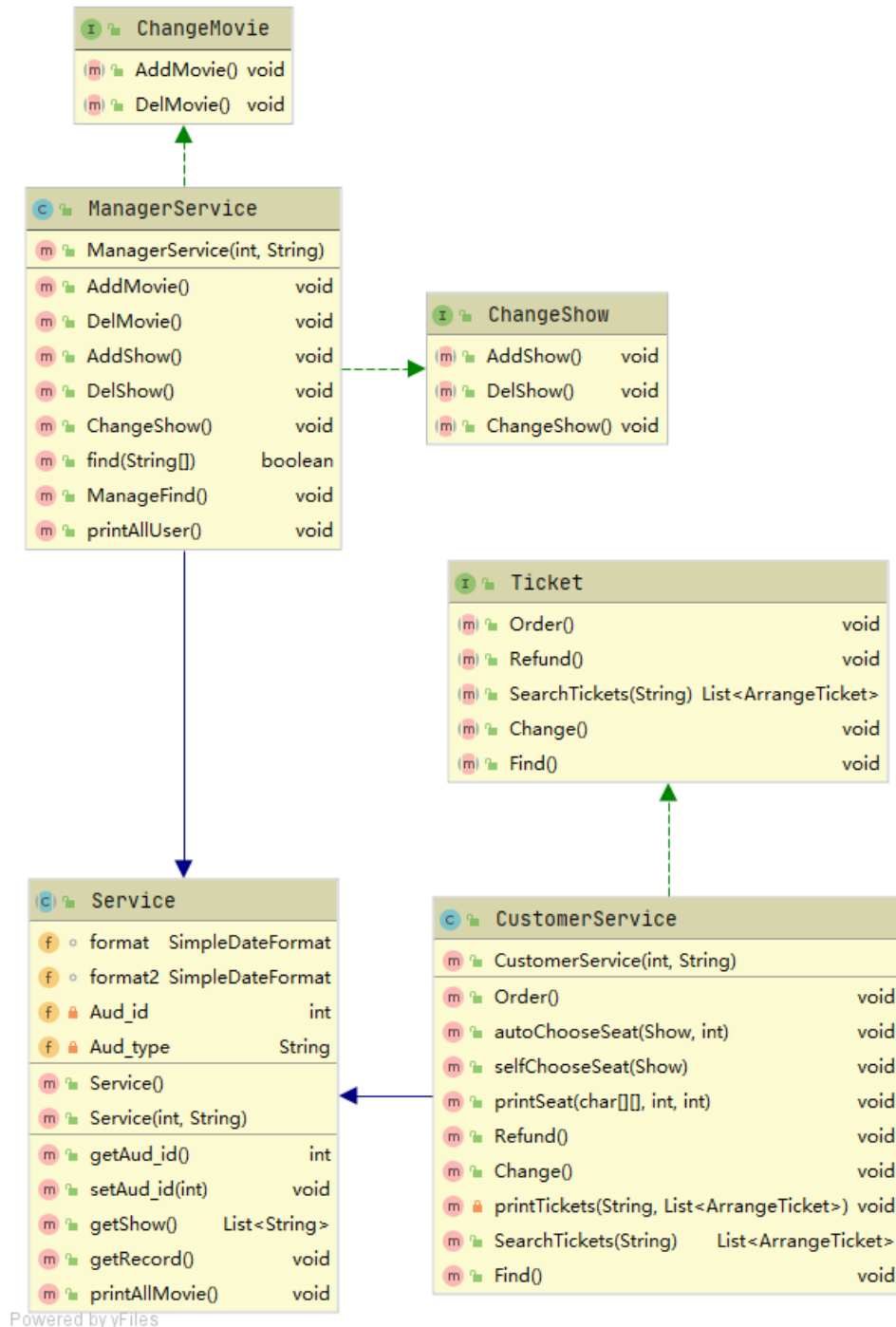
DBOpeartion 包中的每一个 DaoImpl 类都继承自 BaseDao 类, 而且都实现各自预定义的接口 Dao 以获得不同的具体操作, 并且为下一层预留了直接调用进行 sql 操作的接口 (该函数仅为传递作用)。

1. BaseDao 类提供基础的数据库连接操作: 得到数据库连接, 释放资源, 执行 SQL 语句 (增、删、改操作通过插入、更新或删除实现, 不可查。查通过其他接口自行使用 sql 接口实现)。
2. HallDaoImpl 类实现 HallDao 接口, 负责对放映厅对象的操作: 返回大厅列表。
3. UserDaoImpl 类实现 UserDao 接口, 负责对用户对象的操作: 获取用户列表, 更新用户表。
4. ChangeMovieDaoImpl 类实现 ChangeMovieDao 接口, 负责对电影对象进行操作: 返回特定条件电影, 返回电影列表, 更新电影表。
5. ChangeShowDaoImpl 类实现 ChangeShowDao 接口, 负责对放映对象进行操作: 更新放映信息。
6. TicketDaoImpl 类实现 TicketDao 接口, 负责对票务的操作: 返回当前时间结果集, 通过用户 id 或与权限寻找定票情况, 根据特定的条件查找场

次，增删改所定订的电影票。

1.3 服务 (Service) 类

服务类以及接口 UML 图如下：

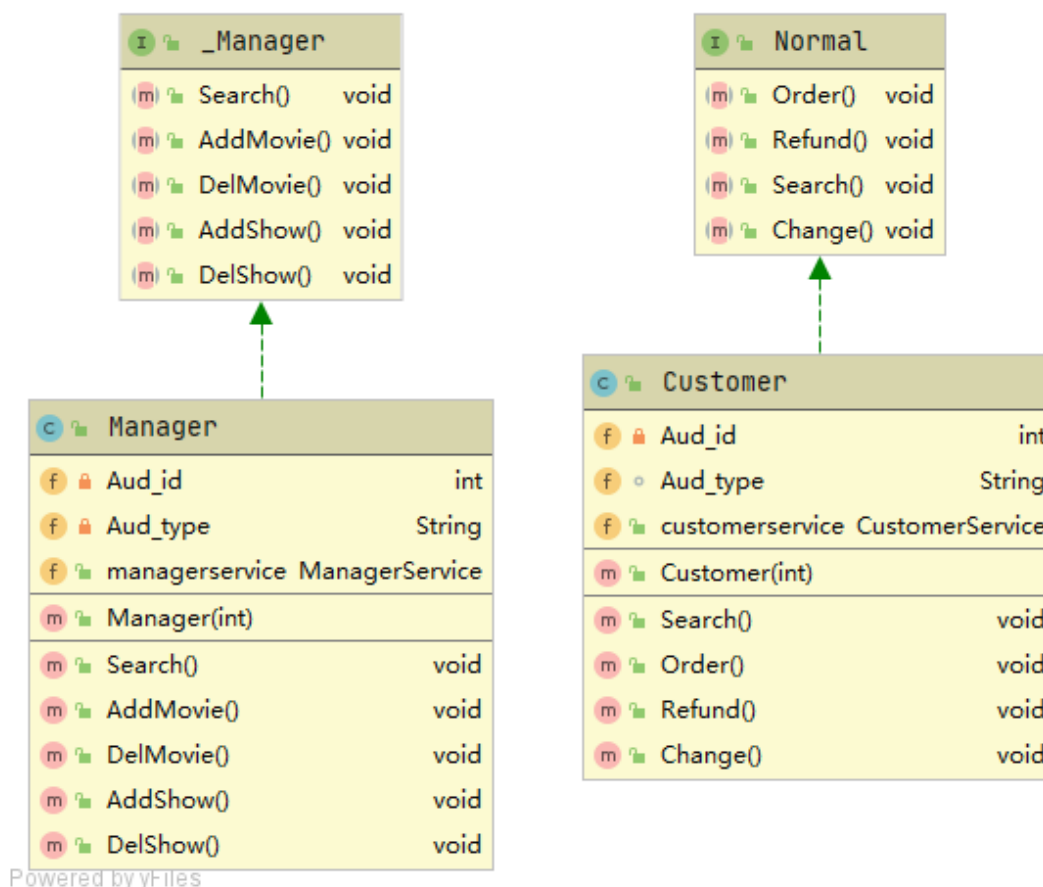


服务包中含有服务类 **Service**，管理员服务类 **MangerService**，普通用户服务类 **CustomerService** 以及对应接口。

1. 服务类 **Service** 为影院服务基类，管理员服务类和用户服务类都继承于服务类。实现：查询电影、查询当前放映电影、多态实现查询订票记录。
2. 管理员服务类 **MangerService** 实现更新放映 **ChangeShow** 和更新电影 **ChangeMovie** 两个接口，实现对电影和当前放映电影的增删改查以及对用户列表的查询。
3. 普通用户服务类 **CustomerService** 实现票务 **Ticket** 接口，实现购票的自动选座和手动选座（包含了对座位信息的输出）、退票、改票、查询上映电影以及购票记录。

1.4 用户功能抽象（User）类

用户功能抽象类 UML 类图如下：

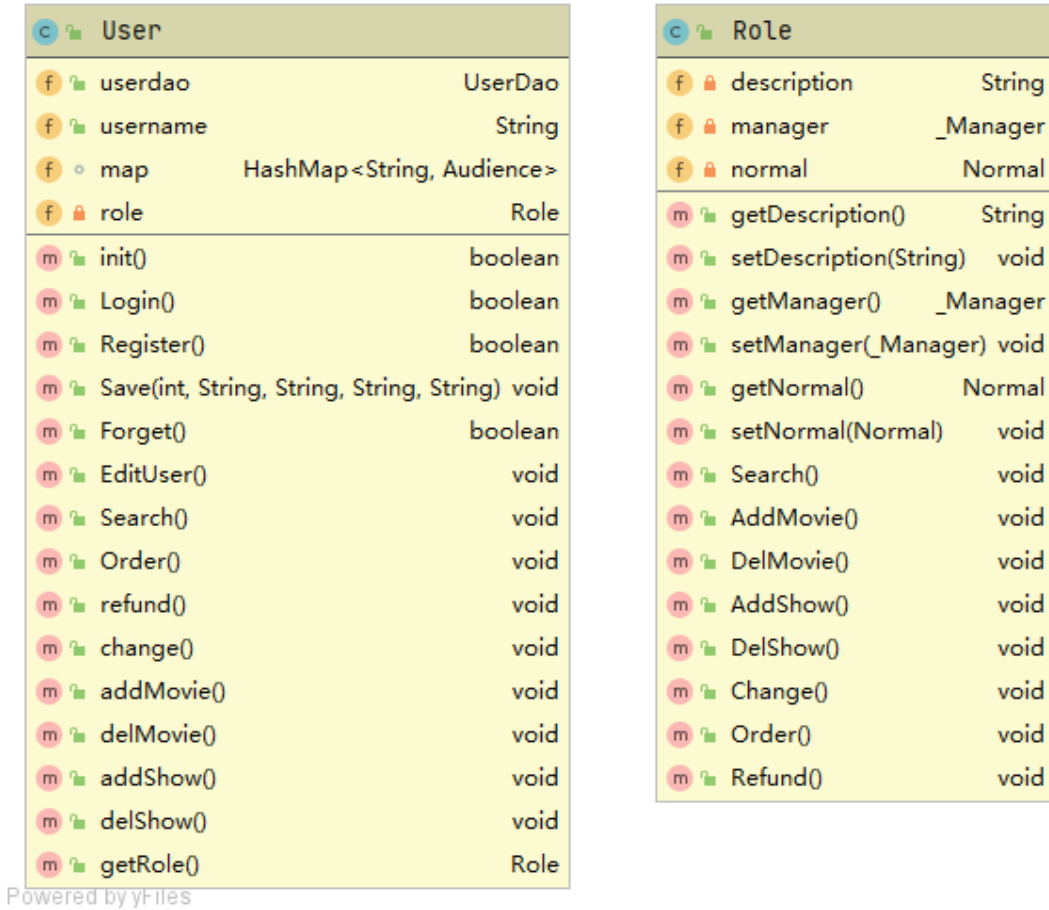


功能抽象类分为普通用户功能类和管理员功能类，分别有各自的接口。

1. 普通用户类中包含了用户服务类，从而普通用户类通过调用实现查询、订票退票换票功能。
2. 管理员类中包含了管理员服务类，从而管理员类通过调用实现对电影和上映的增删改查功能。

1.5 权限-用户 (Authority) 类

权限-用户类 UML 类图如下：

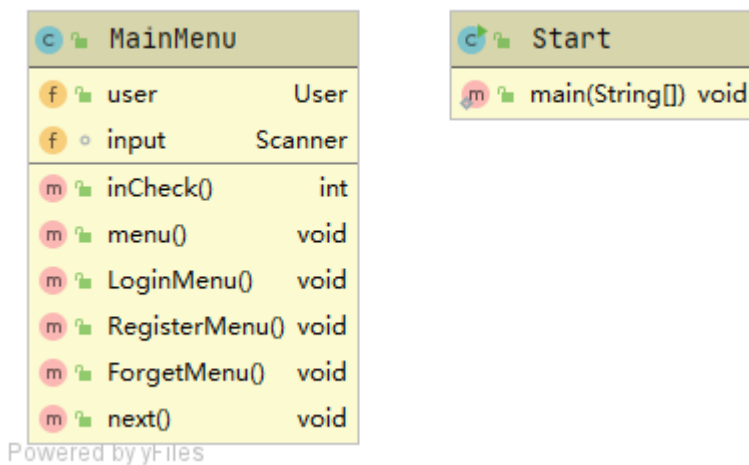


权限类实现权限管理，用户类中实现登录注册功能且包含权限类。

1. 权限类用于检测当前用户的权限是否能进行操作调用，若验证不超过则给出提示。
2. 用户类包含权限类，调用权限类提供的函数经过验证后执行操作。同时集成了登录、注册、忘记密码的用户操作和修改用户信息的管理员操作。

1.6 主界面 (Menu) 类

主界面类 UML 类图如下：

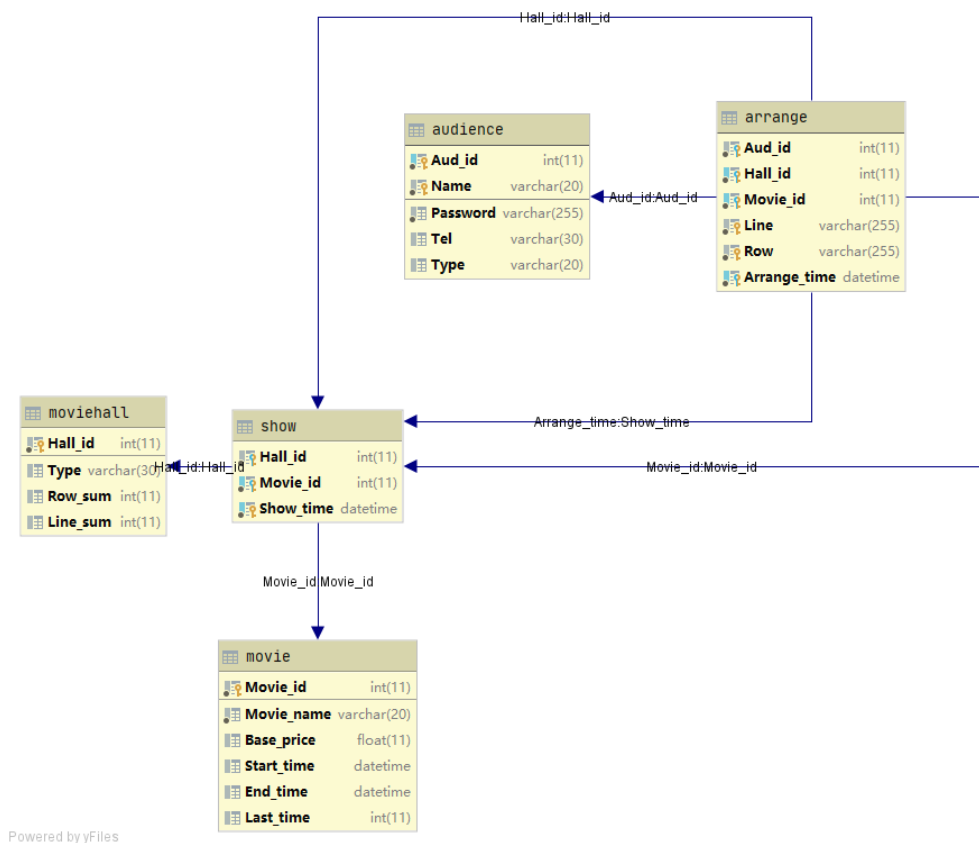


Start 类为程序入口，调用新的 MainMenu 对象的 menu（）函数，形成多级选择菜单。

2 系统数据库设计

2.1 各数据库表以及功能

本项目数据库一共使用了五张表，数据库 UML 总图如下：



每张表的功能如下图所示：

表名	描述	功能说明
Movie	电影信息表	存储电影信息
MovieHall	放映厅信息表	存储放映厅信息
Audience	用户信息表	存储用户信息
Show	放映信息表	存储放映信息
Arrange	订票信息表	存储订票信息

2.2 电影信息表

下表所示是电影信息表的详细说明，其中 Movie_id 是主键

编号	名称	描述	数据类型	大小	备注
1	Movie_id	电影序号	int	11	主键
2	Movie_name	电影名	varchar	20	
3	Base_price	电影价格	float	11	
4	Start_time	上映时间	datetime		
5	End_time	结束时间	datetime		
6	Last_time	电影时长	int	11	

2.3 影厅信息表

下表所示是影厅信息表的详细说明，其中 Hall_id 是主键。

编号	名称	描述	数据类型	大小	备注
1	Hall_id	影厅编号	int	11	主键
2	Type	影厅类型	varchar	30	
3	Row_sum	总列数	int	11	
4	Line_sum	总行数	int	11	

2.4 观众信息表

下表所示是影厅信息表的详细说明，其中 Aud_id 是主键。Type 是观众类型，逻辑上仅可为 User 或 Manager。

编号	名称	描述	数据类型	大小	备注
1	Aud_id	用户编号	int	11	主键
2	Name	用户名	varchar	20	
3	Password	密码	varchar	255	
4	Tel	手机号	varchar	30	

5	Type	观众类型	varchar	20	User/Manager
---	------	------	---------	----	--------------

2.5 放映信息表

如下表所示是放映信息表的详细说明，其中(Hall_id, Movie_id, show_time)的集合构成主键，Hall_id 是表 MovieHall 的外键，Movie_id 是表 Movie 的外键。

编号	名称	描述	数据类型	大小	备注
1	Hall_id	影厅编号	int	11	外键
2	Movie_id	电影编号	int	11	外键
3	Show_time	放映时间	datetime		

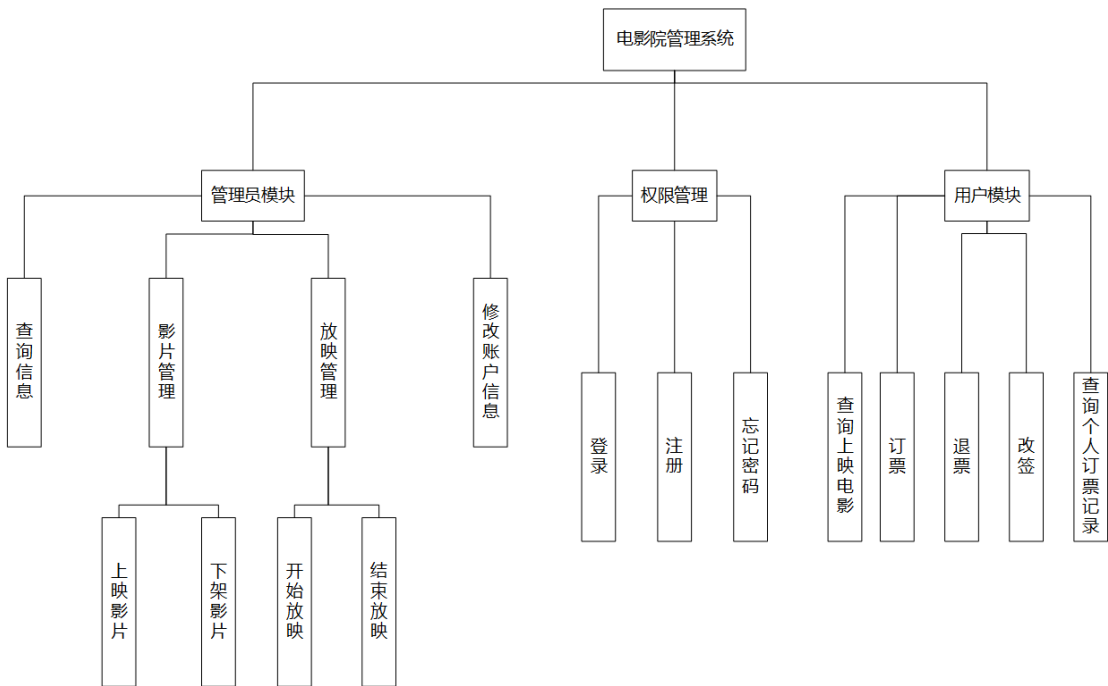
2.6 订票信息表

下表所示是订票信息表的详细说明，其中可以通过 Line 和 Row 确定用户订票的位置，(Aud_id, Hall_id, Movie_id, Line, Row, arrange_time)是主键，Aud_id 是表 Audience 的外键 Aud_id，Hall_id 是表 Show 的外键 Hall_id，Movie_id 是表 Show 的外键 Movid_id，Arrange_time 是表 Show 的外键 Show_time。外键绑定关系这里使用了“ON DELETE CASCADE”“ON UPDATE CASCADE”，如果 Show 表的外键对应数据被删除或者更新时（对应管理员将放映结束的操作），将关联数据完全删除或者相应地更新。

编号	名称	描述	数据类型	大小	备注
1	Aud_id	用户编号	int	11	外键
2	Hall_id	影厅编号	int	11	外键，级联
3	Movie_id	电影编号	int	11	外键，级联
4	Line	排	varchar	255	用户选座时写入
5	Row	列	varchar	255	用户选座时写入
6	Arrange_time	放映时间	datetime		外键，级联

3 主要功能设计

3.1 功能框图



3.2 功能说明

权限管理模块支持用户登录、注册、普通用户找回密码以及管理员修改用户信息。

1. 通过用户名及密码登录，读取数据库内信息确定身份是用户还是管理员。登陆时密码错误最多为3次，多则退出登录界面。
2. 注册需要输入对应的信息，包括用户名，密码，手机号，密码需要经过确认才能注册成功。
3. 普通用户忘记密码功能需要输入用户名和用户手机号来验证信息，从而读取当前密码。
4. 管理员修改用户信息，需要先输入用户编号和用户名查询是否有此用户，若有则继续输入该用户的新信息（用户编号不可变）。

管理员模块支持上架电影，下架电影，添加放映信息，更新放映信息，查找用户信息，查找电影信息，查找放映记录，查找订票记录等对于整个系统以及对应数据库的增删改查功能。

1. 管理员通过输入对应信息上架电影和添加放映，输入对应电影名下架电影。
2. 可以显示所有注册过的用户信息以及他们的订票记录，可以查找电影的放映记录以及当前电影的信息。

3. 删除放映场次时，一旦有用户订了该场次的票则票将会被删除。

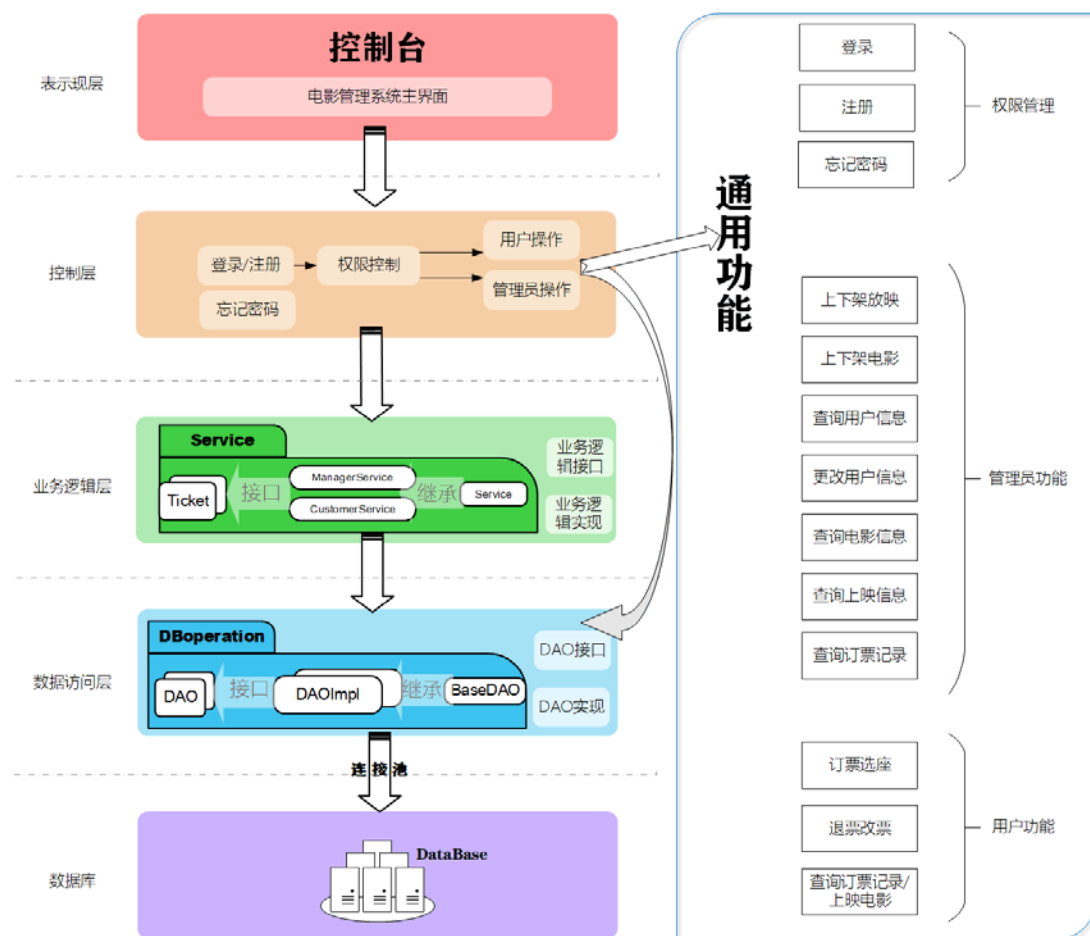
用户模块支持订票，自动选座和手动选座，退票，改签，查找本人订票记录和显示当前上映电影功能。选座时将输出影厅当前空位数量和总位置数量，方便客户购票。

1. 自动选座分为优化和顺序选座。默认方式是优化选座，必选中连续座位，当不满足条件时降级为顺序选座。
2. 手动选座将输出座位图，用户通过输入排和列选择座位。
3. 退票为输入对应电影名和对应场次来删除订票记录。
4. 改签为输入对应电影名和对应场次来寻找到所定场次，并选择同电影其他场次更改订票记录。
5. 查找本人订票记录即为按照当前用户编号寻找订票记录并输出。
6. 显示当前上映电影功能为输出当前上映电影的信息。

4 主要代码设计

4.1 总体架构描述

采用了四层的分层架构设计模式，项目分为了四层：数据访问层，业务逻辑层，控制层和表示层。分层架构图如下图。



4.2 数据访问层

数据访问层用于处理与数据库有关的操作，我们使用了 DAO 模式封装各种对 JDBC 所提供的接口的操作，实现数据的增加、删除、修改、查询等操作层。通过传入参数得到各种类型对应的泛型集合 List，并将操作结果或泛型集合 List 反馈到业务逻辑以及预留了一些给业务逻辑层和控制层处理有关修改数据的接口。

以下是数据访问层的核心代码：

```
//基类 BaseDao 以及各接口定义
public class BaseDao {...}
public interface UserDao {...}
public interface TicketDao {...}
public interface HallDao {...}
public interface ChangeShowDao {...}
public interface ChangeMovieDao {...}

//DAO 接口具体实现
public class UserDaoImpl extends BaseDao implements UserDao {...}
```

```

public class HallDaoImpl extends BaseDao implements HallDao {...}

public class TicketDaoImpl extends BaseDao implements TicketDao {...}

public class ChangeMovieDaoImpl extends BaseDao implements ChangeMovieDao {...}

public class ChangeShowDaoImpl extends BaseDao implements ChangeShowDao {...}

```

因为涉及到的操作的模式都是大同小异，这里举例较为复杂的查询订票记录函数：

```

/**
 * @param Aud_id 用户 id
 * @return 通过用户 id 或与权限寻找所订票：电影名 厅号 厅类型 时间 用户名 座位 票价
 */
@Override
public List<List<String>> findTicket(int Aud_id,String Aud_type) {
    List<List<String>> showList = new ArrayList<>();
    try {
        Date date = new Date();
        Timestamp timestamp = new Timestamp(date.getTime());
        String preparedSql;
        if (Aud_type.equals("Manager")) { //是管理员则获取全部订票记录
            preparedSql = "select Movie.Movie_name,Moviehall.Hall_id,Moviehall.type,arrange_time,name,line,row,base_
price as ticket_price " +
                "from Arrange join Movie on Arrange.movie_id = Movie.movie_id " +
                "join audience on audience.aud_id = arrange.aud_id " +
                "join MovieHall on arrange.hall_id=MovieHall.Hall_id ";
        } else { //非管理员则仅查询本 id 下的订票记录
            preparedSql = "select Movie.Movie_name,Moviehall.Hall_id,Moviehall.type,arrange_time,name,line,row,base_
price as ticket_price " +
                "from Arrange join Movie on Arrange.movie_id = Movie.movie_id " +
                "join audience on audience.aud_id = arrange.aud_id " +
                "join MovieHall on arrange.hall_id=MovieHall.Hall_id " +
                "where Arrange.aud_id=" + Aud_id;
        }
        conn = getConn();
        prepareSql = conn.prepareStatement(preparedSql);
        rs = prepareSql.executeQuery(); //执行
        while (rs.next()) { //对结果集合进行写入
            List<String> show = new ArrayList<>();
            show.add(rs.getString(1) + "");
            show.add(rs.getInt(2) + "");
            show.add(rs.getString(3) + "");
            show.add(rs.getTimestamp(4) + "");
            show.add(rs.getString(5) + "");
            show.add(rs.getInt(6) + "");
            show.add(rs.getInt(7) + "");
            show.add(rs.getFloat(8) + "");
            showList.add(show);
        }
    } catch (SQLException | ClassNotFoundException e) {
        e.printStackTrace();
    } finally {
        super.closeAll(conn, prepareSql, rs);
    }
    return showList;
}

```

可见，核心模式即为 preparedSql 作为预编译的数据库语句（若引入参数则使用 param 对象进行对参数的赋值，从而拼接成为完整的功能语句），执行语句并将得到的结果放入泛型集合或其他容器中，从而传递给业务逻辑层和控制层。

另外，比较典型的传递型函数（为业务逻辑层和控制层预留数据库操作接口的函数）为更新函数：

```

@Override
public int updateArrange(String sql, String[] param) {
    return super.executeSQL(sql, param);
}

```

4.3 业务逻辑层

业务逻辑层用于实现具体的功能，使用数据访问层的接口得到从数据库中得到的数据并进行操作，包括选座功能，以及输入数据进行添加删除修改操作。使用了管理员服务和普通用户服务均继承自服务类。

核心代码：管理员服务有修改电影接口和修改影厅接口对电影和当前放映电影的增删改查以及对用户列表的查询，普通用户服务有票务接口选取自动选座，手动选座，查询上映电影以及购票记录。

实现代码概述如下：

```
public interface Ticket {...} //票务接口

public interface ChangeShow {...} //修改上映接口

public interface ChangeMovie {...} //修改电影接口

public abstract class Service {...} //服务类

public class CustomerService extends Service implements Ticket {...} //用户服务类

public class ManagerService extends Service implements ChangeShow, ChangeMovie {...} //管理员服务类
```

其中自动选座的功能是重点代码，代码将自动选座分为两种情况考虑。第一种为条件允许实现连续座位选择，第二种不满足情况则进入顺序选择（有空位就选择）：

```
public void autoChooseSeat(Show show, int num) {
    //得到当前影厅行数和列数
    String sql = "select * from MovieHall where Hall_id = ?";
    String[] param = {show.getHall_id() + ""};
    HallDaoImpl hallDaoimpl = new HallDaoImpl();
    List<Hall> hallList = hallDaoimpl.getHall(sql, param);
    int row = hallList.get(0).getHallRowSum();
    int line = hallList.get(0).getHallLineSum();

    //得到当前选票数据
    sql = "select * from arrange where arrange.hall_id = ? and arrange.arrange_time = ?";
    String[] param1 = {show.getHall_id() + "", show.getShow_time() + ""};
    TicketDaoImpl ticket = new TicketDaoImpl();
    List<ArrangeTicket> arrangeTicketList = ticket.search(sql, param1);

    //生成选座数组
    char[][] chars = new char[line][row];
    for (int i = 0; i < line; i++) {
        for (int j = 0; j < row; j++) {
            chars[i][j] = '_';
        }
    }
    for (ArrangeTicket arrangeTicket : arrangeTicketList) {
        chars[arrangeTicket.getLine() - 1][arrangeTicket.getRow() - 1] = '*';
    }

    //-----
    //开始优化选座的处理
    int autoLine = 0;
    int autoRow = 0;
    //缩小选座区域
    int font, back, left, right;
    font = line / 4;
    back = line * 3 / 4;
    left = row / 4;
    right = row * 3 / 4;

    int temp_i = font;
    int temp_j = left - 1;
    boolean flag = false;

    for (int i = font; i <= back; i++) {
        for (int j = left; j <= right; j++) {
```



```

        if (chars[i][j] == '*' && !flag) { // 前面够否?
            if (j - left >= num) {
                autoLine = i;
                autoRow = left;
                break;
            }
            temp_i = i;
            temp_j = j;
            flag = true;
        } else if (chars[i][j] == '*' && flag) { // 中间够否?
            if (j - temp_j - 1 >= num) {
                autoLine = temp_i;
                autoRow = temp_j + 1;
                break;
            } else {
                temp_i = i;
                temp_j = j;
            }
        }
    }
    if (right - temp_j - 1 >= num) { // 后面够否? 整行空
        autoLine = temp_i;
        autoRow = temp_j + 1;
        break;
    } else {
        temp_i = i + 1;
        temp_j = left - 1;
        flag = false;
    }
}
// 优化选座处理完毕, 接下来进行优化购票是否越界的检测, 如果越界则执行顺序购票
// -----
int number = 0;
autoLine++;
autoRow++;
if (row < num || autoRow + num > row || autoLine > line) // 越界检测
{
    System.out.println("无法优化选座, 进入顺序选座模式! ");
    for (int i = 0; i < line; i++) {
        for (int j = 0; j < row && number < num; j++) {
            if (chars[i][j] != '*') {
                String[] param2 = {super.getAud_id() + "", show.getHall_id() + "", show.getMovie_id() + "", (i+1) + "", (j+1) + "", show.getShow_time() + ""};
                sql = "insert into arrange(Aud_id,Hall_id,Movie_id,line,row,Arrange_time) values(?,?,?,?,?,?)";
                int count = ticket.insertArrange(sql, param2);
                if (count > 0) {
                    System.out.println("顺序购票成功! 在第" + (i+1) + "排第" + (j+1) + "号");
                } else System.out.println("顺序购票失败! ");
                number++;
            }
        }
    }
    return;
} // 无法最优选座

// 未越界: 优化购票: 同时订多张连座的实现
String[] param2 = {super.getAud_id() + "", show.getHall_id() + "", show.getMovie_id() + "", autoLine + "", (autoRow + number) + "", show.getShow_time() + ""};
while (number != num) {
    sql = "insert into arrange(Aud_id,Hall_id,Movie_id,line,row,Arrange_time) values(?,?,?,?,?,?)";
    param2[4] = (autoRow + number) + "";

    int count = ticket.insertArrange(sql, param2);
    if (count > 0) {
        System.out.println("优化购票成功! 在第" + autoLine + "排第" + (autoRow + number) + "号");
    } else System.out.println("优化购票失败! " + Arrays.toString(param2));
    number++;
}
}

```

4.4 控制层

控制层主要实现用户的登陆、注册、忘记密码（登录界面）、修改账户信息（管理员）以及权限管理功能，根据登录用户的 Type 确定用户的身份是观众还是管理员，从而赋予相应的权限。

Role 类负责权限管理，Role 类调用用户接口以及管理员接口，在这两个接口实现中分别调用了用户业务接口和管理员业务接口，实现了控制层的功能。

User 类负责调用 DAO 接口获取用户信息到 HashMap 中，避免读取用户数据

时频繁遍历数据库。同时登陆、注册、忘记密码（登录界面）、修改账户信息（管理员）等功能的操作原语因为借助了 `HashMap` 易于管理数据而实现于此层。

此层的操作对于表示层的编程者来说是半透明状态的（可以通过 `getRole` 来获得 `Role` 对象从而调用方法）。

```
public class Role {
    private String description; // 角色名
    private _Manager manager = null; // 管理权限
    private Normal normal = null; // 顾客权限
    /**
     * 查询（实现分用户类型查询）
     */
    public void Search() {
        if (normal == null && manager == null) {
            System.out.println("您还没有登录，请登陆后再操作");
        } else if (normal != null) {
            normal.Search();
        } else {
            manager.Search();
        }
    }
}

// 管理员服务
public void AddMovie() {
    if (manager == null) {
        System.out.println("您没有管理员权限");
        return;
    }
    manager.AddMovie();
}
public void DelMovie() {...}
public void AddShow() {...}
public void DelShow() {...}

// 普通用户服务
public void Change() {...}
public void Order() {...}
public void Refund() {...}
}

public class User {
    public UserDao userDao = new UserDaoImpl();
    public String username;
    HashMap<String, Audience> map = new HashMap<>();
    private Role role = null;

    /**
     * 读取用户信息
     * @return 用户信息表
     */
    public boolean init() {
        List<Audience> audienceList = userDao.getAllUser();
        for (Audience value : audienceList) {
            Audience audience = new Audience();
            audience = value;
            map.put(audience.getAud_name(), audience);
        }
        return map.size() > 0;
    }

    /**
     * 用户登陆，读取信息实现管理员 or 用户角色的分配
     * @return 是否登陆成功
     */
    public boolean Login() {...}

    /**
     * 注册函数
     * 新增用户放到泛型集合里
     * @return 是否成功注册
     */
    public boolean Register() {...}

    /**
     * 忘记密码
     * @return 是否成功获得密码
     */
    public boolean Forget() {...}

    /**
     * 修改用户信息（允许修改权限）
     */
    public void EditUser() {...}
}
```

```

public void Search() {...}
public void Order() {...}
public void refund() {...}
public void change() {...}
public void addMovie() {...}
public void delMovie() {...}
public void addShow() {...}
public void delShow() {...}
public Role getRole() {...}
}

```

4.5 表示层

使用控制台进行交互，使用者通过输入序号选择菜单选项进行操作。本层由主界面类的 MainMenu 类和 Start 类实现。

```

public class Start {
    public static void main(String[] args) {
        MainMenu menu = new MainMenu();
        menu.menu();
    }
}

public class MainMenu {
    public User user = new User();
    public int inCheck() {...} // 检查输入选项
    public void menu() {...} // 初始菜单
    public void LoginMenu() {...} // 登陆
    public void RegisterMenu() {...} // 注册
    public void ForgetMenu() {...} // 忘记密码
    public void next() {...} // 登陆完毕的功能菜单
}

```

5.1 程序流程图

```
-----电影管理系统-----
1. 用户登陆      2. 用户注册      3. 忘记密码      4. 退出
1
-----登陆页面-----
请输入用户名
5
请输入密码
5
用户名不存在！
```

5.2.1 登陆

只有用户名和密码正确输入才能登录成功，如果用户不存在会提醒。

(a) 用户名不存在：

```
-----登陆页面-----  
请输入用户名  
5  
请输入密码  
5  
用户名不存在！
```

(b) 密码错误，登录密码最多输入三次，否则登录失败。

```
-----电影管理系统-----  
1. 用户登陆    2. 用户注册    3. 忘记密码    4. 退出  
1  
-----登陆页面-----  
请输入用户名  
hhh  
请输入密码  
1  
密码错误！重新输入  
2  
密码错误！重新输入  
3  
输入错误三次，本次登陆失败
```

(c) 登陆成功

```
-----电影管理系统-----  
1. 用户登陆    2. 用户注册    3. 忘记密码    4. 退出  
1  
-----登陆页面-----  
请输入用户名  
hhh  
请输入密码  
x74rtw05  
-----Loading...-----  
-----观众系统-----  
欢迎，用户hhh，请选择您的服务  
1. 订票    2. 退票    3. 改签    4. 查询电影&购票记录    5. 退出登陆
```

5.2.2 注册

新注册的用户需要输入用户名，密码以及手机号，需要二次确认密码。

(a) 注册成功

```
-----电影管理系统-----
1. 用户登陆    2. 用户注册    3. 忘记密码    4. 退出
2
-----注册页面-----
请输入要注册的用户名(用户名长度3~20):
shysensi
请输入密码
77777
请再次输入密码:
77777
请输入您的电话
77777
用户信息保存成功
注册成功!
```

(b) 用户名已存在导致失败

```
-----电影管理系统-----
1. 用户登陆    2. 用户注册    3. 忘记密码    4. 退出
2
-----注册页面-----
请输入要注册的用户名(用户名长度3~20):
shysensi
请输入密码
x74rtw05
用户名已存在
注册失败，请重新注册!
```

(c) 用户名不合规导致失败

```
-----电影管理系统-----
1. 用户登陆    2. 用户注册    3. 忘记密码    4. 退出
2
-----注册页面-----
请输入要注册的用户名(用户名长度3~20):
1
请输入密码
1
请输入正确长度的用户名!
注册失败，请重新注册!
```

5.2.3 忘记密码

普通用户忘记密码功能需要输入用户名和用户手机号来验证信息，从而读取当前密码。

(a) 信息正确，找回成功

```
-----电影管理系统-----
1.用户登陆    2.用户注册    3.忘记密码    4.退出
3
-----忘记密码-----
请输入用户名:
shysensi
请输入预留的手机号码:
77777
密码为: 77777
找回密码成功!
```

(b) 手机号不正确，找回失败

```
-----电影管理系统-----
1.用户登陆    2.用户注册    3.忘记密码    4.退出
3
-----忘记密码-----
请输入用户名:
hhh
请输入预留的手机号码:
1
找回密码失败，请重试或联系管理员!
```

(c) 用户名不存在，找回失败

```
-----电影管理系统-----
1.用户登陆    2.用户注册    3.忘记密码    4.退出
3
-----忘记密码-----
请输入用户名:
2
请输入预留的手机号码:
2
用户名不存在!
找回密码失败，请重试或联系管理员!
```

5.3 管理员服务

管理员有查询，管理影片，管理放映的权限。查询分为四种，根据用户选择查询类型的不同返回不同的结果并输出。

管理员界面：

-----管理系统-----

欢迎，管理员StuG，请选择您的操作

1. 查询信息 2. 影片管理 3. 放映管理 4. 修改账户信息 5. 退出登陆

!

5.3.1 查询信息

查询分为四种，查询所有放映记录，查询所有电影，查询所有用户，查询所有订票记录。

查询界面：

1

-----管理员查询-----

请输入查询类型 1. 查询所有放映记录 2. 查询所有电影 3. 查询所有用户 4. 查询所有订票记录 5. 返回

(a) 查询放映记录：

-----管理员查询-----

请输入查询类型 1. 查询所有放映记录 2. 查询所有电影 3. 查询所有用户 4. 查询所有订票记录 5. 返回

1

序号	电影名	影厅号	时间	影厅类型	电影时长	票价
1	辉夜大小姐	1号厅	2020-09-20 10:00:00.0	激光巨幕	120分钟	20.0元
2	辉夜大小姐	2号厅	2020-09-20 10:00:00.0	杜比	120分钟	20.0元
3	小妇人	5号厅	2020-09-21 10:00:00.0	激光	110分钟	30.0元

(b) 查询所有电影

-----管理员查询-----

请输入查询类型 1. 查询所有放映记录 2. 查询所有电影 3. 查询所有用户 4. 查询所有订票记录 5. 返回

2

序号	电影名	电影票价	上架时间	下映时间	电影时长
1	辉夜大小姐	20.0元	2020-08-25 00:00:00.0	2020-10-20 00:00:00.0	120分钟
2	小妇人	30.0元	2020-08-25 00:00:00.0	2020-09-25 00:00:00.0	110分钟
3	千与千寻	35.0元	2020-06-21 00:00:00.0	2019-07-10 00:00:00.0	100分钟
4	复仇者联盟	30.0元	2012-05-04 00:00:00.0	2012-05-31 00:00:00.0	120分钟
5	命运之夜	30.0元	2020-08-25 00:00:00.0	2008-05-31 00:00:00.0	80分钟
6	名侦探柯南	30.0元	2020-08-25 00:00:00.0	2020-08-31 00:00:00.0	90分钟

(c) 查询所有用户

-----管理员查询-----

请输入查询类型 1. 查询所有放映记录 2. 查询所有电影 3. 查询所有用户 4. 查询所有订票记录 5. 返回

3

序号	用户名	密码	手机号	用户类型
1	StuG	x74rtw05	18687839701	Manager
2	朝喜	2	2	Manager
4	1	1	1	Manager
5	hhh	x74rtw05	199999	User
5	shysensi	77777	77777	User
6	369	123	123	User

(d) 查询所有订票记录

-----管理员查询-----

请输入查询类型 1. 查询所有放映记录 2. 查询所有电影 3. 查询所有用户 4. 查询所有订票记录 5. 返回

4

序号	电影名	影厅号	影厅类型	时间	用户名	排	列	票价
1	辉夜大小姐	1号厅	激光巨幕	2020-09-20 10:00:00.0	shysensi	3	10	20.0元
2	辉夜大小姐	1号厅	激光巨幕	2020-09-20 10:00:00.0	shysensi	3	6	20.0元
3	辉夜大小姐	1号厅	激光巨幕	2020-09-20 10:00:00.0	shysensi	3	7	20.0元
4	辉夜大小姐	1号厅	激光巨幕	2020-09-20 10:00:00.0	shysensi	3	8	20.0元
5	辉夜大小姐	1号厅	激光巨幕	2020-09-20 10:00:00.0	shysensi	3	9	20.0元
6	辉夜大小姐	1号厅	激光巨幕	2020-09-20 10:00:00.0	555	8	8	20.0元
7	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	10	20.0元
8	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	11	20.0元
9	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	12	20.0元
10	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	6	20.0元
11	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	7	20.0元
12	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	1	30.0元
13	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	10	30.0元
14	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	11	30.0元
15	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	12	30.0元
16	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	13	30.0元
17	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	14	30.0元
18	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	15	30.0元
19	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	16	30.0元
20	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	17	30.0元
21	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	18	30.0元
22	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	19	30.0元
23	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	2	30.0元
24	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	20	30.0元
25	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	3	30.0元
26	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	4	30.0元
27	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	5	30.0元
28	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	6	30.0元
29	小妇人	5号厅	激光	2020-09-21 10:00:00.0	shysensi	1	7	30.0元

5.3.2 影片管理

可选择上架或下架影片。

欢迎，管理员StuG，请选择您的操作

1. 查询信息 2. 影片管理 3. 放映管理 4. 修改账户信息 5. 退出登陆

2

-----影片管理系统-----

1. 上架影片 2. 下架影片 3. 返回

(a) 上架：管理员输入信息，并存入数据库，输出添加成功。

-----影片管理系统-----

1. 上架影片 2. 下架影片 3. 返回

1

请输入电影id:

7

请输入电影名:

星际穿越

请输入电影票价:

60

请输入上映时间:

2020-08-20

请输入下架时间:

2020-11-20

请输入影片时长:

120

成功添加：星际穿越影片

(b) 下架：管理员通过查询电影信息获得电影 id，输入电影 id 并删除该电影信息。

```
-----管理系统-----
欢迎，管理员StuG，请选择您的操作
1.查询信息    2.影片管理    3.放映管理    4.修改账户信息    5.退出登陆
2
-----影片管理系统-----
1.上架影片    2.下架影片    3.返回
2
请输入电影id:
8
成功删除：test0影片
```

5.3.3 放映管理

可选择上映或下映影片。

```
-----放映管理系统-----
1.开始放映    2.结束放映    3.返回
```

(a) 上映：管理员输入信息，并存入数据库，输出添加成功。

```
-----放映管理系统-----
1.开始放映    2.结束放映    3.返回
1
电影信息：
序号    电影名    电影票价    上架时间    下映时间    电影时长
1        辉夜大小姐    |    20.0元    2020-08-25 00:00:00.0    2020-10-20 00:00:00.0    120分钟
2        小妇人        |    30.0元    2020-08-25 00:00:00.0    2020-09-25 00:00:00.0    110分钟
3        千与千寻      |    35.0元    2020-06-21 00:00:00.0    2019-07-10 00:00:00.0    100分钟
4        复仇者联盟    |    30.0元    2012-05-04 00:00:00.0    2012-05-31 00:00:00.0    120分钟
5        命运之夜      |    30.0元    2020-08-25 00:00:00.0    2008-05-31 00:00:00.0    80分钟
6        名侦探柯南    |    30.0元    2020-08-25 00:00:00.0    2020-08-31 00:00:00.0    90分钟
7        星际穿越      |    60.0元    2020-08-20 00:00:00.0    2020-11-20 00:00:00.0    120分钟
请输入电影id:
7
请输入放映厅id:
2
请输入播放时间:
2020-09-20 00:00:00
成功添加：2号大厅2020-09-20 00:00:00.0场次电影
```

(b) 下映：管理员通过查询电影信息获得电影 id 和放映厅 id 以及放映时间，输入电影 id 和放映厅 id 以及放映时间并删除该场次的放映，若有观众订票将被一起删除。

```

-----放映管理系统-----
1. 开始放映    2. 结束放映    3. 返回
2
电影信息：
序号      电影名      电影票价      上架时间      下映时间      电影时长
1          辉夜大小姐      |    20.0元      2020-08-25 00:00:00.0      2020-10-20 00:00:00.0      120分钟
2          小妇人      |    30.0元      2020-08-25 00:00:00.0      2020-09-25 00:00:00.0      110分钟
3          千与千寻      |    35.0元      2020-06-21 00:00:00.0      2019-07-10 00:00:00.0      100分钟
4          复仇者联盟      |    30.0元      2012-05-04 00:00:00.0      2012-05-31 00:00:00.0      120分钟
5          命运之夜      |    30.0元      2020-08-25 00:00:00.0      2008-05-31 00:00:00.0      80分钟
6          名侦探柯南      |    30.0元      2020-08-25 00:00:00.0      2020-08-31 00:00:00.0      90分钟
7          星际穿越      |    60.0元      2020-08-20 00:00:00.0      2020-11-20 00:00:00.0      120分钟

放映信息：
序号      电影名      影厅号      时间      影厅类型      电影时长      票价
1          辉夜大小姐      1号厅      2020-09-20 10:00:00.0      激光巨幕      120分钟      20.0元
2          辉夜大小姐      2号厅      2020-09-20 10:00:00.0      杜比      120分钟      20.0元
3          小妇人      5号厅      2020-09-21 10:00:00.0      激光      110分钟      30.0元
4          星际穿越      2号厅      2020-09-20 00:00:00.0      杜比      120分钟      60.0元

请输入电影id:
7
请输入放映厅id:
2
请输入播放时间:
2020-09-20 00:00:00.0
成功删除：2号大厅2020-09-20 00:00:00.0场次电影

```

5.3.4 修改账户信息

管理员修改用户信息，需要先输入用户编号和用户名查询是否有此用户，若有则继续输入该用户的新信息（用户编号不可变）。

（a）修改成功

```

-----管理系统-----
欢迎，管理员StuG，请选择您的操作
1. 查询信息    2. 影片管理    3. 放映管理    4. 修改账户信息    5. 退出登陆
4
请输入原用户id:
0
请输入原用户名:
0
找到用户：0-0
请输入现用户名:
test0
请输入现用户电话号码:
000
请输入现用户权限：（Manager/User）
User
请输入现用户密码:
0
成功修改！ id:0    用户名: test0    密码: 0    手机号码: 000 类型: User

```

（b）未找到用户，修改失败

```

-----管理系统-----
欢迎，管理员StuG，请选择您的操作
1.查询信息    2.影片管理    3.放映管理    4.修改账户信息    5.退出登陆
4
请输入原用户id:
10
请输入原用户名:
10
没有该用户!

```

5.4 普通用户服务

用户有订票，退票，改签，自动选座，手动选座，查询的权限，查询包括，查询包括查询个人订票记录以及当前上映电影。

普通用户界面：

```

-----观众系统-----
欢迎，用户shysensi，请选择您的服务
1.订票    2.退票    3.改签    4.查询电影&购票记录    5.退出登陆

```

5.4.1. 订票

首先打印当前上映的电影场次，用户可以看到电影的票价，放映信息以及电影相关信息。

```

-----观众系统-----
欢迎，用户shysensi，请选择您的服务
1.订票    2.退票    3.改签    4.查询电影&购票记录    5.退出登陆
1

```

序号	电影名	影厅号	时间	影厅类型	电影时长	票价
1	辉夜大小姐	1号厅	2020-09-20 10:00:00.0	激光巨幕	120分钟	20.0元
2	辉夜大小姐	2号厅	2020-09-20 10:00:00.0	杜比	120分钟	20.0元
3	小妇人	5号厅	2020-09-21 10:00:00.0	激光	110分钟	30.0元

请输入电影名：

用户输入电影名，并根据场次选择序号以及票数，若票数小于剩余票数，系统会询问是否手动选座，若票数超过则订票失败。若进入选座模式后输入 y 则进入手动选座界面，否则自动选座，若自动选座则根据具体情况选择优化（连续）选座或者顺序选座，之后打印选座信息。

（a）选择电影名称，得到场次，等待输入选择序号

```

请输入电影名:
辉夜大小姐

```

序号	电影名	厅号	时间
1	辉夜大小姐	1号厅	2020-09-20 10:00:00.0
2	辉夜大小姐	2号厅	2020-09-20 10:00:00.0

请选择场次（填序号）：(0返回)

(b) 输入序号返回所定票信息，进入票数选择

请输入电影名：

辉夜大小姐

序号	电影名	厅号	时间
1	辉夜大小姐	1号厅	2020-09-20 10:00:00.0
2	辉夜大小姐	2号厅	2020-09-20 10:00:00.0

请选择场次（填序号）：(0返回)

2

您已选择：

2 辉夜大小姐 2号厅 2020-09-20 10:00:00.0

该场次已选座/总容量为：0/200，剩余量为200

请选择预订的电影票数

(c) 选择票数后选择手动或自动选票

您已选择：

1 辉夜大小姐 1号厅 2020-09-20 10:00:00.0

该场次已选座/总容量为：6/200，剩余量为194

请选择预订的电影票数

1

是否手动选座：(y/n)

(c.1) 超出最大余票数量导致订票失败

请输入电影名：

辉夜大小姐

序号	电影名	厅号	时间
1	辉夜大小姐	1号厅	2020-09-20 10:00:00.0
2	辉夜大小姐	2号厅	2020-09-20 10:00:00.0

请选择场次（填序号）：(0返回)

2

您已选择：

2 辉夜大小姐 2号厅 2020-09-20 10:00:00.0

该场次已选座/总容量为：0/200，剩余量为200

请选择预订的电影票数

210

超出最大电影票余量，订票失败！

(d.1) 手动选座，输入行和列购票

是否手动选座：(y/n)

y

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
第1排																				
第2排																				
第3排																				
第4排																				
第5排																				
第6排																				
第7排																				
第8排																				
第9排																				
第10排																				

_可选，*不可选

请选择您的座位排

10

请选择您的座位列

10

购票成功!

(d.2) 自动选座-优化(连座)模式

您已选择:

2 辉夜大小姐 2号厅 2020-09-20 10:00:00.0

请选择预订的电影票数

10

是否手动选座：(y/n)

n

优化购票成功! 在第3排第6号

优化购票成功! 在第3排第7号

优化购票成功! 在第3排第8号

优化购票成功! 在第3排第9号

优化购票成功! 在第3排第10号

优化购票成功! 在第3排第11号

优化购票成功! 在第3排第12号

优化购票成功! 在第3排第13号

优化购票成功! 在第3排第14号

优化购票成功! 在第3排第15号

(d.3) 自动选座-顺序模式

您已选择：
1 辉夜大小姐 1号厅 2020-09-20 10:00:00.0

请选择预订的电影票数

21

是否手动选座：(y/n)

n

无法优化选座，进入顺序选座模式！

顺序购票成功！在第1排第1号

顺序购票成功！在第1排第2号

顺序购票成功！在第1排第3号

顺序购票成功！在第1排第4号

顺序购票成功！在第1排第5号

顺序购票成功！在第1排第6号

顺序购票成功！在第1排第7号

顺序购票成功！在第1排第8号

顺序购票成功！在第1排第9号

顺序购票成功！在第1排第10号

顺序购票成功！在第1排第11号

顺序购票成功！在第1排第12号

顺序购票成功！在第1排第13号

顺序购票成功！在第1排第14号

顺序购票成功！在第1排第15号

顺序购票成功！在第1排第16号

顺序购票成功！在第1排第17号

顺序购票成功！在第1排第18号

顺序购票成功！在第1排第19号

顺序购票成功！在第1排第20号

顺序购票成功！在第2排第1号

5.4.2. 退票

系统会提示用户输入电影名，系统返回用户关于该电影的订票记录，用户输入序号，系统会提示退票成功，是否继续退票。

-----观众系统-----

欢迎，用户555，请选择您的服务

1. 订票 2. 退票 3. 改签 4. 查询电影购票记录 5. 退出登陆

2

订票记录为：

序号	电影名	影厅号	影厅类型	时间	用户名	排	列	票价
1	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	10	20.0元
2	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	6	20.0元
3	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	7	20.0元
4	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	8	20.0元
5	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	9	20.0元

请输入电影名：

辉夜大小姐

序号	电影名	厅号	排	列	时间
1	辉夜大小姐	2号厅	第3排	第10列	2020-09-20 10:00:00.0
2	辉夜大小姐	2号厅	第3排	第6列	2020-09-20 10:00:00.0
3	辉夜大小姐	2号厅	第3排	第7列	2020-09-20 10:00:00.0
4	辉夜大小姐	2号厅	第3排	第8列	2020-09-20 10:00:00.0
5	辉夜大小姐	2号厅	第3排	第9列	2020-09-20 10:00:00.0

请输入退票序号

5

退票成功是否继续退票(y/n)?

n

5. 4. 3. 改签

系统首先输出该用户的所有订票记录，用户输入序号，系统返回该电影的其他场次，提示用户输入改签后的场次，用户输入场次号，只能手动选座。改签后输出改签成功。

```
-----观众系统-----
欢迎，用户555，请选择您的服务
1.订票  2.退票  3.改签  4.查询电影&购票记录  5.退出登陆
3
订票记录为：
序号    电影名    影厅号    影厅类型    时间    用户名    排    列    票价
1        辉夜大小姐    2号厅    杜比    2020-09-20 10:00:00.0    555    3    10    20.0元
2        辉夜大小姐    2号厅    杜比    2020-09-20 10:00:00.0    555    3    11    20.0元
3        辉夜大小姐    2号厅    杜比    2020-09-20 10:00:00.0    555    3    12    20.0元
4        辉夜大小姐    2号厅    杜比    2020-09-20 10:00:00.0    555    3    6    20.0元
5        辉夜大小姐    2号厅    杜比    2020-09-20 10:00:00.0    555    3    7    20.0元
6        辉夜大小姐    2号厅    杜比    2020-09-20 10:00:00.0    555    3    8    20.0元

请输入电影名：
辉夜大小姐
序号    电影名    厅号    排    列    时间
1        辉夜大小姐    2号厅    第3排    第10列    2020-09-20 10:00:00.0
2        辉夜大小姐    2号厅    第3排    第11列    2020-09-20 10:00:00.0
3        辉夜大小姐    2号厅    第3排    第12列    2020-09-20 10:00:00.0
4        辉夜大小姐    2号厅    第3排    第6列    2020-09-20 10:00:00.0
5        辉夜大小姐    2号厅    第3排    第7列    2020-09-20 10:00:00.0
6        辉夜大小姐    2号厅    第3排    第8列    2020-09-20 10:00:00.0

选择要改签的票序号
6
序号    电影名    厅号    时间
1        辉夜大小姐    1号厅    2020-09-20 10:00:00.0
2        辉夜大小姐    2号厅    2020-09-20 10:00:00.0

请选择场次（序号）：
1
您已选择：
1  辉夜大小姐    1号厅    2020-09-20 10:00:00.0
| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|16|17|18|19|20|
第1排 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
第2排 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
第3排 | - | - | - | - | - | * | * | * | * | - | - | - | - | - | - | - | - | - |
第4排 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
第5排 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
第6排 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
第7排 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
第8排 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
第9排 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
第10排 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
_可选，*不可选

请选择您的座位排
8
请选择您的座位列
8
购票成功！
-----观众系统-----
欢迎，用户555，请选择您的服务
1.订票  2.退票  3.改签  4.查询电影&购票记录  5.退出登陆
4
```

5. 4. 4. 查询上映电影&个人购票记录

查询个人订票记录以及当前上映电影。

（a）查询当前上映电影

```
-----观众系统-----
欢迎，用户555，请选择您的服务
1.订票  2.退票  3.改签  4.查询电影&购票记录  5.退出登陆
4
-----普通用户查询-----
请输入查询类型 1.查询当前上映电影 2.查询订票记录
1
序号    电影名    影厅号    时间    影厅类型    电影时长    票价
1        辉夜大小姐    1号厅    2020-09-20 10:00:00.0    激光巨幕    120分钟    20.0元
2        辉夜大小姐    2号厅    2020-09-20 10:00:00.0    杜比    120分钟    20.0元
3        小妇人    5号厅    2020-09-21 10:00:00.0    激光    110分钟    30.0元
```

（b）查询个人订票记录


```

-----观众系统-----
欢迎，用户555，请选择您的服务
1.订票  2.退票  3.改签  4.查询电影购票记录  5.退出登陆
4
-----普通用户查询-----
请输入查询类型 1.查询当前上映电影 2.查询订票记录
2

```

序号	电影名	影厅号	影厅类型	时间	用户名	排	列	票价
1	辉夜大小姐	1号厅	激光巨幕	2020-09-20 10:00:00.0	555	8	8	20.0元
2	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	10	20.0元
3	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	11	20.0元
4	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	12	20.0元
5	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	6	20.0元
6	辉夜大小姐	2号厅	杜比	2020-09-20 10:00:00.0	555	3	7	20.0元

5.5 退出

```

-----电影管理系统-----
1.用户登陆  2.用户注册  3.忘记密码  4.退出
4
欢迎下次光临！

Process finished with exit code 0

```

6 对课程的体会

6.1 赵书彬

6.1.1 心得

漫长的小学期+暑假里，我在华为云课堂里伴随着老师的视频和教程学习了关于 Java 这门语言的基础知识，结合海量的编程作业，我对学到的知识进行了进一步巩固，同时自身的编码能力和对项目的理解能力都有了提高。Java 和 C++ 一样，都是面向对象的编程，不同的是 Java 中有接口，泛型，反射等，而这些多出来的高级部分在 C++ 中也隐隐约约有一些体现（虽然没有实现），这是我们值得思考的点：高级语言的高级之处在何处。

其中收获最大的 DAO 模式以及分层架构的设计思想——对于每一层的编程，我们需要关心的事情是不同的：数据访问层用于处理与数据库有关的操作，业务逻辑层用于实现具体的功能，使用数据访问层的接口得到从数据库中得到的数据并进行操作，控制层主要实现用户的具体分类操作（行为模拟），表示层就是用户直接与程序进行交互的地方。

大作业项目的组队进行的好处在于可以提前感受团队项目的工作模式：需要有好的版本控制方法和好的分工合作方法，乃至对代码进行集中调试时出现 bug 的沟通和解决方法。这些都是需要在项目进行的实际过程中不断学习和探讨的，

在我们的项目进行的过程中这些问题得到了妥当的解决（Git 实现版本控制，注释写清和代码风格统一实现良好的沟通），从而使得项目顺利完成。

6.1.2 分工

数据库设计，大作业文档文字部分撰写，控制层设计以及实现（用户类），数据访问层设计以及实现，业务逻辑层设计以及实现（普通用户）。

6.2 刘晨欣

6.2.1 心得

这次小学期进行了 Java 课程的学习和对课程内容的实践，虽然是线上课程，我还是收获了很多经验和知识。首先，和之前学过的 C++ 语言相比，同样是面向对象，java 语言也有其独特之处，比如它是单继承的，但是可以实现多个接口；用包代替了命名空间；不必提前声明；所有数据方法都放在类中等等。前几周的课程内容主要是基本语法和面向对象的介绍，对我们来说容易上手，而后面讲到接口、泛型、SQL 连接、DAO 模式是比较新的内容，需要实践练习，也是大作业项目中用到很多次的知识。比如使用接口来实现封装，连接数据库的 JDBC，相比使用文件操作，有了更好的存储数据的容器。进行大作业项目的过程中，我认识到了分层架构的重要性和便利性，好的架构使我们思路更加清晰，也使分工更加简洁明了。而且不仅仅有利于程序的编写，对我们绘制各类图表也有很大的帮助。对于一个项目来说，千里之堤溃于蚁穴，最重要的是及时调整心态、耐心调试。另外由于考试周的影响，我们对 java 的学习在中途停止了一段时间，对学习到的新知识应用起来不够熟练。纸上得来终觉浅，绝知此事要躬行。这让我得到了一个教训，学习新的语言时应当边学边练，对知识的掌握需要大量的实践练习。其次，学习使用基本的办公软件也很有必要，这样才能条理清晰地阐述清楚整个工程的逻辑思路和架构。

6.2.2 分工

大作业文档图片表格部分编辑，UML 图设计，实体类设计以及实现，控制层设计以及实现（权限规则类），业务逻辑层设计以及实现（接口设计+管理员类），表示层设计以及实现。