

分布式电影订票系统 系统测试文档

分布式电影订票系统设计小组

成员：李佳钰

王文琪

徐兰天

于明飞

目 录

1 系统通用类测试	3
1.1 分布式数据库通用类测试	3
1.2 系统通用类测试报告	6
2 用户管理模块测试	8
2.1 用户管理模块测试用例	8
2.2 用户管理模块测试报告	10
3 电影信息管理模块测试	12
3.1 物品信息模块白盒测试用例	12
3.2 物品信息模块黑盒测试用例	12
3.3 物品信息模块测试报告	13
4 主页面模块测试	15
4.1 主页面模块测试用例	15
4.2 主页面模块测试报告	16
5 小 结.....	17

1 系统通用类测试

1.1 分布式数据库通用类测试

(1) 白盒测试用例

```
C:\Windows\system32>ping 10.108.6.180

正在 Ping 10.108.6.180 具有 32 字节的数据:
来自 10.108.6.180 的回复: 字节=32 时间<1ms TTL=64
来自 10.108.6.180 的回复: 字节=32 时间<1ms TTL=64
来自 10.108.6.180 的回复: 字节=32 时间<1ms TTL=64
来自 10.108.6.180 的回复: 字节=32 时间<1ms TTL=64

10.108.6.180 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms

C:\Windows\system32>net stop mysql
没有启动 MySQL 服务。

请键入 NET HELPMSG 3521 以获得更多的帮助。

C:\Windows\system32>net start mysql
MySQL 服务正在启动。
MySQL 服务已经启动成功。

C:\Windows\system32>mysql -uroot -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.18 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> stop slave;
Query OK, 0 rows affected (0.01 sec)
```

图 1.1 测试主从机的通信

```

D:\MySQL\mysql-8.0.18-winx64\bin>mysql -uroot -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.18 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show master status;
+-----+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000015 |      155 |               |                   |                   |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> _

```

图 1.2 主服务器状态信息

```

mysql> change master to master_host='10.108.6.180',master_user='root',master_password='mysql',master_port=3306,master_log_file='mysql-bin.000015',master_log_pos=155;
Query OK, 0 rows affected, 1 warning (0.04 sec)

mysql> start slave;
Query OK, 0 rows affected (0.01 sec)

mysql> show slave status\G;
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 10.108.6.180
Master_User: root
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000015
Read_Master_Log_Pos: 155
Relay_Log_File: DESKTOP-2TCF00A-relay-bin.000002
Relay_Log_Pos: 322
Relay_Master_Log_File: mysql-bin.000015
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: mymovie, mymovie2
Replicate_Do_Table:
Replicate_Ignore_DB:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 155
Relay_Log_Space: 540
Until_Condition: None

```

图 1.3 从服务器状态信息

```
schema.xml
1  <?DOCTYPE mycat:schema SYSTEM "schema.dtd">
2  <mycat:schema xmlns:mycat="http://io.mycat/">
3    <schema name="mymovie" checkSQLSchema="true" sqlMaxLimit="100">
4      <table name="sell" primaryKey="payStyle" autoIncrement="false" dataNode="dn1,dn2"
5        rule="sharding-by-intfile" />
6      <table name="person" primaryKey="personID" autoIncrement="false" dataNode="fn1"/>
7      <table name="movie" primaryKey="movieID" autoIncrement="false" dataNode="fn2"/>
8    </schema>
9    <dataNode name="dn1" dataHost="localhost1" database="mymovie" />
10   <dataNode name="dn2" dataHost="localhost1" database="mymovie2" />
11   <dataNode name="fn1" dataHost="localhost2" database="mymovie" />
12   <dataNode name="fn2" dataHost="localhost2" database="mymovie" />
13   <dataHost name="localhost1" maxCon="1000" minCon="10" balance="1"
14     writeType="0" dbType="mysql" dbDriver="native" switchType="1" slaveThreshold="100">
15     <heartbeat>select user()</heartbeat>
16     <!-- can have multi write hosts -->
17     <writeHost host="hostM1" url="localhost:3306" user="root"
18       password="mysql">
19     <!-- can have multi read hosts -->
20     <readHost host="hostS2" url="10.108.7.163:3306" user="root" password="20154418" />
21   </dataHost>
22   <dataHost name="localhost2" maxCon="1000" minCon="10" balance="1"
23     writeType="0" dbType="mysql" dbDriver="native" switchType="1" slaveThreshold="100">
24     <heartbeat>select user()</heartbeat>
25     <writeHost host="hostM1" url="localhost:3306" user="root"
26       password="mysql">
27     <readHost host="hostM1" url="localhost:3306" user="root"
28       password="mysql" />
29   </dataHost>
30 </mycat:schema>
```

图 1.4 mycat 中间件配置 schma.xml

```
schema.xml  server.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE mycat:server SYSTEM "server.dtd">
3  <mycat:server xmlns:mycat="http://io.mycat/">
4    <system>
5      <property name="useSqlStat">0</property>
6      <property name="useGlobleTableCheck">0</property>
7      <property name="sequenceHandlerType">2</property>
8      <property name="processorBufferPoolType">0</property>
9      <property name="handleDistributedTransactions">0</property>
10     <property name="serverPort">8066</property>
11     <property name="useOffHeapForMerge">1</property>
12     <property name="memoryPageSize">1m</property>
13     <property name="spillsFileBufferSize">1k</property>
14     <property name="useStreamOutput">0</property>
15     <property name="systemReserveMemorySize">384m</property>
16     <property name="useZKSwitch">true</property>
17   </system>
18   <user name="root">
19     <property name="password">mysql</property>
20     <property name="schemas">mymovie</property>
21   </user>
22   <user name="user">
23     <property name="password">user</property>
24     <property name="schemas">mymovie</property>
25     <property name="readOnly">true</property>
26   </user>
27 </mycat:server>
```

图 1.5 mycat 中间件配置 server.xml

(2) 黑盒测试用例

出于安全性考虑，普通用户权限下每个用户只能查看自己的购票信息记录。

用户信息列表

用户名	电影名	支付方式	操作
李四	中国女排	0	<button>删除</button>
李四	中国女排	1	<button>删除</button>
共: < 2> 条记录			

图 1.6 a 不同用户只能显示各自的购票信息

用户信息列表

用户名	电影名	支付方式	操作
于明飞	中国机长	0	<button>删除</button>
于明飞	中国机长	1	<button>删除</button>
共: < 2> 条记录			

图 1.6 b 不同用户只能显示各自的购票信息

1.2 系统通用类测试报告

对部分测试结果给出测试报告。

系统通用类	用户对电影信息的增加和删除		
测试描述			
用例目的	测试后台数据库功能的正确性		
前提条件	用户已注册并登录、对数据库成功实现水平分片和主从复制		
输入/动作	期望的输出/响应	实际情况	
添加购票信息	数据库中 mymovie 的 sell 表和 mymovie2 的 sell 表中根据分片规则新增了相应的数据，电影余票量相应减少，主从数据库实现数据复制	成功	

删除购票信息	数据库中 mymovie 的 sell 表和 mymovie2 的 sell 表中根据分片规则 删除相应的数据，电影余 票量相应更新，主从数据 库实现数据同步	成功
修改购票信息	在数据库分表中修改相应 数据，主从数据库实现数 据同步	成功
查找购票信息	在数据库分表中查找相应 的数据并返回	成功

表 1.1 系统通用类测试报告

2 用户管理模块测试

2.1 用户管理模块测试用例

登陆和注册用例：



The login form is titled "登录" (Login) in a large, bold, black font. It contains three input fields: the first contains the text "张三" (Zhang San), the second contains three dots "...", and the third contains the text "管理员" (Administrator). Below the input fields is a checkbox with the label "记住用户名和密码" (Remember username and password). At the bottom are two buttons: a light blue button labeled "登入" (Login) and a dark blue button labeled "注册" (Register).

图 2.1 登陆示意图



The registration form is titled "注册" (Register) in a large, bold, black font. It contains three input fields: the first contains the text "徐兰天" (Xu Lantian), the second contains three dots "...", and the third contains three dots "...". At the bottom is a dark blue button labeled "注册" (Register).

图 2.2 注册示意图

用户信息管理用例：

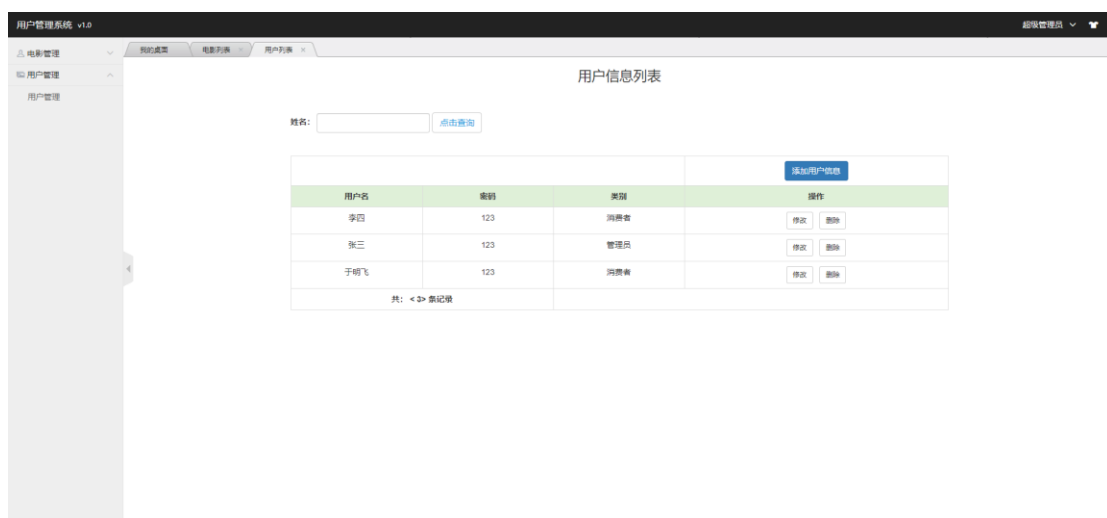


图 2.3 查询用户信息

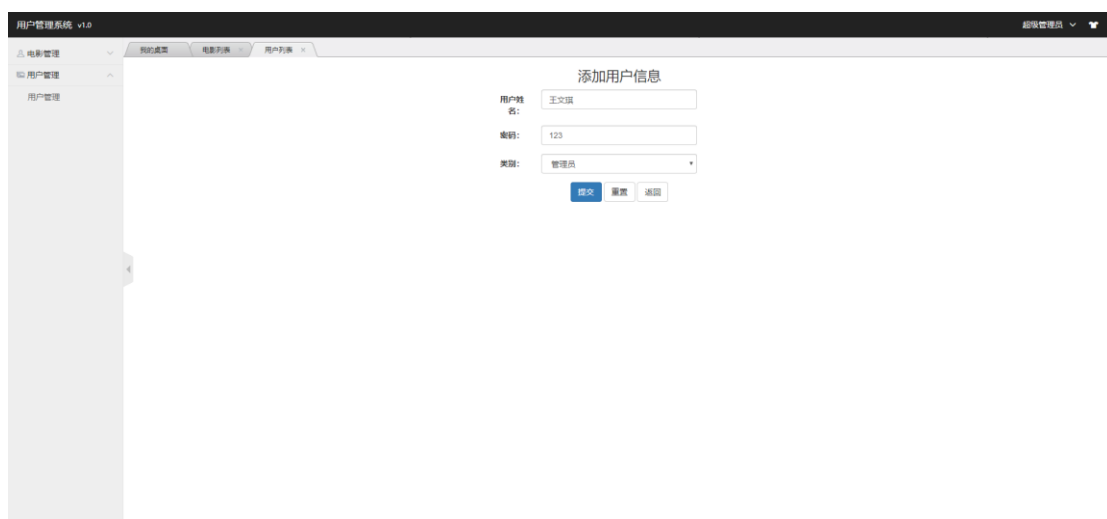


图 2.4 添加用户信息

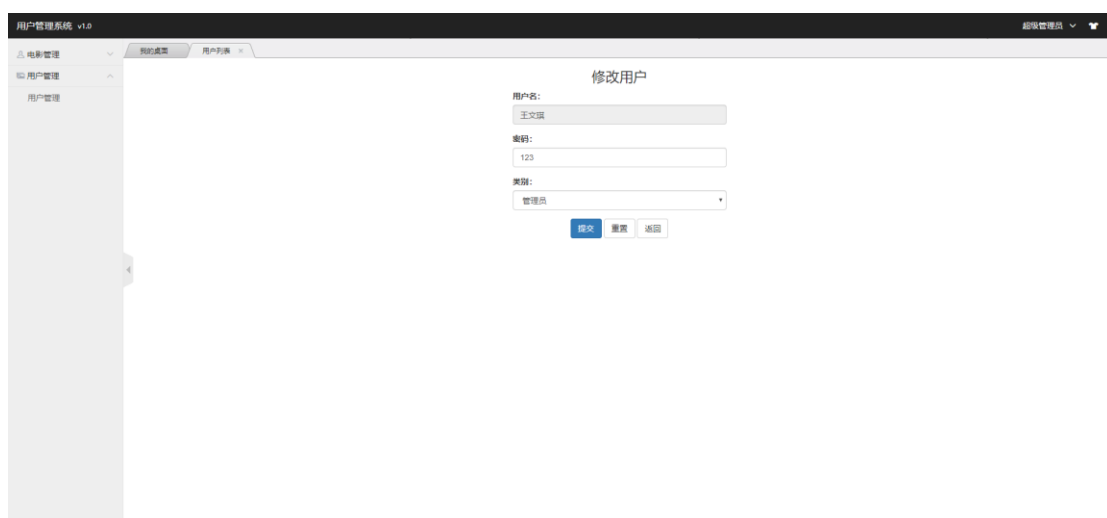


图 2.5 修改用户信息



图 2.6 删除用户信息

2.2 用户管理模块测试报告

用户登录功能描述	用户通过账号、密码进行登录		
用例目的	测试登录功能的正确性		
前提条件	账号已经提前注册		
输入/动作	期望的输出/响应	实际情况	
输入账号、密码后，点击登录	成功登陆可使用其他功能	成功	
输入账号后，点击修改密码	成功进入修改密码界面	成功	
输入原密码、新密码、重新输入新密码后，点击确认	提示：修改密码成功	成功	
点击取消	退出修改密码界面	成功	
点击退出	成功退出登录界面	成功	
用户注册功能描述	用户填写个人信息注册账号		
用例目的	测试注册功能正确性		
前提条件	给账号、用户名未被使用过		

输入/动作	期望的输出/响应	实际情况
点击注册	界面跳转至用户协议	成功
点击同意	跳转至填写个人信息界面	成功
输入账号	无	成功
输入密码	无	成功
确认密码	无	成功
点击注册	跳转至注册成功界面	成功
用户信息管理功能描述	对用户信息进行增删改查操作	
用例目的	测试用户信息管理功能正确性	
前提条件	对用户信息的操作符合系统要求	
输入/动作	期望的输出/响应	实际情况
点击添加用户信息	界面跳转至用户协议	成功
点击删除用户信息	删除相应的用户记录	成功
点击修改用户信息	跳转至修改用户信息界面	成功
点击查询用户信息	跳转至用户信息列表	成功
输入用户名	无	成功
输入密码	无	成功
输入用户类别	无	成功

表 2.1 用户管理模块测试报告

3 电影信息管理模块测试

3.1 电影信息管理模块白盒测试用例

新电影信息发布成功：

电影信息表中存储电影的信息，包括电影的 ID（主键）、电影名称、电影上映时间、票价、总票量、余票量。

对象	movie @mymovie (localhost...				
开始事务	文本	筛选	排序	导入	导出
movieID	movieName	movieTime	moviePrice	movieNum	movieRemain
1	中国女排	2020-01-02 10:00:00	50	50	48
2	闪光少女	2020-01-05 10:00:00	40	30	30
3	中国机长	2020-02-13 10:00:00	50	60	58

图 3.1 电影信息

3.2 电影信息管理模块黑盒测试用例

新增电影信息：

用户管理系统 v1.0

添加电影信息

movieName: 中国机长

movieTime: 2020/02/13

moviePrice: 50

movieNum: 60

movieRemain: 60

提交 重置 返回

图 3.2 添加电影信息

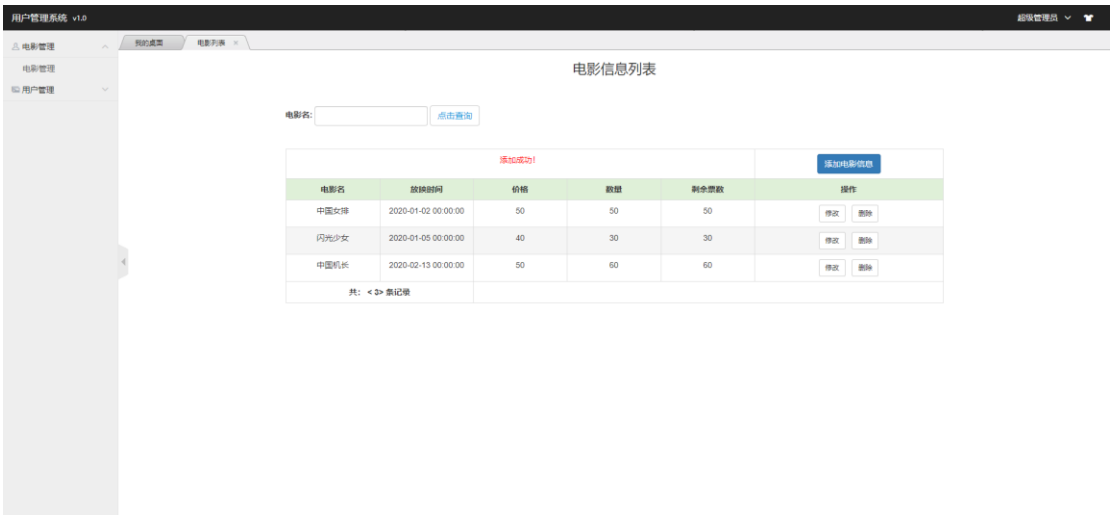


图 3.3 电影信息列表

删除电影信息：



图 3.4 删除操作后电影信息列表

3.3 电影信息管理模块测试报告

对部分测试结果给出测试报告。

物品信息模块测试报告如表 3.1 所示。

功能信息发布描述	用户发布上映电影信息	
用例目的	测试信息发布功能的正确性	
前提条件	用户已注册并登录、用户拥有相应权限	
输入/动作	期望的输出/响应	实际情况
输入不完整的必填信息	提示错误	成功

输入完整的必填信息	成功发布信息	成功
点击“重置”	清空已输入的信息	成功
提交后	跳转电影信息管理界面	成功
功能信息修改/删除描述	用户修改/删除已发布的电影信息	
用例目的	测试信息修改/删除功能	
前提条件	用户已注册并登陆、用户拥有相应权限	
输入/动作	期望的输出/响应	实际情况
删除所选信息	提示：确定要删除吗？	成功
删除后	刷新个人信息管理界面	成功
修改所选信息	进入信息填写界面	成功
提交后	修改成功	成功
功能信息查找描述	用户通过关键字对电影信息查找	
用例目的	测试信息查找功能	
前提条件	用户已注册并登录	
输入/动作	期望的输出/响应	实际情况
输入关键字、选项	显示符合条件的信息列表	成功
平台内无任何信息发布，仍查找信息	显示空的信息列表	成功

表 3.1 电影信息管理模块测试报告

4 主界面模块测试

4.1 主界面模块测试用例

管理员操作界面：

管理员权限下可以对全局用户订单和用户信息进行管理。



图 4.1 管理员操作界面

普通用户操作界面：

普通用户权限下可以查看上映的电影、购买电影票。



图 4.2 普通用户操作界面

用户个性化设置界面：

针对用户的个人喜好可以改变前端界面的样式，这里只实现了改变界面的颜色。



图 4.3 个性化样式选择界面

登出界面：



图 4.4 登出操作界面

4.2 主界面模块测试报告

功能描述	用户使用该系统进行操作的显示页面	
用例目的	测试主页面功能的正确性	
前提条件	账号已经提前注册并在登录状态下操作	
输入/动作	期望的输出/响应	实际情况
下拉菜单中选择功能界面	跳转到相应界面	成功
改变界面的样式	改变界面的显示颜色	成功
点击登出按钮	安全退出系统	成功

表 4.1 主界面模块测试报告

5 小 结

分布式电影订票系统在传统单机系统的基础上,利用 Mycat 中间件实现分布式数据库的部署,其中主要实现分库分表功能来提高系统的查询效率和安全性。此外我们利用日中成功实现了主从数据库的设置,提高了系统的安全性和可用性。系统测试文档针对该系统的各个模块进行测试,分析并检验各个模块的执行情况,此过程关系到系统真正的使用,因此至关重要。