## 2.2 模拟法庭模块设计

### 2.2.1 设计目的和原则

#### 2.2.1.1设计目的

学生选择典型案例，模拟担当法官、公诉人、原告、被告、代理人、证人、书记员等各种角色，通过对所选择案件的模拟庭审全过程，使学生熟悉司法审判的实际程序，熟悉与案件相关的实体法和程序法，加深学生对有关程序法的准确理解，锻炼学生在真实的环境里从事法律实务的能力；通过“亲身、亲历”参与模拟法庭活动，培养和锻炼学生发现问题、分析问题和解决问题的能力，提高学生语言表达能力、组织协调能力，提升学生的专业技能和实践技能，同时还检验学生专业知识学习成果、法学理论掌握程度，学生反应能力与思辩能力等

#### 2.2.1.2设计原则

1. **开闭原则。**

即一个软件实体应该对外扩展开发，对内修改关闭。

用抽象构建框架，用实现扩展细节。不以改动原有类的方式来实现新需求，而是应该以实现事先抽象出来的接口的方式来实现。

优点：在于可以在不改动原有代码的前提下给程序扩展功能。增加了程序的可扩展性，同时也降低了程序的维护成本。

1. **单一职责原则。**

即一个类只允许有一个职责，只有一个导致该类变更的原因。

如果一个类具有多种职责，就会有多种导致这个类变化的原因，从而导致这个类的维护变得困难。在软件开发中，随着需求的不断增加，可能会给原来的类添加一些本来不属于它的一些职责，从而违反了单一职责原则。如果我们发现当前类的职责不仅仅有一个，就应该将本来不属于该类真正的职责分离出去。不仅仅是类，函数也要遵循单一职责原则。

优点：如果类与方法的职责划分的很清晰，不但可以提高代码的可读性，更实际性地更降低了程序出错的风险，降低了程序的维护成本。

1. **依赖倒置原则。**

依赖抽象而不是依赖实现。抽象不应该依赖细节，细节应该依赖抽象。高层模块不能依赖低层模块，二者都应该依赖抽象。

针对接口编程，而不是针对实现编程。尽量不要从具体的类派生，而是以继承抽象类或实现接口来实现。关于高层模块与低层模块的划分可以按照决策能力的高低进行划分。业务层自然就处于上层模块，逻辑层和数据层自然就归类为底层。

优点：通过抽象来搭建框架，建立类和类的关联，以减少类间的耦合性。而且以抽象搭建的系统要比以具体实现搭建的系统更加稳定，扩展性更高，同时也便于维护。

1. **迪米特法则**

一个对象应该对尽可能少的对象有接触，也就是只接触那些真正需要接触的对象。

一个类应该只和它的成员变量，方法的输入，返回参数中的类作交流，而不应该引入其他的类。

优点：实践迪米特法则可以良好地降低类与类之间的耦合，减少类与类之间的关联程度，让类与类之间的协作更加直接。

1. **里氏替换原则。**

所有引用基类的地方必须能透明地使用其子类的对象，也就是说子类对象可以替换其父类对象，而程序执行效果不变。

在继承体系中，子类中可以增加自己特有的方法，也可以实现父类的抽象方法，但是不能重写父类的非抽象方法，否则该继承关系就不是一个正确的继承关系。

优点：可以检验继承使用的正确性，约束继承在使用上的泛滥。

### 2.2.2 设计思想

#### 2.2.2.1软件体系结构风格

风格是一种带有倾向性的模式，本软件采用的是**浏览/服务器（B/S）风格**。具体结构为：浏览器/Web服务器/数据库服务器。

在这种架构下，用户工作界面通过WWW浏览器来实现，极少部分[事务](https://baike.baidu.com/item/%E4%BA%8B%E5%8A%A1" \t "https://baike.baidu.com/item/BS%E6%9E%B6%E6%9E%84/_blank)逻辑在前端实现，但是主要事务逻辑在服务器端实现，形成三层3-tier结构，WEB浏览器是客户端最主要的应用软件。用户通过客户端上的浏览器输入相应网址，即可打开软件《一锤定音》，使用模拟法庭功能创建法庭房间和加入法庭房间，选取自己所想扮演的法庭评审角色，也可以选取自己所需模拟的具体案件并开始模拟法庭的运作。系统安装、修改和维护全在服务器端解决，这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器，服务器安装[Oracle](https://baike.baidu.com/item/Oracle" \t "https://baike.baidu.com/item/BS%E6%9E%B6%E6%9E%84/_blank)、[Sybase](https://baike.baidu.com/item/Sybase" \t "https://baike.baidu.com/item/BS%E6%9E%B6%E6%9E%84/_blank)、[Informix](https://baike.baidu.com/item/Informix" \t "https://baike.baidu.com/item/BS%E6%9E%B6%E6%9E%84/_blank)或 [SQL Server](https://baike.baidu.com/item/SQL Server" \t "https://baike.baidu.com/item/BS%E6%9E%B6%E6%9E%84/_blank)等数据库。浏览器通过[Web Server](https://baike.baidu.com/item/Web Server" \t "https://baike.baidu.com/item/BS%E6%9E%B6%E6%9E%84/_blank)同数据库进行数据交互。这样就大大简化了客户端电脑载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本。

#### 2.2.2.2软件体系结构框架

框架是特定应用领域问题的体系结构模式，本软件采用的是**J2EE体系结构框架**，J2EE是分层结构，中间的表示层、业务层、集成层包含应用程序构件，客户层和资源层处于应用程序的外围。

客户层：用户通过客户层与系统交互，本软件采用的是浏览/服务器（B/S）风格，因此客户层为Web浏览器客户端。

资源层：在本软件中，资源层为服务器上的数据库。

表示层：用户通过表示层来访问应用程序。在本软件中，表示层由用户界面和运行于Web服务器上的过程组成。

业务层：业务层负责确认和执行业务规则和事务，并管理从资源层加载到应用程序高速缓存中的业务对象，主要表现为确认软件各项功能是否正确运行。

集成层：负责建立和维护与数据源的连接，主要体现在通过JDBS与数据库进行通信。

#### 2.2.2.3设计模式

设计模式为软件系统的子系统、构件或者构件之间的关系提供一个精练之后的解决方案，描述了在特定环境下，用于解决通用软件设计问题的构件以及这些构件相互通信时的各种结构。本软件采用的是**观察者模式**。

观察者模式中关键的对象为主题和观察者，它定义了对象间的一种一对多的依赖关系，当主题的状态发生改变时，所有依赖于它的观察者都得到通知，并被自动更新。在本软件中，所有使用模拟法庭功能的用户都是主题，同时也是其他用户的观察者。观察者模式降低了目标与观察者之间的耦合关系，两者之间是抽象耦合关系，符合依赖倒置原则。并且在目标与观察者之间建立了一套触发机制。

### 2.2.3 基本处理流程

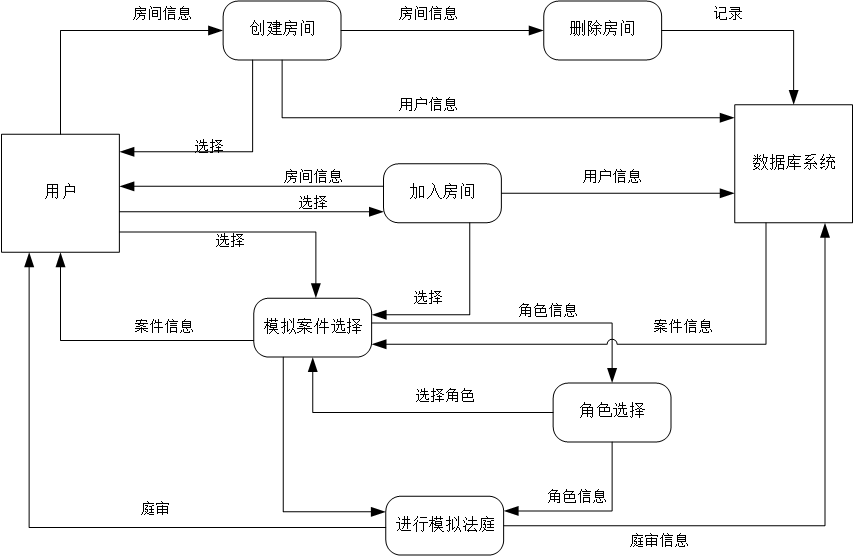


图 1模拟法庭功能数据流图

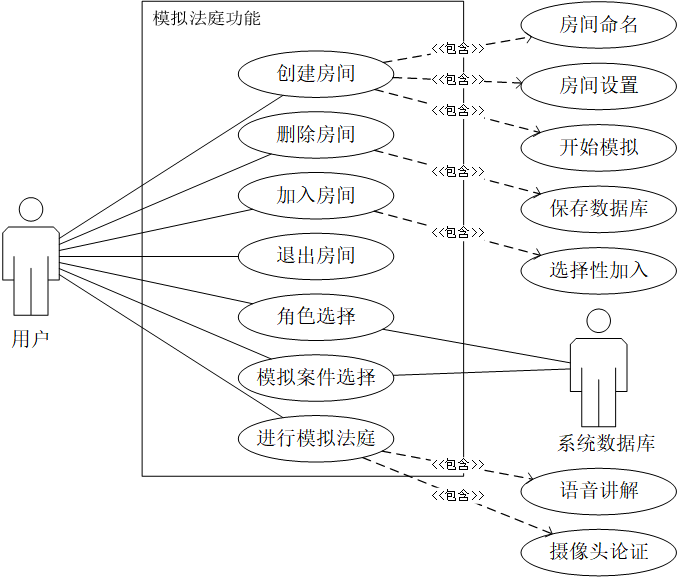


图 2模拟法庭功能用例图

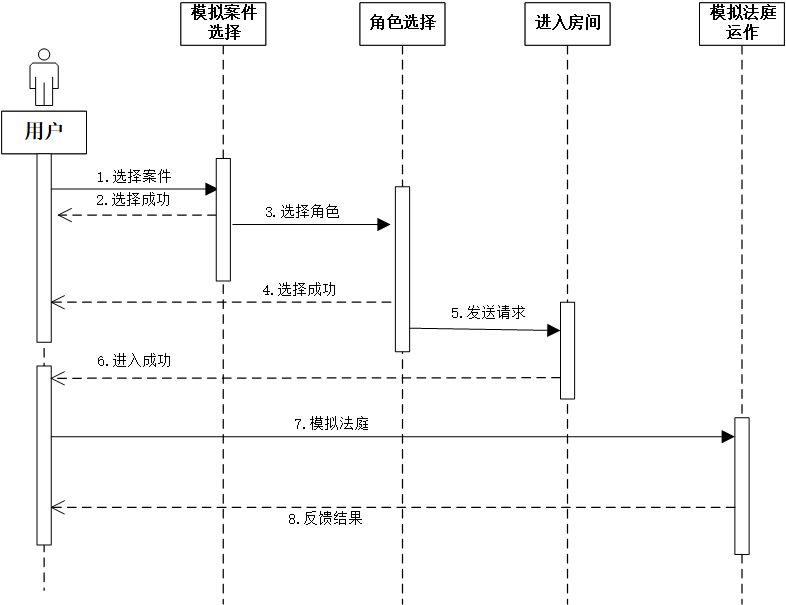


图 3模拟法庭功能顺序图

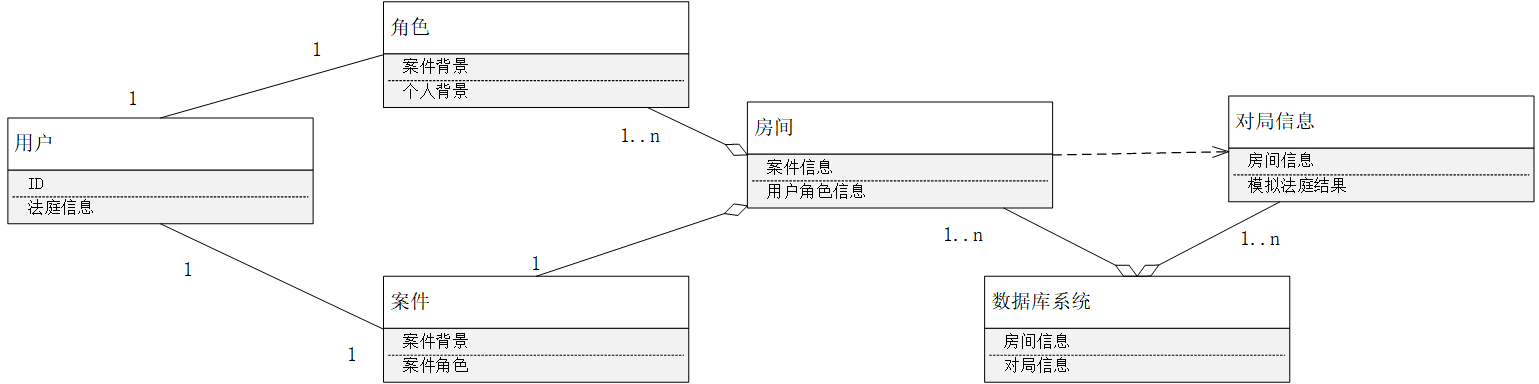


图 4模拟法庭功能类图

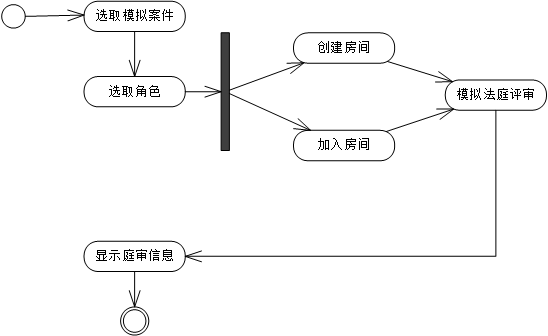


图 5模拟法庭功能活动图

### 2.2.4 数据字典

1.数据项

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 数据项名 | 数据项含义 | 数据类型 | 取值范围 | 取值含义 |
| A1 | 用户昵称 | 用户在软件中展示的昵称 | Char |  | 用户自拟 |
| A2 | 用户ID | 每位用户的唯一标识 | Char | 0~999999 | 随机生成用户的唯一ID |
| A3 | 用户类型 | 整型 | 1位 | ‘0‘或’1‘ | ‘0’为管理员，  ‘1‘为普通用户 |
| A4 | 案件标题 | 每个案件的标题 | Char |  | 对案件的分类及梗概 |
| A5 | 案件角色 | 案件中每个人物的姓名 | Char |  | 标识不同角色 |
| A6 | 案件信息 | 每个案件的案情 | Char |  | 对案情的描述 |
| A7 | 案件角色信息 | 案件中每个人物的简介 | Char |  | 标识不同角色 |
| A8 | 庭审角色 | 庭审时法庭中的各项角色 | Char |  | 发挥每位庭审角色特定职能 |
| A9 | 房间信息 | 案件信息及用户信息 | Char |  | 记录此房间的各项信息 |
| A10 | 庭审结果 | 由法官宣布的最终评审结果 | Char |  | 庭审结果显示本案获胜方 |

1. 数据流

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 数据流名 | 组成 | 来源 | 去向 |
| F1 | 用户信息 | A1+A2+A3 | E1+E3+E4 | C1 |
| F2 | 案件 | A4+A5+A6+A7 | E2 | C2 |
| F3 | 角色 | A6+A7 | E2 | C2 |
| F4 | 房间 | A1+A2+A3+A4+A5+A6+A7 | E1+E2+E3+E4 | C1+C2 |
| F5 | 庭审 | A1+A2+A3+A4+A5+A6+A7+A8+A9 | E1+E2+E3+E4 | C1+C2 |

1. 数据存储

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 数据项名称 | 组成 | 组成方式 | 查询要求 | 简述 |
| C1 | 用户表 | A1+A2+A3 | A1升序 | 立即 | 暂无 |
| C2 | 案件表 | A4+A5+A6+A7 | A4升序 | 立即 | 暂无 |
| C3 | 庭审表 | A1+A2+A3+A4+A5+A6+A7+A8+A9 | A1升序 | 立即 | 暂无 |

1. 外部输入

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 数据项名称 | 输出数据流 | 简述 |
| E1 | 用户 | F1+F4+F5 | 暂无 |
| E2 | 案件信息 | F2+F3+F4+F5 | 暂无 |
| E3 | 注册系统 | F1 | 暂无 |
| E4 | 登录系统 | F1 | 暂无 |