

**《Web课程设计》项目设计报告**



**团队编号**  101

**团队名称**  海报突击队

**院 系**  机械与电气工程学院

**专 业**  计算机科学与技术

**团队成员** 陈俊龙、顾煜、侯志宁

**团队成员** 桂婷、邓棉茵、洪志炫

**填写日期** 2024-07-12

# 大学生海报设计竞赛系统项目设计

## **1.1** 需求分析

### **1.1.1 项目背景**

为了展示大学生的艺术才华，激发其创新意识和实践能力，我们决定开发一套大学生海报设计竞赛系统。大学生可以创建或加入队伍进行竞赛、查看赛事、查看作品、分享作品、点赞作品等。管理员可以发布竞赛、设置获奖、发布新闻、数据统计等。评委可以查看作品，设置分数，设置评语等。

为了方便，我们给这个管理系统取一个英文缩写名称：PDCS，也就是Poster Design Competition System系统。开设的海报设计竞赛系统希望达到如表1-1所示目的。

表1-1 PDCS业务前景表

|  |  |
| --- | --- |
| **接口** | **目标** |
| P01 | 大学生可以创建或加入队伍进行竞赛、查看赛事、查看作品、分享作品、点赞作品等。 |
| P02 | 管理员可以发布竞赛、设置获奖、发布新闻、数据统计等。 |
| P03 | 评委可以查看作品，设置分数，设置评语等。 |

### **1.1.2 业务流程分析**

首先通过和项目相关人员访谈、收集整理资料来掌握现有业务流程，然后结合业务前景对流程进行优化。下面分析PDCS项目核心业务流程，注意业务流程分析只聚焦做什么，而不是如何做。图1-1是PDCS项目的核心业务流程图，重点在于对业务情况的整体梳理。

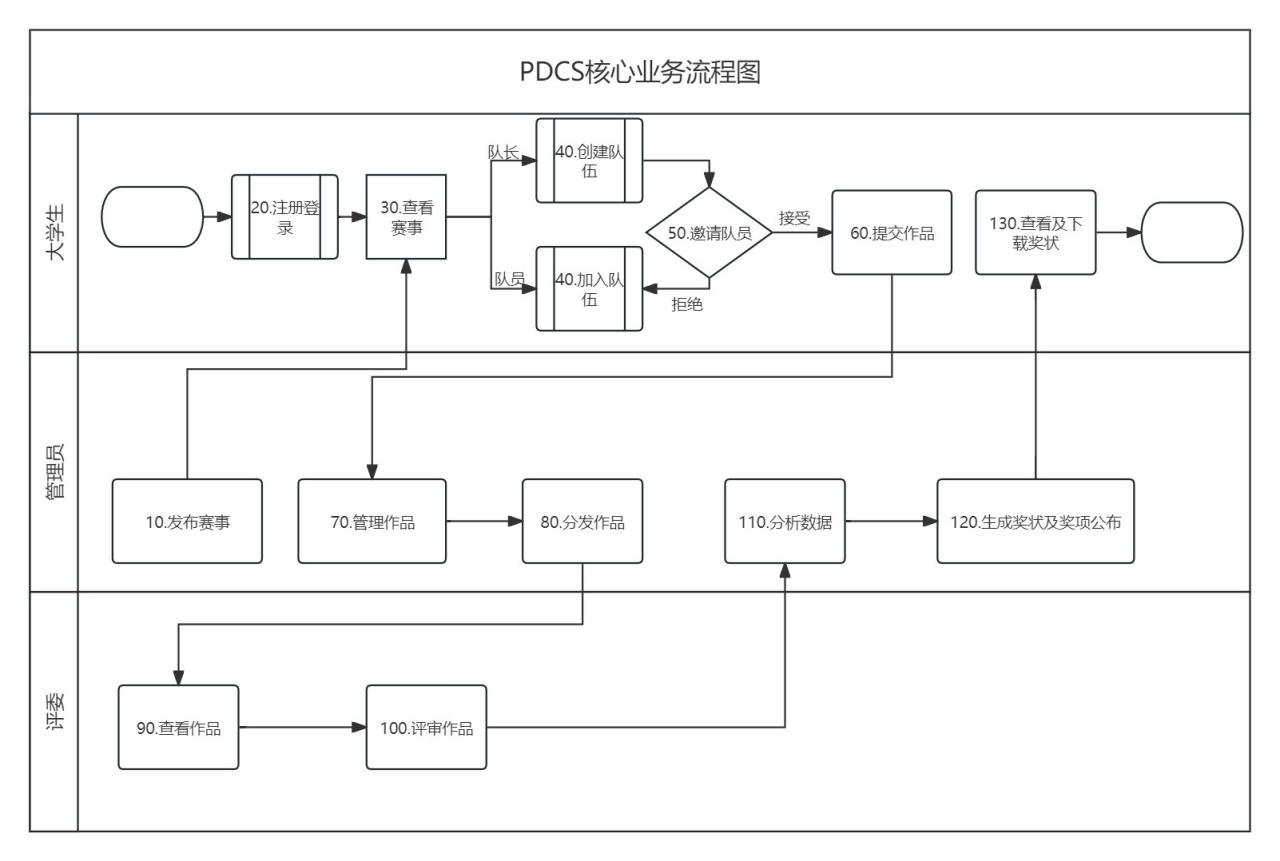


图1-1 核心业务流程图

项目中大量细节的内容，用流程图来表示会显得非常繁琐，此时可以用表1-2的方式对流程图进行描述。为了方便将描述和流程图对照起来，图1-1中给图标编了号。

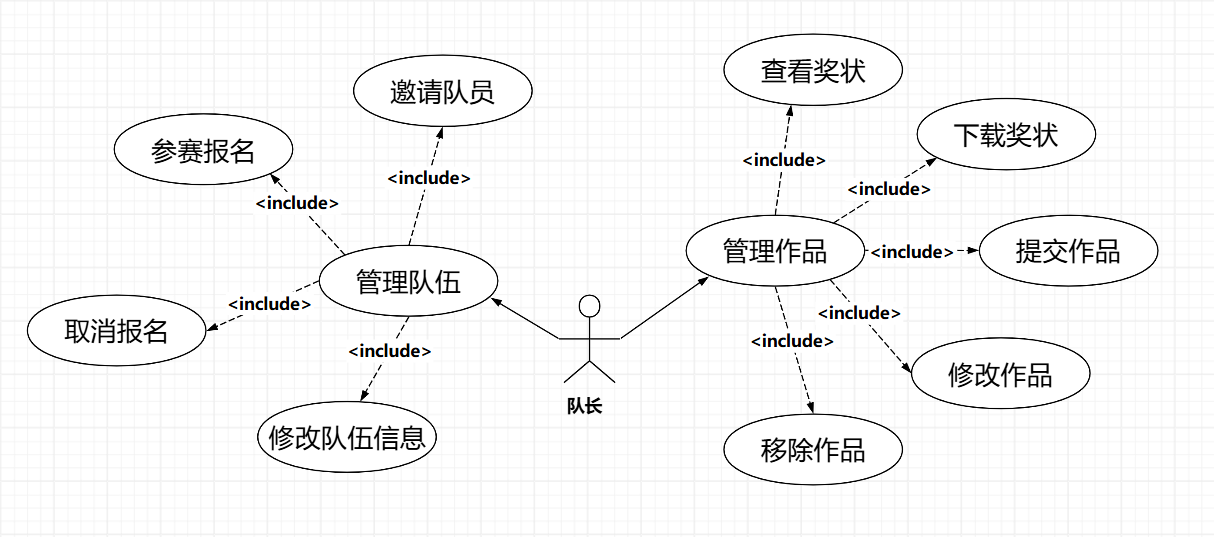
表1-2 业务流程描述

|  |  |  |
| --- | --- | --- |
| **编号** | **责任人** | **说明** |
| 10 | 管理员 | 管理员发布赛事 |
| 20 | 大学生 | 大学生注册并登录竞赛系统 |
| 30 | 大学生 | 大学生可以查看赛事 |
| 40 | 大学生 | 大学生分为队长和队员，队长创建队伍，队员加入队伍 |
| 50 | 大学生 | 大学生中队长身份可以邀请队员加入队伍 |
| 60 | 大学生 | 大学生邀请成功组建队伍可以提交作品 |
| 70 | 管理员 | 管理员对提交上来的作品进行管理 |
| 80 | 管理员 | 管理员将作品分发给评委 |
| 90 | 评委 | 评委可以查看分发的作品 |
| 100 | 评委 | 评委对作品可以进行评审 |
| 110 | 管理员 | 管理员对评委评审完的作品进行分析 |
| 120 | 管理员 | 管理员给作品生成奖状及奖项公布 |
| 130 | 大学生 | 大学生可以查看奖状及下载 |

### **1.1.3 用例分析**

根据业务流程，结合前景分析得到PDCS项目的用例。由于用例较多，为了避免单个用例图过于复杂，根据业务逻辑将用例划分到多个用例图。

大学生用例图给出针对大学生这个角色的用例图，如图1-2所示。



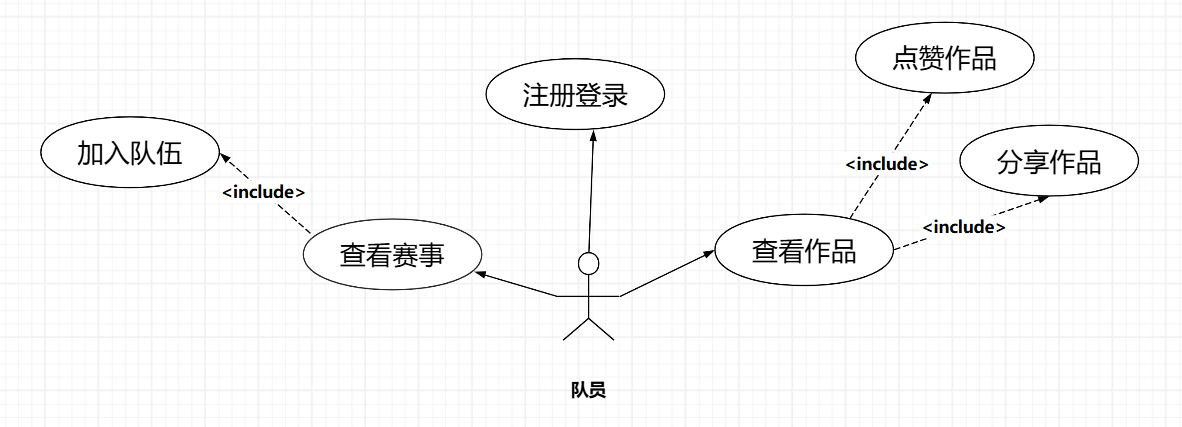


图1-2 大学生用例图

图1-3给出了后台管理用例图，用例名称通常用动词。

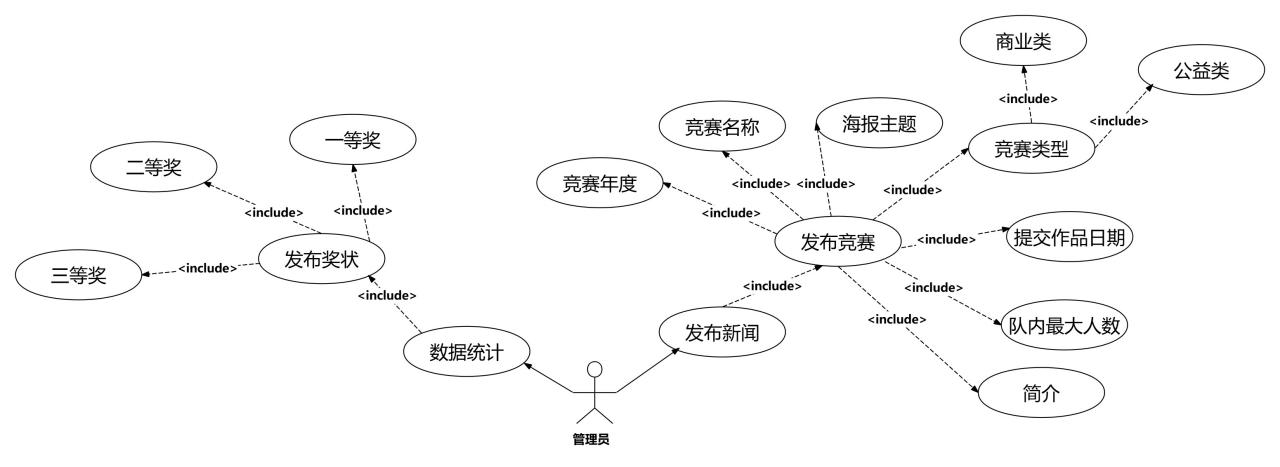


图1-3 后台管理用例图

图1-4给出了评委用例图，用例名称通常用动词。

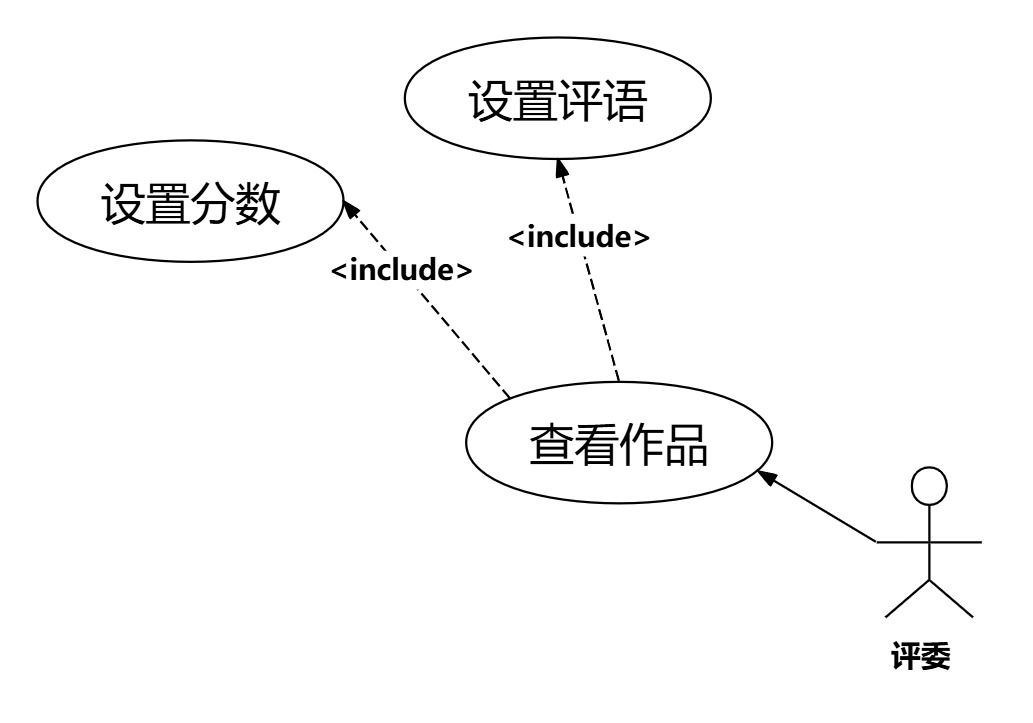


图1-4 评委用例图

## **1.2** 系统设计

### **1.2.1 功能模块设计**

图1-5是PDCS项目功能模块图，由于功能模块比较多，所以采用分层设计，将所有功能分为“基础功能”、“学生功能”、“管理员功能”、“评委功能”和“竞赛中心”五大部分。



图1-5功能模块图

表1-3给出了PDCS项目功能模块简要说明。

表1-3 功能模块说明

|  |  |  |
| --- | --- | --- |
| **模块** | **功能** | **功能描述** |
| 基本功能 | 用户注册 | 所有人都可以注册成为参与人员、管理员、评委 |
| 登录/注销 | 参与人员登录海报设计竞赛系统和退出海报设计竞赛系统 |
| 密码修改 | 所有人都可以修改登录的密码 |
| 队长功能 | 创建队伍 | 队长可以创建队伍 |
| 报名参赛 | 队长可以报名参赛 |
| 邀请队员 | 队长可以邀请队员 |
| 修改队伍信息 | 队长可以修改队伍信息 |
| 取消报名 | 队长可以取消报名 |
| 修改作品 | 队长可以修改作品 |
| 移除作品 | 队长可以移除作品 |
| 提交作品 | 每个队长都可以提交参加比赛的作品 |
| 下载奖状 | 队长可以下载奖状 |
| 提交团队基本信息 | 队长可以提交团队基本信息 |
| 选择参赛子类 | 队长可以选择参赛子类 |
| 添加团队成员 | 队长可以添加团队成员 |
| 踢出队员 | 队长可以踢出队员 |
| 查看奖状 | 每个队长都可以查看自己队伍是否获得奖状 |
| 队员功能 | 注册登录 | 队员可以注册登录 |
| 加入队伍 | 队员可以加入队伍 |
| 查看赛事 | 队员可以查看赛事 |
| 分享作品 | 队员可以分享作品 |
| 点赞作品 | 队员可以点赞作品 |
| 查看作品 | 队员可以查看作品 |
| 管理员功能 | 赛事发布 | 管理员能够发布竞赛、发布获奖情况的新闻 |
| 发布奖项 | 管理员能够发布奖项 |
| 发布新闻 | 管理员能够发布新闻 |
| 数据统计 | 管理员能够数据统计 |
| 设置获奖 | 管理员可以根据评委的打分设置获奖 |
| 评委功能 | 设置分数 | 评委可以给作品打分 |
| 查看作品 | 评委可以查看作品 |
| 设置评语 | 评委可以给作品设置评语 |
| 竞赛中心 | 竞赛明细 | 包含了竞赛的详细介绍 |
| 竞赛类型 | 包含了竞赛的类型 |

### **1.2.2 MVC开发模式**

MVC是一种软件的开发模式，目前主流的软件系统开发都受到这种思想的指导。

M即模型（Model）是表示系统业务处理相关代码组件的集合。在MVC的三个部件中，模型拥有最多的处理任务。同一个模型能为多个视图提供数据，由于应用于模型的代码只需写一次就可以被多个视图重用，所以减少了代码的重复性。

V即视图（View）是指用户看到并与之交互的界面。比如由HTML元素组成的网页界面，或者软件的客户端界面。MVC的好处之一在于它能为应用程序处理很多不同的视图。在视图中其实没有真正的处理发生，它只是作为一种输出数据并允许用户操作的方式。

C即控制器（Controller），控制器接受用户的输入并调用模型和视图去完成用户的需求，控制器本身不输出任何东西和做任何处理。它只是接收请求并决定调用哪个模型去处理请求，然后再确定用哪个视图来显示返回的数据。

MVC开发模式有如下四个优点：

1. 耦合性低

视图和模型分离，这样就允许更改视图层代码而不用重新编译模型和控制器代码，同样，一个应用的业务流程或者业务规则的改变只需要改动模型层即可。因为模型与控制器和视图相分离，所以很容易改变应用程序的模型层和业务规则。

1. 重用性高

MVC模式允许使用各种不同样式的视图来访问同一个服务器端的代码，因为多个视图能共享一个模型，它包括任何Web浏览器或者无线浏览器（wap）。由于模型返回的数据没有与界面代码混合，所以同样数据能被不同的界面使用。

1. 开发效率高，生命周期成本低

MVC模式下，开发和维护接口代码的技术含量降低。使用MVC模式使开发时间得到相当大的缩减，它使开发人员（Java开发人员）聚焦于业务逻辑，界面程序员（HTML和JSP开发人员）集中精力于表现形式上。

1. 可维护性高

MVC模式下各个层次耦合性低，有利于项目的后期维护和修改。

PDCS项目，采用Java Web开发技术，结合MVC开发模式，项目中各个组件职责划分如图1-6所示。

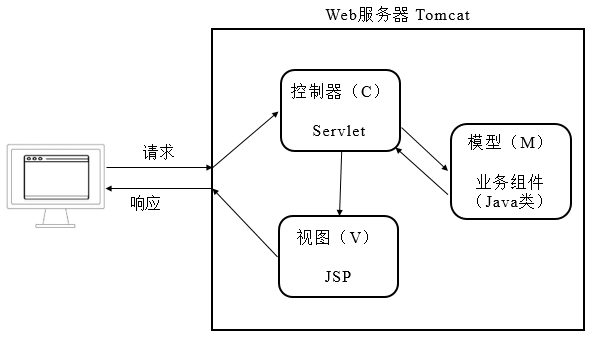


图1-6 MVC开发模式

Servlet组件承担了控制器（Controller）的角色，它们的常规工作流程如下：

* 接收客户端的请求信息。
* 调用执行业务模型，获取其执行后返回的数据。
* 把数据传递给视图，将视图展示给客户端。

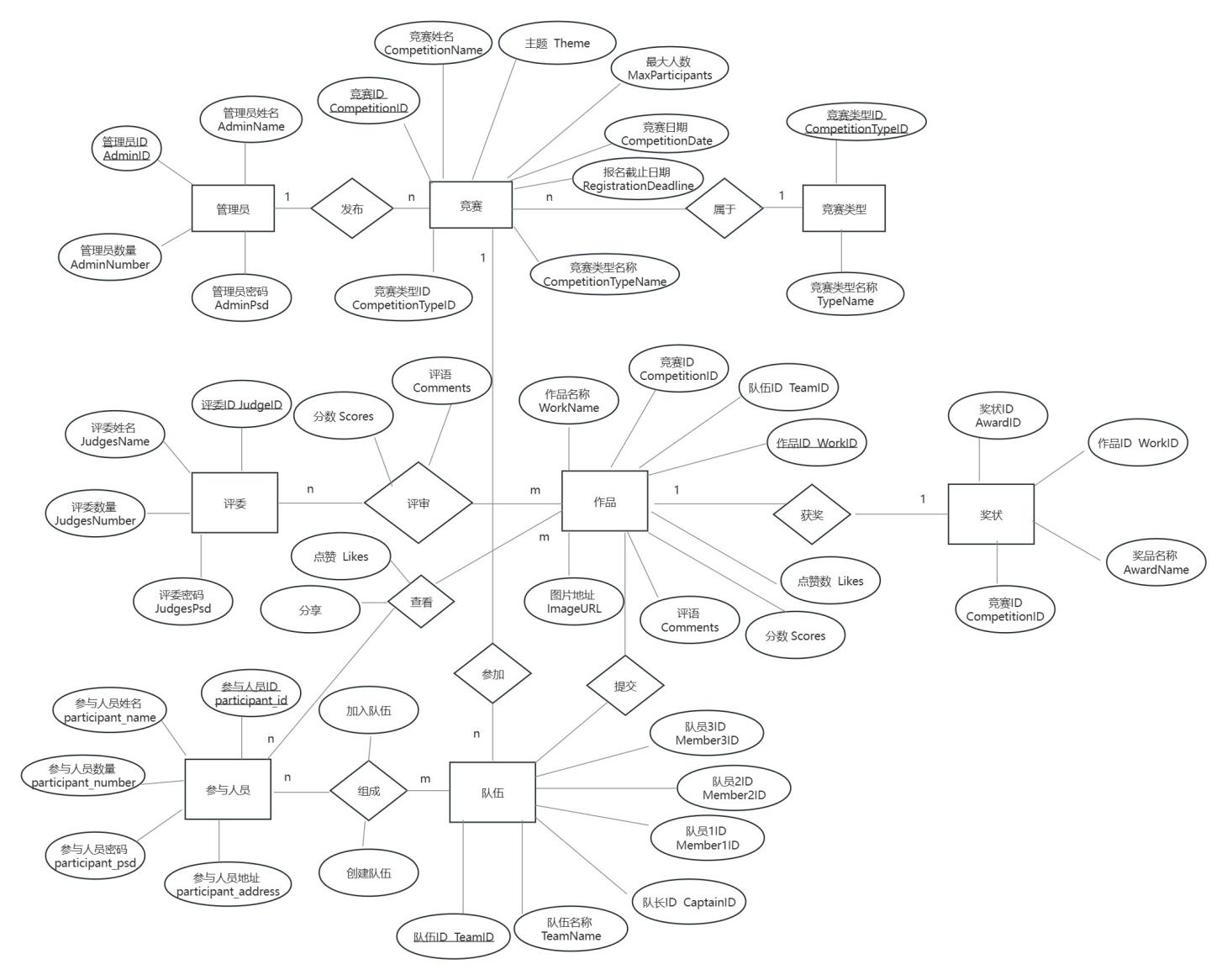
JSP组件承担了视图（View）的角色，它们的主要工作是使用EL或者JSTL在页面中展示数据。项目在后续加入JQuery框架的Ajax请求时，会使用JavaScript在页面上展示数据。

项目中其他的组件，例如数据库存取类、业务数据类、业务逻辑类、工具类等内容，承担了模型（Model）的角色，它们的主要工作是处理项目的业务逻辑，返回结果数据。

## **1.3 数据库设计**

### **1.3.1 概念模型**

根据前面的需求分析和系统设计，可以得到如图1-7所示的ER图和如图1-8所示的实体类图。



101

图1-7 ER图

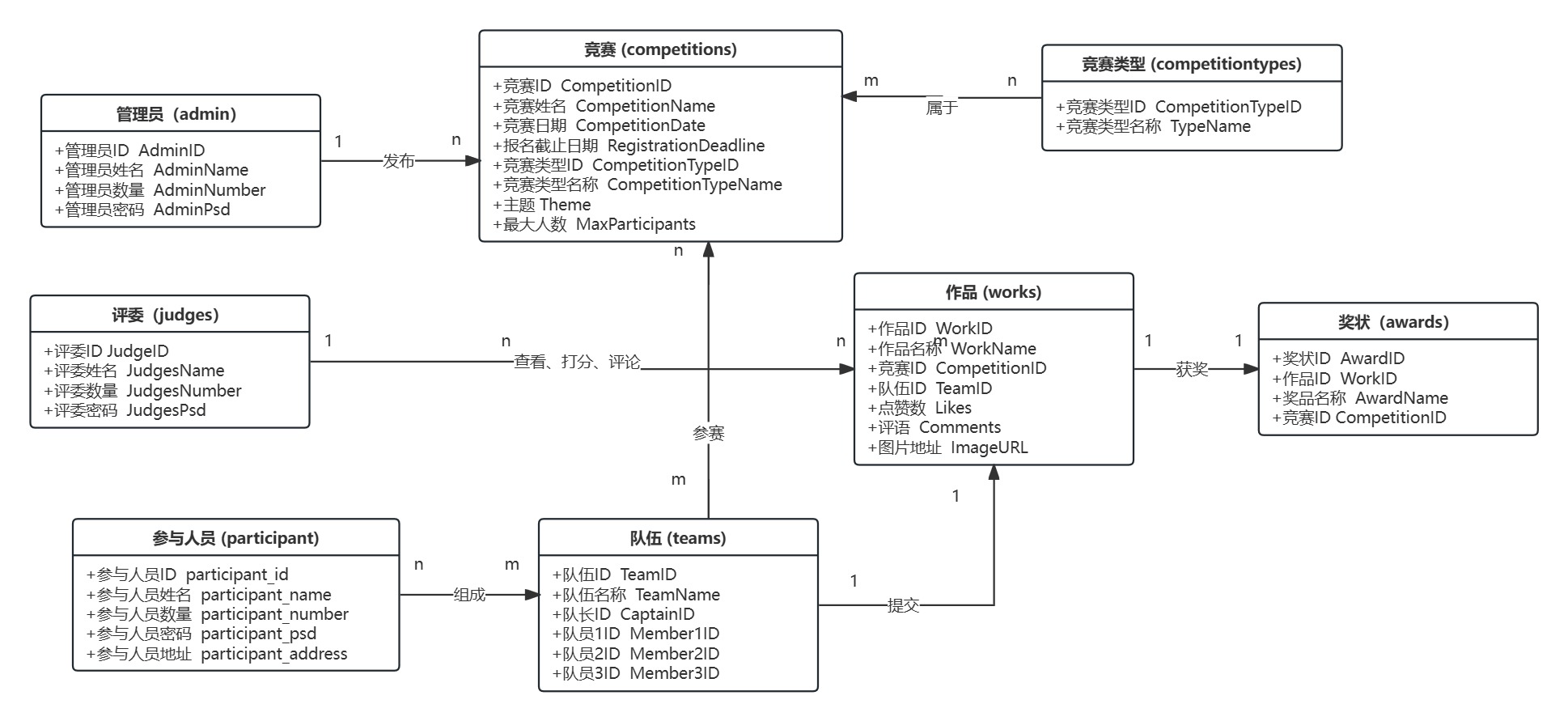


图1-8实体类图

概念模型设计过程中的实体类图需要确保满足具体的业务需求：

* 管理员可以发布多个竞赛
* 评委可以给查看多个作品并进行打分和评论
* 参与人员可以组成多个队伍参加多个竞赛
* 一支队伍在一个竞赛中只能提交一份作品
* 一份作品最多获得一张奖状

### **1.3.2 关系模型**

PDCS项目关系模型采用表格方式描述，关系模型的设计基于需求分析和实体类图，特别注意关系模型如何体现实体类之间的关系，也就是表中外键字段的说明。

如表1-4所示，管理员表（admin）保存系统中所有管理员的信息。

表1‑4管理员表admin

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 属性 | 说明 |
| AdminID | Int | PK，IDENTITY | 管理员Id。(自增型) |
| AdminName | Varchar(50) | NOT NULL | 管理员姓名 |
| AdminNumber | Varchar(50) | NOT NULL | 管理员数量 |
| AdminPsd | Varchar(50) | NOT NULL | 管理员密码 |

如表1-5所示，奖状表（awards）保存奖状信息。

表1‑5奖状表awards

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 属性 | 说明 |
| AwardID | Int | PK，IDENTITY | 奖状ID。(自增型) |
| WorkID | Int | FK,NOT NULL | 作品ID。 |
| AwardName | Varchar (100) | NOT NULL | 奖品名称。 |
| CompetitionID | Int | FK,NOT NULL | 竞赛ID。 |

如表1-6所示，竞赛表（competitions）保存系统中所有竞赛的基本信息。

表1‑6竞赛表competitions

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 属性 | 说明 |
| CompetitionID | Int | PK，IDENTITY | 竞赛ID。(自增型) |
| CompetitionName | Varchar(255) | NOT NULL | 竞赛姓名 |
| CompetitionDate | DateTime | NOT NULL | 竞赛日期 |
| RegistrationDeadline | DateTime | NOT NULL | 报名截止日期 |
| CompetitionTypeID | Int | DEFAULT NULL, | 竞赛类型ID |
| CompetitionTypeName | Varchar(255) | DEFAULT NULL, | 竞赛类型名称 |
| CompetitionDescription | Text | NOT NULL | 竞赛描述 |
| MaxParticipants | Int | DEFAULT NULL, | 最大人数 |
| Theme | Varchar(255) | DEFAULT NULL, | 主题 |

如表1-7所示，竞赛类型表（competitiontypes）保存竞赛类型信息，用于竞赛的管理。

表1‑7 竞赛类型表competitiontypes

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 属性 | 说明 |
| CompetitionTypeID | Int | PK，IDENTITY | 竞赛类型ID。(自增型) |
| TypeName | Varchar(255) | NOT NULL | 竞赛类型名称 |

如表1-8所示，评委表（judges）保存评委信息。

表1‑8评委表judges

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 属性 | 说明 |
| JudgeID | Int | PK，IDENTITY | 评委ID。(自增型) |
| JudgesName | Varchar(255) | NOT NULL | 评委姓名 |
| JudgesNumber | Varchar(255) | NOT NULL | 评委数量 |
| JudgesPsd | Varchar(255) | NOT NULL | 评委密码 |

如表1-9所示，参与人员表（participant）保存参与人员信息。

表1‑9参与人员表participant

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 属性 | 说明 |
| participant\_id | Int | PK，IDENTITY | 参与人员ID。(自增型) |
| participant\_name | Varchar(255) | NOT NULL | 参与人员姓名 |
| participant\_number | Varchar(50) | NOT NULL | 参与人员数量 |
| participant\_psd | Varchar(50) | NOT NULL | 参与人员密码 |
| participant\_address | Varchar(255) | NOT NULL | 参与人员地址 |

如表1-10所示，队伍表（teams）保存队伍信息。

表1‑10队伍表teams

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 属性 | 说明 |
| TeamID | Int | PK，IDENTITY | 队伍ID。(自增型) |
| TeamName | Varchar(255) | NOT NULL | 队伍名称 |
| CaptainID | Int | PK,NOT NULL | 队长ID |
| Member1ID | Int | PK,NOT NULL | 队员1ID |
| Member2ID | Int | PK,NOT NULL | 队员2ID |
| Member3ID | Int | PK,NOT NULL | 队员3ID |
| CompetitionID | Int | NOT NULL | 竞赛ID |

如表1-11所示，作品表（works）保存作品信息。

表1‑11作品表works

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 属性 | 说明 |
| WorkID | Int | PK，IDENTITY | 作品ID。(自增型) |
| WorkName | Varchar(255) | NOT NULL | 作品名称 |
| CompetitionID | Int | NOT NULL | 竞赛ID |
| TeamID | Int | NOT NULL | 队伍ID |
| Likes | Int | NOT NULL | 点赞数 |
| Comments | Text | PK,NOT NULL | 评语 |
| ImageURL | Varchar(255) | NOT NULL | 图片地址 |
| Score | Int | NOT NULL | 作品分数 |

### **1.3.2 物理设计**

完成关系模型的设计后，针对所采用的数据库系统完成物理设计，也就是确定数据库的存储结构、确定表结构等内容。

首先确定数据库系统采用MySQL，数据库名为pdcs。下面给出具体的数据库创建脚本代码，开发人员在数据库客户端中执行pdcs.sql文件中的代码即可完成数据库物理实现。

源程序：pdcs.sql数据库脚本文件

-- 创建数据库

CREATE DATABASE PDCS charset = utf8;

USE PDCS;

-- 创建管理员表

CREATE TABLE admin(

AdminID INT(11) NOT NULL AUTO\_INCREMENT,

AdminName VARCHAR(255) NOT NULL,

AdminNumber VARCHAR(255) NOT NULL,

AdminPsd VARCHAR(255) NOT NULL,

PRIMARY KEY (AdminID),

UNIQUE KEY AdminNumber (AdminNumber)

) ENGINE=INNODB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8;

-- 创建奖状表

CREATE TABLE awards(

AwardID INT(11) NOT NULL AUTO\_INCREMENT,

WorkID INT(11) NOT NULL,

AwardName VARCHAR(255) NOT NULL,

CompetitionID INT(11) NOT NULL,

Winner1ID INT(11) DEFAULT NULL,

Winner2ID INT(11) DEFAULT NULL,

Winner3ID INT(11) DEFAULT NULL,

Winner4ID INT(11) DEFAULT NULL,

AwardImageURL varchar(255) DEFAULT NULL,

PRIMARY KEY (AwardID)

) ENGINE=INNODB DEFAULT CHARSET=utf8;

-- 创建竞赛类型表

CREATE TABLE competitiontypes(

CompetitionTypeID INT(11) NOT NULL AUTO\_INCREMENT,

TypeName VARCHAR(255) NOT NULL,

PRIMARY KEY (CompetitionTypeID)

) ENGINE=INNODB DEFAULT CHARSET=utf8;

-- 创建竞赛表

CREATE TABLE competitions(

CompetitionID INT(11) NOT NULL AUTO\_INCREMENT,

CompetitionName VARCHAR(255) NOT NULL,

CompetitionDate DATE NOT NULL,

RegistrationDeadline DATE NOT NULL,

CompetitionTypeID INT(11) DEFAULT NULL,

CompetitionTypeName VARCHAR(255) DEFAULT NULL,

CompetitionDescription TEXT NOT NULL,

MaxParticipants INT(11) DEFAULT NULL,

Theme VARCHAR(255) DEFAULT NULL,

PRIMARY KEY (CompetitionID),

KEY fk\_competition\_type (CompetitionTypeID),

CONSTRAINT fk\_competition\_type FOREIGN KEY (CompetitionTypeID) REFERENCES competitiontypes (CompetitionTypeID)

) ENGINE=INNODB DEFAULT CHARSET=utf8;

-- 创建评委表

CREATE TABLE judges(

JudgeID INT(11) NOT NULL AUTO\_INCREMENT,

JudgesName VARCHAR(255) NOT NULL,

JudgesNumber VARCHAR(255) NOT NULL,

JudgesPsd VARCHAR(255) NOT NULL,

PRIMARY KEY (JudgeID),

UNIQUE KEY JudgesNumber (JudgesNumber)

) ENGINE=INNODB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8;

-- 创建参赛人员表

CREATE TABLE participant(

participant\_id INT(11) NOT NULL AUTO\_INCREMENT,

participant\_name VARCHAR(255) NOT NULL,

participant\_number VARCHAR(50) NOT NULL,

participant\_psd VARCHAR(50) NOT NULL,

participant\_address VARCHAR(255) NOT NULL,

enter INT(11) DEFAULT NULL,

PRIMARY KEY (participant\_id)

) ENGINE=INNODB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8;

-- 创建队伍表

CREATE TABLE teams(

TeamID INT(11) NOT NULL AUTO\_INCREMENT,

TeamName VARCHAR(255) NOT NULL,

CaptainID INT(11) NOT NULL,

Member1ID INT(11) DEFAULT NULL,

Member2ID INT(11) DEFAULT NULL,

Member3ID INT(11) DEFAULT NULL,

CompetitionID INT(11) DEFAULT NULL,

PRIMARY KEY (TeamID),

KEY CaptainID (CaptainID),

KEY Member1ID (Member1ID),

KEY Member2ID (Member2ID),

KEY Member3ID (Member3ID),

CONSTRAINT teams\_ibfk\_1 FOREIGN KEY (CaptainID) REFERENCES participant (participant\_id),

CONSTRAINT teams\_ibfk\_2 FOREIGN KEY (Member1ID) REFERENCES participant (participant\_id),

CONSTRAINT teams\_ibfk\_3 FOREIGN KEY (Member2ID) REFERENCES participant (participant\_id),

CONSTRAINT teams\_ibfk\_4 FOREIGN KEY (Member3ID) REFERENCES participant (participant\_id)

) ENGINE=INNODB DEFAULT CHARSET=utf8;

-- 创建作品表

CREATE TABLE works(

WorkID INT(11) NOT NULL AUTO\_INCREMENT,

WorkName VARCHAR(255) NOT NULL,

CompetitionID INT(11) NOT NULL,

TeamID INT(11) NOT NULL,

Likes INT(11) DEFAULT '0',

Comments TEXT,

ImageURL VARCHAR(255) DEFAULT NULL,

Score INT(11) DEFAULT NULL,

PRIMARY KEY (WorkID),

KEY CompetitionID (CompetitionID),

KEY TeamID (TeamID),

CONSTRAINT works\_ibfk\_1 FOREIGN KEY (CompetitionID) REFERENCES competitions (CompetitionID),

CONSTRAINT works\_ibfk\_2 FOREIGN KEY (TeamID) REFERENCES teams (TeamID)

) ENGINE=INNODB DEFAULT CHARSET=utf8;

为了在系统开发过程中帮助开发人员进行代码调试，需在数据库中初始化一批测试数据，具体数据在测试数据脚本文件pdcs\_data.sql中给出。

源程序：pdcs\_data.sql测试数据脚本文件

USE PDCS;

-- 插入管理员数据

INSERT INTO admin VALUES ('1', 'Alice', 'admin@1234.com', '1234');

INSERT INTO admin VALUES ('2', 'Bob', 'admin@1213.com', '1211');

-- 插入竞赛类型数据

INSERT INTO competitiontypes (CompetitionTypeID, TypeName) VALUES (1, '公益类');

INSERT INTO competitiontypes (CompetitionTypeID, TypeName) VALUES (2, '国防类');

INSERT INTO competitiontypes (CompetitionTypeID, TypeName) VALUES (3, '商业类');

-- 插入评委数据

INSERT INTO judges VALUES ('1 ', 'Jhon','1224.com','1234');

INSERT INTO judges VALUES ('2 ', 'Mike','1432.com','9879');

-- 插入参与人员数据

INSERT INTO `participant` VALUES ('1', 'Jim', '3415.com', '1234', '二仙桥', '0');

INSERT INTO `participant` VALUES ('2', 'Kiko', '2456.com', '1111', '成华大道', '0');

INSERT INTO `participant` VALUES ('4', '朱杨华', '1267.com', '7890', '绍兴', '0');

INSERT INTO `participant` VALUES ('5', 'Lili', '9868.com', '6521', '天海市', '0');

INSERT INTO `participant` VALUES ('9', 'hlj', '1457.com', '1111', '楼外楼', '0');

INSERT INTO `participant` VALUES ('15', '刘德华', '7777.com', '1111', '香港', '0');

INSERT INTO `participant` VALUES ('16', '张三', '7891.com', '1234', '北京', '0');

INSERT INTO `participant` VALUES ('17', '李四', '1234.com', '1234', '上海', '0');

INSERT INTO `participant` VALUES ('18', '王五', '5678.com', '1234', '广州', '0');

INSERT INTO `participant` VALUES ('19', '赵六', '9012.com', '1234', '深圳', '0');

INSERT INTO `participant` VALUES ('20', '孙七', '3456.com', '1234', '杭州', '0');

INSERT INTO `participant` VALUES ('21', '周八', '7890.com', '1234', '成都', '0');

INSERT INTO `participant` VALUES ('22', '吴九', '1357.com', '1234', '武汉', '0');

INSERT INTO `participant` VALUES ('23', '郑十', '2460.com', '1234', '西安', '0');

INSERT INTO `participant` VALUES ('24', '钱十一', '1111.com', '1234', '南京', '0');

INSERT INTO `participant` VALUES ('25', '孙十二', '2222.com', '1234', '重庆', '0');

INSERT INTO `participant` VALUES ('26', '李白', '1999.com', '9999', '大唐', '0');

INSERT INTO `participant` VALUES ('27', '张三', 'lisi@qq.com', '123', 'aaaa', '0');

INSERT INTO `participant` VALUES ('28', '白起', '1666.com', '9999', 'aaaa', '0');

-- 插入奖状数据

INSERT INTO `awards` VALUES ('1', '1', '一等奖', '1', '1', '2', '4', '5', '1.png');

INSERT INTO `awards` VALUES ('2', '2', '一等奖', '5', '4', '9', '15', '16', '4.png');

INSERT INTO `awards` VALUES ('3', '3', '一等奖', '4', '5', '17', '18', '19', '3.png');

INSERT INTO `awards` VALUES ('4', '4', '一等奖', '6', '2', '20', '21', '22', '5.png');

INSERT INTO `awards` VALUES ('5', '5', '一等奖', '7', '9', '23', '24', '25', '6.png');

INSERT INTO `awards` VALUES ('6', '6', '一等奖', '10', '1', '4', '5', null, '7.png');

INSERT INTO `awards` VALUES ('7', '7', '一等奖', '13', '15', '9', '2', null, '8.png');

INSERT INTO `awards` VALUES ('8', '8', '一等奖', '19', '19', '18', '17', null, '9.png');

INSERT INTO `awards` VALUES ('9', '9', '一等奖', '2', '1', '2', '4', '5', '2.png');

-- 插入竞赛数据

INSERT INTO `competitions` VALUES ('1', '2022年第7届大学生海洋公益海报竞赛', '2022-10-09', '2022-07-10', '1', '公益类', '2022年第7届大学生海洋公益海报竞赛是一项旨在提升公众海洋保护意识的年度盛事。本次竞赛鼓励学生们运用创意设计，展现海洋生态的多样性与脆弱性，倡导可持续发展的理念。参赛者将通过海报这一直观媒介，传达保护海洋环境的紧迫性和重要性。我们诚邀全国在校大学生积极参与，用艺术的力量为海洋发声，共同守护我们的蓝色星球。', '4', '海洋');

INSERT INTO `competitions` VALUES ('2', '2024年第15届大学生建军节海报', '2024-01-02', '2023-12-01', '2', '国防类', '2024年第15届大学生建军节海报竞赛涌现出众多创意与情感并重的作品。这些海报不仅以其鲜明的色彩和独特的设计吸引了观众的目光，更深刻地传达了对军人精神的敬意和对和平的珍视。学生们的创作充分展现了他们对历史的理解、对英雄的崇敬以及对未来的憧憬。每一幅海报都是对建军节深刻的思考和美好愿景的展现，它们让我们铭记过去，珍惜和平，激励我们共同为一个更加繁荣和谐的社会贡献力量。', '4', '建军');

INSERT INTO `competitions` VALUES ('3', '2021年第30届大学生泡芙宣传竞赛', '2024-09-10', '2024-07-06', '3', '商业类', '2021年第30届大学生泡芙宣传竞赛是一场集创意与美食于一体的盛会。本次竞赛旨在激发学生们的创新思维和艺术设计能力，通过海报的形式展现泡芙的诱人魅力和文化内涵。我们鼓励参赛者以独特的视角和创新的手法，捕捉泡芙的美味瞬间，传达甜品带来的幸福感。竞赛不仅为学生们提供了展示才华的平台，也为推广美食文化、提升设计水平搭建了桥梁。让我们一同期待这场视觉与味觉的双重盛宴。', '4', '泡芙');

INSERT INTO `competitions` VALUES ('4', '2024年第11届国防科技创新海报大赛', '2024-01-15', '2023-11-15', '2', '国防类', '科技创新，国防之魂。第11届国防科技创新海报大赛致力于激发青年学子的创新精神和爱国情怀。我们鼓励参赛者运用现代设计理念，结合国防科技元素，创作出能够展现国家力量和科技成就的海报作品，共同推动国防科技进步与和平发展。', '4', '国防');

INSERT INTO `competitions` VALUES ('5', '2024年第6届军事历史知识海报竞赛', '2024-02-20', '2023-12-20', '2', '国防类', '历史是最好的教科书。第6届军事历史知识海报竞赛旨在通过视觉艺术的形式，让学生们深入了解军事历史，传承红色基因，弘扬民族精神。我们期待参赛者用海报讲述那些激励人心的历史瞬间，增强民族自豪感和历史使命感。', '4', '军事历史');

INSERT INTO `competitions` VALUES ('6', '2024年第1届绿色地球环保海报设计海报大赛', '2024-07-10', '2024-05-10', '1', '公益类', '绿色地球，你我共建。首届绿色地球环保海报设计大赛以提升公众环保意识为宗旨，鼓励学生运用创意设计，表达对自然环境保护的关注和思考。我们希望参赛者的作品能够唤起人们心中对绿色生活的向往，共同参与到环境保护的伟大事业中。', '4', '环保');

INSERT INTO `competitions` VALUES ('7', '2024年第14届关爱留守儿童公益广告海报大赛', '2024-07-11', '2024-05-11', '1', '公益类', '留守不孤单，关爱在行动。第14届关爱留守儿童公益广告海报大赛聚焦于这一特殊群体，通过公益海报的形式，传递社会的温暖与关怀。我们期望参赛者的作品能够触动人心，唤起更多人对留守儿童问题的关注，并积极参与到关爱行动中来。', '4', '关爱');

INSERT INTO `competitions` VALUES ('8', '2024年第3届国际品牌设计海报大赛', '2024-08-12', '2024-05-12', '3', '商业类', '品牌故事，设计发声。第3届国际品牌设计海报大赛以国际化视野为背景，邀请全球设计爱好者展现国际品牌的文化内涵和设计理念。我们鼓励参赛者深入挖掘品牌故事，通过海报设计传递品牌价值，探索设计在品牌传播中的力量。', '4', '品牌设计');

INSERT INTO `competitions` VALUES ('9', '2024第8届年电子商务创新海报大赛', '2024-08-13', '2024-05-13', '3', '商业类', '电商新浪潮，创新引领未来。第8届电子商务创新海报大赛聚焦于电子商务领域的创新发展，鼓励学生探索电商新模式、新趋势。我们期待参赛者用海报展示电商如何便捷生活、推动经济发展，激发更多人对电商创新的思考和参与。', '4', '电子商务');

INSERT INTO `competitions` VALUES ('10', '2024年第11届春季大学生编程挑战海报赛', '2024-06-01', '2024-03-01', '3', '商业类', '春回大地，代码飞扬。第11届春季大学生编程挑战海报赛以编程挑战为主题，邀请编程爱好者通过海报展现编程语言的魅力和创新思维。我们希望参赛者的作品能够激发更多人对编程学习的兴趣，共同探索技术改变世界的无限可能。', '4', '编程');

INSERT INTO `competitions` VALUES ('11', '2024年第4届夏季大学生建筑设计海报大赛', '2024-08-31', '2024-08-01', '1', '公益类', '夏日炎炎，设计清凉。第4届夏季大学生建筑设计海报大赛以夏日为灵感，鼓励学生探索建筑设计与自然环境的和谐共生。我们期待参赛者用海报展示如何通过创新设计为人们带来清凉舒适的生活空间，展现建筑设计的美学与实用性。', '4', '建筑');

INSERT INTO `competitions` VALUES ('12', '2024年第5届大学生夏季军歌节海报大赛', '2024-08-17', '2024-07-20', '2', '国防类', '以军歌传承精神，以海报展现力量。第5届大学生夏季军歌节海报大赛不仅是一场视觉盛宴，更是一次心灵的触动。我们鼓励学生们通过海报设计，捕捉军歌背后的深厚情感和历史意义，传递军人的坚毅与勇敢，激发全社会的爱国热情和集体荣誉感。', '4', '音乐节');

INSERT INTO `competitions` VALUES ('13', '2024年第9届国防科技挑战海报赛', '2024-07-08', '2024-04-08', '2', '国防类', '科技创新，国防未来。第9届国防科技挑战海报赛旨在激发青年学生的创新热情和科技梦想。通过海报，我们希望学生们能够展示国防科技的前沿发展，探索科技与国防结合的无限可能，为国家的和平与安全贡献青春智慧。', '4', '国防科技');

INSERT INTO `competitions` VALUES ('14', '2024年第13届军事模拟战略海报大赛', '2024-07-18', '2024-04-18', '2', '国防类', '战略智慧，沙场秋点兵。第13届军事模拟战略海报大赛邀请学生们运用战略思维，设计出能够体现军事模拟深度和广度的海报。我们期待作品能够展现军事策略的精妙，启发人们对军事艺术的深入思考和学习。', '4', '军事战略');

INSERT INTO `competitions` VALUES ('15', '2024年第21届城市环保创新海报大赛', '2024-07-12', '2024-04-12', '1', '公益类', '绿色城市，创新生活。第21届城市环保创新海报大赛聚焦于城市可持续发展的创新理念。我们鼓励学生们提出并展示他们的环保解决方案，通过海报设计引领环保新潮流，促进生态文明建设，共同为地球的未来贡献力量。', '4', '城市环保');

INSERT INTO `competitions` VALUES ('16', '2024年第22届文化遗产保护海报竞赛', '2024-07-13', '2024-04-13', '1', '公益类', '文化遗产，民族之根。第22届文化遗产保护海报竞赛致力于提高公众对文化遗产重要性的认识。我们邀请学生们通过海报设计，展现文化遗产的独特魅力，激发对传统与现代融合的思考，共同守护我们宝贵的文化财富。', '4', '文化遗产');

INSERT INTO `competitions` VALUES ('17', '2024年第20届国际品牌营销海报大赛', '2024-08-14', '2024-07-24', '3', '商业类', '品牌故事，全球视野。第20届国际品牌营销海报大赛以国际品牌为焦点，鼓励学生们探索品牌背后的深刻故事和全球影响力。我们期待学生们的创意海报能够展现品牌的文化内涵和市场策略，提升品牌的全球竞争力。', '4', '品牌营销');

INSERT INTO `competitions` VALUES ('18', '2024年第1届青年创业挑战海报赛', '2024-08-15', '2024-07-15', '3', '商业类', '创业之路，挑战与机遇并存。首届青年创业挑战海报赛旨在激发青年的创业激情和创新思维。我们鼓励学生们通过海报设计，展示他们的创业理念、项目特色和市场潜力，激励更多青年勇敢追求创业梦想。', '4', '青年创业');

INSERT INTO `competitions` VALUES ('19', '2024年第11届春季大学生艺术海报展', '2024-06-30', '2024-03-30', '3', '商业类', '春天的艺术，创意的绽放。第11届春季大学生艺术海报展是一场展示大学生艺术才华的盛会。我们邀请学生们以春天为主题，通过海报设计表达对自然美、艺术美和生活美的感悟，展现春天的生机与艺术的创造力。', '4', '艺术展');

INSERT INTO `competitions` VALUES ('20', '2024年第14届夏季大学生编程马拉松', '2024-08-31', '2024-07-30', '1', '公益类', '编码盛夏，智慧对决。第14届夏季大学生编程马拉松是一场技术的较量，一场创新的盛宴。我们邀请编程高手们在限定时间内挑战极限，展现逻辑之美。这不仅是一场编程竞赛，更是一次思维与速度的碰撞，一次团队协作与个人能力的双重考验。', '4', '编程马拉松');

INSERT INTO `competitions` VALUES ('21', '2024年第17届夏季大学生编程马拉松海报大赛', '2024-08-31', '2024-07-30', '1', '公益类', '海报映照智慧，编码点燃激情。第17届夏季大学生编程马拉松海报大赛以创意为笔，以技术为墨，邀请学生们设计出能够代表编程精神的海报。我们期待作品捕捉编程马拉松的紧张与兴奋，展现参赛者们的专注与热情，传递科技改变世界的力量。', '4', '编程马拉松');

INSERT INTO `competitions` VALUES ('22', '2024年第18届大学生夏季电影海报大赛', '2024-08-17', '2024-07-27', '2', '国防类', '银幕之下，海报之上。第18届大学生夏季电影海报大赛是一场视觉与故事的交融。我们鼓励学生们以电影为灵感，设计出能够讲述故事、传达情感的海报。这不仅是对电影艺术的致敬，更是一次创意与设计技巧的展示，一次视觉叙事的探索。', '4', '电影节');

INSERT INTO `competitions` VALUES ('23', '2024年第19届大学生夏季摄影海报大赛', '2024-07-18', '2024-04-18', '1', '公益类', '镜头捕捉瞬间，海报讲述故事。第19届大学生夏季摄影海报大赛邀请摄影爱好者们通过镜头探索世界，用海报展示瞬间之美。我们期待作品展现摄影的独特视角，讲述触动人心的故事，传递摄影艺术的魅力与情感深度。', '4', '摄影');

-- 插入队伍数据

INSERT INTO `teams` VALUES ('10', '海洋卫士', '1', '2', '4', '5', '1');

INSERT INTO `teams` VALUES ('11', '历史传承者', '4', '9', '15', '16', '5');

INSERT INTO `teams` VALUES ('12', '科技探索者', '5', '17', '18', '19', '4');

INSERT INTO `teams` VALUES ('13', '环保使者', '2', '20', '21', '22', '6');

INSERT INTO `teams` VALUES ('14', '关爱大使', '9', '23', '24', '25', '7');

INSERT INTO `teams` VALUES ('15', '编程挑战者', '1', '4', '5', null, '10');

INSERT INTO `teams` VALUES ('16', '国防创新者', '15', '9', '2', null, '13');

INSERT INTO `teams` VALUES ('17', '艺术春天', '19', '18', '17', null, '19');

INSERT INTO `teams` VALUES ('18', '建军强国', '1', '2', '4', '5', '2');

-- 插入作品数据

INSERT INTO `works` VALUES ('1', '海洋生态海报', '1', '10', '0', null, null, '95');

INSERT INTO `works` VALUES ('2', '历史传承海报', '5', '11', '0', null, null, '95');

INSERT INTO `works` VALUES ('3', '科技创新海报', '4', '12', '0', null, null, '95');

INSERT INTO `works` VALUES ('4', '环保行动海报', '6', '13', '0', null, null, '95');

INSERT INTO `works` VALUES ('5', '关爱成长海报', '7', '14', '0', null, null, '95');

INSERT INTO `works` VALUES ('6', '编程艺术海报', '10', '15', '0', null, null, '95');

INSERT INTO `works` VALUES ('7', '国防教育海报', '13', '16', '0', null, null, '95');

INSERT INTO `works` VALUES ('8', '艺术春天海报', '19', '17', '0', null, null, '95');

INSERT INTO `works` VALUES ('9', '建军节庆祝海报', '2', '18', '0', null, null, '95');

## **1.4** 项目架构搭建

### **1.4.1 创建项目**

在IDEA菜单栏中，选择File →New →Project命令，在呈现的对话框选择Java Enterprise项目类型，输入项目名称为PDCS、位置D:\IDEA\program\PDCS，选择模块结构为Web application，选择应用服务器Tomcat Server，如图1-7所示。

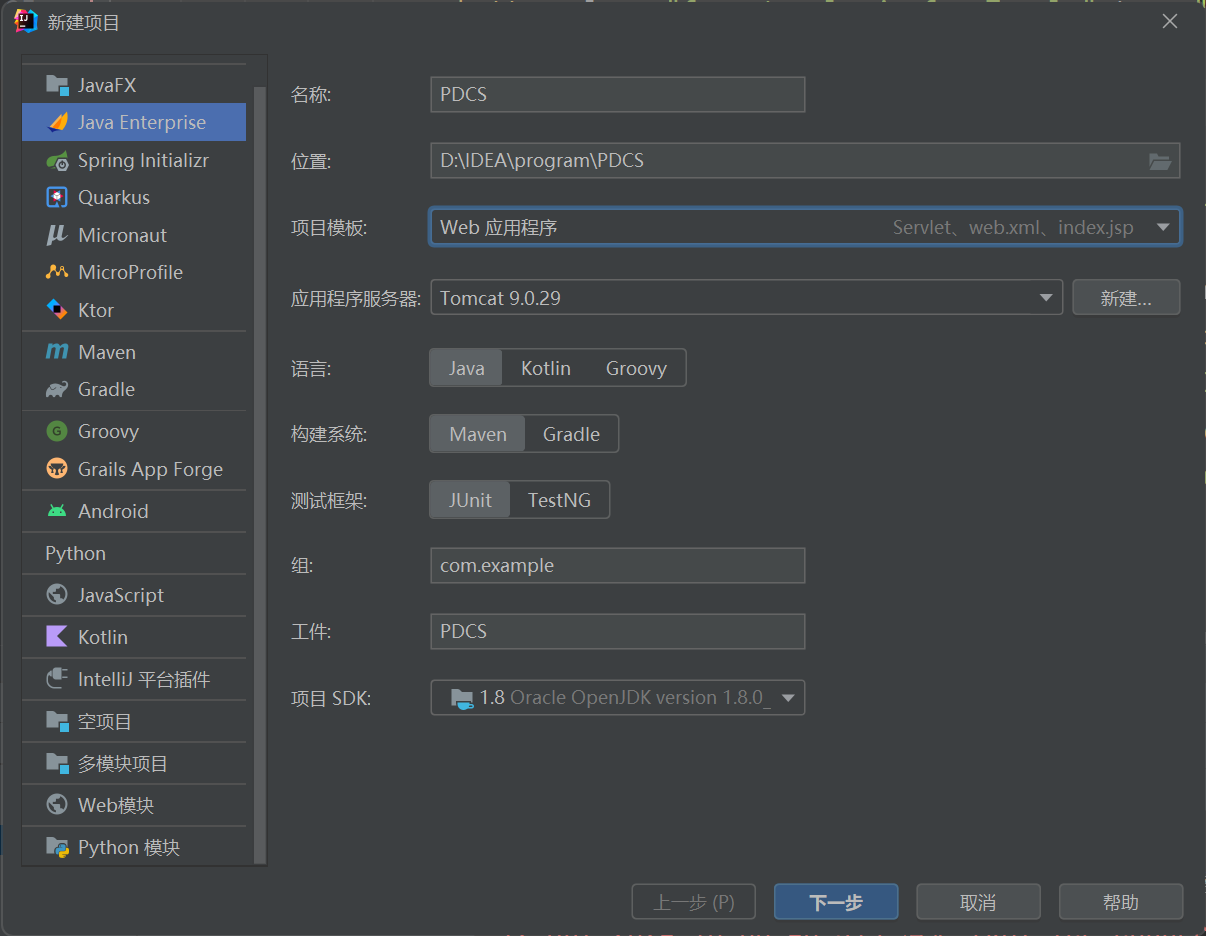


图1-7 新建PDCS项目界面1

然后单击Next按钮，进入项目配置界面，如图1-8所示。

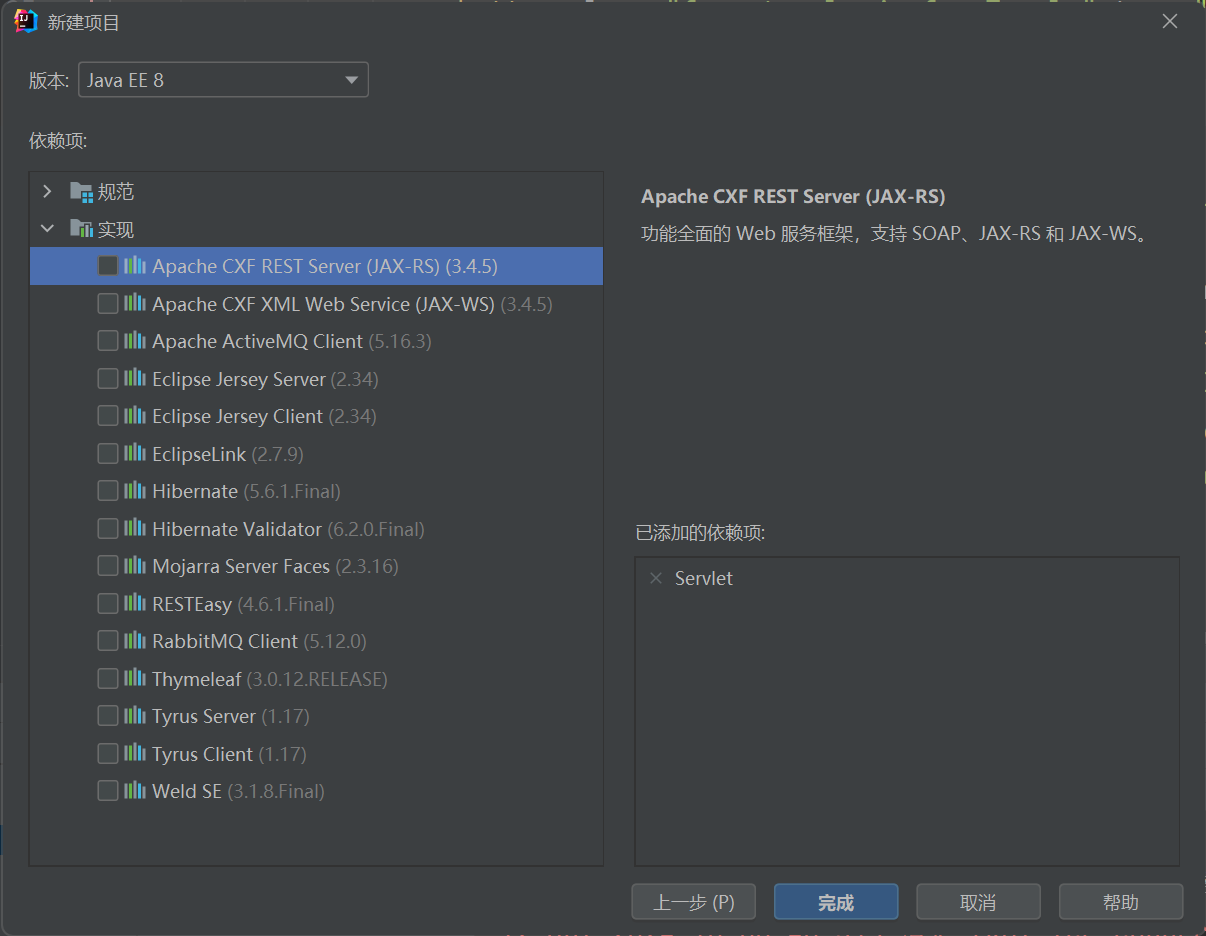


图1-8 新建PDCS项目界面2

最后单击Finish按钮，完成项目创建。项目PDCS初始的文件结构，如图1-9所示。

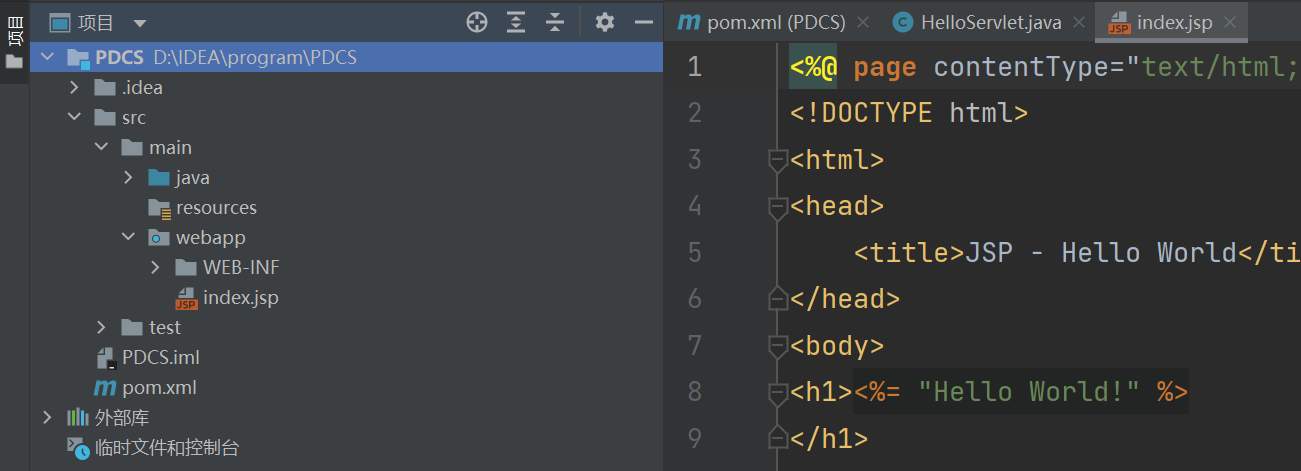


图1-9 PDCS项目初始文件结构

### **1.4.2 项目架构**

依据MVC开发模式，完善项目架构。在PDCS项目的文件夹src\main\java上右击，选择New→Package，在弹出的窗口中输入com.example.servlet，回车创建Package。同理创建com.example.domain、com.example.dao、com.example.utils，如图1-10所示。

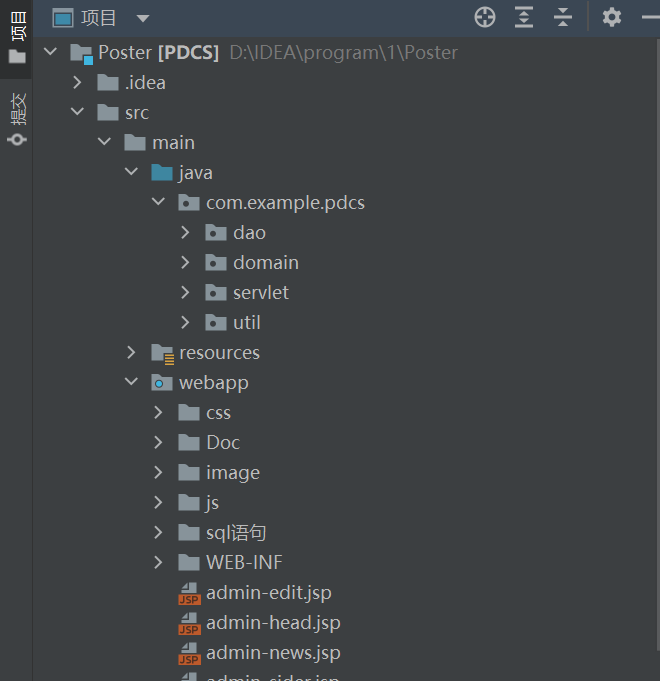


图1-10 PDCS项目架构

PDCS项目中MVC模式说明，com.example.servlet包中编写Servlet代码，代表控制层（C）；webapp站点中编写JSP代码，代表视图层（V）；com.example.domain包中编写业务数据类、业务逻辑类代码，com.example.dao包中编写数据库存取类代码，它们代表模型层（M）。详细说明如表1-9所示。

表1-9 项目架构说明表

|  |  |  |
| --- | --- | --- |
| **包Package/文件夹** | **包含内容** | **备注** |
| com.example.domain | 大学生海报设计竞赛系统业务数据实体类、业务逻辑处理类 | MVC开发模式中模型层（M） |
| com.example.dao | 数据库存储代码类 | MVC开发模式中模型层（M） |
| webapp | 大学生海报设计竞赛系统网站JSP页面 | MVC开发模式中视图层（V） |
| com.example.servlet | 大学生海报设计竞赛系统网站Servlet类 | MVC开发模式中控制层（C） |
| com.example.utils | 通用工具类 |  |

基于MVC开发模式的项目架构，没有固定的结构和命名规范，开发人员可以根据项目的复杂程度自定义结构，符合MVC思想即可。

### **1.4.3 配置jar包**

在PDCS项目开发过程中，使用了多个第三方jar包，具体清单如表1-10所示。

表1-10 第三方jar包清单

|  |  |  |
| --- | --- | --- |
| **jar名称** | **说明** | **版本** |
| mysql-connector-java | MySQL数据库驱动 | 8.0.28 |
| druid | JDBC数据源 | 1.2.8 |
| commons-logging | 日志操作 | 1.2 |
| spring-jdbc | JDBCTemplate数据库工具 | 5.3.17 |
| fastjson2 | JSON解析和生成器 | 2.0.18 |
| taglibs-standard-spec | JSTL接口 | 1.2.1 |
| taglibs-standard-impl | JSTL实现 | 1.2.1 |
| commons-io | 文件上传基础 | 2.2 |
| commons-fileupload | 文件上传 | 1.4 |
| javax.servlet-api | Servlet API | 4.0.1 |
| junit-jupiter-api | JUnit Jupiter API | ${junit.version} |
| junit-jupiter-engine | JUnit Jupiter Engine | ${junit.version} |

第三方jar包在PDCS项目中由maven工具管理，开发人员配置项目中的pom.xml文件，maven工具自动从网络中下载相应jar包到项目中。

源程序：pom.xml文件中依赖配置代码

<dependencies>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter-api</artifactId>

<version>${junit.version}</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter-engine</artifactId>

<version>${junit.version}</version>

<scope>test</scope>

</dependency>

</dependencies>

编写pom.xml文件中的第三方jar包依赖配置代码，一般情况开发人员不是直接编写，而是从网络中复制配置代码。站点MVNrepository就是提供这类服务的网站，其网址为：https://mvnrepository.com/。在这个网站上，开发人员搜索jar包的关键字即可获取相应的配置代码。

通常第三方jar包配置完成后，在IDEA的项目中可以查看相应的jar包文件，如图1-11所示。

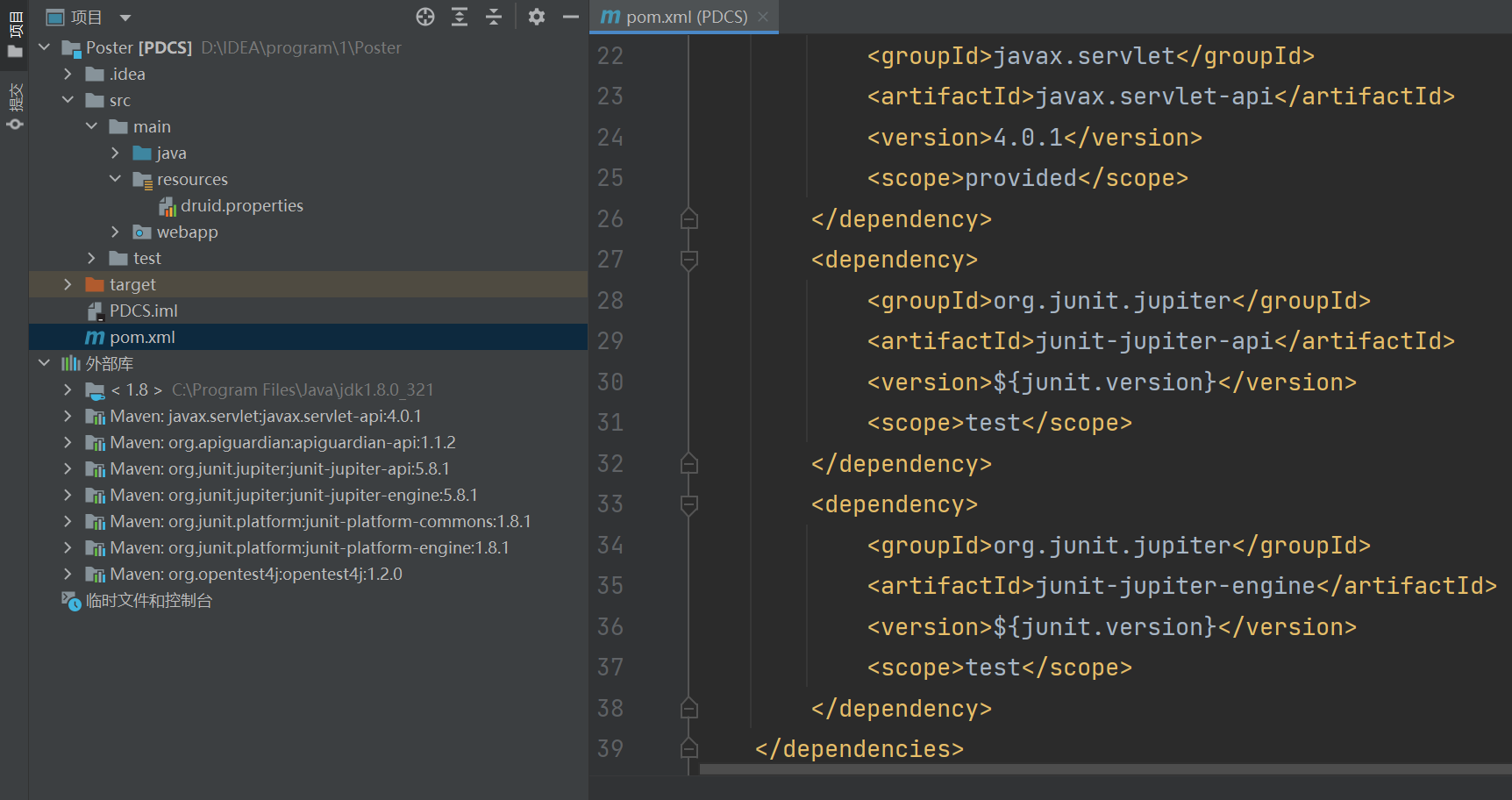


图1-11 PDCS项目依赖的第三方jar包

### **1.4.4 网站结构**

PDCS项目中默认已有webapp站点，需要进一步初始化站点的文件夹，文件夹清单如表1-11所示。

表1-11 webapp站点文件夹清单

|  |  |
| --- | --- |
| **文件夹名称** | **说明** |
| css | 存放样式文件，\*.css |
| image | 存放图片文件，\*.png \*.ico |
| js | 存放JavaScript脚本文件, \*.js |
| doc | 存放文档文件，\*.docx |
| sql | 存放数据库文件，\*.sql |
| WEB-INF | Web站点的安全文件夹, 存放web.xml等文件 |

在PDCS项目的文件夹webapp上右击，选择New→Directory，在弹出的窗口中输入css，回车创建文件夹，同理创建文件夹img、js、doc。创建完成后webapp文件夹结构如图1-12所示。

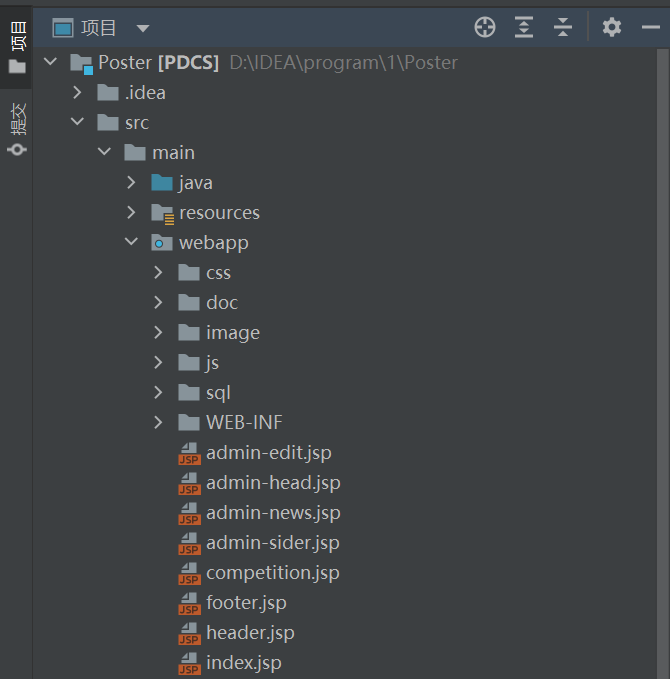


图1-12 webapp文件夹结构

PDCS项目中，各种身份信息存放在MySQL数据库中，图片等在数据库中只保存了文件名。

### **1.4.5 数据库连接配置**

PDCS项目中使用druid数据源连接池以及JDBCTemplate工具，需要在项目中配置数据库连接信息文件。

在PDCS项目的文件夹src\main\resources上右击，选择New→File，在弹出的窗口中输入druid.properties，回车创建文件。同理在文件夹src\test\resources创建文件druid.properties。在这两个新文件中编写配置信息，配置代码如图1-13所示。

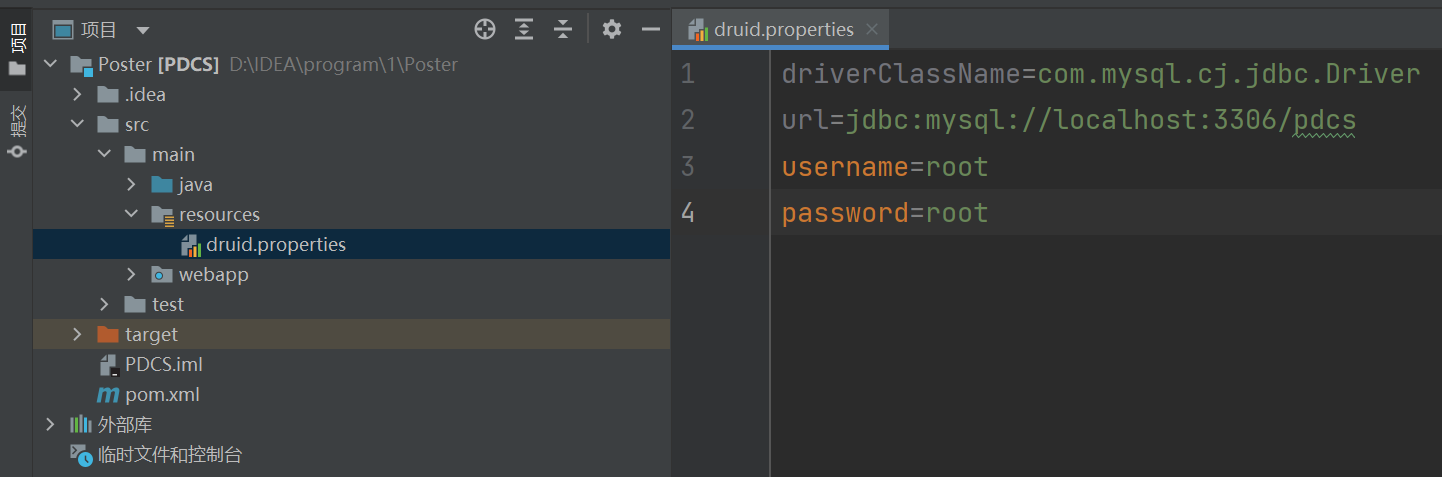


图1-13 数据库连接配置文件

PDCS项目中使用druid数据源连接池，需要设计一个工具类供项目中的数据存取类使用，该工具类命名为JDBCUtils。

在PDCS项目的com.example.utils包上右击，选择New→Java Class，在弹出的窗口中输入JDBCUtils，回车创建文件并输入源代码。

源程序：JDBCUtils.java文件

package com.example.utils;

import javax.sql.DataSource;

import java.io.IOException;

import java.io.InputStream;

import java.sql.Connection;

import java.sql.SQLException;

import java.util.Properties;

import com.alibaba.druid.pool.DruidDataSourceFactory;

public class JDBCUtils {

private static DataSource ds ;

static {

try {

//1.加载配置文件

Properties pro = new Properties();

//使用ClassLoader加载配置文件，获取字节输入流

InputStream is = JDBCUtils.class.getClassLoader()

.getResourceAsStream("druid.properties");

pro.load(is);

//2.初始化连接池对象

ds = DruidDataSourceFactory.createDataSource(pro);

} catch (IOException e) {

throw new RuntimeException("找不到druid.properties文件，"

+"请在resources文件夹中创建druid.properties文件", e);

} catch (Exception e) {

throw new RuntimeException("数据库初始化异常", e);

}

}

public static DataSource getDataSource(){

return ds;

}

public static Connection getConnection() throws SQLException {

return ds.getConnection();

}

}

通过上述步骤的实施，PDCS项目的架构已经搭建完毕。