

比赛对阵管理系统项目

系统实现和测试说明书

目录

1. 文档介绍..... 3

1.1. 编写目的..... 3

1.2. 文档范围..... 3

1.3. 读者对象..... 3

1.4. 术语与缩写解释..... 3

1.5. 参考资料..... 4

2. 总体设计..... 4

2.1. 系统说明..... 4

2.2. 项目背景..... 4

2.3. 软件结构..... 5

2.4. 数据库设计..... 7

3. 详细设计..... 10

3.1. 人机界面设计..... 10

3.1.1. 注册登录模块..... 10

3.1.2. 首页模块..... 11

3.1.3. 个人信息模块..... 12

3.1.4. 比赛管理模块..... 13

3.1.5. 人员管理模块..... 14

3.1.6. 人员管理模块..... 16

3.2. 关键技术..... 18

3.2.1. 注册登录模块..... 18

3.2.2. 首页模块..... 19

3.2.3. 个人信息模块..... 19

3.2.4. 比赛管理模块..... 20

3.2.5. 人员管理模块..... 21

3.2.6. 成绩管理模块..... 22

4. 系统测试..... 22

1. 文档介绍

1.1. 编写目的

本文档描述软件产品系统设计说明书的目的是：

- 1) 定义软件总体结构，作为软件开发人员进行初步设计的基础；
- 2) 提供性能要求、概要设计、详细设计、测试报告和用户影响的信息，作为软件人员进行软件结构设计和编码的基础；
- 3) 作为软件成功交付、编写用户手册的依据。

1.2. 文档范围

比赛对阵管理系统项目设计说明书主要是对该系统中功能及功能操作方式和测试情况进行基本的描述。

1.3. 读者对象

编写详细设计人员、程序开发人员以及系统使用人员。

1.4. 术语与缩写解释

术语或缩写	解释
MVT 架构	模型—视图—控制器，是一种软件设计模式
数据中心 Data Center	以各类数据为核心，依托成熟的存储、数据库、GIS、网络等技术，按照统一标准，建立的具有信息管理、分析、查询、统计及服务的一体化数据管理体系。
数据管理 Data Management	利用数据库、数据仓库、元数据和网络等技术，建立分布式、集中式或集中加分布式数据管理系统，开展数据接收、组织存储、运行维护、更新、共享交换等工作，实现对数据资源的有效组织和应用。
数据维护 Data Maintenance	在制定维护方案基础上，对数据和数据库进行的日常维护与监控、备份与恢复、应急处理和监督管理等，从而保护数据的安全性和可移植性。
用户	本系统的使用者

1.5. 参考资料

序号	文档名称	起止页码	主要负责人	发布日期
1	《计算机数据库技术在信息管理中的应用》	9+15	曾英	2020(06)
2	《MVC 架构下网站的设计与实现》	188-191	刘桃丽	2020,30(02)
3	《基于 Django 的 Python Web 开发》	37-40	白昌盛	2019,31(24)

2. 总体设计

2.1. 系统说明

这是一款比赛对阵管理系统，任务由北京科技大学软件工程课程组的殷绪成老师发布，开发者为计算机科学与技术计 172 班学生刘梓瑄，学号为 41724039，面向用户群为北京科技大学全体在校师生。

- 项目名称：比赛对阵管理系统
- 任务提出者：殷绪成老师
- 开发者：计 172 刘梓瑄 41724039
- 用户群：北京科技大学全体在校师生

2.2. 项目背景

近年来，北京科技大学给学生提供着丰富广泛、种类多样的学术竞赛和体育竞赛活动，旨在通过各类赛事，丰富广大在校学生的课余生活，培养学生能力和兴趣，以及鼓励学生们积极参加比赛活动。但是，北京科技大学现各赛事管理流程较为混乱，赛事信息的发布多来源于公众号、本科教学网等途径，不少学生反映赛事信息不集中，查找困难，容易错过等情况。且比赛的报名机制也较为分散，

大量赛事报名途径各异，易对学生造成误导，且不方便赛事承办单位进行统一管理。赛事公告等信息发布不及时，无法保障参赛学生及时获取有关赛事的最新信息。

本项目旨在解决上述提到的潜在问题，使用赛事管理系统对校内各赛事进行统一管理，优化学生的参赛，比赛全流程，提高赛事管理员的管理效率，在降低管理员负担的同时方便广大在校学生参与赛事活动。

2.3. 软件结构

总体采用 Django 的 Web 框架搭配轻量级数据库 sqlite，前端采用 CSS、HTML 和 JavaScript 完成页面设计，部分嵌套 JQuery 函数库完成界面设计。

Django 的 MVT 设计模式包括四个模块：

(1) Models.py 用于创建数据库模型，是对数据库的上层封装，大大简化了编码过程中对数据库的增、删、改、查操作，与 MVC 架构中的 Model 功能类似。

(2) Views.py 是主要的功能模块，负责业务逻辑处理，与 Templates 进行数据交换，与 MVC 中的 Control 控制器功能类似。

(3) Templates 文件夹中的保存的模版文件，用于生成最终前端向用户展示的 HTML 页面。

(4) url.py 则用于定义整个系统或某个子模块的路由表，指定了 URL 与 views.py 的映射关系。urls.py 根据用户发起的请求，调用 views.py 中对应的函数，与数据模型以及模版进行交互，响应用户请求。

根据需求分析设计说明书，得到了对应的层次结构图如图 2.1 所示：

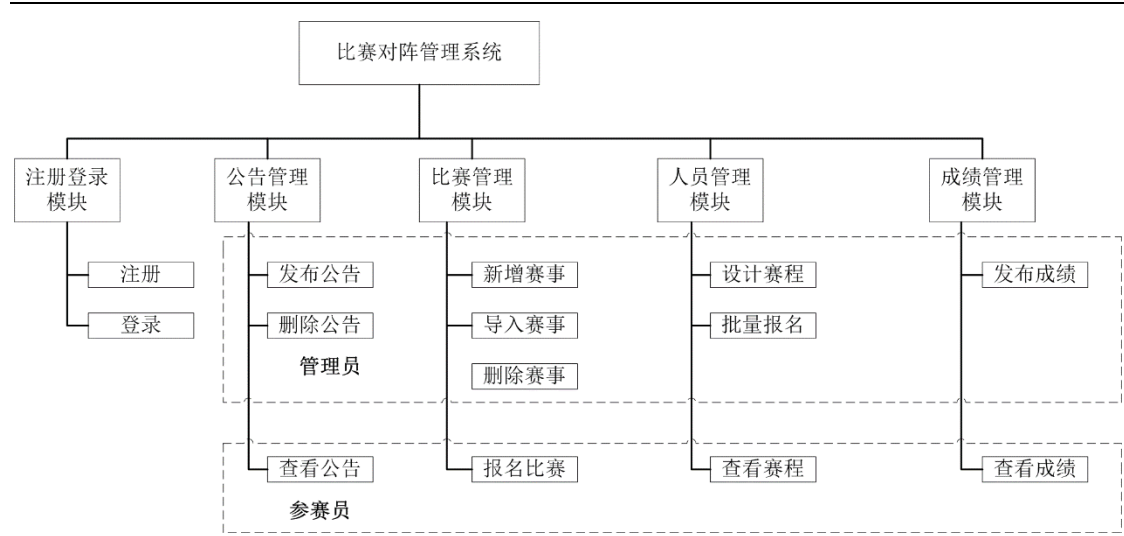


图 2.1 层次结构图

根据功能性需求，设计出人机交互页面如图 2.2 所示：



图 2.2 人机交互界面示意图

根据该系统的工作模式和其提供给用户的主要功能，比赛对阵管理系统的用例图如图 2.3 所示：

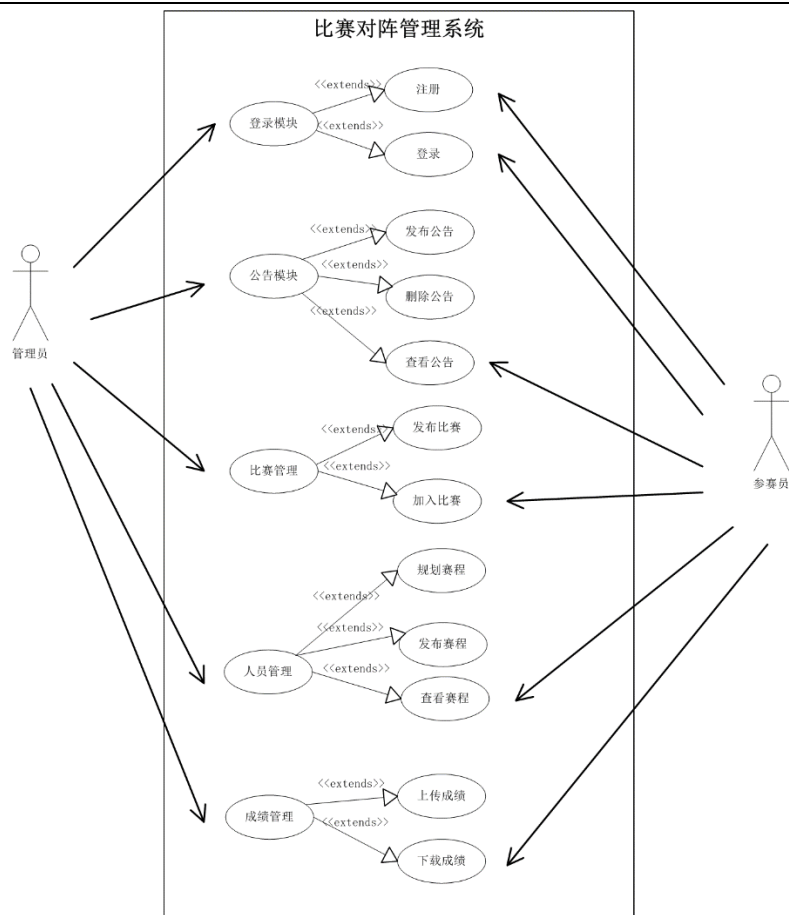


图 2.3 系统用例图

2.4. 数据库设计

本项目设计中采用 Django 自带的轻量级数据库 sqlite，开发工具选用 Pycharm，需在使用前在 Pycharm 中安装 sqlite 的数据库插件。

- 1) 数据库系统：sqlite3
- 2) 设计工具：Pycharm 2020.1
- 3) 表结构：

表 1 学生信息表

字段名	描述	数据类型	是否主键	备注
stu_id	学生 ID	char(50)	PK	
stu_name	学生姓名	char(50)		
stu_college	学生所在学院	char(50)		
stu_mail	学生邮箱	char(50)		
identity	学生身份	char(50)		

stu_password	学生密码	char(50)		
stu_gender	学生性别	char(10)		

表 2 管理员信息表

字段名	描述	数据类型	是否主键	备注
adm_id	管理员 ID	char(50)	PK	
adm_name	管理员姓名	char(50)		
adm_college	管理员所在学院	char(50)		
adm_mail	管理员邮箱	char(50)		
identity	管理员身份	char(50)		
adm_password	管理员密码	char(50)		
adm_gender	管理员性别	char(10)		

表 3 赛事模型表

字段名	描述	数据类型	是否主键	备注
com_id	赛事 ID	char(50)	PK	
com_name	赛事名称	char(50)		
com_starttime	赛事初赛时间	char(50)		
com_endtime	赛事决赛时间	char(50)		
com_type	赛事类别	char(50)		
com_college	赛事承办学院	char(50)		
com_stu	关联学生类	char(50)	FK	关联外键
com_total	报名人数	char(50)		
com_mark	赛事成绩单	char(50)		
com_infor	赛程信息	char(50)		

表 4 公告模型表

字段名	描述	数据类型	是否主键	备注
noti_namei	公告名称	char(50)	PK	

4) 表关系：

赛事和学生之间存在多对多关联关系（一项比赛可被多名学生报名，一名学生可报名参加多项赛事），它们通过外键“com_stu”完成关联，sqlite 会生成一张关联数据表，记录学生报名参加赛事的情况。

表 5 关联关系表

字段名	描述	数据类型	是否主键	备注
id	ID	char(50)	PK	
competitions_id	赛事 id			
students_id	学生 id			

5) 物理设计

根据以上逻辑关系，得到以下表：

表 1:

	id	stu_id	stu_name	stu_college	stu_mail	identity	stu_password	stu_gender
1	3	41724066	王小薇	8	41724066@ustb.edu.cn	1	wangxiaowei	1
2	4	41724001	李兰	5	lilan@ustb.edu.cn	1	lilan123456	1
3	5	41724003	江平	9	41724003@xs.ustb.edu.cn	1	jiangping123456	0
4	6	41724002	吴凯	5	41724040@ustb.edu.cn	1	wukai123456	0
5	7	41724037	王二	2	41724037@xs.ustb.edu.cn	1	wanger123456	0
6	8	41724052	梁育	5	41724052@xs.ustb.edu.cn	1	liangyu123456	1
7	33	41724080	史云	2	41724080@xs.ustb.edu.cn	1	shiyun123456	1
8	37	41727001	钱博	2	41727001@xs.ustb.edu.cn	None	123456	0
9	38	41727002	杨开	3	41727002@xs.ustb.edu.cn	None	123456	1
10	39	41727003	欧阳辉	7	41727003@xs.ustb.edu.cn	None	123456	0
11	40	41724088	宋玉	3	41724088@xs.ustb.edu.cn	1	songyu123456	0

表 2:

	id	adm_id	adm_name	adm_college	adm_mail	identity	adm_password	adm_gender
1	15	0001	a赵四	5	zhaosi@ustb.edu.cn	0	zhaosi123456	1
2	16	0008	a王尔	2	wanger@ustb.edu.cn	0	wanger123456	0
3	17	0010	a王涛	2	wangtao@ustb.edu.cn	0	wangtao123456	0
4	18	0012	a刘红	2	liuhong@ustb.edu.cn	0	liuhong123456	1

表 3:

	id	com_id	com_starttime	com_endtime	com_type	com_college	com_total	com_name	com_mark	com_i
1	8	None	2020-04-25	2020-05-25	0	4	17	北京大学算法竞赛成绩单.xls		赛程信息
2	9	None	2019-03-26	2020-04-30	1	9	5	北京大学英语竞赛成绩单.xlsx	<null>	
3	10	None	2020-01-22	2020-03-30	0	5	3	北京大学物理竞赛成绩单.xlsx	<null>	
4	11	None	2020-01-26	2020-09-01	1	5	9	北京大学数学竞赛成绩单.xlsx	<null>	
5	12	None	2019-05-26	2020-08-30	1	2	2	北京大学羽毛球竞赛成绩单.xlsx	<null>	
6	13	None	2017-05-26	2018-05-26	0	7	5	北京大学游泳比赛成绩单.xlsx	<null>	
7	14	None	2019-05-26	2020-05-26	1	8	4	北京大学人文知识竞赛成绩单.xlsx	<null>	
8	37	None	2019-06-08	2019-06-08	1	1	1	北京大学英语竞赛成绩单.xlsx	<null>	
9	40	None	2019-06-08	2020-07-08	1	2	1	北京大学马拉松比赛成绩单.xls		赛程信息

表 4:

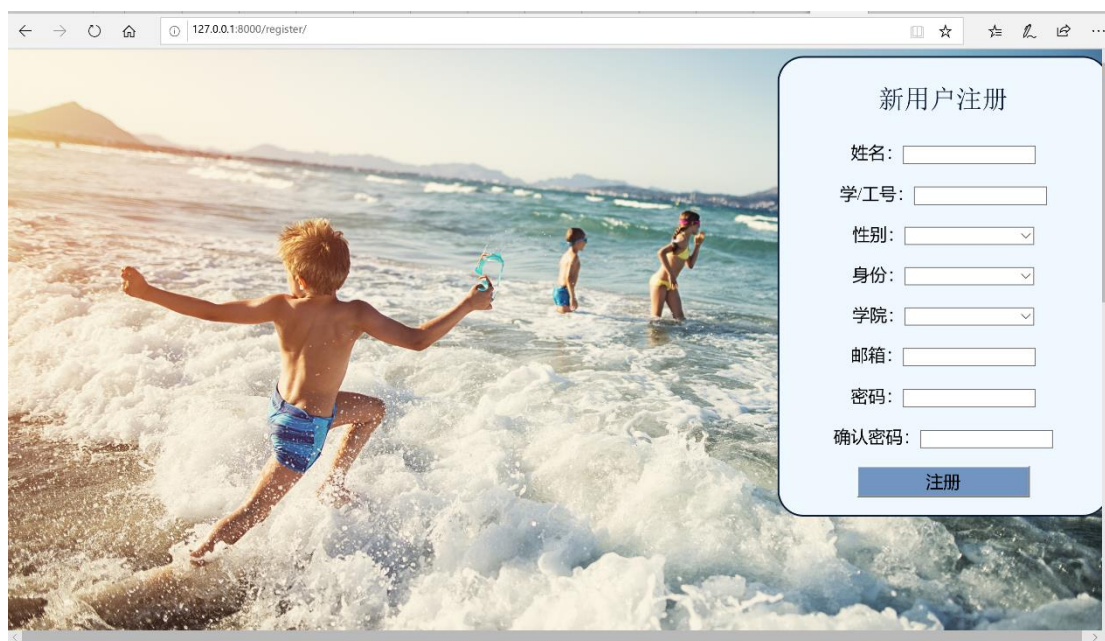
	id	competitions_id	students_id
1	1	8	4
2	2	9	4
3	3	10	4
4	4	14	4
5	5	8	5
6	6	9	5
7	7	13	4
8	8	13	5
9	9	13	6
10	10	13	3

3. 详细设计

3.1. 人机界面设计

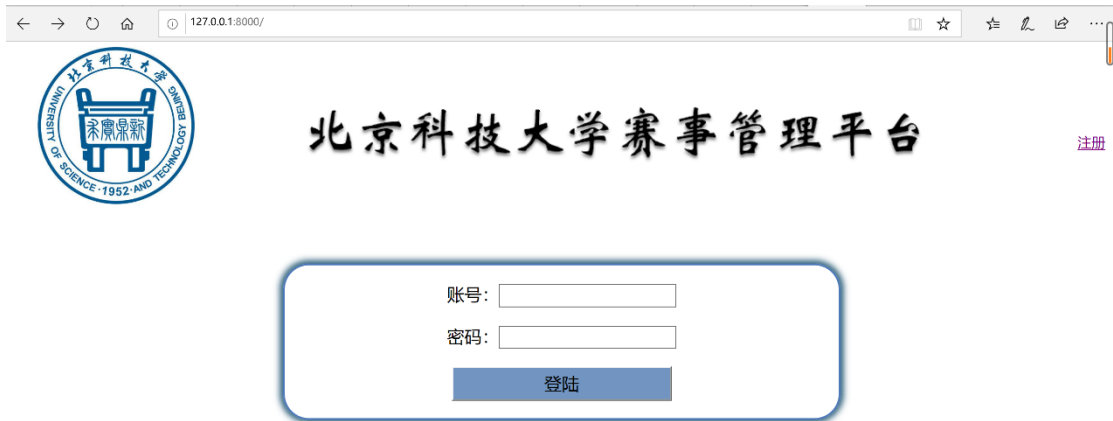
3.1.1. 注册登录模块

1) 注册部分



支持功能：系统使用人员根据自己唯一的学/工号对个人信息进行注册，获得本系统的使用权。

2) 登陆部分



支持功能：系统使用人员用户名密码匹配登录验证，未注册人员注册页面跳转。

3.1.2. 首页模块

1) 管理员部分



支持功能：管理员发布赛事公告、下载查看赛事公告、删除已发布的赛事公告。

2) 参赛员部分



支持功能：参赛员验证登录情况、下载已发布赛事公告。

3.1.3. 个人信息模块

1) 管理员部分



支持功能：管理员验证个人身份，查看存储在库中的个人信息。

2) 参赛员部分



支持功能：参赛人员验证个人身份，查看存储在库中的个人信息。

3.1.4. 比赛管理模块

1) 管理员部分





支持功能：管理员查看已经发布的比赛，通过 EXCEL 批量发布赛事信息，删除已经发布的比赛，逆序展示赛事信息，记录每项赛事的已报名人数，新增赛事。

2) 参赛员部分



支持功能：参赛人员查看已发布的赛事，报名已发布的比赛。

3.1.5. 人员管理模块

1) 管理员部分

北京科技大学软件工程课程所有
未经许可，不可全部或部分发表、复制、使用于任何目的





支持功能：管理员以 EXCEL 的形式发布比赛赛程信息，通过 EXCEL 表格对某项赛事进行批量人员报名，自定义设计比赛赛程。

2) 参赛员部分



支持功能：参赛人员下载并查看已发布的赛程信息。

3.1.6. 人员管理模块

1) 管理员部分



支持功能：管理员上传赛事成绩及排名信息。

2) 参赛员部分



支持功能：参赛人员下载并查看已经发布的赛事成绩单。

3.2. 关键技术

3.2.1. 注册登录模块

- 1) 注册的基本思想是，通过用户在注册页面前端输入的个人信息，提交 form 表单，根据注册用户名前是否有字段 ‘a’ 进行标注，后台在库中新建对应的（参赛员/管理员）数据库对象，且注册的识别标志是用户的学号/工号，作为他们在数据库中的唯一标识字段。
- 2) 登录模块的实现：

进行查找用户并比对密码是否正确调用的是 Django 中的登录函数 `authenticate`，如果查找不到或密码不正确则会返回 `None`。此外该系统用的加密算法是 Django 默认的 `pbkdf2_sha256` 加密算法。

参赛员和管理员的账户类型的区分依靠账号中的第一个字符来进行区分，管理员的账号会在前面加 ‘a’，而参赛员则没有这个字符。

登录模块的主要算法流程图如图 3.2.1 所示。

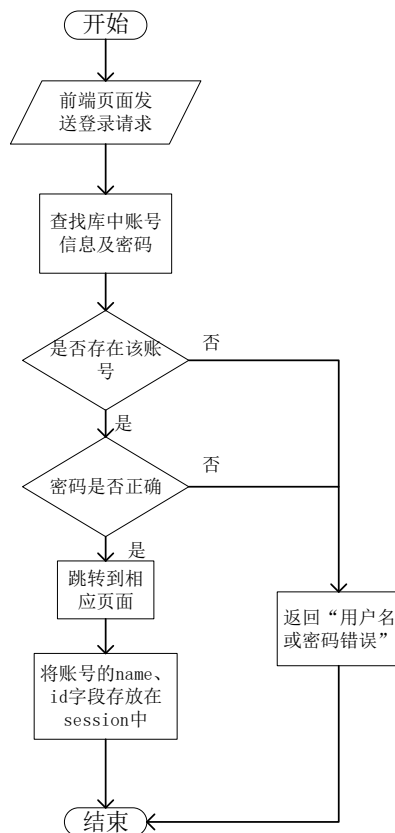


图 3.2.1 登录模块流程图

3.2.2. 首页模块

首页主要由两部分组成：公告栏部分和首部。现分别对这两部分进行分析。

(1) 首部

首部主要的功能有导航栏和退出登录两个功能，通过点击导航栏中各个模块，会跳转到相应的模块中，点击退出登录则会将登录是在 session 中存储的账号和密码删除并跳转回到登录页面。导航栏的跳转主要靠点击超链接，然后通过 url 响应相应的模块。退出登录是调用了 Django 中的登出函数 logout 函数并设置了如果用户不通过点击退出登录这个按钮，则会在两周后（Django 的默认失效时间）自动清除保存的用户数据即自动登出。

(2) 公告栏部分

本系统公告部分主要功能是提供赛事公告信息的管理途径，后台中对赛事公告的管理是单独维护一张赛事公告表，管理人员比参赛人员拥有更多的权限，管理人员可以通过前端的‘file’控件上传公告文件，上传的文件进入本地文件夹下进行保存，文件名入库。在前端进行展示时通过库中存储的文件名进入本地目录下进行查找，管理人员还可以删除已经发布的赛事公告，调用 Django 的 delete() 函数，对数据库中该项进行删除。

参赛员端只具有下载查看已发布赛事公告的权限，通过展示的文件名下载本地文件夹下的已上传文件。

3.2.3. 个人信息模块

个人信息模块未做特殊设计，仅为用户信息展示。处于安全性的考虑，参赛人员的个人信息对于学生而言是无法修改的，若录入信息有误应联系管理员在管理器 Admin 处修改。与个人信息模块相关的模型有 Students、Admins，两种模型分别对应两种类型的用户（参赛人员、管理人员）。提供登录功能的模型为 Django 内置的 User 对象，其登录的用户名（或唯一标识码）分别对应 Students 和 Admins 模型的 id 属性(stu_id、adm_id)，而 Students 和 Admins 模型为 User

对象的信息扩展，这样可以对两类用户进行区分。

3.2.4. 比赛管理模块

比赛管理模块是本对阵系统的主要功能模块，主要用于管理员发布比赛，参赛员报名比赛。

1) 管理员

首先对所有已发布的赛事进行一个按初赛日期的逆序展示，这一功能是由 Django 提供的 `order_by()` 函数实现；

第二个功能点是发布单项赛事，其实现逻辑和注册登录模块中的新用户注册功能相似，通过前端用户填写的信息，form 表单提交至后台，比赛名称作为 Primary Key 作为比赛项目在库中的唯一标识，不允许相同名称的比赛进行二次发布。发布成功后调用 `redirect()` 函数刷新赛事管理页面，同样按初赛时期逆序展示，以提醒用户功能发布成功。

最后，本项目考虑到实际的应用情况，由管理员手动发布单项赛事效率较低且加重了管理员的负担，该系统支持由 excel 批量发布比赛的功能。这一功能的实现借助 python 的 `xlwt` 库函数，以识别和读取 excel 表格中的字段，按行读取字段成功后，新建 Competitions 对象，按字段信息更新库中的数据，以此实现通过 excel 表批量导入比赛的功能。

2) 参赛员

首先对所有已发布的赛事进行一个按初赛日期的逆序展示，这一功能是由 Django 提供的 `order_by()` 函数实现；且标注出当前用户已报名的赛事，这一功能的实现借助数据库设计中提到的多对多关联表，该表用于记录学生报名比赛的信息，首先查找该项记录(`stu_id`, `competitions_id`)在该关联表中是否存在，若存在，标记该用户已经报名这项比赛，否则向用户展示报名按钮。

报名按钮点击后提供给用户二维码报名功能，跳转至一个新的二维码页面，用户需扫描二维码完成报名，由于本项目没有部署在服务器上运行，无法时时动态生成报名二维码，系统中二维码的扫描结果会提示用

户“已成功完成报名”，此时刷新原页面，会展示已经成功报名，该项记录会更新数据库中关联表的信息。

3.2.5. 人员管理模块

1) 管理员

首先对所有已发布的赛事进行一个展示，对应后侧 ‘file’ 浏览框用于上传赛程信息。

点击“管理”，session 上传当前比赛 id 用于完成对当前赛事的管理，展示已报名的学生信息，通过多对多关联表中按 competitions_id 查找所有项后在前端进行展示。考虑现实中本系统的应用情况，涉及发布一项赛事后对大量人员进行统一报名的功能，该系统在此模块下支持通过 excel 批量导入参赛人员信息。这一功能的实现借助 python 的 xlwt 库函数，以识别和读取 excel 表格中的字段，按行读取字段成功后，新建 Students 对象，新建对象后，本对象同时获得该系统的使用权（下次登陆时无须进行新用户注册），按字段信息更新库中 Students 表和多对多关联表 competitions_com_stu，新建学生报名参加比赛项，的数据，以此实现通过 excel 表批量导入参赛人员的功能。

为了更大程度上使本系统贴合实际需求，本模块下还提供了自定义比赛赛程的信息，利用数据库中的用户信息和 python 的 xlwt 库，读取数据库后台信息，前端由用户输入，form 表单提交给后台用户的设计原则，生成 excel 表格作为对应的自定义赛程信息。

2) 参赛员

后台中对赛程信息的管理是通过赛事 Competitions 模型的 com_infor 字段，管理人员可以通过前端的‘file’ 控件上传公告文件，此时 session 读取的信息是当前比赛的 ‘com_id’ 字段，上传的文件进入本地文件夹下进行保存，文件名入库。在前端进行展示时通过库中存储的文件名进入本地目录下进行查找。

参赛员端只具有下载查看已发布赛程信息的权限，通过展示的文件

名下载本地文件夹下的已上传文件。

3.2.6. 成绩管理模块

本系统的成绩单信息均通过 Excel 表格的形式进行统一管理。

后台中对成绩单的管理是通过赛事 Competitions 模型的 com_mark 字段，管理人员可以通过前端的‘file’控件上传公告文件，上传的文件进入本地文件夹下进行保存，文件名入库。在前端进行展示时通过库中存储的文件名进入本地目录下进行查找。

参赛员端只具有下载查看已发布成绩信息的权限，通过展示的文件名下载本地文件夹下的已上传文件。

4. 系统测试

在完成项目设计后，项目进入测试阶段，重点针对该项目的功能、性能、可用性、安全性、可部署性、可移植性、运行速度等方面进行了大量测试，测试结果如下表所示：

比赛对阵管理系统		
测试名称	测试类别	测试过程与结果
功能测试	链接测试	项目内所有超链接条目均可成功跳转，无孤立页面
	表单测试	用户提交操作完整，提交给服务器信息正确
	Cookies 测试	Cookies 工作正常，可按预定时间进行保存
	设计语言测试	HTML+CSS+JavaScript 客户端和服务端未出现问题
	数据库测试	数据一致性正确，输出正确；增删改查功能实现正确
性能测试	连接速度测试	页面响应速度正常，未引起数据丢失
	负载测试	未能有大量负载测试条件，小规模内运行正常
	压力测试	未能有破坏性压力测试条件，目前未出现崩溃情况
可用性测试	导航测试	导航栏直观简洁，可满足用户基本需求
	图形测试	图片尺寸和质量切合系统，达到了美化目的
	内容测试	项目可提供准确、可靠的信息
	整体界面测试	用户体验舒适，界面操作良好、设计风格统一

客户端 兼容性测试	平台测试	目前仅支持 Windows 操作系统下的用户配置
	浏览器测试	兼容于多款主流浏览器：IE、Firefox、Chrome 等
安全性测试	用户登录验证	可正确识别匹配用户名密码，不支持未登录用户操作
	超时测试	暂未提供用户超时登录功能
	日志文件测试	日志文件正确记录、全程可追踪
	安全套接字测试	加密过程正确，可保证信息完整性
	服务器端脚本测试	暂未完成
部署方便性 测试	环境测试	用户需部署：Python+sqlite+Django 的运行环境
	运行简易性测试	管理员运行脚本文件访问，普通用户用 URL 地址访问

图 4.1 比赛对阵管理系统测试结果一览表

已进行的测试中未能 100%完成项目的测试需求，但至少覆盖了 85%以上项目中需要进行的软件测试部分。我会继续跟进项目的测试，并于后期进行回归性测试，以提供给使用者完美的用户体验，提升该项目的鲁棒性，使之能成功应用于更多在线学习的 Web 项目场合。