



赛事管家：分布式射箭比赛管理系统

— 设计说明书

参赛编号：190411

参赛学校：上海对外经贸大学

参赛成员：吴烜圣 郭菁菁 杨非池

2019 年 3 月 19 日

1 简介.....	3
1.1 项目背景及创意.....	3
1.1.1 创意来源.....	3
1.1.2 需求分析.....	3
1.1.3 类似项目分析.....	3
1.1.4 项目创意——基于“B/S + C/S 混合部署架构”的射箭赛事管理系统.....	4
1.1.5 项目应用效果——应用于“2018 中国大学生射箭（射艺）锦标赛”.....	4
1.2 项目规划.....	5
1.2.1 项目目标.....	5
1.2.2 人力资源计划.....	5
1.2.3 软硬件资源计划.....	5
2 总体设计.....	6
2.1 “B/S + C/S 混合部署架构”设计.....	6
2.2 系统功能.....	6
2.2.1 功能概述.....	6
2.2.2 功能说明.....	7
2.3 系统开发/运行平台.....	8
2.4 关键技术.....	8
2.4.1 系统架构——“B/S + C/S”的架构设计.....	8
2.4.2 数据储存和操作——SQLite3 与 DaPy 库.....	9
2.4.3 友好的用户交互界面——wxPython 库.....	9
2.4.4 云服务器后端——Flask 库.....	9
2.5 作品特色.....	9
3 详细设计.....	10
3.1 系统架构.....	10
3.1.1 技术架构.....	10
3.1.2 功能模块.....	11
3.1.3 关键功能/算法.....	11
3.2 数据结构.....	12
3.2.1 持久化数据.....	12
3.2.2 临时数据.....	15
3.2.3 缓存数据.....	15
3.3 客户端界面.....	16
3.3.1 界面设计风格.....	16
3.3.2 客户端主要功能页面.....	16
3.4 网页端界面.....	20
3.4.1 界面设计风格.....	20
3.4.2 网页端主要功能界面.....	20
4 系统安装及使用说明.....	23
5 总结.....	25
6 附录.....	26
6.1 名词定义.....	26
6.2 参考资料.....	26
6.3 源代码清单.....	26

1 简介

1.1 项目背景及创意

1.1.1 创意来源

射艺将中华民族体育、艺术与文化相容，通过“寓教于射”的形式来强化射者“心志之和”和社会“天下之和”的理念。我校上海对外经贸大学近年来积极推进各项射艺赛事，从2015年举办的第一届中国大学生射箭射艺邀请赛至2017年底，共计举办了三场百人以上的大型射箭比赛，另外举办了多场50-100人的中等规模比赛。随着我校射箭比赛举办的常态化以及赛事规模的不断升级，专门为赛事的顺利承办而设计自有的射箭比赛赛事管理系统显得尤为重要。

1.1.2 需求分析

射箭比赛管理系统要求能尽可能地辅助赛事管理者组织一场射艺比赛。一场射艺比赛包含数个环节，分别是：1)“比赛报名”指赛事组委会通过某种渠道采集有意愿参赛人员的相关信息，并对参赛人员的资质进行审核；2)“排位赛编排”指采集到参赛人员的相关信息后，在正式比赛日之前编排好个人排位赛的靶位安排；3)“首轮淘汰赛编排”指排位赛结束后，依据排位赛成绩编排出第一轮淘汰赛的靶位安排；4)“逐轮淘汰赛编排”指依据上一轮淘汰赛的成绩，编排下一轮淘汰赛的靶位安排；5)“成绩报表”指系统能够自动地生成整场比赛所有场次记录的成绩报表。对于任何一场正规的锦标赛级别的体育赛事而言，都要求保留有运动员签名、裁判员签名及记分员签名的纸质版成绩单。

上述工作如果由人工配合Excel软件进行操作不仅低效，而且正确率较低。因此，主流的射箭比赛（商业、非商业）都会采用赛事管理系统作为软件方面的支持。作为一款赛事管理系统，其用户群体有两类：1)以记分员、裁判员为代表的赛事管理者；2)以运动员和教练员为代表的赛事参与者。大部分情况下，赛事管理者是赛事管理系统的直接使用者，而赛事参与者则是赛事管理系统的间接使用者。

结合我校以往赛事管理方面的经验，我们认为一款优秀的射箭比赛赛事管理系统应该包括发布比赛信息、参赛报名、自动编排、比赛成绩录入、生成统计报表这五大部分。其中，发布比赛信息按照“比赛前”、“比赛中”和“比赛后”分别对应于公告赛事文件、公布比赛成绩和发布比赛成绩册三个部分。系统自动编排包含排位赛、首轮淘汰赛和逐轮淘汰赛三个部分的比赛场次。

1.1.3 类似项目分析

现在国内常用的赛事系统只有两套。一种是由国家体育总局组织开发的“单节点架构”系统，而另一种是由某些商业联赛组织开发的“B/S架构”系统。

国家体育总局提供的“单节点模式”系统流程为：1)由记分员使用纸质成绩单记录；2)待该轮次比赛结束后，汇总纸质成绩单；3)编排人员通过一台设备录入数据。这种模式的赛事软件虽然符合国际箭联对锦标赛关于“纸质版成绩转录电脑”的要求，但因为需要单人在短时间内通过一台客户端处理大量数据，所以实践中错误率较高。另外，这种单节点模式对于大型赛事来说效率偏低，两个场次之间的等待时长可能多达一小时，不利于安排紧凑的时间表。

商业联赛组织开发的“B/S架构”系统的流程为：1)多名记分员直接使用手机软件录入分数；2)比赛记录上传云端服务器；3)编排人员在后端下载比赛记录进行编排。为了能够能应对大规模的比赛，该公司放弃了作为凭证的纸质版计分表。这种改动不符合锦标赛的严谨性与规范性，导致其仅能应用于商业联赛，从根本上失去了在正规比赛中大规模推广的可能性。另外，其B/S

架构过分依赖畅通的广域网连接，易受到比赛场地的网络状况影响。在实践中，户外比赛场地常常伴随着不通畅的网络、较差的移动信号，该软件由于不能在本机储存数据，很难在全国尤其是偏远欠发达地区推广，从外部扼杀了该软件大规模使用的可能性。

1.1.4 项目创意——“B/S + C/S 混合部署架构”的射箭赛事管理系统

利用团队首创的“**B/S + C/S 混合部署架构**”，“**赛事管家**”是第一个同时满足“必须有纸质版的成绩单”（**严谨**）、“400 名运动员比赛结束后 10 分钟内出结果”（**高效**）和“不受制于比赛场地网络状况”（**稳定**）三个条件的射箭比赛赛事管理系统。赛事管家通过提供一系列的赛前赛后管理功能（比赛信息公示、在线报名、靶位编排、实时成绩查询、成绩录入、赛事成绩册），让赛事管理者和赛事参与者都有着更为沉浸式的赛事体验。

所谓“**B/S + C/S 混合部署架构**”即用户可以依据承办比赛的规模、场地网络环境、志愿者数量、服务水平以及预算等多方面条件将整套系统拆解为：单节点模式、C/S 架构、“B/S 架构群+C/S 架构群”和“B/S + C/S 混合部署”四种形式。图 1 的流程图详细说明了赛事管家一站式服务的实现过程，具体的架构实现方法请参考 2.1 章节的“B/S + C/S 混合部署架构”设计部分。

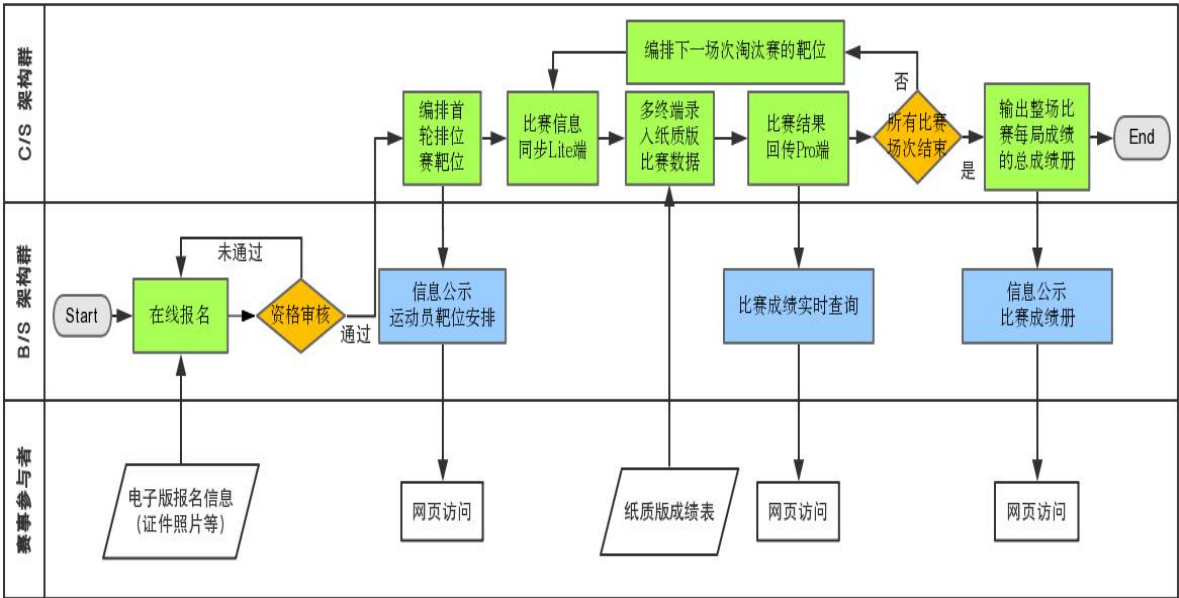


图 1 基于“B/S + C/S 混合部署架构”的一站式射箭比赛赛事管理服务

1.1.5 项目应用效果

赛事管家顺利支持了由中国大学生体育协会主办、上海市大学生体育协会和上海对外经贸大学承办的、超过 400 名运动员参加的“2018 第二届中国大学生射箭（射艺）锦标赛”。2018 年 10 月 10 日，“信息公示”功能启动。10 月 15 日，为期两个星期的“线上报名”功能开放，共采集 72 只代表队超 410 名运动员的报名信息。11 月 9 日-11 日，赛事管家以完全符合大赛时间安排表的表现支持了整场锦标赛的顺利进行。11 月 10 日中午 2 点，“在线成绩查询”功能迎来了峰值访问量，成绩查询的同时在线人数达到 1430 人。11 月 11 日晚 8 点，官网正式发布了整场比赛的成绩册，本届锦标赛宣告顺利闭幕。赛事组委会对本系统给予了高度评价。

另外，赛事管家在公开测试阶段（2018 年 5 月）以赛事编排、成绩录入、成绩报表三个功能支持了 80 余名运动员的“2018 中国大学生射箭（射艺）锦标赛——预选赛”。该场比赛组委会同时还应用了一套基于 B/S 架构的赛事系统——拓世助手。在比赛中由于网络中断导致“拓世助手”服务中断，而赛事管家以其优越的稳定性支撑起了整场比赛的管理服务。这次危机中表现出的稳定性是 2018 年 11 月的大赛中组委会最终选择赛事管家的重要原因。

1.2 项目规划

1.2.1 项目目标

本项目的目标是自 2017 年 12 月 1 日至 2018 年 10 月 1 日的 11 个月内完成“赛事管家”射箭比赛赛事管理系统的研发工作并且完成使用人员的培训工作。其需要具备的基本功能共有六个，分别为信息公示、在线报名、靶位编排、实施成绩查询、成绩录入和赛事成绩册。以下是对于上述功能一些具体要求：1) 在成绩录入方面，系统能够具备 200 人的成绩录入工作在 15 分钟以内完成的能力；2) 靶位编排功能要求完全自动化，输入相应的配置参数后迅速生成；3) 要确保无网络环境下的系统可用性。

整个项目按照阶段性检验成果可以分成若干个子任务，每个任务的具体信息如表 1 所示。

表 1 项目子任务安排表

编号	任务名称	预计用时	开始时间	说明
1	需求分析	4 周	2017-12-1	撰写《功能及界面设计》
2	架构设计	2 周	2018-1-1	绘制架构图；撰写《技术说明》
3	交互体验设计	6 周	2018-1-1	绘制前端、客户端所需美工素材
4	代码编写	16 周	2018-1-15	编写所有的代码文件
5	代码审阅	8 周	2018-5-15	两名 coder 交叉审阅代码
6	单元测试	6 周	2018-7-15	针对每一个功能进行单元测试
7	文档撰写	6 周	2018-7-1	撰写《使用说明书》
8	模拟压力测试(100 名运动员)	1 周	2018-8-1	团队使用虚拟技术内测
9	模拟压力测试(500 名运动员)	1 周	2018-8-8	团队使用虚拟技术内测
10	真实压力测试(100 名运动员)	2 周	2018-8-15	招募志愿者依据历史数据公测
11	真实压力测试(200 名运动员)	3 周	2018-9-1	招募志愿者依据历史数据公测
12	用户培训	1 周	2018-9-21	为赛事组委会培训志愿者
13	交付		2018-10-1	

1.2.2 人力资源计划

本项目的开发团队共有成员三人，分别为吴烜圣、杨非池和郭菁菁。针对各个成员的能力以及技术背景，我们对每个成员进行了具体的分工如下表 2。除了上述的团队内部成员外，团队还可以通过志愿者的方式招募最多 20 名临时成员。

表 2 项目人员分工

成员	分工	任务细节
吴烜圣	产品规划、架构	需求分析、系统架构设计、代码编写、代码审阅
杨非池	编码、测试	代码编写、代码审阅、单元测试
郭菁菁	测试、文档、交互体验设计	设计绘制美工素材、压力测试、文档编写、用户培训

1.2.3 软硬件资源计划

在硬件方面，本团队共配备了 3 台笔记本电脑及 2 台阿里云服务器；在软件方面，操作系统同时有 64 位的 Windows 和 Mac，每台设备都安装了 Python2.7 及相关的依赖环境。拥有付费版的 Office 套件一套，Photoshop 一套。项目管理使用 Teambition 平台的敏捷开发框架进行任务安排和代码管理。初步评估本团队所具备的软硬件资源能够进行交互体验的设计、前端及客户端美工素材的绘制、代码实现、文档撰写、外网环境部署等任务。

2 总体设计

2.1 “B/S + C/S 混合部署架构”设计

赛事管家最基本的组成单元为一台 Pro 版本客户端（以下简称：Pro）。一个 Pro 能够实现“编排靶位”、“成绩录入”和“成绩报表导出”三个最核心的功能，此时该系统类似于国家体育总局提供的单节点系统，适合小规模的比赛场景。

在运动员规模较多时，赛事管家系统可以为 Pro 添加多个仅具有“编排靶位”功能的 Lite 版本客户端（以下简称：Lite）。Pro 和多个 Lite 之间可以通过局域网或者最为传统但极其稳定的 U 盘来同步数据文件，此时整套系统构成了“C/S 架构群”。这种“分布式”处理的设计很大程度上减少了人工处理数据出错的情况，同时具有应对大规模比赛的可能。相比较于商业联赛的 B/S 架构，此架构对于网络的要求较低，则在网络环境较差时的系统仍然可以工作。

为了便于向赛事参与者（运动员、教练员）采集和发布相关的信息，赛事管家还能在云服务器上部署“Web 服务器”和“DB 服务器”。他们能够提供“信息公示”、“在线报名”两个基本功能，这部分的系统被我们称为“B/S 架构群”。

在同时具有上述 C/S 架构群和 B/S 架构群的时候，本地的 Pro 可以直接将每一轮次的比赛数据通过互联网上传至 DB 服务器中，B/S 架构群开始提供“在线成绩查询”功能。公布成绩的网页自适应至手机、电脑、平板等各种平台，参赛者可以随时随地查看自己的成绩。在赛事管家提供“在线成绩查询”功能时，整套系统同时拥有两种架构群，并在业务逻辑上实现了两架构群之间的耦合，我们称为“B/S + C/S 混合部署架构”。“赛事管家”真正做到了“裁判不用急，选手不用等”，我们可以把有关比赛的一切交给这个可靠的“管家”。

2.2 系统功能

本系统具备“比赛信息公示”、“在线比赛报名”、“靶位编排”、“成绩录入”、“成绩报表导出”、“在线成绩查询”等 6 个核心功能。相较于传统的“纸笔记录、人工统计”的做法，赛事管家以友好的用户界面及良好的兼容性、稳定性，让比赛组织者和裁判员更加高效地完成从赛前到赛后的赛事管理工作，让教练员和参赛选手更便捷及时地了解自己的排名和比赛结果。

2.2.1 功能概述

赛事管家核心框架包括云端服务器、本地 Pro 版客户端、本地 Lite 版客户端，功能包括在线比赛报名、靶位编排、成绩录入、成绩表导出、在线成绩查询等，其功能框架图如图 2 所示。

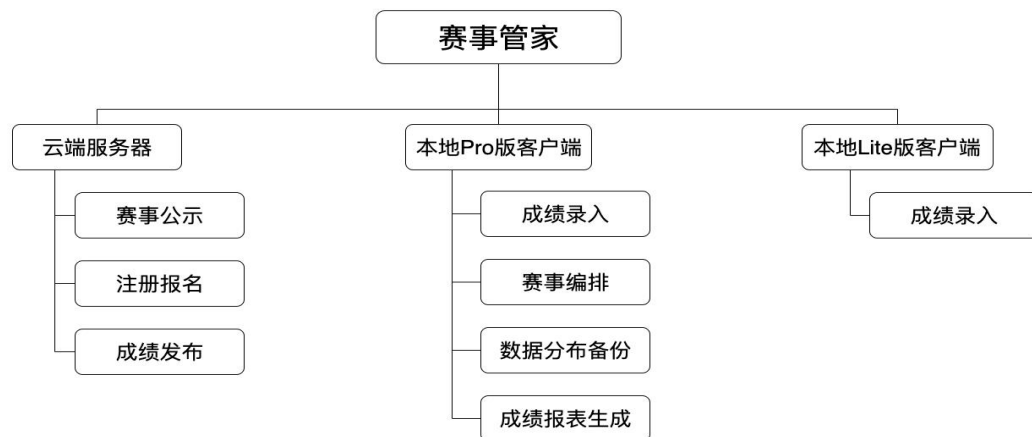


图 2 系统功能框架图

2.2.2 功能说明

1) 在线注册报名（面向运动员）

在开赛前，参赛选手与教练员需在赛事管家官方主页选择相应的赛事进行注册报名。验证报名邮箱后，运动员的注册信息即可储存进云端的数据库中，以便后续登陆查询成绩操作。

2) 成绩录入（面向裁判员）

在赛事管家的 Pro 版和 Lite 版桌面客户端中，裁判员可进行比赛成绩的分布式录入。

- i. “排位赛”成绩录入界面分为左右部分，左边用于用户信息的设定和查询，右侧用于成绩的录入。每输入一组记录，都先在左侧的信息栏中填入运动员信息，而后点击“查询信息”按钮。运动员信息可以只输入其中一项，如果所有信息自动补全，则代表系统探测到了该运动员数据，右侧“提交记录”按钮将会点亮。
- ii. 裁判员可以一次只输入一个距离（10 米靶、20 米靶等）的比赛记录并提交，当裁判员再次查询该运动员信息时，客户端会自动显示已存在的数据记录。
- iii. 点击“提交记录”，若界面成功刷新则表示数据已经被写入。（Lite 版客户端将数据通过 LAN 加密传输给 Pro，Pro 版客户端将数据储存在本地。）

3) 赛事编排（面向赛事管理员）

a) 自动编排排位赛靶位安排表

- i. Pro 版客户端从云端服务器的数据库中下载一份保存着运动员姓名、性别、俱乐部名称的基本数据，并将其存在本地。
- ii. 选择数据文件，设定起始靶位号及单性别最多运动员数量，选择“编排”。
- iii. 系统随机打乱俱乐部次序后，按照“单靶位无同俱乐部成员”、“同俱乐部成员就近安排”、“人数不足的俱乐部在队列末尾”这三个原则自动完成赛事编排。
- iv. 完成编排后，系统会将信息记录保存一份在本地数据库中。与此同时，Pro 版客户端将信息通过 LAN 加密传输给 Lite 版客户端。

b) 编排首轮(个人&团体)淘汰赛靶位编排

完成排位赛成绩录入后，“赛事管家”可以短时间内完成淘汰赛靶位编排。

- i. 管理员需要设置首轮淘汰赛的赛程进度、起始靶位号及单靶位运动员数量。
- ii. 如果不同性别组的赛程进度不同，管理员可以先勾选其中一个性别编排。
- iii. 确认操作后，系统会自动地在本地数据库中建立选择赛程的数据文件，此时裁判员可以开始进行淘汰赛成绩录入。
- iv. 完成编排后，Pro 版客户端会将信息记录保存一份在本地数据库中。与此同时，Pro 版客户端将信息通过 LAN 加密传输给 Lite 版客户端。

c) 编排首轮(个人&团体)淘汰赛靶位编排

完成了首轮淘汰赛的成绩录入后，管理员需要进行逐场的淘汰赛编排。

- i. 相较于首轮淘汰赛靶位安排，系统会自动识别各个性别组正在进行的赛程以及下一赛程。裁判员需要完成每一轮每一位运动员的成绩录入后，才能够进行下一轮的靶位安排。如果出现该名运动员弃赛的情况的，裁判员需要在成绩录入中为该 名运动员填入“0”的成绩。
- ii. 完成编排后，Pro 版客户端会将信息记录保存一份在本地数据库中。与此同时，Pro 版客户端将信息通过 LAN 加密传输给 Lite 版客户端。

4) 成绩查询（面向运动员）

在每一场次比赛结束后，Pro 客户端将储存在本地的所有比赛数据通过 WAN 加密传输至云端。云端后台处理后渲染至 Web 前端，此时运动员可以登陆查看自己的成绩排名和下一个比赛日的靶位安排。

2.3 系统软硬件平台

2.3.1 系统开发平台（含开源/第三方工具）

1) 硬件配置

表 3 开发环境硬件配置

硬件	型号
处理器	Intel Core i9 9900K
主板	TUF B360M-PLUS GAMING S
内存	32GB
显卡	ROG-STRIX-GeForce RTX2080 TI-011G
操作系统	Windows 10 Professional x64
网卡	Intel E10G42BFSR

2) 软件配置

表 4 开发环境第三方工具

依赖库	版本
Python	2.7.15
DaPy	1.8.1.2
Flask	1.0.2
wxPython	4.0
sqlite3	3.27.1
requests	3.0.4

2.3.2 系统运行平台

1) 硬件配置

表 5 最低运行配置

硬件	型号
处理器	Intel Core i3 4010U
主板	Inter(R) 82865G
内存	4GB
显卡	Intel (R) Q33 Express Chipset Family
操作系统	Windows 7 home basic 64 Bit

2) 赛场系统搭建

一般情况下，顺利举行一场大规模的（400 人左右）的传统弓比赛只需要 1 台装有 Pro 版客户端的电脑和 20 台 Lite 版客户端的电脑，即可保证在每场比赛的 10 分钟内生成所有的赛事结果；其中 Pro 版本与云端服务器通过 WAN 通讯而 Lite 版与 Pro 版之间通过搭建 LAN 通讯，Lite 版相互之间不进行通讯；通过阿里云等云服务器平台部署 1 台“单核 2GB 内存”的服务器来提供所有的在线服务。

2.4 关键技术

2.4.1 系统架构——“B/S + C/S”的架构设计

B/S 架构采取浏览器请求，服务器响应的工作模式。在本系统中，参赛运动员和教练员通过赛事管家官网网页进行在线比赛注册报名、查看成绩等功能，架设在云端的服务器使用 Flask 框架对浏览器请求进行处理与响应。

C/S 通过将任务合理分配到 Client 端（Lite 版客户端）和 Server 端（Pro 版客户端），降低了

系统的通讯开销。客户端和服务端端的程序不同，在本系统中，Lite 版客户端主要负责各个靶位的成绩录入，而 Pro 版客户端则主要负责数据收集整合以及赛事安排等任务。

2.4.2 数据储存和操作——SQLite3 与 DaPy 库

SQLite3 是一个轻型的嵌入式数据库引擎，占用资源非常低，用户无需安装 MySQL 等第三方程序即可轻松地管理比赛数据。

DaPy 是一个针对大数据处理的 Python 开源支持库。它支持用户在内存层面迅速地处理大批量的表结构数据，提高了系统的工作效率。本系统使用 DaPy 配以相应的算法，可以在 1 秒内跨十几张数据表生成整场比赛的成绩册。

2.4.3 友好的用户交互界面——wxPython 库

wxPython 是 Python 一套优秀的 GUI 图形库。允许本团队很方便地创建完整的、功能健全的 GUI 用户界面。本系统的界面功能完整、美观大方、操作简单，是“用户友好型”界面的典范。

2.4.4 云服务器后端——Flask 库

Flask 是一个使用 Python 编写的轻量级 Web 应用框架，每秒钟可以响应大约 120 个访问请求。本系统使用 Flask 框架搭建云服务器后端，具有很好的稳定性及功能拓展性。在我们实际使用中，我们的系统经受住了每分钟 40 个以上新增 TCP 长连接的请求。

2.5 作品特色

得益于“C/S + B/S 混合部署架构”，赛事管家是全球首套针对射箭比赛、符合世界射箭联盟赛事要求的分布式赛事管理系统。我们团队致力于为赛事管理者（主办方、裁判员）和赛事参与者（运动员、教练员）提供一站式的赛事服务。

从赛事管理者方面来说，本系统采用 C/S 架构，以“单 Pro 多 Lite”的形式使得比赛数据能够及时有效地被处理，并自动生成成绩表。该结构以本地局域网为载体，巧妙地克服了比赛场馆由于地理环境等因素与公网连接效率较低的困难。

从赛事参与者角度来说，“赛事管家”采用 B/S 架构，以网页形式向运动员与教练员提供在线注册报名、在线成绩查询、在线赛程查询等一系列便捷的功能。

相比于传统的比赛组织模式，本系统依托于局域网但可以拓展至广域网，给予了管理者、参赛者极大程度的便利，顺应了“互联网+”的时代发展潮流，同时也弘扬了中华传统射艺中“射以观德”的价值观。此外，本系统采用的“C/S + B/S 混合部署架构”不仅仅能够应用于传统射箭比赛，也可以推广至所有户外竞技体育项目，具有非常强的现实意义和应用价值。

3 详细设计

3.1 系统结构设计

3.1.1 技术架构

赛事管家通过 http 协议, 将 B/S 架构和 C/S 架构相结合, 构造了一套稳定、易用且高效的赛事管理系统。使用 SQLite3 和 DaPy 作为数据存储和数据操作方面的技术框架。客户端的开发使用了轻量级的 wxPython 图形库, 而服务器端的 API 接口及网页请求主要通过 Flask 框架进行开发。

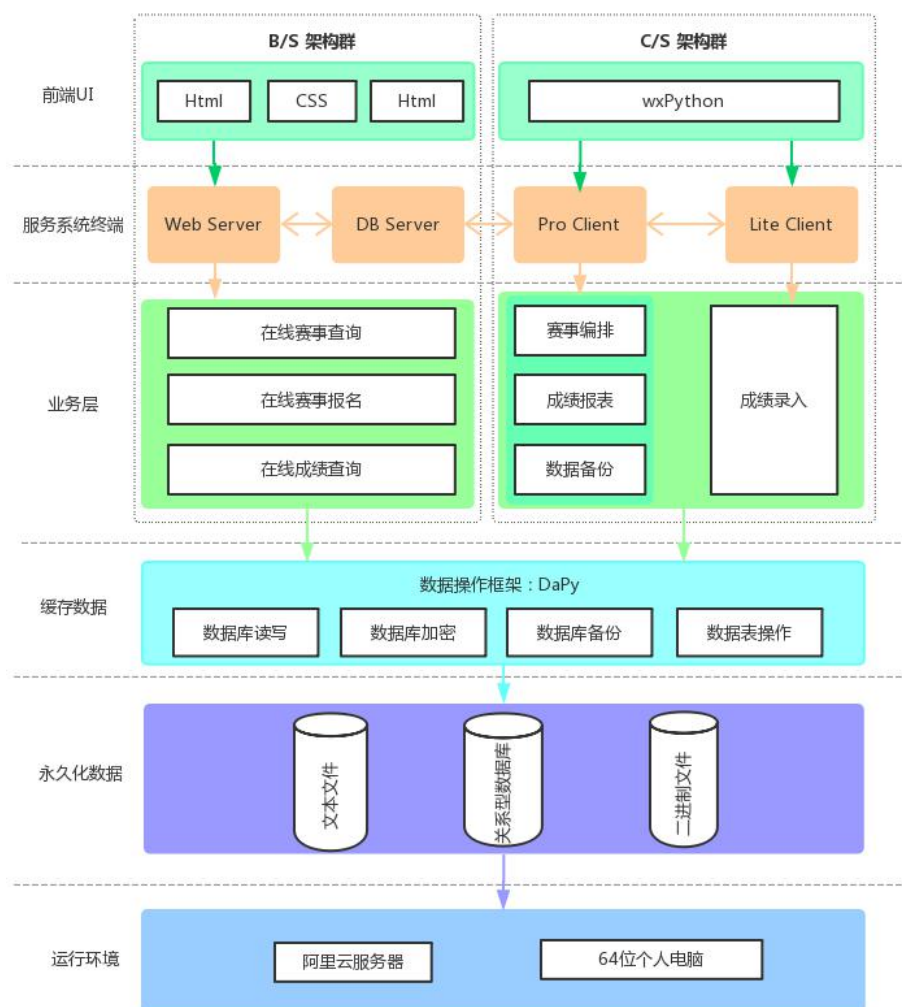


图 3 系统技术架构图

B/S 架构主要负责支持以参赛运动员为群体的主要客户群体, 他们对于参赛的比赛结果以及访问方式有着“及时”和“方便”两种需求。选用 B/S 架构能够降低该客户群体的加入方式成本, 并利于按照客户数量, 灵活地进行系统扩容。

C/S 架构主要负责为以赛事管理人员为代表的客户群体提供电子化的解决方案, 他们对于系统的要求为稳定性、安全性和高效性。在这种情况下, 基于本地局域网构建的分布式架构可以帮助数据的处理提高效率, 同时保证了良好的系统健壮性。

SQLite3是一款轻量级且遵守遵守 ACID 的关系型数据管理库, 它包含在一个相对小的 C 库中。SQLite3 技术帮助 Pro 和 Lite 在免安装的情况下可以有组织地存储比赛数据; 通过分布式的存储技术及相关的协议方案, 使得数据的安全性进一步得到了提升。

DaPy 作为一个简单易用的 Python 数据处理/操作框架，它支持用户在内存层面迅速地处理大批量的表结构数据，提高了系统的工作效率。在本项目中，为了能够在跨表查询的情况下短时间整理出一场比赛的最终成绩册报表，DaPy 毫无疑问是最适合的技术方案。

Flask 是一个开源的 web 微服务支持方案，他可以支持轻量级的网页服务需求。赛事管家系统主要通过 Flask 框架构建了 Web 服务器的相关服务，并构建了 DB 服务器的 API 接口服务。相较于 Django 等其他框架，Flask 的启动更为轻便，并且支持热更新技术。

wxPython 使得我们能够轻松的创建具有健壮、功能强大的图形用户界面的程序，同时还具有良好的跨平台性。相较于 PyQt 等技术方案，wxPython 封装的程序包更为轻便，同时也能满足较好的用户体验。作为对于美术性要求不太高的赛事服务人员而言，PyQt 的自绘功能虽然强大，但却导致较长的开发周期及较大的安装程序。

3.1.2 功能模块设计

系统按照对于不同用户群体，将功能分为了两大部分：在线功能和本地功能。在线功能部分设计的思路为“用户通过自由网络通过访问网页的形式获取关于赛事的实时信息”。因此，在线功能分为“赛事信息公布”、“成绩查询”、“在线报名”三大部分，这些服务都是通过访问基于 DB Server 提供的在线数据支持接口服务器转接 Web Server 网页服务器来维持的。另一方面的本地功能部分的 Pro 客户端主要是完成“赛事编排”、“报表生成”和“数据备份”功能。对于 Lite 客户端，其主要功能为完成相应的数据录入工作，“数据录入”即为其唯一的功能。

基于以上功能，系统主要设计为了四个模块分别为：Web Server、DB Server、Pro Client 和 Lite Client。Web Server 和 Pro Client 通过 http 协议搭建的接口与 DB Server 进行联络；Pro Client 和 Lite Client 之间通过局域网环境下的 http 协议进行联络。在所有的联络过程中，均会调用到相同的加密模块。负责成绩录入的两种客户端，均共享相似的桌面设计模块及数据交换模块。Pro Server 单独享有用于数据报表生成的模块。具体的模块信息请见图 4：

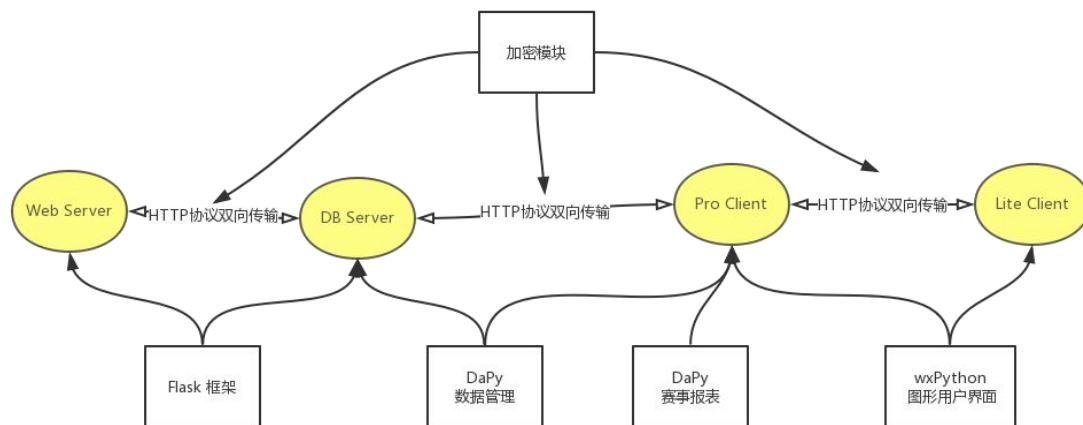


图 4 功能模块架构图

3.1.3 关键功能/算法设计

赛事管家得益于高效的算法来完成赛事编排任务，赛事管家 Pro 客户端，仅用不到 1 秒的时间，就能编排超过 400 名运动员比赛数据。类似的软件，该任务的操作时间大约为 1 分钟。我们仅介绍赛事管家中最为核心的两个功能所使用的算法，“排位赛编排”和“团体淘汰赛编排”，并使用伪代码的形式在“图 5”加以说明。

“排位赛编排”功能有如下要求要求：1）单靶位无同单位运动员；2）同单位运动员就近安排；3）人数不足的运动员使用编号为末尾的靶位。“团体淘汰赛编排”功能要求能力差距悬殊的运动队先进行较量，技术水平相近的运动队后进行较量。

算法 1 运动员信息随机编排比赛靶位

输入: 每队最大参赛人数MaxAthlt, 起始靶位Begin, 单靶位人数Limit, 运动员信息athlete
返回: 运动员编号、靶位号

```

1: function SperateSex(athletes)
2:   SubSheet: boys, girls
3:   boys ← athletes.select("Gender == 男")
4:   girls ← athletes.select("Gender == 女")
5:   return boys, girls
6:
7: function SperateSchool(boys, girls)
8:   List Of String: MaleSchool, FemaleSchool, TotalSchool
9:   MaleSchool ← boys.School
10:  FemaleSchool ← girls.School
11:  TotalSchool ← MaleSchool ∪ FemaleSchool # 男女子运动员均存在的学校
12:  return MaleSchool, FemaleSchool, TotalSchool
13:
14: function CountStudent(student)
15:   Integer: counts
16:   counts ← student.most_common('School') # 人数最多的学校
17:   return counts
18:
19: function GroupAbsent(Schools)
20:   # 按照每所学校不足的人数进行分组
21:   Dict[int] → list: absents ← {0:[], 1:[], 2:[], 3:[]} # 缺少int名运动员
22:   for school in Schools
23:     absent_num ← count(school) - MaxAthlt
24:     absent[count_school].Add(school.Name)
25:   # 对每一个分组的内的学校打乱顺序
26:   for school in absent
27:     random(school)
28:   return absents
29:
30: function GetTargetNo(Begin, students, school, G, athletes)
31:   List Of Integer: No ← []
32:   List Of String: target ← []
33:   Integer: j ← 0
34:   Integer: Totaltimes ← count(students) / (athletes * MaxAthlt)
35:   for T = 1 to (athletes + Totaltimes)
36:     TargetN = Begin - athletes + athletes * T
37:     i ← 0
38:     for tg in [甲, 乙, 丙, 丁, 戊, 己]
39:       i ← i + 1
40:       for k = 0 to athletes
41:         if count(students) != j
42:           No.append(10 + str(index(school)) + G + k)
43:           target.append(TargetN + k + tg)
44:           j ← j + 1
45:       end if
46:   return No, target
47:
48: boys, girls ← SperateSex(athlete)
49: Mschool, Fschool, Tschool ← SperateSchool(boys, girls)
50: mostMschool ← CountStudent(Mschool)
51: mostFschool ← CountStudent(Fschool)
52: absentMschool ← GroupAbsent(Mschool)
53: absentFschool ← GroupAbsent(Fschool)
54: IDMale, TargetMale ← GetTargetNo(Begin, Limit, MSchool, '男', MaxAthlt)
55: IDFemale, TargetFemale ← GetTargetNo(Begin, Limit, FSchool, '女', MaxAthlt)

```

算法 2 基于排位赛成绩编排团体淘汰赛对阵图

输入: 首轮淘汰赛进程名Types, 起始靶位号Begin, 最大参赛人数MaxAthlt, 排位赛记录Records
返回: 进入1/Types淘汰赛的队伍编排信息, 包括入选的运动员、总成绩、靶位号及排位赛排名

```

1: function OrderedList(Integer: Type)
2:   '''给定场次数据, 返回该场次的对阵序列
3:   该序列实行末位淘汰制, 即实力差异最小的选手要在越后面的场次中相遇
4:   参数Type为首轮比赛场次中的数字, 如“32进16淘汰赛”或“1/16淘汰赛”输出16
5:   ...
6:   sums ← 2 * Type + 1 # 赛程需要运动员总数 + 1
7:   List Of Tuples: Group1, Group2
8:   for t = 1 to (Type + 1)
9:     Tuple: bound ← tuple(t, sums - t)
10:    Group1.Add(bound)
11:   for t = 0 to (i / 8 + 1)
12:     Integer: size_G1 ← count(G1)
13:     for k = 0 to (size_G1 / 2)
14:       bound ← tuple(Group1[k], Group1[size_G1 - k - 1])
15:       Group2.Add(bound)
16:       if Group2.is_empty() is True
17:         break;
18:     else
19:       Group1, Group2 = Group2, [];
20:   end if
21:   return Group1中的所有数字 - 1
22:
23: function Arrange_Mix(Records, tgt, Orderlist, max_athlt)
24:   # 混合团体淘汰赛资格
25:   List Of String: Schools
26:   Schools ← 至少有1名男子及1名女子的学校队列
27:   # 资格学校的运动员排位赛成绩表
28:   Sheet: MSchool, FSchool
29:   for sex, sexSchool in [("男", MSchool), ("女", FSchool)]
30:     for record in Records.select("性别 == sex")
31:       if record.运动员学校 in Schools
32:         useful_keys ← tuple(姓名, 总分, 总的", 总"正", 学校);
33:         sexSchool.Add(useful_keys);
34:   # 每所学校最好的一名运动员
35:   Dict[学校] → 运动员信息: bestM, bestF
36:   for school in Schools
37:     for sexBest, sexSchool in [(bestM, MSchool), (bestF, FSchool)]
38:       SubSheet: best
39:       best ← sexSchool.select("学校 == school")
40:       best.sort(("总分", DESC), ('总的"', DESC), ('总"正"', DESC))
41:       sexBest[学校] = best[0]
42:   # 带运动员信息的参赛学校表
43:   Sheet: rank_schools
44:   for school in Schools
45:     Integer: GTotal, GDi, GZheng
46:     GTotal ← bestM[school].总分 + bestF[school].总分
47:     GDi ← bestM[school].的 + bestF[school].的
48:     GZheng ← bestM[school].正 + bestF[school].正
49:     rank_schools.Add([school, GTotal, GDi, GZheng, 男子姓名, 女子姓名])
50:   Integer: length_school ← count(rank_schools)
51:   if length_school < (i + 1) * 2
52:     for times = 0 to ((i + 1) * 2 - length_school)
53:       rank_schools.Add(['轮空', 0, 0, 0, '-', '-'])
54:   end if
55:
56:   # 对入选学校按照团队总分进行排名
57:   rank_schools.sort(('总分', 降序), ('总的"', 降序), ('总"正"', 降序))
58:   # 编排靶位
59:   Dict[靶位] → 中文名: dicPst ← {1: 甲, 2: 乙, 3: 丙, 4: 丁}
60:   Integer: tgtPeople ← 1
61:   Sheet: schedule # 列名: 靶位, 排名, 单位, 运动员(男), 运动员(女)
62:   for j in Orderlist
63:     record: member ← rank_schools[j]
64:     string: tgt ← str(tgt_Mix) + dicPst[tgtPeople] # 3甲
65:     schedule.Add(tgt, j+1, record.学校, record.男子姓名, record.女子姓名)
66:     if tgtPeople == tgt_Max:
67:       tgtPeople ← 1;
68:       tgt_Mix ← tgt_Mix + 1
69:     else
70:       tgtPeople ← tgtPeople + 1
71:   end if
72:   return schedule
73: Order_list ← Orderedlist(Types)
74: schedule ← Arrange_Mix(Records, Begin, Orderlist, MaxAthlt)

```

图 5 核心算法演示

3.2 数据结构设计

本系统中的数据结构按照数据的用途及应用场景区分为“持久化数据”、“临时数据”和“缓存数据”共计三个数据类型。每一个数据类型在不同场景下都选择了综合考虑最优的数据结构进行有组织地存储，具体的设计细节在紧接着的三个小节中进行了讨论。

表 6 数据结构设计表

数据类型	应用场景	使用者	数据结构
持久化数据	比赛记录数据	Pro、 Lite	SQLite3
	参赛报名信息	DB Server	SQLite3
	比赛报表	用户、 Pro	Excel
	加密数据库	Pro	凯撒加密文本文件
临时数据	局域网数据传输	Pro、 Lite	哈希加密文本文件
	赛事数据本地共享至云端	DB Server、 Pro	Pickle 二进制文件
缓存数据	生成赛事报表	Pro	DaPy.SeriesSet 数据框架

3.2.1 持久化数据

在系统中，持久化数据包括了用于系统需要频繁调用但不应加载到内存的 SQLite3 数据库，和便于用户跨应用读取的 Excel 文档两种基本类型；并延伸出了一种用于数据备份的加密数据。

1) 数据库 SQLite3

SQLite3 因“低硬件需求”、“易于部署”和“高性能”的三个特性，被我们认为是最适合本项目的关系型数据库。我们在 DB 服务器、Pro 及 Lite 上均应用了该数据库来存储数据。由于本系统中数据库内部的表结构会因不同的比赛场次动态地进行修改；同时，由于射箭比赛的数据信息庞大，导致每一张表的字段较多。在此仅通过下面的数据库表结构设计图阐述我们设计数据库的思想，并附上两张最基本的数据表。

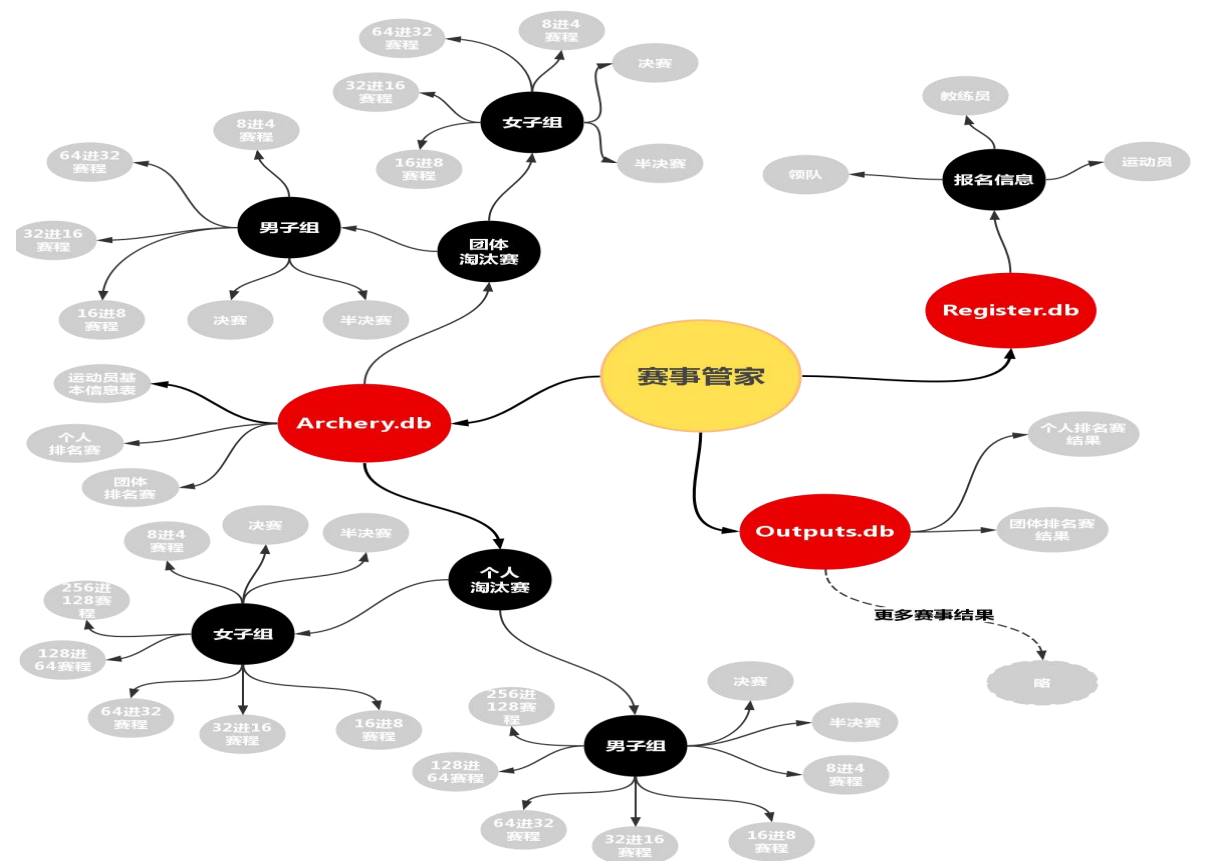


图 6 赛事管家数据库表结构设计

图 6 阐述了赛事管家中，用于客户端数据存储的 Archery.db、用于 DB Server 存储报名信息信息的 Register.db 和用于公示比赛记录的 Outputs.db 三个数据库。Archery.db 和 Outputs.db 中的每一张表都代表了一场场比赛的结果（任意一场排位赛或淘汰赛），但是在 Outputs.db 中记录的字段会比 Archery.db 中同样的一张表少记录了很多字段。

表 7 展示了参赛团体通过网页端注册用户并成功报名后，保存在 DB Server 中的数据信息。在传统的报名模式中，上述信息都是通过发送电子邮件的方式汇总到赛事组委会。组委会在收到邮件后再组织人力录入 Excel 生成数据表，十分耗费人力。在本系统中，所有的报名数据在用户第一次点击“提交报名”功能的时候先以临时存储格式存储在 DB 服务器上，同时系统会自动生成一封包含报名信息的确认邮件至领队的邮箱。当领队核验了邮件中显示的运动员信息后，点击其中的“确认报名”按钮，该队的报名信息才会被最终存入系统。相较于传统方法，该方案更为高效严谨。

表 7 DB Server 中的运动员报名信息表

数据库: Register.db 数据表名: students 中文描述: 报名信息					
字段名称	字段描述	数据类型	长度	非空	备注
Name	姓名	Varchar	10	√	
Gender	性别	Varchar	4	√	
Number	学生证号	varchar	20	√	
IDcode	身份证号	Varchar	18	√	
Tel	联系电话	Varchar	11	√	
Project	项目（反曲、传统）	Varchar	10	√	
Groups	组别（大学、中学、小学）	Varchar	10	√	
Type	参与方式（个人、团体、混团）	Integers	1	√	
School	学校	Varchar	20	√	

表 8 是客户端数据库中存储的运动员基本信息表。该数据表中记录了所有运动员的属性信息，并且记录了由系统在编排排位赛靶位时自动生成的唯一运动员识别码（ID）。ID 中包含了学校信息、运动员性别以及对应的编号。ID 字段解决了在大规模赛事中偶尔发生的运动员重名现象导致的违反数据库记录“唯一性”的条件。因此，ID 字段作为“生成成绩册”等跨表查询功能最为重要的参数，被记录在了所有客户端数据库的基本信息表中。

表 8 客户端中的运动员基本信息表

数据库: Archery.db 数据表名: athlete_Info 中文描述: 运动员信息表					
字段名称	字段描述	数据类型	长度	非空	备注
ID	运动员识别码	Integer	4	√	主键
TargetNo	排位赛靶位号	Varchar	3	√	
Name	姓名	varchar	10	√	
School	单位	varchar	20	√	
Gender	性别	Varchar	4	√	
Rank	排位赛排名	Integer	3		
P_128_Rank	个人 256 进 128 淘汰赛靶位号	varchar	3		
..... 部分赛程表从略					
P_half_Rank	个人半决赛靶位号	Varchar	3		
P_final_Rank	个人决赛靶位号	Varchar	3		
G_64_Rank	团体 128 进 64 淘汰赛靶位号	varchar	3		
..... 部分赛程表从略					
G_half_Rank	团体半决赛靶位号	Varchar	3		
G_final_Rank	团体决赛靶位号	Varchar	3		
M_64_Rank	混团 128 进 64 淘汰赛靶位号	Varchar	3		
..... 部分赛程表从略					
M_half_Rank	混团半决赛靶位号	Varchar	3		
M_final_Rank	混团决赛靶位号	Varchar	3		

2) 文件存储

在本系统中，主要使用了两种文件格式来存储数据，分别为报名所需的证件照片以图片格式（jpg、bmp、png）等格式存储，及生成的赛事靶位编排文件或者赛事结果报告文件以 Excel 文件的形式存储。在本系统中，主要使用了两种文件格式来存储数据，分别为报名所需的证件照片以图片格式（jpg、bmp、png）等格式存储，及生成的赛事靶位编排文件或者赛事结果报告文件以 Excel 文件的形式存储。证件照片存储在服务器上，路径为

“bin/web_server/db/Archery/”文件夹下，在该路径下分布着以学校名命名的众多子文件夹，每一个子文件夹中以报名运动员的姓名命名了该照片。Excel 文件存储在运行目录

“TAEMS/FILES/SCHEDULE”和“TAEMS/FILES/SCORE”两个文件夹，命名的规范均为中文报表名称，如“男子个人半决赛靶位安排.xls”或“个人排名赛总成绩表.xls”。

3) 加密存储

系统在每次关闭时，会将最新的 SQLite3 数据库文件以加密的方式存储至本地，其本质仍是数据库结构，但以二进制按照一定密钥进行凯撒加密。这种数据加密存储的方式使得该 SQLite3 数据库不会被外部的数据库软件或他人恶意连接。

3.2.2 临时数据

临时数据格式被定义为系统间或者模块间用来交换信息用的数据格式。在本系统中，临时数据主要用于两个途径：客户端上的分布式存储以及赛事信息从客户端到云端的发布过程。

1) 分布式传输的临时数据

在将数据分布式分发的过程中，数据持有者会将数据的明文写入 txt 文件。将上述数据明文进行哈希加密，并将上述哈希值拼接一个密钥，对于新的字符串再次计算哈希值。最终会在上述 txt 文件的末尾记录上第二轮加密的哈希值。用户可以通过基于 http 协议的局域网传输协议或者直接通过 U 盘等硬件设备传输该数据文件。接收方在读取到该数据文件后，会先校验该哈希值是否正确，然后再决定是否加载该数据到本地数据库中。

2) 在线服务同步的临时数据

在将数据从本地客户端上传至数据库服务器的过程中，通过 Python 自带的 Pickle 方式，将缓存中的数据以二进制的形式载入本地文件。将此文件以 http 协议的 post 方式上传至数据库服务器后，再次使用 Pickle 加载该数据至缓存，完成数据上传过程。

3.2.3 缓存数据结构

缓存数据结构被定义为存储与内存中用于临时数据分析或处理的数据格式。在本系统中，缓存数据主要用于报表的编制过程。生成报表过程的数据常常要使用多张数据表，同时还需要辅以复杂的逻辑关系。传统的 SQL 语句完成上述问题较为复杂，不利于代码的可读性的维护，效率也较为低下，故我们的缓存数据使用 Python 中的数据处理框架 DaPy 来完成。

我们使用 DaPy 的 DataSet 类构建的实例来存储整个数据集，使用 SeriesSet 类来存储每一张关系型数据表。Read 函数加载整个数据集至缓存，select 函数筛选符合要求的记录，append_col 函数生成报表中的每一列代码，merge 函数和 join 函数来拼接多张表，sort 函数来对数据集中的记录进行排序，最后使用 save 函数将结果保存成 Excel 文件供用户使用。

3.3 客户端系统界面设计



图 7 “赛事管家”客户端

3.3.1 界面设计风格

1) 界面简洁

作为一套实用型赛事系统，客户端主界面的以简洁明了的白底黑字作为主基调，同时，各项功能的操作按钮按照赛事推进的逻辑顺序依次排列，确保“编排组”裁判人员可以方便快捷地进行功能操作，减少记忆负担。

2) 视觉设计

相应的图形符号被置于各项功能的按键左侧以帮助用户记忆。如：数据管理所对应的图标是用户常用的数据库 data 所常用的代表性图形，方便用户识别。

3) 宣传图

为了宣传射艺文化，给用户带来更强的代入感，射艺运动员赛时的精神风貌也以图片形式呈现在了页面上。而为了减少用户的审美疲劳，增添页面的趣味性，页面上的照片将进行切换，每张照片在页面的停留时间为 2-3 分钟，并且图片素材也会根据时间的推移不断进行更新。

3.3.2 客户端主要功能页面

“赛事管家”的本地客户端面向裁判员及赛事管理员，其一共具备四项主要功能：赛事编排、成绩录入、数据分布式备份以及成绩报表生成。

功能页面依旧与主页面一样，以极简的黑白色调为主，将对应的主要功能名称置于页面左侧，并以大号字体便于用户识别相应功能。而每项主要功能对应的每步操作则按照比赛的推进流程依次排列于页面的右侧，可进行的操作将以黑色字体显示，而因比赛流程无法进行的操作则以灰色字体显示且无法点击。

1) 赛事编排



图 8 赛事管家-赛事编排

Pro 版客户端从云端服务器的数据库中下载好运动员基本数据后将其保存在本地。

用户可点击“编排排位赛靶位”按钮，系统会引导你导入下载好的运动员基本信息，设定起始靶位号及不同性别、学校最多运动员数量，点击“编排”完成首轮排位赛靶位安排，相应运动员靶位编排结果可通过文件夹（Files）中自动生成的 Excel 格式报表进行查看。

在编排首轮个人淘汰赛靶位前，Pro 版客户端需将信息通过 LAN 加密（终端管理的导出导入功能）传输给 Lite 版客户端。在完成排位赛成绩录入后，即可点击“编排首轮个人\团体淘汰赛靶位”按钮，完成下一轮淘汰赛的靶位编排。如果不同性别组的赛程进度不同，管理员可以先勾选其中一个性别编排。

裁判员需要完成每一轮每一位运动员的成绩录入后，才能够进行下一轮的靶位安排。如果出现该名运动员弃赛的情况的，裁判员需要在成绩录入中为该名运动员填入“0”的成绩。

2) 成绩录入



图 9 赛事管家-成绩录入



图 10 成绩录入-个人排名赛

“成绩录入”功能的主界面以“排位赛”与“淘汰赛”的分数录入为划分。作为所有选手的选拔赛，排位赛上方的图标是以记分板与靶子相结合的形式出现，其中，记分板象征分数录入功能，而靶子代表射艺比赛；淘汰赛是经过排位赛选拔后脱颖而出的运动员们相互竞争直至选出冠军的层层比赛，以“VS”作为图标正体现出了射艺高手们互相切磋的激烈之感。

3) 录入排位赛成绩

排位赛的成绩录入界面分为左右部分，左边用于用户信息的设定和查询，右侧用于成绩的录入。每输入一组记录，都先在左侧的信息栏中填入运动员信息，而后点击“查询信息”按钮。运动员信息可以只输入其中一项，如果所有信息自动补全，则代表系统探测到了该运动员数据，右侧“提交记录”按钮将会点亮。



图 11 成绩录入-个人积分制淘汰赛



图 12 成绩录入-个人局胜制淘汰赛

4) 录入淘汰赛成绩

淘汰赛的模式分为局胜制和积分制，“1/4 赛程”后，系统会切入局胜制界面。
淘汰赛赛程界面分为上部的“运动员/运动队信息”和下半部分的“比赛记录”两部分。每录入一组数据都需要填入至少一个运动员/运动队信息，而后点击右侧或者下方的“查询信息”按钮。
如果未识别到该运动员/运动队信息或着当前运动员/运动队已录入过本轮比赛的成绩，系统会弹出对话框进行提示。

点击“提交记录”按钮后界面刷新说明记录被成功写入。

5) 数据分布式存储



图 2 赛事管家-数据库管理



图 3 赛事管家-终端管理

数据分布式备份功能主要以“数据库管理”和“终端管理”两个界面加以使用。“数据库管理”界面主要用于 Pro 与 Lite 客户端自身的数据库备份与重置，而“终端管理”则是用于 Pro 与 Lite 之间的分布式备份。为了便于用户识别区分，“导入数据”与“导出数据”功能图标分别为以不同方向的箭头为导向。

“数据库管理”界面主要负责提高数据的本地容灾备份能力。“备份数据库”：将当前活跃中的数据库进行凯撒加密后备份至另一文件。“重置数据库”：清空整个活跃中的运动员信息数据库。“历史数据”：恢复数据库至最近一次备份后的数据库状态或者本次启动前的状态。



图 13 终端管理-数据导入

图 14 终端管理-数据导出



图 15 终端管理-联机工作

“终端管理”界面主要用于控制各客户端之间的数据传输功能。图 10 所示的“导入数据”：选中一个外部文件后先对外部文件中的数据内容进行格式确认，紧接着进行使用密钥进行哈希验证确保数据未被修改。通过上述何效验后，系统将根据当前数据库是否已包含该数据文件，提供“导入”、“覆盖”和“合并”功能。图 11 所示的“导出数据”：启动页面时加载当前数据库中表列表，选择对应赛程后，将数据哈希加密并导出成.bat 文件并保存于“ShareData”文件夹下。用户可以使用 U 盘或者联机工作的方式同步该数据文件至其他终端。图 12 的“联机工作”：具有接收数据、分享数据两个核心功能，用户不用设定防火墙，只需要录入相应的 IP 地址，即可发送赛事文件。分享的文件为实时数据表中数据。

6) 成绩报表导出



图 16 赛事管家-记录管理

图 17 赛事管家-成绩云端同步

成绩报表生成功能通过“记录管理”界面实现。点击“记录导出”按键即可导出排位赛成绩总表。导出的名次将按照总分、总“正”、总“的”的顺序降次排序，里面详细记录了排位赛中每一箭的记录，导出的成绩表将保存在“Files/SCORES”文件夹下。同时，用户还能将数据上传至云端 DB 服务器，以及在整场比赛结束后迅速导出整份成绩册。

3.4 网页前端界面设计

3.4.1 界面设计风格



图 18 “赛事管家”网页端

“赛事系统”的网页端页面主要服务于参加射艺锦标赛的运动员。“赛事管家”的网页端（面向运动员）一共具备三项主要功能：赛事公示、注册报名与成绩查询。

主页面以大海与天空接壤之景为背景，代表了射艺这项中国传统运动远大的发展图景。为了清晰的将网页信息展现给用户，背景的亮度加以调低，以蓝灰色为主色，并统一将白色作为文字颜色。为了简化运动员赛事报名与成绩查询等功能的使用，相应功能按键直接以白框蓝字的形式呈现在页面上，方便运动员直接进行操作，如：“现在报名”、“在线查询”等。同时，为了若用户有意向使用 Kitgram 项目组的其他软件，也可点击页面右上角的图标进行用户注册以获取更多信息。

3.4.2 网页端主要功能界面

1) 赛事公示

如图 5 所示，相关赛事信息如：比赛时间，比赛地点、赛事规程等都以醒目的白色字体直接展示在“赛事管家”网页端主界面上。具体赛事规程可通过点击“下载”按钮进行查看。在每场次的比赛报名结束后，相应的秩序册下载按钮可以使用，各个比赛队可以按照秩序册的信息参与比赛。

2) 注册报名

运动员可通过点击网页端主界面的“现在报名”按钮进行射艺锦标赛的注册报名。为保

持网页系统的整体一致性，报名页面延用了主页面的背景。

[illegible]

图 19 报名页面

2018中国大学生射箭(射艺)锦标赛-报名确认函

我

我 2018-10-25 16:15
发至 834157651

🔍

🔄

📄

☰

详情

尊敬的**西北政法大学**代表队：

你们好！

自2015年开始，中国大学生射箭（射艺）赛赛事渐开先河，并逐渐成长为中国大学生射箭（射艺）锦标赛。2017年，我们迎来了52所高校代表队的291名顶尖运动员参加的冠军赛，共决出16个项目的冠军。在今年初夏之际，1055名学生运动员及184名教练员和工作人员参加了本届锦标赛的预选赛。本次锦标赛将继续践行“射以观德”的精神，促进地区射艺文化相互交流，同时为各位选手提供一个检验训练成果、提升技术素养的平台。

世界锦标赛将于11月9日至11月11日在上海对外经贸大学（松江校区）举办，你们的到来必定能让比赛增添许多，预祝你们能在比赛中取得满意的成绩。

我们已经收到贵队代表的报名信息，其中包含有**22名教练**及**9名运动员**信息，详情请阅读下面的报名表。如单上记录信息与输入的报名信息不符，请单击：[链接提示](#)。若缺少某些报名信息可能是由于未上传证件照片或运动员信息未填写完全导致收到您的报名信息后，后台工作人员将核实身份信息。如果有不符合的情况，我们将与您取得联系！我们将在所有运动项目赛前审核完毕，向领队人邮箱发送《报名信息通知》或《比赛查询通知》，请及时查收。

如果这不是您的操作，请忽略本邮件，抱歉。

顺颂秋安

上海对外经贸大学 观塘射艺队
2018年10月25日

附表1 - 教练报名表

姓名	邮箱	电话
领队 王华	834157651@qq.com	18706857913
教练 丁雨	834157651@qq.com	18706857913
教练 赵晨	834157651@qq.com	18706857913

附表2 - 运动员报名表

姓名	性别	项目	组别
运动员 陈浩	男	传统弓能大项组	大学组
运动员 李亚东	男	传统弓能大项组	大学组
运动员 冯佳乐	男	传统弓能大项组	大学组
运动员 赵梦婷	女	传统弓能大项组	大学组
运动员 李子豪	男	传统弓能大项组	大学组
运动员 李金雷	女	传统弓能大项组	大学组
运动员 戴林文	男	反曲弓能大项组	大学组
运动员 杨子豪	男	反曲弓能大项组	大学组
运动员 金源	男	反曲弓能大项组	大学组

确认报名

图 20 报名确认函

在报名过程中，除了运动员姓名，性别，学号，学校等相关信息的填写，运动员还需上传个人证件照片（学生证、身份证）用于后续的后台工作人员对运动员资格的审核。填写完毕报名信息后便可点击“提交申请”按钮，将运动员数据传至后台。再提交结束后为了确认这次报名的邮件地址合法，如图 13 所示的“报名确认函”将以即刻邮件形式发至各领队邮箱。运动员与教练员可再次审核报名信息进行确认，邮件中的重要信息用以加粗字体与附表的形式以使用户更快确认。如果邮件内容无误，待点击确认报名连接后，他们的比赛报名信息将被录入进数据库。

3) 实时成绩查询

2018中国大学生射箭（射艺）锦标赛-实时成绩查询

大学乙组女子个人排名赛	大学乙组男子个人排名赛	大学乙组女子个人32进16淘汰赛	大学乙组男子个人32进16淘汰赛	大学乙组女子个人16进8淘汰赛
大学乙组男子个人8进4淘汰赛	大学乙组女子个人8进4淘汰赛	大学甲组男子个人排名赛	大学甲组女子个人排名赛	大学甲组女子个人64进32淘汰赛
大学甲组男子个人64进32淘汰赛	大学甲组男子团体16进8淘汰赛	大学甲组女子团体16进8淘汰赛	大学甲组混合团体16进8淘汰赛	大学甲组男子个人32进16淘汰赛
大学甲组女子个人32进16淘汰赛	大学甲组女子个人16进8淘汰赛	大学甲组男子个人16进8淘汰赛	大学甲组男子个人8进4淘汰赛	大学甲组女子个人8进4淘汰赛

编号	姓名	的	正	鹄	侯	流失	总分	结果
1110	阿斯根	1	5	5	1	0	46	1
1310	孙凯强	1	7	2	2	0	48	1
1411	童良晶	0	3	7	2	0	38	1
1611	王雨	0	6	6	0	0	48	0
2010	阿斯尔	1	5	6	0	0	48	1
2012	王硕	0	5	6	1	0	44	0
2110	陈云迁	0	3	4	4	1	31	0

图 21 成绩查询

在比赛过程中,后台人员通过“赛事管家”客户端将运动员成绩实时上传至云端。运动员从“赛事系统”网页版主页,点击“在线查询”按钮进入成绩查询页面。为了更为清晰合理的为运

动员提供成绩查询，该功能页面可选定不同比赛场次、比赛组别进行查询，查询结果包括不同距离总分与全距离总分。同时，通过点击蓝色字段名，系统将按照各字段名进行排序。运动员可查询到自己的实时排名，并更好的调节自身的赛时状态以更好地迎接下一场次的比赛。

比赛结束后，上传比赛成绩册。根据比赛弓种（传统弓与反曲弓）以及不同比赛单位（大学、中学、小学），秩序册将分为大学组、中小学组、反曲组三项以便查询，用户只需根据自己所在比赛单位及弓种下载对应秩序册，即可查看各学校、个人及团体的全距离排位表，各距离成绩一览表以及比赛对阵图。

4 系统安装及使用说明

1) 安装说明

- a) 请确保您的计算机使用 Windows 64 位操作系统;
- b) 直接运行程序, 初次授权验证后会引导您注册管理员用户;
- c) 安装完成, 祝您使用愉快!

2) Pro 版客户端授权申请

- a) 通过运行绑定于指定 U 盘的“授权颁布系统”申请授权码;
- b) 通过联系开发者邮箱购买授权码。

1) 在线注册报名（面向运动员）

在开赛前, 参赛选手与教练员需要在赛事管家官方主页选择相应的赛事进行注册报名。验证报名邮箱后, 运动员的注册信息即储存进云端的数据库中, 以便后续登陆查询成绩操作。

2) 成绩录入（面向裁判员）

在“赛事管家”系统的 Pro 版和 Lite 版桌面客户端中, 裁判员可以进行比赛成绩的实时录入。

- i. 成绩录入界面分为左右部分, 左边用于用户信息的设定和查询, 右侧用于成绩的录入。每输入一组记录, 都先在左侧的信息栏中填入运动员信息, 而后点击“查询信息”按钮。运动员信息可以只输入其中一项, 如果所有信息自动补全, 则代表系统探测到了该运动员数据, 右侧“提交记录”按钮将会点亮。
- ii. 裁判员可以一次只输入一个距离 (10 米靶、20 米靶等) 的比赛记录并提交, 当裁判员再次查询该运动员信息时, 客户端会自动显示已存在的数据记录。
- iii. 点击“提交记录”, 若界面成功刷新则表示数据已经被写入。(Lite 版客户端将数据通过 LAN 加密传输给 Pro, Pro 版客户端将数据储存在本地。)

3) 赛事编排（面向赛事管理员）

a) 自动编排排位赛靶位安排表

- i. Pro 版客户端从云端服务器的数据库中下载一份保存着运动员姓名、性别、俱乐部名称的基本数据, 并将其存在本地。
- ii. 选择数据文件, 设定起始靶位号及单性别最多运动员数量, 选择“编排”。
- iii. 系统随机打乱俱乐部次序后, 按照“单靶位无同俱乐部成员”、“同俱乐部成员就近安排”、“人数不足的俱乐部在队列末尾”这三个原则自动完成赛事编排。
- iv. 完成编排后, 系统会将信息记录保存一份在本地数据库中。与此同时, Pro 版客户端将信息通过 LAN 加密传输给 Lite 版客户端。

b) 编排首轮(个人&团体)淘汰赛靶位编排

完成排位赛成绩录入后, “赛事管家”可以短时间内完成淘汰赛靶位编排。

- i. 管理员需要设置首轮淘汰赛的赛程进度、起始靶位号及单靶位运动员数量。
- ii. 如果不同性别组的赛程进度不同, 管理员可以先勾选其中一个性别编排。
- iii. 确认操作后, 系统会自动地在本地数据库中建立选择赛程的数据文件, 此时裁判员可以开始进行淘汰赛成绩录入。
- iv. 完成编排后, Pro 版客户端会将信息记录保存一份在本地数据库中。与此同时, Pro 版客户端将信息通过 LAN 加密传输给 Lite 版客户端。

c) 编排首轮(个人&团体)淘汰赛靶位编排

完成了首轮淘汰赛的成绩录入后，管理员需要进行逐场的淘汰赛编排。

i. 相较于首轮淘汰赛靶位安排，系统会自动识别各个性别组正在进行的赛程以及下一赛程。裁判员需要完成每一轮每一位运动员的成绩录入后，才能够进行下一轮的靶位安排。如果出现该名运动员弃赛的情况的，裁判员需要在成绩录入中为该 名运动员填入“0”的成绩。

ii. 完成编排后，Pro 版客户端会将信息记录保存一份在本地数据库中。与此同时，Pro 版客户端将信息通过 LAN 加密传输给 Lite 版客户端。

4) 成绩查询（面向运动员）

在每一场次比赛结束后，Pro 客户端将储存在本地的所有比赛数据通过 WAN 加密传输至云端。云端后台处理后渲染至 Web 前端，此时运动员可以登陆查看自己的成绩排名和下一个比赛日的靶位安排。

5 总结

传统射艺作为一项竞技体育运动，需要通过举办高质量的赛事来推动其更好的传播和发扬。CBA 主席姚明也在采访中曾表示，“运动员水平是需要大量的比赛才能提高的”。所以无论是为了推广这项运动，还是提高运动员的专业技能，一场高质量的比赛都是至关重要的。

然而，想要推广传统弓赛事更大规模的进行，必须要有强大的后勤管理能力。传统“纸笔记录，人工统计”的枯燥工作使得编排组、裁判员职位总是让人敬而远之，赛事的推广和发展也因此被遏制。

令人高兴的是，我们看到了一些商业公司基于移动端平台提供了他们的解决方案。这些解决方案的出现虽然在一定程度上带动了传统射艺比赛向着更远更深的方向发展。但是由于技术方案本身的局限性，市面上现存的平台存在着稳定性较低和数据严谨性不强的缺点。

我们认为，新的系统主要需要关注稳定性、安全性及高效性三个方面，同时提供一种与赛事参与者实时交互的新渠道。因此，我们团队希望开发一套基于笔记本桌面端平台，同时支持拓展网页服务的赛事管理系统以提供自己的解决方案。

新的解决方案的设计思想思想被我们成为“B/S + C/S 混合部署架构”。从最基本的单节点模式开始，一台搭载 Pro 的个人电脑，便可以胜任一个中小型规模的赛事编排组工作。为了提高效率，赛事管家增加了 Lite 版，并构造“单 Pro 多 Lite”的 C/S 架构群。由于大部分的比赛场地网络信号较差且通讯设备部署成本高，我们考虑使用局域网通讯；当网络环境临时中断时，系统也支持使用 U 盘进行数据同步。在交互渠道方面，系统通过额外部署的 DB 服务器和 Web 服务器形成的 B/S 架构群拉近了赛事管理者与赛事参与者之间的距离。在广域网连接状况较好时，C/S 架构群中的核心管理 Pro 可以与 B/S 架构群中的 DB 服务器进行数据同步，让运动员可以实时查询到最新的比赛数据。

在未来，我们团队考虑将 OCR 功能加入系统，能支持识别人工手动填写的成绩单。此外，我们还打算拓展支持比赛的种类，将建立在“B/S + C/S 混合部署架构”的“传统弓赛事管家”推广到“反曲弓赛事管家”、“高尔夫球赛事管家”，甚至“帆船赛事管家”等各种户外项目中。

6 附录

6.1 名词定义

表 9 名词含义说明

名词/缩写	说明
B/S	Browser/Server（客户机/服务器）
C/S	Client/Server（客户机/服务器）
Pro	Professional（专业版客户端）
Lite	Lite（轻量版客户端）

6.2 参考资料

- [1] 计算机软件产品开发文件编制指南. 中华人民共和国国家标准 GB8567-88. 国家标准局, 1988 年 1 月 7 日.
- [2] 谢希仁. 计算机网络（第四版）. 大连理工大学出版社, 2006.8.
- [3] 严霄凤, 高炽扬. 美国联邦信息安全风险管理框架及其相关标准研究. 信息安全与通信保密, 2(2009): 40-44.
- [4] Salakhutdinov & Geoff Hinton, Training a deep autoencoder or a classifier on MNIST digits, 2006.

6.3 源代码清单

TAEMS	/client_pro	/Source/
		/Launcher.py
		/accredit_function.pyc
		/desktop.pyc
		/global_functions.pyc
		/global_variables.pyc
		/Installer.pyc
		/L3_Dialog.pyc
		/L4_Dialog.pyc
		/threads.pyc
		/filePrinter.pyc
		/upload.pyc
	/server_db	/database/
		/flask_sqlite_web/
		/app.py
		/db_init.pyc
		/db_server.pyc
		/requirements.txt
		/README.md
	/server_web	/db
		/static
		/templates
		/temporary
		/tk
		/web_server.py