

同濟大學

毕业设计（论文）

同濟大學

TONGJI UNIVERSITY

本科毕业设计(论文)

课 题 名 称 基于 LeanCloud 的运动约伴网站的设计与实现

副 标 题

学院（系）

软件学院

专业

软件工程

学号

1252871

学生姓名

郑天齐

指导教师

唐剑锋

日期

2016-05-30

基于 LeanCloud 的运动约伴网站的设计与实现

摘要

当今大学生体质健康状况随着 2015 年的体质健康报告的出炉愈发引起人们的关注，本网站设计针对现下大学生运动少，缺乏运动伙伴，且偏向于使用在线网站或应用获取信息的习惯，设计并实现了这一集成个人信息，照片展示，好友添加，运动约伴，在线聊天功能的在线运动约伴的平台。本文主要介绍了该系统的设计及实现过程。文章开头首先介绍了课题的背景、意义以及在国内外的发展趋势，提出了设计中的主要研究内容。其次介绍了项目实现过程中所使用的软件设计架构以及所使用的各项软硬件技术。接着研究了系统的总体目标以及对系统进行了整体的设计，对各功能模块进行了划分。最后一部分则详细介绍了系统各个模块的实现过程，包括功能描述、使用情况说明、页面展示以及关键算法的流程。

关键词：运动，HTML5，JavaScript，bootstrap，LeanCloud，API

The design and implementation of sports engagement website based on LeanCloud

ABSTRACT

After the publication of the Report of Body Building & Health Status of College Students, college students' health status has become a public concern. This design focus on this issue and the habit of students to get information from web, implemented this website contains personal information, albums, friend engagement, sport engagement, online chat function. At first the article illustrates the background of this graduate design, together with its significance and development tendency inside-out China, put forward the main topic of design. Second, the article introduces the framework, technologies used in development. Third, the article introduces the main target of the design, the global design of the project and partition of functional modules. The last part of the article introduces the implementation of each module in detail.

Key words: sports, HTML, JavaScript, bootstrap, LeanCloud, API

装

订

线

目 录

目 录	3
1 引 言	6
1.1 课题背景	6
1.1.1 现实需求	6
1.1.2 技术背景	7
1.2 课题的研究意义	7
1.3 课题的研究内容	7
1.4 文章结构	7
2 相关技术	9
2.1 JavaScript	9
2.2 HTML	9
2.3 LeanCloud 云引擎及 API	9
2.4 LeanCloud	10
2.5 开发环境	10
2.6 本章小结	11
3 需求分析	12
3.1 功能性需求分析	12
3.1.1 登录管理	12
3.1.2 用户信息管理	12
3.1.3 相册管理	13
3.1.4 好友管理	13
3.1.5 约伴管理	13
3.1.6 在线聊天	13
3.2 用例分析	13
3.2.1 用户注册	13
3.2.2 用户信息管理	14
3.2.3 好友管理	15
3.2.4 约伴管理	15
3.2.5 请求处理	15
3.2.6 数据库管理	16
3.3 非功能性需求分析	16
3.3.1 系统可行性	17
3.3.2 性能需求	17
3.3.3 安全性需求	17
3.3.4 可维护性	17
3.3.5 可扩展性	17
3.4 本章小结	18
4 网站设计	19
4.1 网站总体架构	19
4.2 功能模块设计	19
4.3.1 用户登录注册模块	19
4.3.2 用户信息模块	20
4.3.3 相册模块	21
4.3.4 好友模块	21
4.3.5 约伴模块	22

4.3.6 在线聊天模块.....	23
4.3 前端设计.....	23
4.4.1 设计原则.....	23
4.4.2 主要页面.....	23
4.4 数据库设计.....	24
4.4.1 用户信息表设计.....	24
4.4.2 好友申请信息表设计.....	25
4.4.3 学校及校区信息表设计.....	25
4.4.4 运动信息表设计.....	25
4.4.5 标签信息表设计.....	26
4.4.6 约伴信息表设计.....	26
4.4.7 文件信息表设计.....	26
4.4.8 会话信息表设计.....	26
4.4.9 E-R 图.....	27
4.5 本章小结.....	29
5 系统实现.....	30
5.1 数据库建立.....	30
5.2 用户登录注册模块实现.....	30
5.2.1 功能概述.....	30
5.2.2 部分代码.....	30
5.2.3 界面.....	31
5.2.4 前端部分代码.....	32
5.3 用户信息模块实现.....	33
5.3.1 功能概述.....	33
5.3.2 部分代码.....	34
5.3.3 界面.....	35
5.3.4 前端部分代码.....	36
5.4 相册模块实现.....	37
5.4.1 功能概述.....	37
5.4.2 部分代码.....	37
5.4.3 界面.....	38
5.4.4 前端部分代码.....	39
5.5 好友模块实现.....	40
5.5.1 功能概述.....	40
5.5.2 部分代码.....	40
5.5.3 界面.....	41
5.5.4 前端部分代码.....	42
5.6 约伴模块实现.....	43
5.6.1 功能概述.....	43
5.6.2 部分代码.....	43
5.6.3 界面.....	44
5.6.4 前端部分代码.....	45
5.7 在线聊天模块实现.....	45
5.7.1 功能概述.....	45
5.7.2 部分代码.....	45
5.7.3 界面.....	47
5.7.4 前端部分代码.....	48
5.8 本章小结.....	48

6 系统测试.....	49
6.1 登录注册模块测试.....	49
6.2 用户信息模块测试.....	52
6.3 相册模块测试.....	53
6.4 好友模块测试.....	53
6.5 约伴模块测试.....	55
6.6 在线聊天模块测试.....	56
6.7 本章小结.....	57
7 结论与展望.....	58
7.1 结论.....	58
7.2 展望.....	59
参考文献.....	60
谢 辞.....	61

装

订

线

1 引言

1.1 课题背景

1.1.1 现实需求

2011年9月2日教育部、国家体育总局联合国家六部委共同颁布的《2010年全国学生体质与健康调研结果》显示，中小学生体质状况在出现好转，但是大学生人群的身体素质仍旧在下滑。高校体育作为大学生学校运动的最后一站，是学校体育向社会体育过度的关键阶段，在大学时期养成良好的体育锻炼习惯有助于学生将来进入社会后坚持体育锻炼。现在，大学生课业压力减小，体育锻炼条件提高，但是大学生的体育锻炼现状依旧不乐观。大学生体育锻炼并不是局限在体育设施方面的原因，更多的还是缺乏恒心与动力。

《大学生运动承诺与锻炼坚持的关系研究》（以下简称《研究》）一文中把大学生的体育锻炼行为看成一个动态的过程，并按照跨理论模型中的“变化阶段”的划分，把大学生的体育锻炼参与现状分为5个阶段：准备阶段、意向阶段、前意向阶段、行动阶段、维持获得阶段：

表 1.1 行为变化各阶段及其定义

变化阶段	定义
前意向阶段	目前没有进行有规律的体育锻炼，在今后的6个月内也不打算开始
意向阶段	目前没有进行有规律的体育锻炼，但在以后的6个月内开始
准备阶段	目前有时断链，但没有形成规律
行动阶段	目前正在进行有规律的体育锻炼，但仅仅是在近6个月内开始的
维持获得阶段	目前正在进行有规律的体育锻炼，并以持续了6个月以上

《研究》指出大学生参与体育锻炼在行为阶段上的人数分布比例为准备阶段46.8%、意向阶段18.58%、前意向阶段12.30%、行动阶段13.27%、维持获得阶段9.05%。这个数据表明偶尔参与体育锻炼的人最多，能够坚持体育锻炼的人最少。大学生体育锻炼的减少，造成的影响可想而知，那就是大学生的身体素质下降。2014年，国家体育总局、教育部、科技部、国家民委、民政部、财政部、农业部、卫生计生委、国家统计局和全国总工会等10个部门，联合在全国31个省（区、市）进行了第4次国民体质监测工作。国家体育总局发布了《2014年国民体质检测公报》，检测结果显示，2014年全国达到《国民体质测定标准》“合格”等级以上的人数百分比为89.6%，与2010年相比增长了0.7个百分点，但大学生的身体素质仍继续呈现下降趋势。

1.1.2 技术背景

基于计算机网络的各种网络应用系统通过网络中对数字信息的综合采集、存储、传输、处理和利用而在全球范围把人类社会更紧密地联系起来，并以不可抗拒之势影响和冲击着人类社会政治、经济、军事和日常工作、生活的各个方面。因此，计算机网络将注定成为 21 世纪全球信息社会最重要的基础设施。

同时，JavaScript 技术以及 HTML 技术的成熟使得轻量化的网站开发成为可能，兼之 LeanCloud 的云引擎通过提供大量简单易用的 API 及相应的云数据库服务为网站开发提供了极其便利的平台，简化了开发过程中对后台数据的处理难度。

由此，使用 LeanCloud 云引擎结合 HTML 与 JavaScript 技术的网站开发成为一个易行的方案并具有极高的实践价值。

1.2 课题的研究意义

2015 年的国家大学生体质健康报告（《中国青少年体育发展报告（2015）》）中明确指出当今大学生的体育锻炼，体质健康情况及身体发展状况不容乐观，这与大学生在校园里缺少运动的积极性，伙伴及能够便利的帮助他们找到合适的伙伴的方式有着极大的关系，《研究》中指出大学生趋向于准备开始运动阶段的人数超过 40%。因此，本设计着眼于建立一个便于使用的平台帮助大学生寻找有共同兴趣爱好的伙伴一起运动，同时通过提供的个人信息展示，兴趣爱好标签及相册功能展示自我，寻找志同道合的运动伙伴一起健身锻炼，增强身体素质。

1.3 课题的研究内容

LeanCloud 作为一个新兴的云服务提供者，其云引擎及相应的 API 相对来说比较成熟，但是依旧存在在调用过程中的各类问题需要详细查阅技术文档来处理，同时项目在云引擎上的部署及调试也是文档中需要仔细阅读的部分并辅以实际行动。

本网站用 JavaScript 脚本语言处理前端网页获取的操作请求并与引擎的 API 进行交互。API 作为新兴网络服务的一个类型，越来越多的网站和服务提供商开始为开发者们提供开源的 API 或闭源的 API 接口，其便利的功能为广大开发者所接受，利用 API 能够大大简化后台的数据处理难度，大量的演算被放置在服务商提供的服务器上进行，既能够有效缓解后台的压力，也能够为用户提供更加快捷的交互体验。学习和熟练掌握并使用这类便利的开发工具是一个软件开发者很有必要的能力。

本网站使用 HTML 超文本标记语言作为前端开发的语言，HTML 以其强大的扩展性以及同 CSS 文件结合后优秀的前端 UI 编辑能力为广大开发者所喜爱，同时利用 JS 脚本语言开发出的各类动画、动态效果也能够使得前端网页更加美观，提高用户的操作流畅性和用户体验。网站的前端开发离不开 HTML，是一个很重要的学习研究内容。

1.4 文章结构

本文共分为七章，各章内容如下：

第一章为绪论，阐述了本课题的研究背景，运动社交类网站的需求及技术背景，还有课题的意义和内容，以及所需学习和运用的技术。

第二章具体介绍了设计本运动社交网站的相关技术和开发环境，其中以 JavaScript 技术为主。

第三章为本设计的需求分析，包括功能性分析和非功能性分析，并给出了用例图及流程图。

第四章为本运动约伴网站的详细设计，包括页面逻辑，页面事物处理模块，网站主题，数据库建立。

第五章讲解了本设计的具体实现，介绍了相应的功能模块的实现以及相对应的部分代码，并展示出相应的界面。

第六章对实现的网站进行了一系列测试，并给出了具体的测试用例。

第七章对本文进行了总结，并且根据本设计的目标及现有功能提出进一步的改进方向及展望。

装

订

线

2 相关技术

2.1 JavaScript

JavaScript 一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。它的解释器被称为 JavaScript 引擎，为浏览器的一部分，广泛用于客户端的脚本语言，最早是在 HTML（标准通用标记语言下的一个应用）网页上使用，用来给 HTML 网页增加动态功能。

在 1995 年时，由 Netscape 公司的 Brendan Eich，在网景导航者浏览器上首次设计实现而成。因为 Netscape 与 Sun 合作，Netscape 管理层希望它外观看起来像 Java，因此取名为 JavaScript。

JavaScript 是一种属于网络的脚本语言，已经被广泛用于 Web 应用开发，常用来为网页添加各式各样的动态功能，为用户提供更流畅美观的浏览效果。通常 JavaScript 脚本是通过嵌入在 HTML 中来实现自身的功能的。

是一种解释性脚本语言（代码不进行预编译）。

主要用来向 HTML（标准通用标记语言下的一个应用）页面添加交互行为。

可以直接嵌入 HTML 页面，但写成单独的 JS 文件有利于结构和行为的分离。

跨平台特性，在绝大多数浏览器的支持下，可以在多种平台下运行（如 Windows、Linux、Mac、Android、iOS 等）。

JavaScript 脚本语言同其他语言一样，有它自身的基本数据类型，表达式和算术运算符及程序的基本程序框架。JavaScript 提供了四种基本的数据类型和两种特殊数据类型用来处理数据和文字。而变量提供存放信息的地方，表达式则可以完成较复杂的信息处理。

2.2 HTML

超文本标记语言，标准通用标记语言下的一个应用。

“超文本”就是指页面内可以包含图片、链接，甚至音乐、程序等非文字元素。

超文本标记语言的结构包括“头”部分（英语：Head）、和“主体”部分（英语：Body），其中“头”部提供关于网页的信息，“主体”部分提供网页的具体内容。

超级文本标记语言文档制作不是很复杂，但功能强大，支持不同数据格式的文件嵌入，这也是万维网（WWW）盛行的原因之一，其主要特点如下：

(1) 简易性：超级文本标记语言版本升级采用超集方式，从而更加灵活方便。

(2) 可扩展性：超级文本标记语言的广泛应用带来了加强功能，增加标识符等要求，超级文本标记语言采取子类元素的方式，为系统扩展带来保证。

(3) 平台无关性：虽然个人计算机大行其道，但使用 MAC 等其他机器的大有人在，超级文本标记语言可以使用在广泛的平台上，这也是万维网（WWW）盛行的另一个原因。

(4) 通用性：另外，HTML 是网络的通用语言，一种简单、通用的全置标记语言。它允许网页制作人建立文本与图片相结合的复杂页面，这些页面可以被网上任何其他人浏览到，无论使用的是什么类型的电脑或浏览器。

2.3 LeanCloud 云引擎及 API

云引擎（LeanEngine）是 LeanCloud 推出的服务端托管平台。提供了多种运行环境（Node.js，

Python 等）来运行服务端程序。你只需要提供提供服务端的业务逻辑（网站或云函数等），而服务端的多实例负载均衡，不中断服务的平滑升级等都由云引擎提供支持。

同时 LeanCloud 提供了一系列的在线存储及实时通信 API 方便开发者调用，在使用在线数据库的同时依托云引擎及其便利的 API 大大简化了开发工作，其 API 提供了便利的借口用于调用引擎提供的后台数据处理服务。

2.4 LeanCloud

LeanCloud 是一个在线云数据库服务提供商，本设计的云引擎来源于其服务，于 2014 年 1 月正式创立，其中数据库服务中，LeanCloud 对大部分场景下的后端需求进行了抽象和统一，我们通过四大系统来实现一个通用、强大、可定制的 BaaS(Backend as a Service) 服务：

（1）面向对象的海量数据库 前后端交互的主体，都是「数据」，不管结果多少，属性具体含义如何，它们都可以抽象成统一的「对象」来处理。LeanCloud 支持存储任意类型的对象，支持对象的增、删、改、查等多种操作，并且开发者无需担心数据规模的大小和访问流量的多少，可以简单将 LeanCloud 云端看成是一个面向对象的海量数据库来使用。

（2）大文件存储和分发 任何一款产品，不管是网站、应用还是游戏，都有一些素材或者文件需要存储和分发。与应用内数据不一样，这些文件因为它的体积较大，为了获得更快捷的用户体验，一般都还需要 CDN 服务。LeanCloud 存储系统完整涵盖了大文件存储和分发的需求。

（3）LeanEngine 完成特定业务逻辑 LeanCloud 提供的数据库操作 API 能覆盖大部分业务的需求，但是凡事总有例外，这些标准 API 有时候并不能完全满足某些特定需求，这时候怎么办？LeanCloud 还提供了「LeanEngine」自定义服务端业务逻辑的功能。LeanEngine 与大家熟知的 Google App Engine 相似，允许开发者写很少的一部分代码，来完成业务特有逻辑。这些代码会被部署到 LeanCloud 云端，与 LeanCloud 标准服务一起执行，来实现特殊需求。

（4）离线数据分析平台 对于完全构建在 LeanCloud 上的产品来讲，在运行一段时间之后会积累大量的业务数据，这时候产品和运营层面都会产生一些数据挖掘或商业智能分析的需求，此时如何才能简便地操作云端，看到数据背后隐藏的趋势和价值呢？为此我们推出了分布式的「离线数据分析系统」，支持在应用数据集上进行各种处理和操作。离线分析系统是完全的分布式、实时计算系统，其执行效率和处理的数据规模远在 Hadoop MapReduce 之上。

2.5 开发环境

表 2.1 开发环境一览表

参数类型	参数选择	使用原因
操作系统	Windows 10	大部分软件对 Windows 操作系统支持比较好，对加速开发有一定的益处。
运行环境	JDK 1.8	JDK 1.8 是相对较新的 JDK 版本，支持 JavaScript 开发。
数据库	LeanCloud	LeanCloud 是关系型在线云数据库，易于使用，易于部署的同时，性能也相当优秀。
引擎	LeanEngine	云引擎（LeanEngine）是 LeanCloud 推出的服务端托管平台。提供了多种运行环境（Node.js, Python 等）来运行服务端程序。你

		只需要提供提供服务端的业务逻辑（网站或云函数等），而服务端的多实例负载均衡，不中断服务的平滑升级等都由云引擎提供支持。
应用服务器	Tomcat 8	Tomcat 是目前比较流行的 Web 应用服务器，因其技术先进，性能稳定，并且是免费使用，受到 Java 爱好者的许可。
IDE	NetBeans	NetBeans 是开源软件开发集成环境，是一个开放框架，可扩展的开发平台，可以用于 Java、C/C++，PHP 等语言的开发，本身是一个开发平台，可以通过扩展插件来扩展功能。
Web 浏览器	Chrome	Web 标准实现最好的浏览器，且性能优于其它浏览器。

2.6 本章小结

本章主要介绍了这个基于 LeanCloud 的运动约伴网站中所使用到的主要技术。

本网站使用 HTML 作为前端语言，JavaScript 脚本语言作为前端服务处理及交互方式，LeanCloud 提供的 API 作为后台数据服务处理方案，并使用 LeanCloud 云数据库作为应用数据存储。

3 需求分析

通过对应问题及其环境的理解与分析，为问题涉及的信息、功能及系统行为建立模型，将用户需求精确化、完全化，最终形成需求规格说明，这一系列的活动即构成软件开发生命周期的需求分析阶段。

需求分析是介于系统分析和软件设计阶段之间的桥梁。一方面，需求分析以系统规格说明和项目规划作为分析活动的基本出发点，并从软件角度对它们进行检查与调整；另一方面，需求规格说明又是软件设计、实现、测试直至维护的主要基础。良好的分析活动有助于避免或尽早剔除早期错误，从而提高软件生产率，降低开发成本，改进软件质量。

本章节将网站设计实现需要满足的需求分为两类，第一类为功能性需求，即本网站需要能够在网页上通过代码实现的具体功能；第二类为非功能性需求分析，即网站的可行性，可扩展性，可维护性等与网站具体功能无关的需求的分析。

3.1 功能性需求分析

为了方便用户使用本系统完成注册、登录以及个人信息登记展示，寻找相同兴趣爱好的好友进行约伴运动，网站需要实现如下功能：

- （1）登录管理，提供登录注册功能；
- （2）用户信息管理，提供用户个人信息修改功能；
- （3）相册管理，提供照片上传及查看修改功能；
- （4）好友管理，提供查询、添加、删除好友的功能；
- （5）约伴管理，提供查询有相同兴趣爱好的用户并进行约伴的功能；
- （6）在线聊天，提供约伴成功的两个用户在线进行聊天的功能。

3.1.1 登录管理

用户在使用本网站时首先需要进行注册，只有注册用户才能够使用网站的全部功能；用户在注册成功后会受到提示，并可以进行登录操作。

非注册用户在登录操作时会被提示未注册并拒绝进入下一级页面，而注册时如果出现重复用户名或非法输入、两次密码输入不一致的情况，该注册均视为失败并要求用户再次注册。

完整的登录注册页面应包括简洁易用的输入框及登录、注册按钮，注册页面应包含注册所需要用户提供的必要信息的各个输入框体及选择注册或返回上一级的按钮。

3.1.2 用户信息管理

用户信息管理提供给网站使用者对自己信息进行编辑修改的功能。用户信息包括该登录用户的用户名即昵称，学校信息，个人运动爱好信息，自我标签，头像几部分信息，注册用户在成功登录后应有权修改自己的各类信息，并保存以便于查看。

一个完整的信息管理页面应包含修改用户名的文本框，查找自己学校的文本框及选择框体，修改标签的复选框及上传头像和保存修改的对应按钮，便于用户进行操作。

3.1.3 相册管理

照片的展示是一个用户展现自己运动轨迹的重要部分，一个完整的相册管理页面应能够为登录用户提供查看自己相册，同时上传照片，删除照片的功能，在页面上能够显示照片的缩略图，在用户点击照片后能够打开大图进行查看，同时提供删除功能。

页面上应有简洁易用的上传、删除操作按钮便于用户使用该功能。

3.1.4 好友管理

好友功能是一个社交网站中必不可少的功能，一个合理的好友管理模块中应为登录用户提供按照用户输入的姓名进行模糊搜索其他符合条件的用户的功能，在搜索结束后，搜索结果应以标签的形式展现结果用户的基本信息，并提供添加好友的按钮便于用户进行操作。

用户同时可以进入好友申请管理页面查看自己接收到的好友申请，并通过该页面的按钮对申请进行处理。同时已经是好友的两个人不能再次进行好友申请。

用户登录后在好友信息管理页面上能够查看自己的好友的信息，同时通过好友标签能够进入好友的信息页面查看好友的详细信息，以及好友的相册。

3.1.5 约伴管理

约伴功能是本网站的重点功能，用户应能够通过一个简单易用的下拉框选择自己想要进行的运动类型，通过搜索网页上会显示出符合该条件的其他用户，登录用户便可以通过约伴按钮发送自己的约伴请求给对方。但是同样的两个用户在上一个请求尚未被处理的情况下应该不能重复进行约伴。

用户在约伴信息管理页面中可以对自己收到的请求进行管理，选择接受该请求或是拒绝。

3.1.6 在线聊天

在线聊天功能是一个社交网站需要具备的功能，用户在接受了一个新的约伴请求之后网站后台应自动为约伴成功的两位用户创建一个单人聊天室，用户可以在聊天室页面中选择想要进入的聊天室或是删除该聊天室。

用户进入聊天室后应能够和对方用户进行实时聊天或留言等待对方查看。

3.2 用例分析

3.2.1 用户注册

在本用例中，用户首先选择注册操作，再按照要求输入电话，姓名及密码。

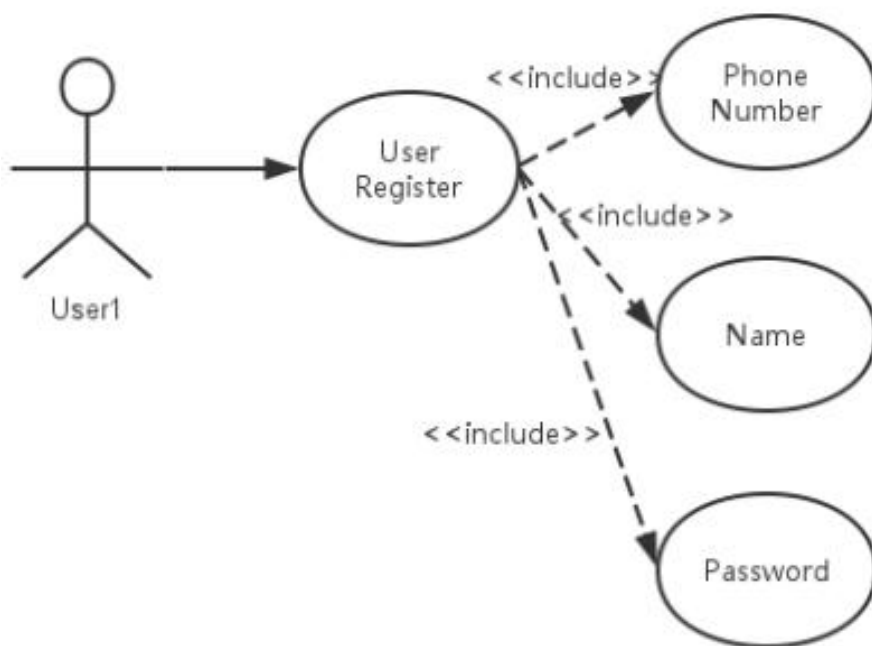


图 3.1 用户注册用例图

3.2.2 用户信息管理

在本用例中，用户需要进行的操作包括登录，进入用户信息管理界面，并根据选择不同对自身的信息进行编辑和更新，包括个人信息更新，上传照片，更新运动。其中个人信息更新中又包括更改名字，上传头像，编辑标签，更改学校，更新按钮点击几个动作。

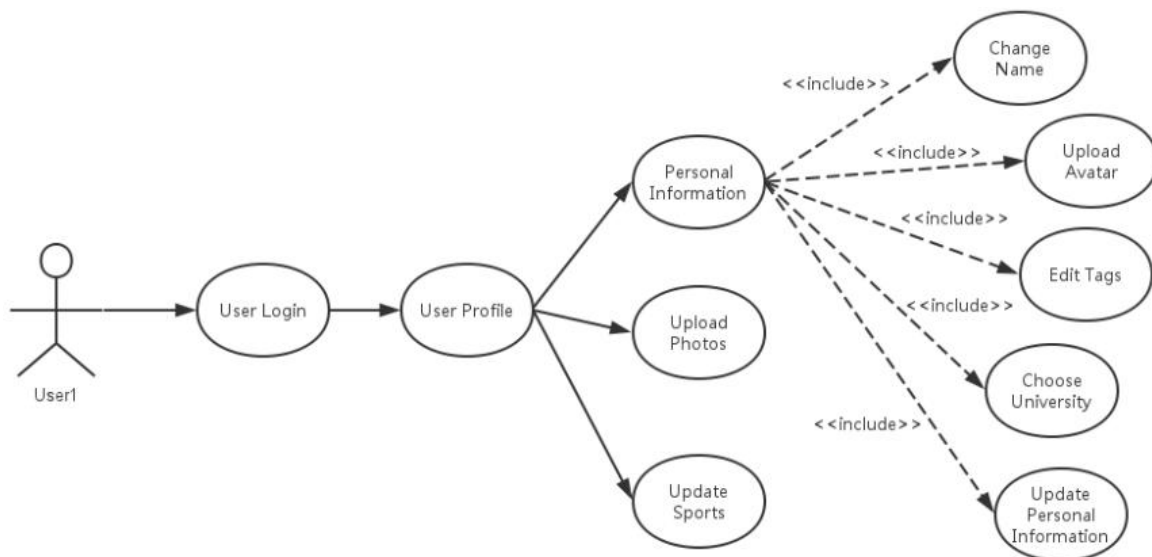


图 3.2 用户信息管理用例图

3.2.3 好友管理

在本用例中，用户登录后选择好友管理标签，进入后通过选择添加或管理好友操作可对好友列表进行更新，添加操作中需要输入姓名，选择查询结果，发送请求，等待回复。好友列表操作中，用户可选择删除好友，查看好友相册，查看好友信息。

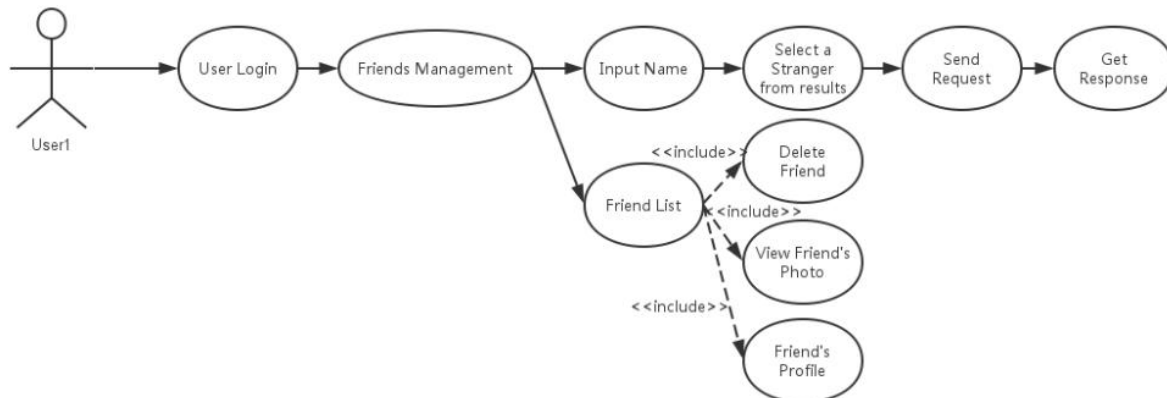


图 3.3 好友管理用例图

3.2.4 约伴管理

在本用例中，用户登录后可以选择发送请求，从查询到的结果中选择一个目标用户并将约伴请求发送出去，等待回复。

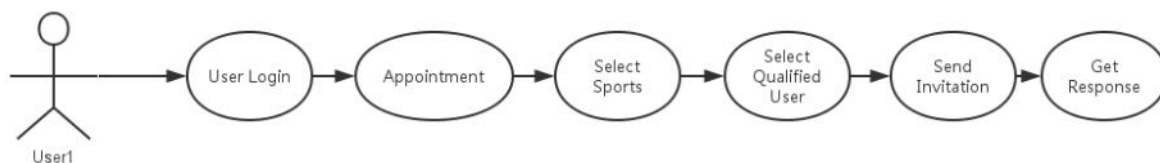


图 3.4 约伴管理用例图

3.2.5 请求处理

在本用例中，用户登录后可以进入好友请求或约伴请求处理自己收到的各类请求，两类请求都包含接受或拒绝的操作，在用户做出选择后该处理结果将会被反馈到请求发送方。其中，约伴请求接受后后台将自动创建一个聊天室，好友请求接受后将自动添加请求发送方到登录用户的好友列表中去。

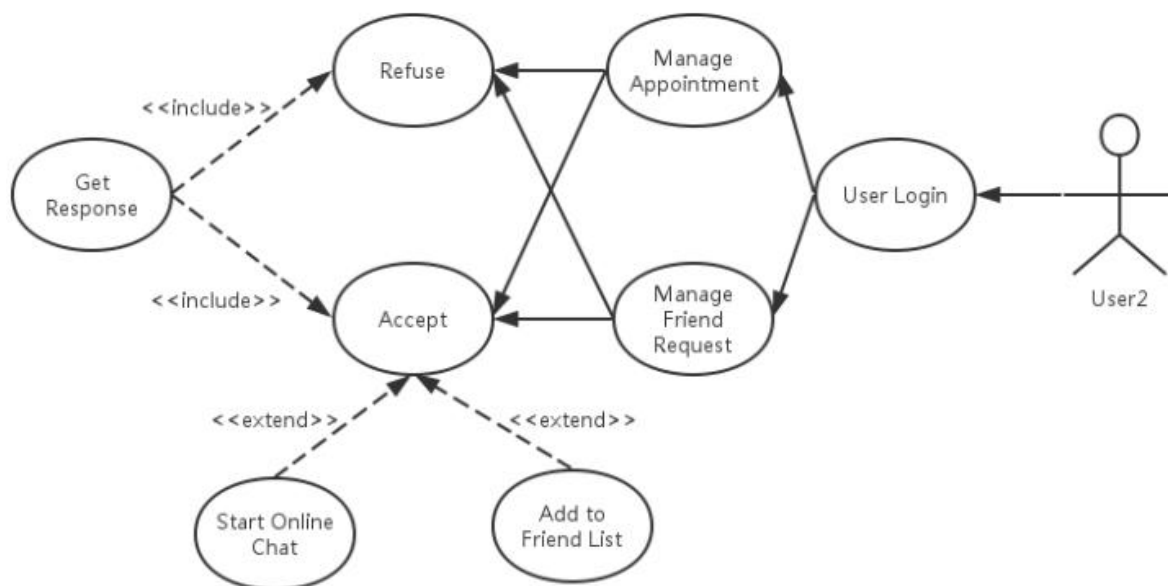


图 3.5 请求处理用例图

3.2.6 数据库管理

本用例中，管理员通过后台进入数据库管理，可以对数据库中的用户数据进行维护，并能够对学校进行添加和修改保证用户能够正确查找到自己的学校。

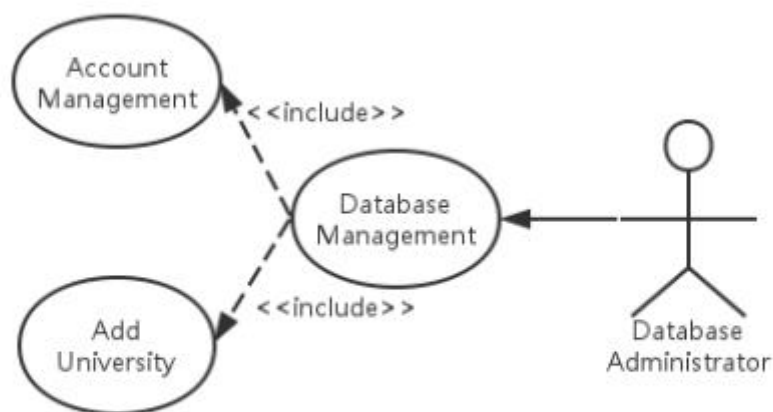


图 3.6 数据库管理用例图

3.3 非功能性需求分析

所谓非功能性需求，是指软件产品为满足用户业务需求而必须具有且除功能需求以外的特性。软件产品本身不是孤立存在的，还涉及到诸多外在环境的影响。非功能性需求必须考虑软件既要可用，又要易用。软件产品的非功能性需求包括系统的可行性、性能需求、安全性、可维护

性和可扩展性等。

3.3.1 系统可行性

（1）技术可行性：本设计基于 LeanCloud 云引擎提供的 API 与后台进行交互，同时使用 JavaScript 脚本语言及 HTML 进行编写，在技术上不存在很高的难度；同时由于功能模块的分离及 HTML 优秀的可扩展性，后期维护及功能扩展也极其便利。

（2）经济可行性：本网站在本人的笔记本电脑上进行开发，无需任何其他平台的支持或赞助，经济成本几乎为零，不存在问题。

（3）操作可行性：本网站采用图形化的操作界面，用户只需在规定的位置输入信息并通过按钮进行各类操作，无需具体的数据库或编程语言类知识，完全可行。

3.3.2 性能需求

社交类网站往往要求较高的响应效率，避免用户长时间的等待，本网站由于使用了相应的 API 来处理各类后台数据，不需要用户在操作时进行加载或等待本地响应，唯一的限制因素是本地网络状况，在网络不佳的情况下才会出现长时间的延迟或等待的情况，因此在性能方面的需求很容易满足。

3.3.3 安全性需求

应用系统的用户管理、权限管理充分利用操作系统和数据库的安全性。对于用户选择保密的信息（如电话号码，邮箱地址等）应做好保密工作，另外对于恶意注册和虚假注册的用户发出警告或不授予部分权限。

应用软件时有完整的日志记录。提供应用系统的运行记录，提供分析报告，以便及时审查日志文件，分析告警信息，掌握运行状况，对系统可能发生的故障做好应急方案。如果系统发生崩溃，用户进行重启后应能恢复主要数据，不会造成重要数据丢失（如好友信息丢失）。

3.3.4 可维护性

软件可维护性即维护人员对该软件进行维护的难易程度,具体包括理解、改正、改动和改进该软件的难易程度。本网站使用了高内聚低耦合的设计，因此大大提高了本系统的可维护性。高内聚低耦合的设计使得软件产品复用率高、黏合度低，即在加入新功能时改动对其他模块的影响小，改动范围小；或者在改动已有功能时，不会因为一个小小的改动引起看似无关的地方发生故障。

3.3.5 可扩展性

软件可扩展性是指设计良好的代码允许更多的功能在必要时可以被插入到适当的位置中。这样做的目的是为了应对未来可能需要进行的修改，而造成代码被过度工程化地开发。本网站使用了可扩展性极强的 HTML 语言作为基础，后期开发中只需要添加相应的标签即可扩展新的功能，极其便利。

3.4 本章小结

通过对本社交网站的理解与分析，为系统涉及的信息、功能及系统行为建立模型，确定系统的功能和执行力。首先本系统应该包括登录管理、用户信息管理、相册管理、好友管理、约伴管理、在线聊天六大功能。其次，设计本系统的时候，应该考虑到系统的可行性、性能需求、安全性、可维护性和可扩展性等非功能性需求。最后，完成用例图。

4 网站设计

4.1 网站总体架构

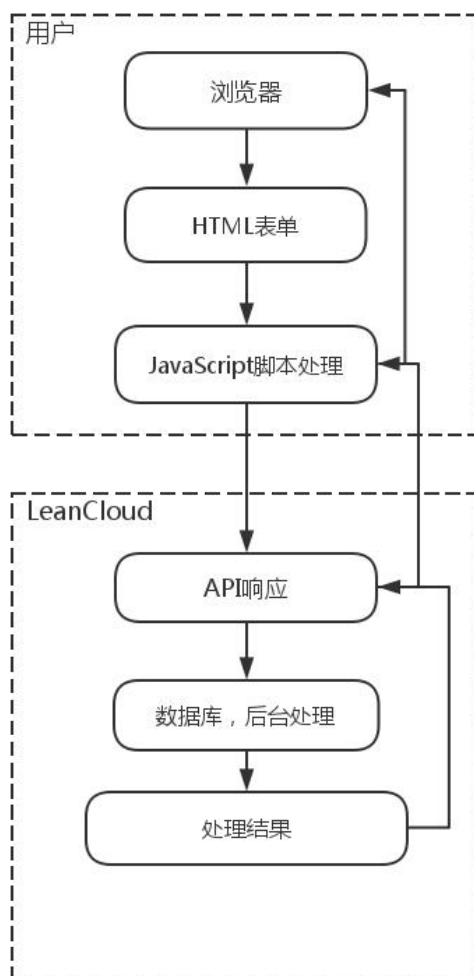


图 4.1 架构图

如图所示，整个 web 应用分为两部分，用户端在浏览器上通过页面前端提交 HTML 表单到脚本进行处理，脚本内通过 API 与云端进行通信，云端后台对提交的数据根据调用的 API 不同进行相应的操作，并进行响应回调，传回的数据经由脚本处理显示在 HTML 端，呈现在用户的浏览器上。

4.2 功能模块设计

4.3.1 用户登录注册模块

用户在使用本网站时应当能够顺利注册账号并进行登录操作，该模块设计以方便用户进行操作为主要目标。

表 4.1 用户登录注册功能表

	用户登录注册
功能描述	1. 识别用户输入的用户名和密码，并提交 2. 根据用户输入的信息进行判断是否符合要求并提交
访问的数据库表	用户信息表_User
进行的操作	1. 提交用户的输入 2. 执行数据库对应操作 3. 根据返回结果进行页面跳转
产生的结果	1. 跳转到相应的界面 2. 用户账号创建并赋予新用户登录网站的权利
结果存储位置或输出	1. 结果表现在跳转页面上，不同的结果，跳转页面不同 2. 新用户信息存储位置为_User 表，将新建数据条目并存储注册时提供的用户名，姓名及密码（保密）

4.3.2 用户信息模块

本模块负责执行用户对个人信息的所有修改操作，包括姓名，学校，性别，标签，运动以及头像修改。考虑到用户对个人信息的修改可能存在遗漏，本模块在除头像修改的部分中要求一次性全部填写，不能留白，并在用户选择修改时自动检查是否存在遗漏并提示用户补全信息。

其中，标签会在页面加载时从数据表 tags 中读取并动态生成复选框显示在页面上，学校信息需要用户输入关键词进行模糊搜索并提供选择框让用户对学校进行选择，运动信息将从 sports 表中读取，用户需要对自己热爱的运动进行一个简单地评级并进行保存即可。

表 4.2 用户信息功能表

	用户信息
功能描述	1. 获取用户需要修改的信息内容 2. 经过对相应信息标的查询，将对于结果显示在页面上 3. 自动检查用户所修改的信息是否存在未填写内容 4. 根据用户输入的信息对相应表进行操作
访问的数据库表	_User, _File, Campus, tags, sports
进行的操作	1. 检查用户的输入 2. 获取对应信息 3. 提交用户的输入 4. 执行数据库对应操作 5. 根据返回的结果进行页面跳转
产生的结果	1. 跳转到相应页面

	2. _User 表及_File 表中相应内容根据提交的数据产生更新
结果存储位置或输出	1. 结果表现在跳转页面上，结果不同，跳转的页面也不同 2. 用户更新的信息存储在_User 表及_File 表中

4.3.3 相册模块

该模块负责执行用户对相册的一系列操作，包括上传照片，删除照片，以及对相册的查看。为了方便用户操作，每张照片的缩略图都配置了一个独立的按钮，方便进行操作。

同时也包含查看好友相册的功能，经过好友模块处理进入查看好友相册页面时加载，将从_File 中读取对应当前选择的好友的相册列表信息并能够被打开查看，但无法进行编辑。

表 4.3 相册功能表

	相册
功能描述	1. 获取用户上传的照片 2. 获取用户在页面上对照片的选择 3. 对相应的选择进行操作
访问的数据库表	_User, _File
进行的操作	1. 上传照片 2. 照片删除 3. 打开照片大图
产生的结果	1. 照片成功上传到_File 表中，同时_User 表相应的条目产生更新，页面刷新 2. 照片删除成功，_User 表相应条目产生更新，页面刷新
结果存储位置或输出	1. 结果表现在跳转页面上，结果不同，跳转的页面也不同 2. 用户更新的信息存储在_User 及_File 表中

4.3.4 好友模块

该模块负责好友的查询，添加，删除以及查看功能。用户首先需要通过页面给定目标好友的姓名关键字，进行模糊搜索后用户可以选择添加相应的好友到自己的好友列表中，并发送一个通知给对方，接受方应能够接收到通知并决定是否也添加该请求发送方到自己的好友列表中去。同时用户还可以对自己的好友进行删除或查看详细信息的操作。

表 4.4 好友功能表

	好友
功能描述	1. 获取用户输入对_User 表进行模糊搜索 2. 发送好友添加通知 3. 打开好友信息页面

	4. 删除好友
	5. 接受/拒绝好友请求
访问的数据库表	_User, AddRequest
进行的操作	1. 获取用户输入并进行模糊搜索 2. 发送好友通知并添加好友 3. 删除好友 4. 接受好友请求，将发送者添加至自己的好友列表 5. 拒绝好友请求
产生的结果	1. 跳转到相应页面，目标好友添加至好友列表，_User 表更新，AddRequest 表更新 2. 跳转至相应页面，好友删除，_User 表更新 3. 接受/拒绝好友请求，_User 表更新，AddRequest 表对应数据条目更新
结果存储位置或输出	1. 结果表现在跳转页面上，结果不同，跳转的页面也不同 2. 用户更新的信息存储在_User, AddRequest 表中

4.3.5 约伴模块

该模块负责运动约伴的功能，用户应能够使用对应页面中的运动类型搜索符合条件的用户并获取他们的列表，根据列表上提供的按钮选择是否邀请目标用户进行运动约伴；用户同时可以进入对应页面处理自己的约伴请求，选择是否接受；一旦用户选择接受请求，将自动创建一个聊天室并存储。

表 4.5 约伴功能表

	约伴
功能描述	1. 获取用户输入 2. 进行模糊搜索，显示符合条件的结果列表 3. 发送约伴请求 4. 处理约伴请求
访问的数据库表	_User, EngagementStrangers, _Conversation
进行的操作	1. 获得用户输入并进行模糊搜索 2. 对相应数据表进行搜索，显示搜索结果 3. 添加约伴信息到相应数据表 4. 根据用户选择更新相应数据表
产生的结果	1. 页面刷新并显示新数据 2. EngagementStrangers 表产生更新

3. _Conversation 表产生更新

结果存储位置或输出	1. 结果表现在跳转页面上，结果不同，跳转的页面也不同 2. 用户更新的信息存储在_User，EngagementStrangers，_Conversation 表中
-----------	--

4.3.6 在线聊天模块

该模块直接调用云引擎提供的后台方法及相应 API，处理在线会话请求。

表 4.6 在线聊天功能表

在线聊天	
功能描述	1. 用户通过聊天室进行在线会话
访问的数据库表	_Conversation
进行的操作	1. 用户进入聊天室页面 2. 发送消息至后台 3. 显示获得的消息
产生的结果	1. 页面更新显示消息
结果存储位置或输出	1. 结果表现在跳转页面上，结果不同，跳转的页面也不同

4.3 前端设计

4.4.1 设计原则

前端页面设计以简洁明了为主，尽量减少按钮，文字区域，通过简单的按钮为用户提供更好的操作体验，并以单色，扁平化的设计风格靠近当今潮流。

4.4.2 主要页面

由于本网站立足于运动，因此整个主页的主题是以宁静为主的户外为主，其余页面使用统一的配色进行搭配，突出鲜明的颜色区分，便于用户区分各区域的功能。

页面包括：

- （1）登录页面
- （2）注册页面
- （3）个人主页
- （4）个人信息页面
- （5）运动信息页面
- （6）相册页面

- (7) 好友列表页面
- (8) 好友请求页面
- (9) 好友主页
- (10) 约伴页面
- (11) 约伴请求页面
- (12) 聊天室页面

4.4 数据库设计

数据库设计(Database Design)是指根据用户的需求,在某一具体的数据库管理系统上,设计数据库的结构和建立数据库的过程。数据库系统需要操作系统的支持。

数据库设计是建立数据库及其应用系统的技术,是信息系统开发和建设中的核心技术。由于数据库应用系统的复杂性,为了支持相关程序运行,数据库设计就变得异常复杂,因此最佳设计不可能一蹴而就,而只能是一种“反复探寻,逐步求精”的过程,也就是规划和结构化数据库中的数据对象以及这些数据对象之间关系的过程。

系统中的具体信息、数据要求等内容可抽象成如下数据项及数据结构:

- (1) 用户信息: 用户名, 性别, 好友, 学校, 密码, 标签, 照片, 运动, 头像
- (2) 好友申请: 请求方, 被请求方, 处理状态
- (3) 学校及校区信息: 校区, 学校, 全称
- (4) 运动信息: 运动名称
- (5) 标签信息: 标签编号, 标签名称
- (6) 约伴信息: 邀请方, 被邀请方, 运动类型, 处理状态
- (7) 文件: 存储路径
- (8) 会话信息: 发起者标识, 接受者标识

4.4.1 用户信息表设计

该表用于存放用户的各类主要信息, `objectId` 作为主键, 其中 `friends` 数据类型为 `relation`, 其成员为指向其他用户数据主键的指针表; `sP_campus` 为用户学校, 其类型为指向学校 `Campus` 表的主键的指针。

表 4.7 用户信息表_User

字段名	数据类型	字段说明	备注
<code>objectId</code>	<code>String</code>	用户识别	自动生成
<code>sP_userName</code>	<code>String</code>	用户名	
<code>username</code>	<code>String</code>	用户账户名	注册用手机号
<code>friends</code>	<code>Relation</code>	好友列表	
<code>password</code>	<code>String</code>	用户密码	后台不可见
<code>sP_campus</code>	<code>Pointer</code>	用户所在学校	

sP_sex	String	用户性别
sP_tagList	Array	用户标签表
sP_photoIdList	Array	用户照片
sP_sportList	Array	用户选择运动种类
avatar	File	用户头像

4.4.2 好友申请信息表设计

好友申请信息表中保存该申请的发起方和接收方的信息，以及该次请求的处理结果的 flag 标识。

表 4.8 好友申请信息表 AddRequest

字段名	数据类型	字段说明	备注
objectId	String	申请信息识别	自动生成
fromUser	Pointer	请求方	
toUser	Pointer	被请求方	
status	Number	是否接受请求状态	

4.4.3 学校及校区信息表设计

学校及校区信息表为用户查询自己所在的校区提供依据，其中查询基础为学校 school 字段，进行模糊搜索后网页将显示查询到的校区信息 campus 字段，并将该条目的指针作为结果保存到 _User 表的相应字段中去。

表 4.9 学校及校区信息表 Campus

字段名	数据类型	字段说明	备注
objectId	String	校区识别	自动生成
campus	String	校区	
school	String	学校	
schoolFullName	String	学校+校区全称	

4.4.4 运动信息表设计

运动信息表为用户查询添加自己的运动提供依据，用户只能从给定的运动类型中做选择，同时管理员可以向运动中添加内容。

表 4.10 运动信息表 sports

字段名	数据类型	字段说明	备注
sportId	String	运动识别	自动生成
sportName	String	运动名词	

4.4.5 标签信息表设计

标签信息表为用户添加自己的标签提供查询依据，用户只能从给定的标签中进行选择，同时管理员可以向标签中添加内容。

表 4.11 标签信息表 tags

字段名	数据类型	字段说明	备注
objectId	String	标签识别	自动生成
tagContent	String	标签内容	
tagID	Number	标签编号	

4.4.6 约伴信息表设计

约伴信息表记录该次约伴的发起人和接收人，并记录运动类型和处理状态，便于用户查看自己的邀约情况是查询。

表 4.12 约伴信息表 EngagementStrangers

字段名	数据类型	字段说明	备注
objectId	String	标签识别	自动生成
toId	Pointer	被邀请用标识	
status	Number	处理状态标识	
sportType	Number	运动标识	
fromId	Pointer	邀请人标识	

4.4.7 文件信息表设计

文件信息表中存放的是用户上传的头像及照片内容，每个文件在云端被自动给定 url，在上传成功后该 url 会被存储到文件信息表中，同时如果存储内容为头像，则指向该头像数据条目的指针会被保存到_User 表中的相应位置；如果是照片，该数据条目的 objectId 会被添加到对应用户的相册字符串数组中。

表 4.13 文件信息表_File

字段名	数据类型	字段说明	备注
objectId	String	标签识别	自动生成
url	String	图片存储路径	

4.4.8 会话信息表设计

会话信息表是云引擎在创建一个会话时自动添加条目的表，云引擎会将会话双方的 id 作为该会话的标识存放到表中作为查询的依据。

表 4.14 会话信息表_Conversation

字段名	数据类型	字段说明	备注
objectId	String	标签识别	自动生成
name	String	发起者 id	
c	String	接受者 id	

4.4.9 E-R 图

E-R 图描述了本网站的数据库模型及其相互关系，及所涉及到的实体，根据每个数据表的不同，本文将 E-R 图分别画出便于查看。

在_User 表中，存在多个实体，同时在 friend 字段中，由于其数据类型为 relation 即关系，所以每个用户在该字段中都存在多对多的关系，在图 4.2 中以 n 来标识。

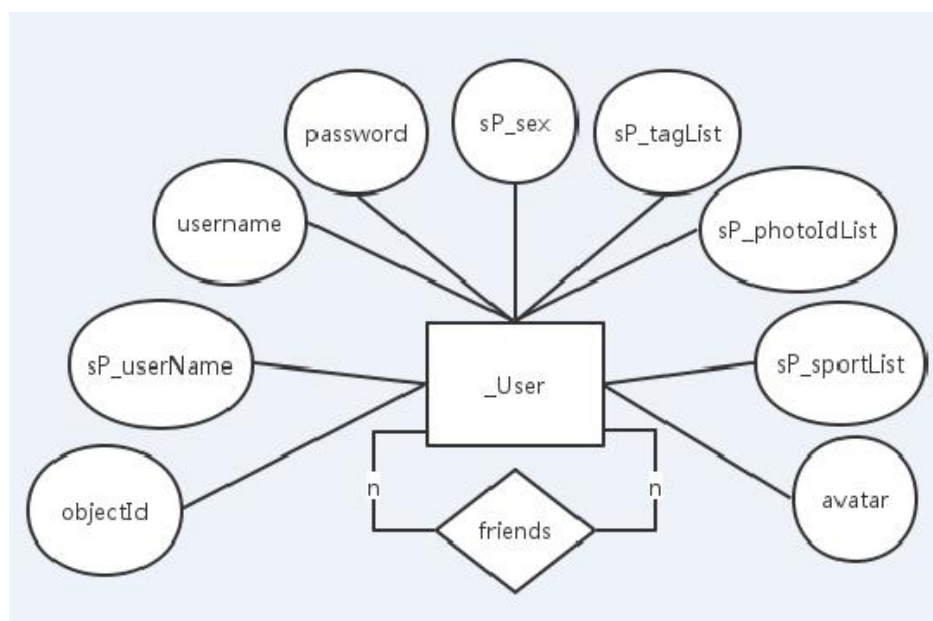


图 4.2 _User 表 E-R 图

在 AddRequest 表中，每个好友请求的发送方必须是单一的，即每一个_User 表中的条目只存在一个发送者，但是可以有多个接收者，故 toUser 是多对多的数据，在图 4.3 中以 n 标识。

在_User 表中，sP_campus 是用户的学校，一个用户只能够有一个学校，故其关系为一对一；该项是一个指向 Campus 中具体条目的指针，故在 Campus 表的 E-R 图即图 4.4 中也是以一对一的关系标识的。

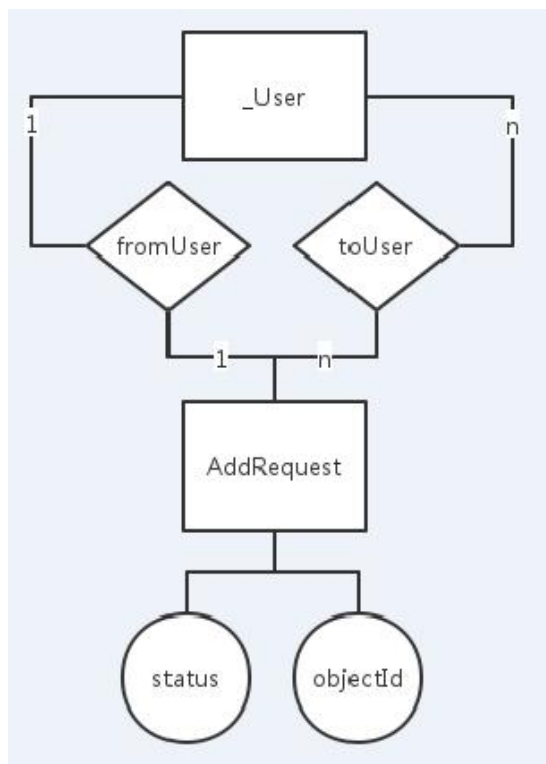


图 4.3 AddRequest 表 E-R 图

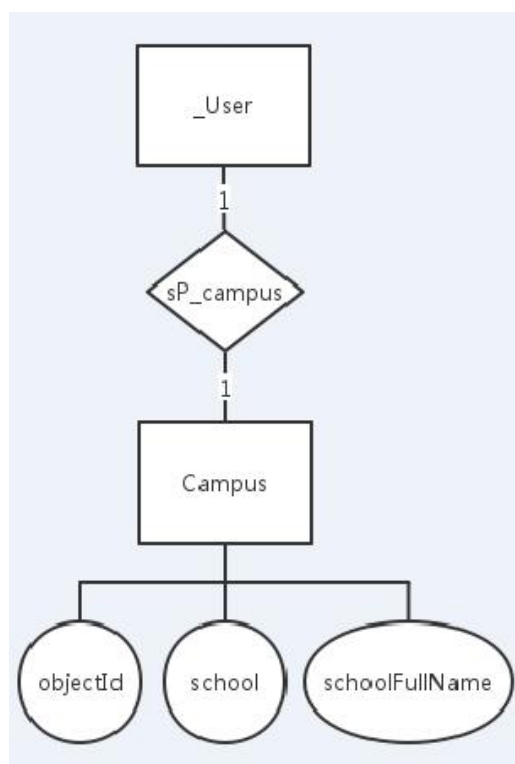


图 4.4 Campus 表 E-R 图

表 sports，表_Conversation，表 tags 中存储的都是可查询的内容，这几张表中只有实体并没有任何关系型的数据，故 E-R 图中只包含相应的实体，如图 4.5、图 4.6、图 4.7 所示。

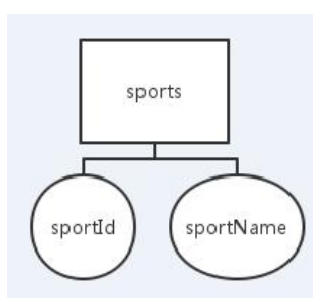


图 4.5 sports 表 E-R 图

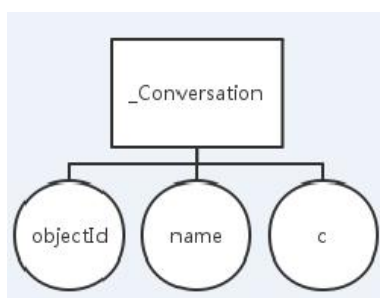


图 4.6 _Conversation 表 E-R 图

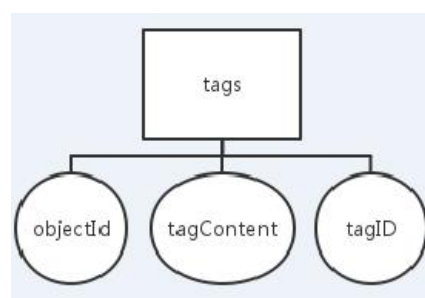


图 4.7 tags 表 E-R 图

表 EngagementStrangers 是约伴信息表，其中 fromId 为指向 _User 表中条目的指针，由于来自同一个用户的请求的发送方只能是该用户，故该条目为一对一的关系；而 toId 为指向 _User 表中其他条目的指针，来自同一个用户的请求可以有多个接收方，故该条目为多对多的关系，在图 4.8 中以 1 和 n 分别进行标识。

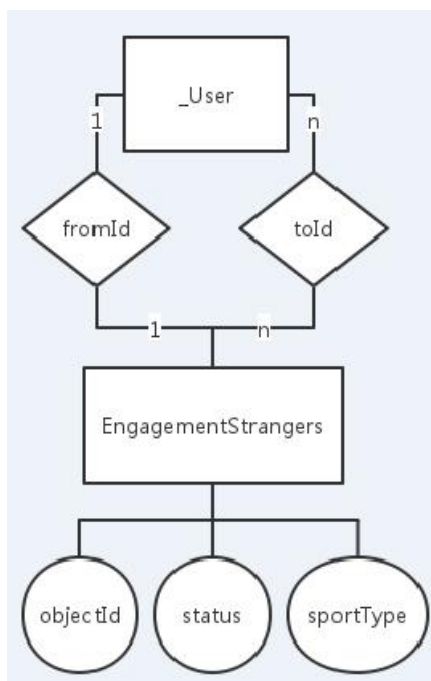


图 4.8 EngagementStrangers 表 E-R 图

4.5 本章小结

本章介绍了网站的详细设计：

首先介绍了网站的总体架构；

其次介绍了主要数据表的设计，并提供了相应的数据内容和类型；

接下来介绍了基于不同模块的功能设计；

最后根据以上设计对前端页面做出了符合网站操作要求的页面设计。

5 系统实现

5.1 数据库建立

数据库建立时直接使用 LeanCloud 云数据库进行设置及生成，无具体代码。建立数据库时通过 LeanCloud 提供的便捷功能，能够快速建立各 Class 即表。编辑数据条目时通过图形化界面能够便利地动态添加新的条目，并对其属性，类型，默认值进行设置。

5.2 用户登录注册模块实现

5.2.1 功能概述

用户在使用本网站时应当能够顺利注册账号并进行登录操作，该模块设计以方便用户进行操作为主要目标。本模块在功能上以调用引擎的 API，通过 JS 脚本传递从前端页面获取的参数并获取返回值，为用户提供一个包含简明易用的登录注册功能的页面。

5.2.2 部分代码

这部分代码是登录模块的实现代码，具体展示了登录操作的处理流程。

```
<script>
//登录
function Login() {
    var name = document.getElementById("name").value;
    var pass = document.getElementById("pass").value;
    if (name == "") {
        $("#name_empty").css("display", "inline");
        $("#name_error").css("display", "none");
        $("#pass_empty").css("display", "none");
        $("#pass_error").css("display", "none");
    }
    else if (pass == "") {
        $("#pass_empty").css("display", "inline");
        $("#pass_error").css("display", "none");
        $("#name_error").css("display", "none");
        $("#name_empty").css("display", "none");
    }
    else {
        AV.initialize(id, key);
        AV.User.logIn(name, pass, {
            success: function (result) {
                var object = result;
            }
        });
    }
}
```

```

        window.location.href = "/home.html";
    },
    error: function (user, error) {
        if (error.message === "Could not find user") {
            $("#name_error").css("display", "inline");
            $("#pass_error").css("display", "none");
            $("#name_empty").css("display", "none");
            $("#pass_empty").css("display", "none");
        }
        else {
            $("#pass_error").css("display", "inline");
            $("#name_error").css("display", "none");
            $("#name_empty").css("display", "none");
            $("#pass_empty").css("display", "none");
        }
    }
});
}
}
</script>

```

5.2.3 界面

本界面为登录界面。

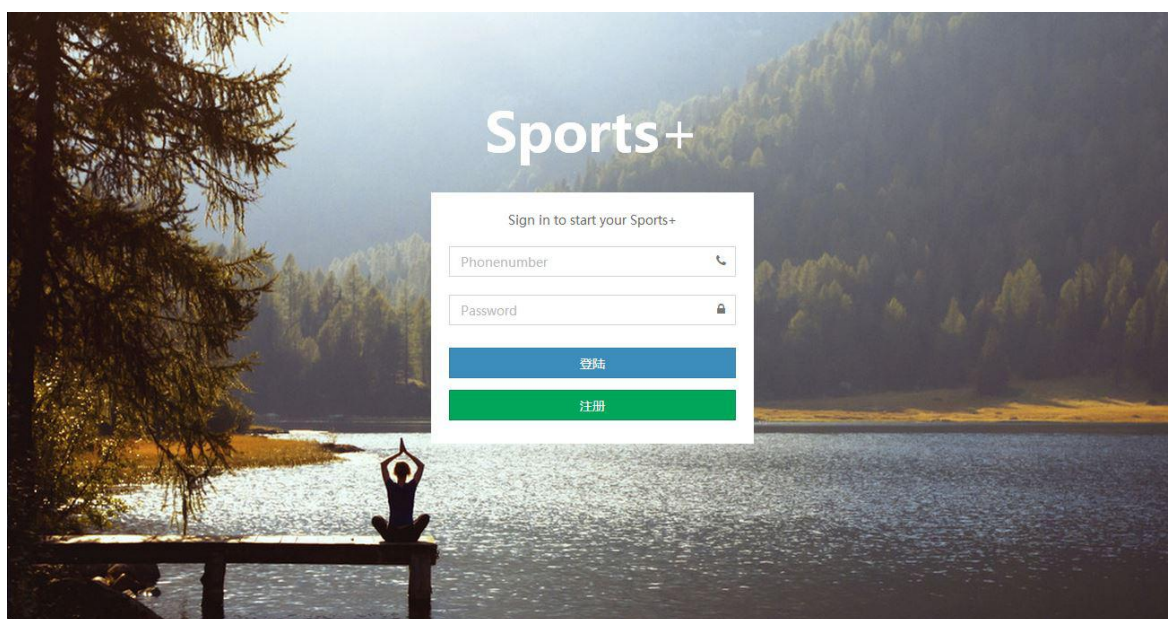


图 5.1 登录界面

本界面为注册界面。

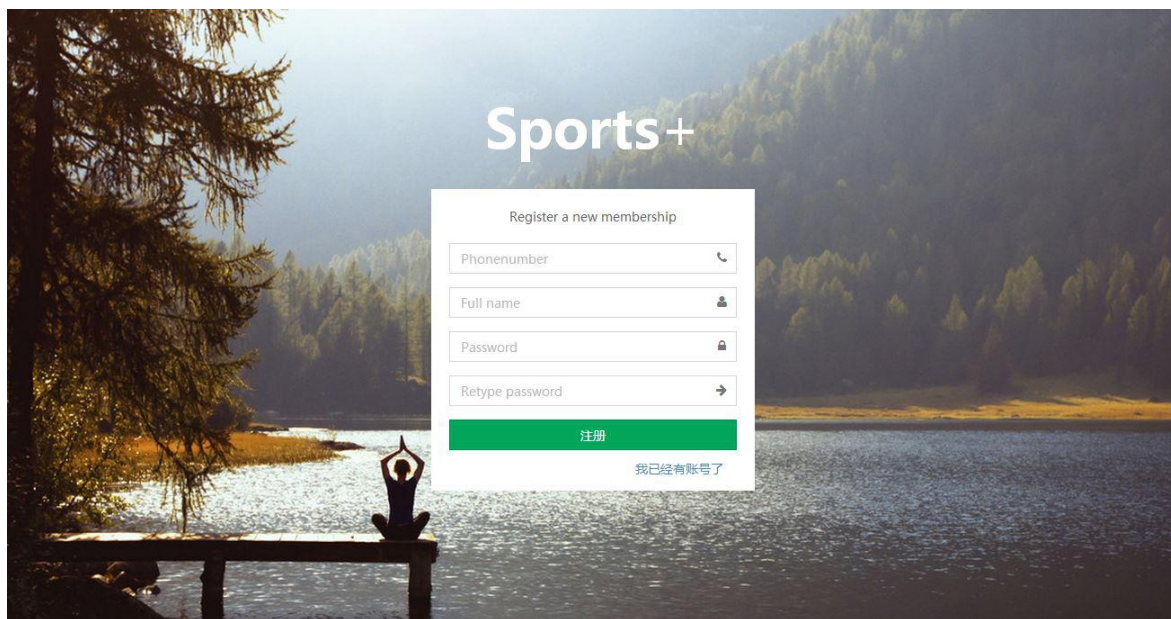


图 5.2 注册界面

5.2.4 前端部分代码

```
<body>
  <div class="login-box">
    <div class="login-logo" style="color: #ffffff;font-size: 60px">
      <b>Sports</b>+
    </div><!-- /.login-logo -->
    <div class="login-box-body" style="height: 280px">
      <p class="login-box-msg">Sign in to start your Sports+</p>
      <form action="/home.html" method="post">
        <div class="form-group has-feedback" style="padding-bottom: 0px;margin-bottom:
0px">
          <input id="name" type="" class="form-control"placeholder="Phonenumber"/>
          <span class="fa fa-phone form-control-feedback" style="padding-top:
9px"></span>
          <span id="name_empty" style="color: red;display: none">请输入账号！
</span>
          <span id="name_error" style="color: red;display: none">账号不存在！
</span>
          <span style="color: #ffffff;font-size: 1px;">1</span>
        </div>
        <div class="form-group has-feedback" style="padding-bottom:0px;margin-bottom:
```

0px">

```

        <input id="pass" type="password" class="form-control"placeholder="Password"/>
        <span class="fa fa-lock form-control-feedback" style="padding-top:9px"></span>
    </div>
    <div class="row">
        <div class="col-xs-12" style="height: 25px;padding-top: 0px">
            <span id="pass_empty" style="color: red;display: none">请输入密码！
</span>
            <span id="pass_error" style="color: red;display: none">密码错误！ </span>
            </div>
            <div class="col-xs-12">
                <a onclick="Login()" class="btn btn-primary btn-block btn-flat">登
陆</a>
            </div>
            <div class="col-xs-12" style="margin-top: 13px">
                <a href="/register.html" class="btn btn-success btn-block btn-flat">
注册</a>
            </div>
        </div>
    </div>
</form>
</div><!-- /.login-box-body -->
</div><!-- /.login-box -->
<!-- jQuery 2.1.4 -->
<script src="/js/jquery-2.1.4.min.js"></script>
<!-- Bootstrap 3.3.2 JS -->
<script src="/js/bootstrap.min.js" type="text/javascript"></script>
<!-- AdminLTE App -->
<script src="/js/app.min.js" type="text/javascript"></script>
</body>
    
```

5.3 用户信息模块实现

5.3.1 功能概述

本模块负责执行用户对个人信息的所有修改操作，包括姓名，学校，性别，标签，运动以及头像修改。考虑到用户对个人信息的修改可能存在遗漏，本模块在除头像修改的部分中要求一次性全部填写，不能留白，并在用户选择修改时自动检查是否存在遗漏并提示用户补全信息。

5.3.2 部分代码

这一部分代码是个人信息修改的实现操作，其中由于 JavaScript 的回调机制，最后的保存即 save()方法的调用安排在所有的 set 操作之后的最内层处理，保证能够将之前所有的 set 操作全部成功发送到服务器。

//修改信息

```
function modify() {
    var user = AV.User.current();
    user.set('sP_userName', document.getElementById("sP_username").value);
    //保存性别
    var chkObjs = document.getElementsByName("optionsRadios");
    if (chkObjs[0].checked) {
        user.set('sP_sex', "男");
    }
    else {
        user.set('sP_sex', "女");
    }
    //保存 tag
    if (getTag() !== "") {
        var str = new Array();
        str = getTag().split(',');
        user.set('sP_tagList', str);
    }
    else
        alert("未选择标签！无法修改");
    //保存我的运动
    if (document.getElementById("campus").disabled === false) {
        if (Campus_index !== "") {
            var keyword = document.getElementById("search_campus").value;
            var Campus = AV.Object.extend("Campus");
            var query = new AV.Query(Campus);
            query.contains("schoolFullName", keyword);
            query.find({
                success: function (results) {
                    user.set('sP_campus', results[Campus_index - 1]);
                    console.log(results[Campus_index - 1].id);
                    user.save().then(function (user) {
```

```

//成功
alert("修改成功");
window.location.href = "/info_show.html";
}, function (error) {
    console.log("something wrong" + error.message);
})
},
error: function (error) {
    console.log(error.message);
}
});
}
else {
    alert("Error for campus index" + Campus_index);
}
}
else {
    alert("学校和学院信息未填写完整故修改未保存！");
}
}
}

```

5.3.3 界面

本界面为主页。



图 5.3 主页

装 订 线



<div class="form-group">

共 61 页 第 36 页

```
<input id="school" type="text" class="form-control">
</div>
<div class="form-group col-md-12" style="padding-left: 0px;margin-left: 0px;padding-right:
0px;margin-right: 0px">
    <select id="campus" class="form-control" disabled>
        <option>请选择学校</option>
    </select>
</div>
```

5.4 相册模块实现

5.4.1 功能概述

该模块负责执行用户对相册的一系列操作，包括上传照片，删除照片，以及对相册的查看。为了方便用户操作，每张照片的缩略图都配置了一个独立的按钮，方便进行操作。

5.4.2 部分代码

这一部分代码实现了对照片的保存操作，通过调用 API 提供的方法，首先创建一个 avFile 对象并存储相应的照片数据进去，之后再保存该对象，返回保存成功的信息。

//保存照片

```
function Save_Photo() {
    var user = AV.User.current();
    if (user) {
        var fileUploadControl = $("#profilePhotoFileUpload")[0];
        if (fileUploadControl.files.length > 0) {
            var file = fileUploadControl.files[0];
            var name = "photo.jpg";
            var avFile = new AV.File(name, file);
            avFile.save().then(function () {
                var photoDictionary = {};
                photoDictionary["fileId"] = avFile.id;
                photoDictionary["fileUrl"] = avFile.url();
                var query = new AV.Query(AV.User);
                query.get(user.id, {
                    success: function (obj) {
                        var results = new Array();
                        if (obj.get("sP_photoIdList") === undefined) {
                            results[0] = photoDictionary;
                        }
                    }
                });
            });
        }
    }
}
```

```

        user.set("sP_photoIdList", results);
        user.save().then(function () {
            alert("成功上传照片！");
            window.location.href = "/my_album.html";
        }, function (error) {
            alert(error.message);
        });
    }
    else {
        results = obj.get("sP_photoIdList");
        results[results.length] = photoDictionary;
        user.set("sP_photoIdList", results);
        user.save().then(function () {
            alert("成功上传照片！");
            window.location.href = "/my_album.html";
        }, function (error) {
            alert(error.message);
        });
    }
    },
    error: function (err) {
        alert("Photo save error: " + err.message);
    }
    });
    },
    function (error) {
        alert("Error in save to File: " + error.message);
    }
    });
    }
}
else {
    alert("Error!");
}
}
}

```

5.4.3 界面

本界面为相册的基础界面。

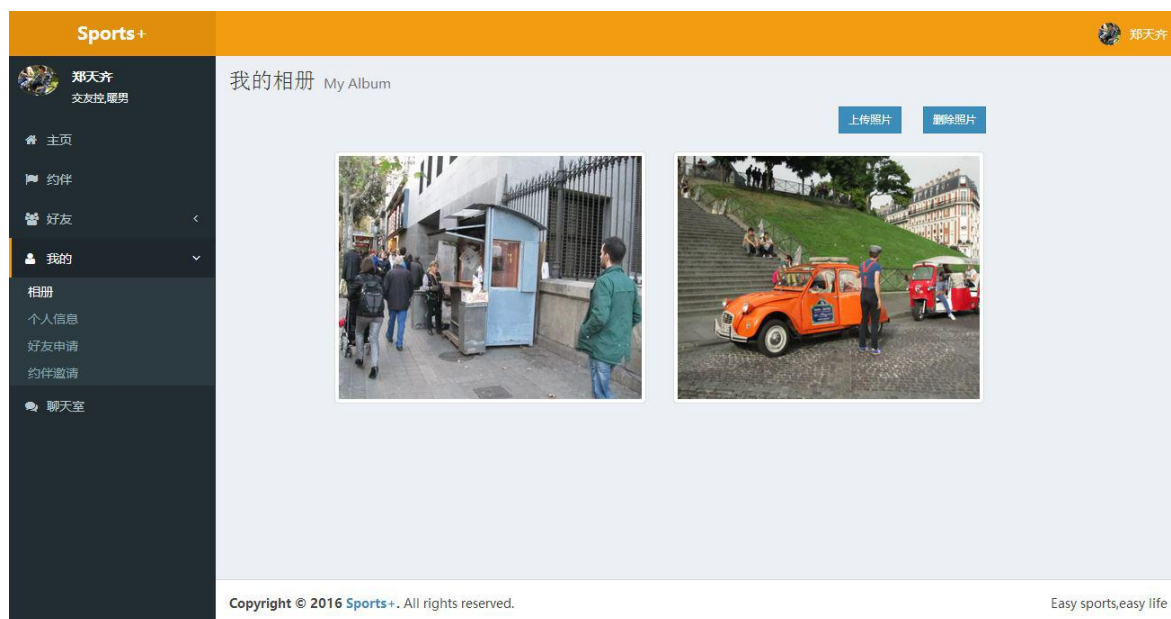


图 5.5 相册界面

5.4.4 前端部分代码

```
<section class="content">
<div class="row">
  <section class="col-md-12">
    <div class="col-md-8 col-md-offset-2 img-thumbnail" style="height: 305px">
      <div class="col-md-6" style="padding-top: 20px">
        
      </div>
      <div class="col-md-6" style="padding-top: 110px;padding-right: 0px;margin-right: 0px">
        <div class="col-md-12" style="margin-bottom: 61px">
          <input type="file" id="profilePhotoFileUpload"><br>
          <a onclick="Save_Photo()" class="btn btn-primarybtn-flat" style="width: 100px">上传照片
        </a>
        </div>
        <div class="col-md-12" style="text-align: right;padding: 0px;margin: 0px">
          <a href="/my_album.html" onclick="" class="btn btn-primary btn-flat" style="width: 70px">
返回</a>
        </div>
      </div>
    </div>
  </div>
</div>
```


5.5 好友模块实现

5.5.1 功能概述

该模块负责好友的查询，添加，删除以及查看功能。用户首先需要通过页面给定目标好友的姓名关键字，进行模糊搜索后用户可以选择添加相应的好友到自己的好友列表中，并发送一个通知给对方，接受方应能够接收到通知并决定是否也添加该请求发送方到自己的好友列表中去。同时用户还可以对自己的好友进行删除或查看详细信息的操作。

5.5.2 部分代码

这部分代码实现了对当前用户的关系数据好友条目中相应的关系的删除操作。首先从该用户的好友关系中找到所对应的好友，之后对该关系进行更改，删除其中的好友条目，并进行保存。
//删除好友

```
function Remove_Friend(username, sP_userName) {
    var user = AV.User.current();
    if (user) {
        var query = new AV.Query(AV.User);
        query.equalTo("username", username);
        query.first({
            success: function (object) {
                if (object !== undefined) {
                    var relation = user.relation('friends');
                    relation.query().find().then(function (list) {
                        console.log(list);
                        for (var i = 0; i < list.length; i++) {
                            if (list[i].get("sP_userName") == sP_userName) {
                                relation.remove(list[i]);
                                user.save().then(function () {
                                    // 成功
                                    alert("成功删除好友！");
                                    window.location.href = "/friends_list.html";
                                }, function (error) {
                                    // 失败
                                    alert(error.message);
                                });
                            }
                        }
                    }, function (error) {
```

```

        alert(error.message);
    });
}
else {
    alert("No such user!");
}
},
error: function (error) {
    alert(error.message);
}
});
}
else {
    alert("Error!");
}
}
}

```

5.5.3 界面

本界面为好友列表的查看界面。



图 5.6 好友列表界面

本界面为添加好友界面。

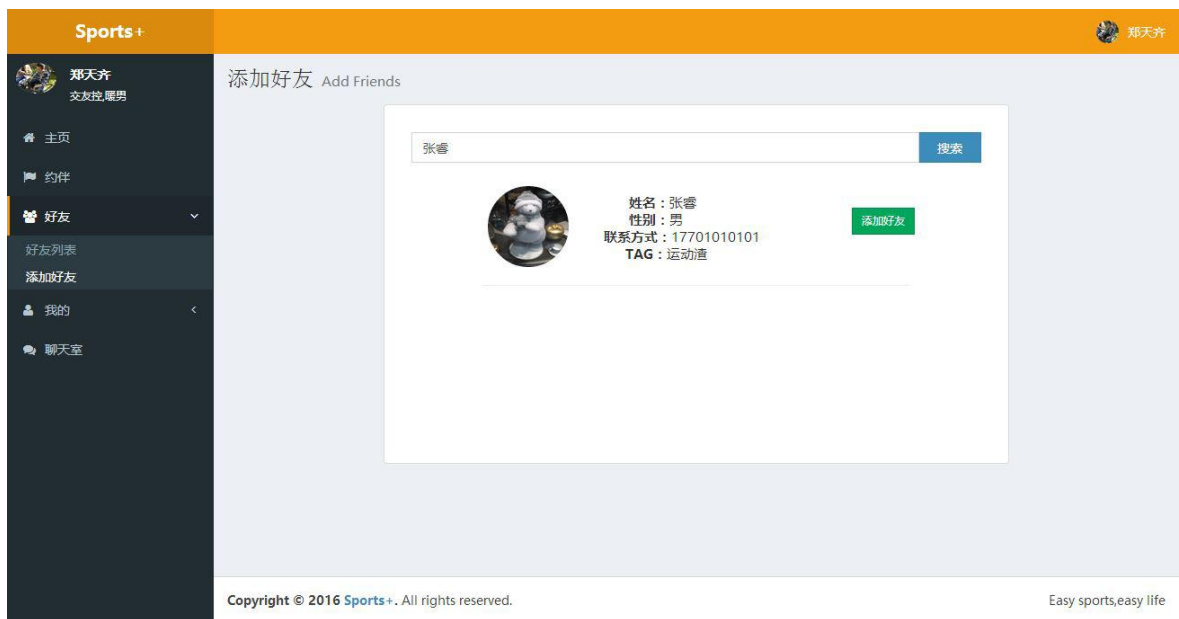


图 5.7 添加好友界面

5.5.4 前端部分代码

```
<ul class="sidebar-menu">
<!-- Optionally, you can add icons to the links -->
<li><a href="/home.html"><i class='fa fa-home'></i> <span>主页</span></a></li>
<li><a href="/my_invitationtable.html"><i class='fa fa-flag'></i> <span>约伴</span></a></li>
<li class="treeview">
  <a href="#"><i class='fa fa-group'></i> <span>好友</span> <i class="fa fa-angle-left
pull-right"></i></a>
  <ul class="treeview-menu">
    <li><a href="/friends_list.html">好友列表</a></li>
    <li><a href="/add_friends.html">添加好友</a></li>
  </ul>
</li>
<li class="treeview active">
  <a href="#"><i class='fa fa-user'></i> <span>我的</span> <i class="fa fa-angle-left
pull-right"></i></a>
  <ul class="treeview-menu">
    <li><a href="/my_album.html">相册</a></li>
    <li><a href="/info_show.html">个人信息</a></li>
    <li class="active"><a href="/friends_requests.html">好友申请</a></li>
    <li><a href="/sports_invitation_list.html">约伴邀请</a></li>
```

```

        </ul>
    </li>
    <li><a href="/chatroom_list.html"><i class='fa fa-comments'></i>
        <span>聊天室</span></a></li>
</ul>

```

5.6 约伴模块实现

5.6.1 功能概述

该模块负责运动约伴的功能，用户应能够使用对应页面中的运动类型搜索符合条件的用户并获取他们的列表，根据列表上提供的按钮选择是否邀请目标用户进行运动约伴；用户同时可以进入对应页面处理自己的约伴请求，选择是否接受；一旦用户选择接受请求，将自动创建一个聊天室并存储。

5.6.2 部分代码

这部分代码实现了要求有相同兴趣爱好的好友的功能，首先将发送请求一方及被请求方的 id 获取到，并将要进行约伴的运动类型进行整理并添加到条目中，最后根据约定的标志值对其进行设置并保存。

//邀请同样爱好的用户

```

function EngagementStrangers(StrangerId, SportType) {
    var AddEngagement = AV.Object.extend("EngagementStrangers");
    var engage = new AddEngagement();
    var query_1 = new AV.Query(AV.User);
    var user = AV.User.current();
    query_1.get(StrangerId, {
        success: function (obj) {
            engage.set("fromId", user);
            engage.set("toId", obj);
            engage.set("sportType", parseInt(SportType));
            engage.set("status", 0);
            engage.save(null, {
                success: function () {
                    alert("请求发送成功！");
                },
                error: function (err) {
                    alert("Engage error " + err.code + " " + err.message);
                }
            });
        }
    });
}

```

```
});
},
error: function (err) {
    alert("Error in add engagement" + err.message);});});
}
```

5.6.3 界面

本界面为约伴搜索的界面。

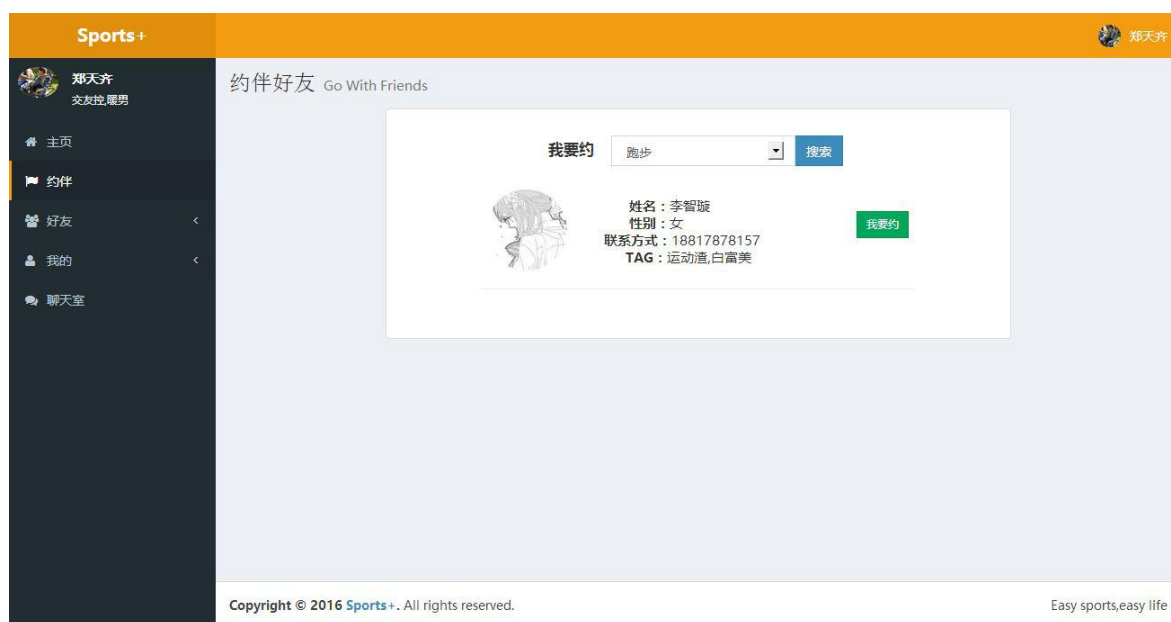


图 5.8 约伴搜索界面

本界面为处理约伴请求的界面。



图 5.9 处理约伴请求界面

装订线

5.7 在线聊天模块实现

该模块直接调用云引擎提供的后台方法及相应 API，处理在线会话请求。

这部分代码实现了让当前用户加入一个已知 id 的聊天，此时可以进行在线的即时聊天，同时在加入聊天时也会显示之前的聊天记录。

共 61 页 第 45 页

```

        appId: id,
        clientId: user.id,
        encodeHTML: true
    });
    realtimeObject.on('open', function() {
        realtimeObject.room(roomid, function(obj) {
            if (obj) {
                room = obj;
                room.join(function() {
                    room.list(function(data) {
                        console.log('当前 Conversation 的成员列表: ', data);
                        realtimeObject.ping(data.slice(0, 20), function(list) {
                            console.log('当前在线的成员列表: ', list);
                        });
                        getLog(function() {
                            console.log('已经加入，可以开始聊天');
                        });
                    });
                });
            }
            room.receive(function(data) {
                if (!msgTime) {
                    msgTime = data.timestamp;
                }
                showMsg(data);
                //$("#chatroom_content").append("<div
class='direct-chat-msg'><div    class='direct-chat-info    clearfix'><span    class='direct-chat-name
pull-left'>"+ getCookie("chatroom_username") + "</span><span class='direct-chat-timestamp pull-right'
style='padding-right:    180px'>"+    formatTime(data.timestamp)    + "</span></div><img
class='direct-chat-img'    src='"+ getCookie("chatroom_user_url")    +' "><div    class='direct-chat-text'
style='padding: 7px;margin-right: 180px;padding-left: 15px;padding-right: 15px'>"+ data.msg.testMsg
+"</div></div>");

                var scroll = document.getElementById("chatroom_content");
                scroll.scrollTop=scroll.scrollHeight;
                console.log(' (' + formatTime(data.timestamp) + ')    ' + data.fromPeerId + ':
' + data.msg.testMsg));
            } else {
                console.log('服务器不存在这个 room');
            }
        });
    });

```

```
        }
    });
});
}
}
```

5.7.3 界面

本界面为搜索聊天室的结果界面。



图 5.10 聊天室界面

本界面为当前聊天的实际界面。

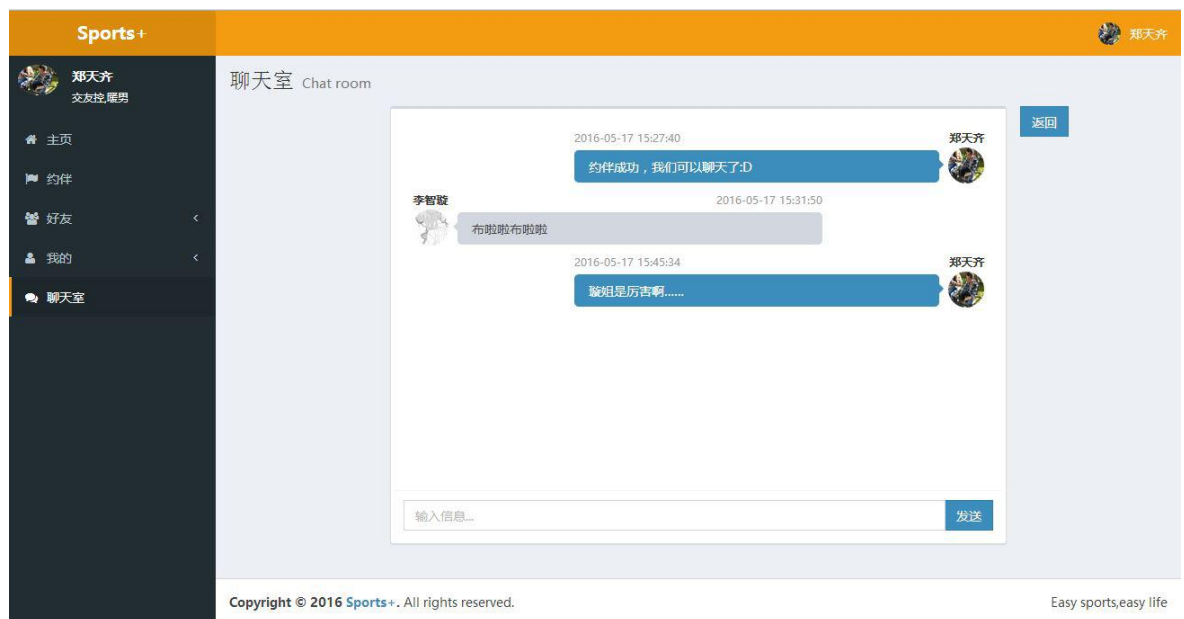


图 5.11 聊天界面

5.7.4 前端部分代码

```
<section class="content">
  <div class="row">
    <section class="col-md-12" id="main">
      <div class="row">
        <div class="col-md-8 col-md-offset-2">
          <div class="box direct-chat direct-chat-primary img-thumbnail">
            <div class="box-body" style="padding: 10px">
              <div id="chatroom_content" class="direct-chat-messages"
style="height: 400px;">
                <span id="msg_end" style="overflow:hidden"></span>

                </div><!--/.direct-chat-messages-->
              </div>
              <div class="box-footer">
                <div class="input-group">
                  <input id="send_message" type="text" name="message"
placeholder="输入信息..." class="form-control"/>
                  <span class="input-group-btn">
                    <a onclick="Send()" class="btn btn-primary btn-flat">发送
</a>

                  </span>
                </div>
              </div><!-- /.box-footer-->
            </div>
          <div class="col-md-2" style="padding: 0px;margin: 0px;text-align: left;">
            <a class="btn btn-primary btn-flat" onclick="back()">返回</a>
          </div>
        </div>
      </div>
    </section>
  </div>
</section>
```

5.8 本章小结

本章节从各个模块出发，根据每个模块的功能性及非功能性需求对其进行了详细设计和实现，并展示了部分代码及完成后的界面，以及部分相应的前端代码。

6 系统测试

软件测试是指在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否满足设计要求进行评估的过程。

本章节按照软件测试的要求，对每个模块都分别设计了具体的测试用例，并按照测试结果对其进行了记录。通过测试，能够从已经实现的网站中找出不足，予以修改和更正，能够为网站开发提供修改的帮助和依据。

6.1 登录注册模块测试

本模块的测试用例设计从登录时输入的用户名和密码的正确与否，匹配程度入手，根据不同的情况给出了测试结果；同时注册页面的测试也是如此，根据用户的输入情况进行测试，查看实现中对用户输入的检测是否有效。

测试用例如下：

表 6.1 登录测试表 1

测试用例编号	User_Login_001
测试功能	登录
重要级别	重要
预置条件	用户已注册
输入	登录账号，密码
操作步骤	输入已注册账号 输入正确密码
预期输出	登录成功，跳转至主页
实际输出	“登陆成功！”

表 6.2 登录测试表 2

测试用例编号	User_Login_002
测试功能	登录
重要级别	重要
预置条件	用户已注册
输入	登录账号，密码
操作步骤	1.输入已注册账号 2.输入错误密码
预期输出	登录失败，提示密码错误
实际输出	“密码错误！”

表 6.3 登录测试表 3

测试用例编号	User_Login_003
测试功能	登录
重要级别	重要
预置条件	用户已注册
输入	登录账号，密码
操作步骤	1.输入账号并不存在 2.输入正确密码
预期输出	登录失败，提示用户不存在，请检查账号输入或注册新账号

实际输出	“用户不存在！”
------	----------

表 6.4 登录测试表 4

测试用例编号	User_Login_004
测试功能	登录
重要级别	重要
预置条件	用户已注册
输入	登录账号，密码
操作步骤	1.输入账号并不存在 2.输入错误密码
预期输出	登录失败，提示用户不存在，请检查账号输入或注册新账号
实际输出	“用户不存在！”

表 6.5 注册测试表 1

测试用例编号	User_Reg_001
测试功能	注册
重要级别	重要
预置条件	用户未注册
输入	注册账号，密码，确认密码，真实姓名
操作步骤	1.输入 11 位数字，即手机号 2.输入密码 3.输入确认密码正确 4.输入正常姓名
预期输出	注册成功，跳转至主页
实际输出	显示注册成功提示信息，并主动跳转到个人主页

表 6.6 注册测试表 2

测试用例编号	User_Reg_002
测试功能	注册
重要级别	重要
预置条件	用户未注册
输入	注册账号，密码，确认密码，真实姓名
操作步骤	输入 11 位数字，即手机号 输入密码 3.输入确认密码与密码不符 4.输入姓名含特殊字符
预期输出	注册失败，提示密码不一致，姓名格式错误
实际输出	显示密码不一致，仍然停留在当前注册页面，不跳转

表 6.7 注册测试表 3

测试用例编号	User_Reg_003
测试功能	注册
重要级别	重要
预置条件	用户未注册
输入	注册账号，密码，确认密码，真实姓名
操作步骤	1.输入数字非 11 位或包含非数字字符 2.输入密码

	3.输入确认密码正确 4.输入姓名含特殊字符
预期输出	注册失败，提示请输入正确的手机号
实际输出	显示输入格式错误，仍然停留在当前注册页面，不跳转

表 6.8 注册测试表 4

测试用例编号	User_Reg_004
测试功能	注册
重要级别	重要
预置条件	用户未注册
输入	注册账号，密码，确认密码，真实姓名
操作步骤	1.输入数字非 11 位或包含非数字字符 2.输入密码 3.输入确认密码与密码不符 4.输入姓名含特殊字符
预期输出	注册失败，提示请输入正确的手机号
实际输出	显示输入格式错误，仍然停留在当前注册页面，不跳转

表 6.9 注册测试表 5

测试用例编号	User_Reg_005
测试功能	注册
重要级别	重要
预置条件	用户未注册
输入	注册账号，密码，确认密码，真实姓名
操作步骤	1.输入数字非 11 位或包含非数字字符 2.输入密码 3.输入确认密码与密码不符 4.输入正常姓名
预期输出	注册失败，提示请输入正确的手机号
实际输出	显示输入格式错误，仍然停留在当前注册页面，不跳转

表 6.10 注册测试表 6

测试用例编号	User_Reg_006
测试功能	注册
重要级别	重要
预置条件	用户未注册
输入	注册账号，密码，确认密码，真实姓名
操作步骤	1.输入已注册的手机号 2.输入密码 3.输入确认密码正确 4.输入正常姓名
预期输出	注册失败，提示用户已存在
实际输出	显示用户已经存在提示信息，仍然停留在当前注册页面，不跳转

表 6.11 注册测试表 7

测试用例编号	User_Reg_007
测试功能	注册
重要级别	重要

预置条件	用户未注册
输入	注册账号，密码，确认密码，真实姓名
操作步骤	1.输入已注册的手机号 2.输入密码 3.输入确认密码与密码不符 4.输入正常姓名
预期输出	注册失败，提示用户已存在
实际输出	显示用户已经存在提示信息，仍然停留在当前注册页面，不跳转

表 6.12 注册测试表 8

测试用例编号	User_Reg_008
测试功能	注册
重要级别	重要
预置条件	用户未注册
输入	注册账号，密码，确认密码，真实姓名
操作步骤	1.输入已注册的手机号 2.输入密码 3.输入确认密码正确 4.输入姓名含特殊字符
预期输出	注册失败，提示用户已存在
实际输出	显示用户已经存在提示信息，仍然停留在当前注册页面，不跳转

6.2 用户信息模块测试

本模块的测试用例设计从用户进入该模块后对个人信息的修改成功与否入手，通过进行测试并比对数据库内的相应表中的条目来查看该模块是否能够成功无误的完成预定的模块功能及目标。

测试用例如下：

表 6.13 用户信息测试表 1

测试用例编号	Person_Management_001
测试功能	修改个人资料
重要级别	重要
预置条件	用户已登录
输入	性别，标签，学校
操作步骤	1. 点击我的信息 2. 选择性别 3. 输入学校信息进行搜索 4. 选择学校 5. 勾选标签
预期输出	所有修改的信息正确显示在个人资料页面
实际输出	用户个人信息修改后未全部正确显示在个人资料页面
差距分析	无

表 6.14 用户信息测试表 2

测试用例编号	Sport_Management_001
测试功能	修改运动信息
重要级别	重要
预置条件	用户已登录

输入	运动，级别
操作步骤	1. 点击我的信息 2. 选择添加运动 3. 勾选运动 4. 选择运动等级
预期输出	所有修改的信息正确显示在个人资料页面
实际输出	用户个人信息修改后全部正确显示在个人资料页面

6.3 相册模块测试

本模块的测试用例设计从检测该模块在相应页面上是否能够正确的上传用户照片，修改用户照片，打开大图进行查看入手，通过选择上传，删除，点击页面中的缩略图查看是否能够打开大图进行查看来测试模块功能是否完成。

测试用例如下：

表 6.15 相册测试表 1

测试用例编号	User_Album_001
测试功能	上传照片
重要级别	普通
预置条件	用户已登录
输入	图片
操作步骤	选择本地文件 点击上传照片
预期输出	上传成功，可浏览
实际输出	显示“成功上传照片”提示信息，并自动跳转到“我的相册”，将最新添加的照片显示在最前面

表 6.16 相册测试表 2

测试用例编号	User_Album_002
测试功能	删除照片
重要级别	普通
预置条件	用户已登录，相册中有照片
输入	图片
操作步骤	1.选择要删除的照片 2.点击删除
预期输出	删除成功
实际输出	点击删除按钮后，会弹出再次确认是否要删除的选择框，确定删除以后显示“成功删除照片”提示信息，并重新加载删除相册的页面，刚才删除的照片已经不显示了。

6.4 好友模块测试

本模块的测试用例设计从用户是否能够正确查看自己的好友，查找符合自己要求的好友并发送好友请求入手，通过建立多个测试账号互相进行好友申请，通过，拒绝并查看好友信息及相册等信息来进行测试。

测试用例如下：

表 6.17 好友添加测试表 1

测试用例编号	Search_Add_001
测试功能	模糊搜索用户名
重要级别	普通
预置条件	用户已登录
输入	用户名或关键字
操作步骤	在搜索栏中输入关键字 点击搜索
预期输出	包含关键字的所有用户名
实际输出	显示出个人信息（姓名、手机号）中包含有搜索关键字的所有用户，同时如果符合搜索条件也会显示用户自己，但无法发送好友请求。

表 6.18 好友添加测试表 2

测试用例编号	Search_Add_002
测试功能	申请加好友
重要级别	普通
预置条件	用户已登录
输入	申请添加
操作步骤	选择要添加的用户 点击加为好友
预期输出	好友申请已发送，等待回复
实际输出	点击添加好友按钮会显示“好友请求已成功发送给***”提示信息，如果已经发送过请求并且对方没有处理请求信息的话，再次发送申请会显示“已经发生过请求”提示信息。

表 6.19 好友管理测试表 1

测试用例编号	Friend_Management_001
测试功能	查看好友
重要级别	重要
预置条件	用户已登录
输入	查看好友
操作步骤	点击好友列表
预期输出	显示用户所有好友
实际输出	显示用户的所有好友信息，并且点击用户头像可以查看好友个人主页。

表 6.20 好友管理测试表 2

测试用例编号	Friend_Management_002
测试功能	删除好友
重要级别	重要
预置条件	用户已登录，已有好友
输入	查看好友
操作步骤	点击好友列表 选择要删除的好友 点击删除
预期输出	删除选中的好友
实际输出	点击删除好友按钮以后会显示再次确认删除的选择框，确认删除后会显示“成功删除好友”提示信息，并且刷新我的好友界面，刚才删除的好友已经不再显示。

表 6.21 好友管理测试表 3

测试用例编号	Friend_Management_003
测试功能	同意好友申请
重要级别	重要
预置条件	用户已登录，收到好友申请
输入	查看好友申请
操作步骤	选择要处理的申请并点击同意
预期输出	好友添加成功，好友列表中显示该好友
实际输出	点击同意按钮以后会显示“成功添加好友”提示信息，并且刷新该好友请求页面，刚才处理的那条好友请求信息已经不会再显示了。

6.5 约伴模块测试

本模块的测试测试用例设计从用户是否能够在对应页面上正确搜索到符合自己要求的约伴目标，正确发送约伴请求，正确处理接收到的约伴请求入手，通过多个测试用户之间相互发送月半请求，拒绝，接受等操作并检查是否正确产生了相应的相应结果来进行测试。

测试用例入下：

表 6.22 约伴测试表 1

测试用例编号	User_Invitation_001
测试功能	发送邀请
重要级别	重要
预置条件	用户已登录
输入	约伴邀请
操作步骤	1.选择约伴功能 2.选择运动种类（下拉框） 3.系统自动筛选符合条件的用户 4.选择用户，点击“我要约”
预期输出	提示请求发送成功
实际输出	点击我要约按钮后显示“请求发送成功”提示信息，并仍然停留在改页面，不刷新。

表 6.23 约伴测试表 2

测试用例编号	User_Invitation_002
测试功能	接收邀请
重要级别	重要
预置条件	用户已登录并受到运动邀请
输入	处理约伴邀请
操作步骤	1.点击约伴邀请 2.选择消息，点击同意
预期输出	提示约伴成功
实际输出	显示“约伴成功”提示信息并刷新该请求页面，刚才处理的信息已经不再显示。

表 6.24 约伴测试表 3

测试用例编号	User_Invitation_003
测试功能	拒绝邀请

重要级别	重要
预置条件	用户已登录并受到运动邀请
输入	处理约伴邀请
操作步骤	1.点击约伴邀请 2.选择消息，点击拒绝
预期输出	提示已拒绝邀约
实际输出	显示“取消约伴”提示信息并刷新该请求页面，刚才处理的信息已经不再显示。

6.6 在线聊天模块测试

本模块的测试用例设计从用户是否能够正常接受聊天室请求，进入聊天室后是否能够正常进行在线聊天，删除及删除提示是否正常入手，通过测试账号之间互相接受约伴请求并进入聊天室，点击删除按钮，发送聊天信息进行在线聊天查看是否能够正常收到在线、离线消息。

测试用例如下：

表 6.25 在线聊天测试表 1

测试用例编号	User_Conversation_001
测试功能	接受聊天
重要级别	重要
预置条件	用户已登录，受到运动邀请并同意
输入	聊天
操作步骤	点击已经同意的邀约 跳入聊天室
预期输出	跳出聊天对话框，可以进行聊天
实际输出	点击相应的聊天室，会进入到该聊天室，显示聊天内容（自动保存最后 20 条历史记录）。

表 6.26 在线聊天测试表 2

测试用例编号	User_Conversation_002
测试功能	删除聊天
重要级别	重要
预置条件	用户已登录，受到运动邀请并同意
输入	聊天
操作步骤	1.点击已经同意的邀约 2.点击删除聊天 3.点击确定
预期输出	提示聊天已删除，刷新后记录已删除
实际输出	点击减号按钮（删除）后，弹出确认删除选择框，确认删除后显示“成功结束聊天”提示信息并刷新聊天室列表页面，刚才删除的聊天室已经不再显示。

表 6.27 在线聊天测试表 3

测试用例编号	User_Conversation_003
测试功能	取消删除聊天
重要级别	重要
预置条件	用户已登录，受到运动邀请并同意
输入	聊天

操作步骤	1.点击已经同意的邀约 2.点击删除聊天 3.点击取消
预期输出	提示已取消删除，刷新后记录任存在
实际输出	点击减号按钮（删除）后，弹出确认删除选择框，取消删除后刷新聊天室列表页面，聊天室仍然显示。

6.7 本章小结

本章根据模块功能性需求及部分非功能性需求对各个已完成实现的模块进行了有计划的测试，通过设计符合需求的测试用例并完成测试对已完成的功能进行了评估，章节中给出了详细的测试用例表。

7 结论与展望

7.1 结论

在 NetBeans 上，以 LeanCloud 云引擎及其 API 为基础，依托 HTML 超文本标记语言与 JavaScript 脚本语言完成了本网站。网站基本上实现了预期的各项功能，达到了任务书中的主要设计内容的各项任务，整个网站也在经过测试和不断地改正之后能够顺利运行。基本上实现了开题报告中的需求设计在程序编写，该网站能够实现如下功能：

（1）用户登录注册

用户能够使用网站提供的界面进行账号的注册和网站的登录，这是一个社交类型网站最基本的功能要求，针对不同的错误情况，网站会给予用户相应的提示便于修正。

（2）用户信息管理

对于网站来说，用户能够通过信息展示展示自我，找到一同运动的伙伴要求网站能够显示其自定义的基本信息。用户在登录网站后，能够获得对自己信息进行修改的权限，用户能够针对自己的姓名，性别，学校，标签及运动信息进行修改和保存，同时也能上传新的头像并保存。

对于学校，管理员能够在后台对其进行动态添加以补充目前没有出现数据库中的学校校区，用户在补充之后通过模糊搜索即可获得自己的校区信息并进行选择。

（3）相册管理

本网站为用户提供了一个便利易用的照片展示平台，用户登录后会获得修改自己相册的权限，用户能够通过网站简介易懂的界面来选择上传照片，删除照片或查看好友相册等功能，但是用户没有修改好友相册权限。

（4）好友管理

社交网站的重点之一便是好友功能，本网站通过简单的界面允许用户快速通过姓名中的关键字查找自己想要找到的朋友并方便的给他们发送好友请求，同时也能够通过好友列表功能查看好友的信息，管理自己的好友。

在好友申请中，网站允许用户决定是否添加对方为自己的好友，也可以随时从自己的好友列表中删除自己不想再见到的好友。

（5）约伴管理

本网站的另一重点便是约伴功能，用户在登录后可以通过选择运动类型的方式快速搜索到拥有同样爱好的注册用户，并根据结果显示的基本信息向自己感兴趣的用户发送约伴请求。

用户在处理约伴请求的过程中可以选择接受也可以选择拒绝，在被拒绝后可以再次约伴，在接受请求后网站会自动为约伴的双方创建一个聊天室，只要聊天室存在，则视为约伴未结束，直到聊天室被删除，双方才能够通过网站再次约伴。

（6）在线聊天

约伴双方能够在网站为其创建的聊天室中进行实时聊天，也可以在对方不在线时进行留言，不在线一方上线后即可在聊天室内接收到对方的留言，这一功能是网站为了方便约伴双方及时进行沟通提供的，能够有效解决没有交流方式的尴尬。

7.2 展望

网站在开发过程中使用到的云引擎 API 只是其中的极小一部分，其中绝大多数的 API 提供了与其他平台互通的功能，如果可以再通过相同的引擎开发 iOS 或 Android 客户端便可以有效的扩大用户群。同时在开发过程中 JavaScript 脚本语言的回调运行方式使得一些功能在实现上存在困难，同时 LeanCloud 数据库后台逻辑中禁止了双向 relation 关系的建立，如果能够找到方法解决回调和 relation 关系的双向建立困难，那么网站的好友功能就可以提供通过申请双方互相加为好友的更加人性化的模式。作为一个轻量化开发的社交网站，网站还存在部分界面没有适配的问题，现存 CSS 文件无法做到在网页上适配缩略图到合适的尺寸，如果后续能够有新的 CSS 配置文件发布就可以解决这个问题了。

其次，本网站只针对大学生的运动健康状况进行了开发，运动种类也较少，随着人们运动及健身方式的不断增加，网站也应当与时俱进，通过不断扩展新的功能比如体感技术等新兴技术来使得用户的运动选择更加的多样化，约伴的难度进一步降低，同时与其他社交平台进行互通，扩大用户群和约伴范围，形成一个依托于社交网络的运动约伴平台，为提高大学生身体素质贡献一份力。

参考文献

1. 《JavaScript 学习指南》；（美）鲍尔斯；2009.10.1；ISBN：9787115214041
2. 《jQuery 高级编程》；[美] 奥特罗；[美] 劳伦斯；2013；ISBN：9787302317845
3. 《Head First HTML with CSS and XHTML》；Elisabeth Freeman；2005.12；ISBN：9780596101978
4. 《Web 开发入门经典:使用 PHP6、Apache 和 MySQL》；薛焱；2009.7；ISBN：9787302203759
5. 《HTML5 从入门到精通》；明日科技；2012.9；ISBN:9787302287582
6. 《JavaScript 实战手册》；（美）麦克法兰德；2009.5.1；ISBN:9787111258674
7. 《CSCE: A Crawler Engine for Cloud Services Discovery on the World Wide Web》；Talal H. Noor; Quan Z. Sheng; Abdullah Alfazi; Anne H. H. Ngu; Jeriel Law; 2013.6.28; ISBN: 978-0-7695-5025-1
- 8.《Improving Precision of Java Script Program Analysis with an Extended Domain of Intervals》; Astrid Younang; Lunjin Lu2015.7.1; INSPEC Accession Number:15476110
9. 《An automated change-detection algorithm for HTML documents based on semantic hierarchies》；Seung-Jin Lim; Yiu-Kai Ng; 2001; ISBN:0-7695-1001-9
10. 《中国青少年体育发展报告（2015）》；2015.8.15

谢 辞

随着毕业设计的工作接近尾声，我的大学生活也即将结束。在这四年中我从一个一问三不知的普通高中生成长为一个具有良好专业素养的软件工程专业毕业生离不开学院老师和前辈们的无私教诲和指导，正如我选择软件学院时所想的一样，既然选择，便不后悔。事实证明我的付出最终收获了完美的回报。感谢四年来给予我帮助和鼓励的所有人，我从你们身上获得了不同的知识，全新的理念，积极地态度，给了我不断前进的动力和目标。

感谢唐剑锋老师，在我的毕业设计过程中无私认真的指导，老师总能够及时与我进行沟通，从选题到具体的设计实现，论文的撰写过程中老师都提出了中肯且切合实际的建议，在我碰到困难挫折时鼓励我，给予我无私的帮助，最终成功完成了毕业设计和论文的撰写。老师认真的态度和乐观的心态也是我一直学习的地方，希望在未来的求学路途上我能够从唐老师身上汲取这些优秀的品质并一以贯之的保持下去。

感谢软件学院的其他老师，你们无私的付出不但教会了我专业的知识，也教导了我为人处世的道理。

感谢室友，让我在四年的大学生活中总是充满了欢声笑语，留下了美好的回忆。

感谢家人，你们的支持是我最大的动力，让我无论何时都能从你们身上汲取温暖，找到归宿感。