

南京信息职业技术学院

毕业设计论文

作者 李大祥 学号 41312P26

系部 计算机与软件学院

专业 计算机应用技术

题目 爱驿站婚恋平台的研究与实现

指导教师 孙仁鹏

评阅教师

完成时间： 2016 年 5 月 14 日

毕业设计(论文)中文摘要

(题目): 爱驿站婚恋平台的研究与实现

摘要: 本次毕业设计是在徐州凡几信息科技有限公司的业务需求背景下,探索一种新型的线下实名认证的婚恋网站平台实现,用真实可靠的相亲模式力求满足当前和未来国内巨大的婚恋相亲市场需求。论文首先论述了爱驿站出现的背景,当前婚恋网站的商业模式和意义,然后详细描述了爱驿站的业务逻辑,并以此为基础设计了网站程序的架构,从而实现以店铺、会员、用户三者为业务核心的婚恋网站交互平台。爱驿站平台能够很好的支撑凡几科技新型的业务需求,O2O模式的实名认证婚恋平台将来都有着巨大的发展潜力和远景。

关键词: PHP O2O 相亲 Thinkphp

毕业设计(论文)外文摘要

Title: Research and Implementation Of Ayizhan Marriage Platform

Abstract: The graduation project is in the business background of the demand Xuzhou where Fanji Technology, Ltd., to explore a new line under the real-name authentication platform dating sites. Where Fanji seeks to use authentic matchmaking marriage strive to meet the huge domestic market demand for current and future blind. Paper first discusses the background of love post appeared, meaning the current business model and dating sites, and then describes in detail the business logic of love inn, as the basis for the architecture of the web application, enabling to store, members, three users those for the core business of dating websites interactive platform. Ayizhan can be a good platform to support science and technology where Fanji new business needs, the real-name authentication platform O2O mode of marriage in the future have a huge potential for development and prospects.

keywords: PHP O2O blind Thinkphp

目录

1 引言.....	1
1.1 课题背景	1
1.2 名词释义	1
1.3 国内外发展现状	2
1.4 课题研究目的和意义	3
1.5 主要内容	3
2 爱驿站业务分析.....	4
2.1 普通用户相关.....	4
2.2 驿站会员相关.....	5
2.3 店铺相关.....	5
2.4 管理员相关.....	6
2.5 本章小结.....	7
3 数据库设计.....	7
3.1 数据库概念分析	7
3.2 数据库设计和生成	9
4 爱驿站平台实现.....	12
4.1 开发环境	12
4.2 系统设计概述.....	13
4.3 关键问题解决方案	14
4.3.1 时区配置	14
4.3.2 图片轮播	14
4.3.3 上传固定格式的图片	15
4.3.4 分页技术	16
4.4 模块划分.....	17
4.4.1 主页模块.....	17
4.4.2 搜索模块.....	18
4.4.3 我的驿站模块.....	21
4.4.4 活动模块.....	21
4.4.5 快报模块.....	24
4.4.6 法律模块	24
4.4.7 店铺模块	25
4.4.8 管理员模块.....	26
5 网站测试和优化.....	27
5.1 功能测试	27
5.2 网站部署	27
5.3 性能与安全优化.....	27
5.3.1 性能优化	27

5.3.2 网站安全优化	28
结论.....	29
致谢.....	30
参考文献.....	30

1 引言

1.1 课题背景

根据国家统计局公布的数据，2014 年末，男性人口 70079 万人，比女性多 3376 万，1980 年到 2014 年期间诞生了近 3000 万“剩男”，婚嫁问题已经成为一个严重的社会问题。婚姻感情状况不稳定，严重影响了个人的心理健康和事业发展，进而对社会产生不良影响。而改革开放以来，随着社会主义现代化、城市化的不断推进，都市中的单身青年男女往往受困于工作的繁忙和沟通渠道的匮乏，互相交流的机会大大减少。这也间接导致了即便相亲成功率也不高的现状，降低了男女恋爱直至结婚组成家庭的可能性。

当过去时代的相亲和红娘媒婆不能满足当前社会的问题需求时，我们需要用当代新的技术、新的思想来解决。上世纪 90 年代以来的互联网技术蓬勃发展，基本实现了跨越空间地理的无障碍交流，相亲行为结合互联网技术诞生的婚恋网站为这一问题的解决带来希望。

目前，婚恋网站一般分为两种类型：一种是以美国的 Match.com 和国内世纪佳缘网为代表，它们主要以搜索好友和线上发信交流为主，进而促进线下交往和恋爱结婚；另一种则是以美国 eHarmony.com 为代表，它们以严格的心理学性格测试为标准，精准推销适合会员的交往对象，从而帮助人们拓展交际圈和找到心仪对象。

1.2 名词释义

O2O: 即把线下商务机会与互联网结合在一起，让互联网成为线下交易的前台的商业模式。

PHP: 一种通用开源脚本语言。语法吸收了 C 语言、Java 和 Perl 的优点特性，利于学习，使用广泛，主要适用于 Web 开发领域。

HTML: 即超文本标记语言，标准通用标记语言下的一个应用。目前最新版本是 HTML5。“超文本”主要指页面内可以包含图片、链接、音乐等非文字元素。

JavaScript: 一种动态类型、弱类型、基于原型的广泛用于客户端的脚本语言，内置支持类型。它的解释器已经成为浏览器的组成部分，被称为 JavaScript 引擎。

早期主要用来给 HTML 网页增加动态功能。

CSS: 即层叠样式表, 一种用来表现 HTML 或 XML (标准通用标记语言的一个子集) 等文件样式的计算机语言。

Thinkphp: 一个免费开源的, 快速、简单的面向对象的轻量级 PHP 开发框架, 广泛应用于中小企业和个人开发者中。

Apache: 全称是 Apache HTTP Server, 世界上最流行的 Web 服务器端软件之一, 由于其良好的跨平台支持和安全性被广泛使用。

MySQL: 目前最流行的关系型数据库管理系统之一, 尤其在 WEB 应用方面表现良好, 目前属于 Oracle 旗下公司。

凡几科技: 徐州凡几信息科技有限公司的简称。

爱驿站: 网址是 www.ayizhan.com, 一家连锁婚恋服务平台, 利用 O2O 模式整合线下门店, 统一服务模式, 旨在帮助所有适龄未婚男女解决相亲难问题。即凡几科技的旗下网站产品。

驿站会员: 本文中特指在爱驿站指定合作店铺中完成实名验证和登记的会员, 参与网站中的活动展示和被搜索业务。爱驿站将帮助其找到相亲的潜在对象。

普通会员: 本文中特指在爱驿站中注册的用户。

普通用户: 本文中指任何访问和使用网站服务的用户。

店铺: 本文中特指与凡几科技签订协议, 在有效期内, 开展相关婚恋实名验证和登记等业务的店家, 凡几科技实际操作中主要与婚纱摄影店合作。

1.3 国内外发展现状

互联网技术最早发源于美国。借此优势, 婚恋网站也最早起源于这里。

Match.com 和 eHarmony.com 是目前国外最具代表性的婚恋网站, 同样都是美国孕育的商业模式。Match.com 于 1995 年上线, 是全球最早创立的婚恋交友网站, 也是如今美国访问人数最多的婚恋网站。主要运营模式是通过线上通信聊天促进沟通, 进而获得更多线下交友机会。现在其注册用户已经超过 1500 万, 面向全球提供至少 15 种语言服务。eHarmony.com 则是由一位心理学博士于 2000 年创立的, 致力于促成以结婚为交往目标的严肃关系。目前已经成为美国最大的婚恋网站之一。它的模式是通过性格测试来进行婚恋匹配, 它将根据新用户回答设定的数百个问题的答案, 判断其个性的不同维度, 并基于此给用户介绍合适的交往

对象。

在美国婚恋网站发展的早期，同一时代的国内的婚恋网站发展状况则由于电脑普及率低、经济条件、社会习俗约束、支付条件不成熟等原因，一直到 2005 年左右才开始渐渐发展起来。国内最大的几家婚恋网站目前主要包含珍爱网、百合网、世纪佳缘等。三家分别定位于人工红娘、婚恋匹配和搜索+发信三种模式。珍爱网截至目前已经有 9000 万注册用户，百合网则有 9500 万，世纪佳缘有 1.35 亿用户，其他的多数婚恋网站则已经逐渐转型或遭遇到资金瓶颈。

虽然已经出现的众多婚恋网站，在社会上产生了较大的影响，一定程度上解决了现有的婚恋交友问题，但是也产生了一些严重的信任危机事件。国外婚恋网站中就出现了部分用户被诈骗集团骗取情财的案例。而相对于国外骗局事件是由于用户缺乏自我保护和防范意识，国内婚恋网站产生的严重问题却更多源自于自身的机制缺陷和不自律，进而发生了众多故意泄露用户资料、会员信息虚假、婚恋骗局等恶性事件。尽管众多婚恋网站采取了所谓的实名制、作出承诺，但是由于企业机制建设不健全、婚恋网站不自律、用户缺乏自我保护和防范意识，我们看到的现况就是国内婚恋网站实名制名存实亡、信任危机事件频发。

1.4 课题研究目的和意义

根据关键词“婚恋网站平台”在知网中查询，结果显示有 118 条期刊和学位论文。其中绝大多数论文倾向于研究当前婚恋网站的缺陷问题和商业模式，而缺少缺乏理论和实践的结合，目前国内的研究并没有大量该方面的全面研究。

鉴于国内婚恋网站存在的种种问题，我认为有必要对婚恋网站市场和新型的解决方法做深入的研究。爱驿站平台是凡几科技公司的推出的婚恋网站平台，致力于解决单身青年婚嫁问题。爱驿站平台应用现在流行的 O2O 概念，线下联合实体店铺，通过店铺验证用户信息真实性，免费为用户登记，对所有网站访客开放搜索服务，以提高用户资料的可信度和相亲过程的成功率，并达到积累用户资源和获取流量入口的目标，为公司后期拓展电子商务和广告业务作铺垫。

1.5 主要内容

本文针对婚恋网站的发展现状，结合徐州凡几信息科技有限公司的 O2O 婚恋网站商业模式，对爱驿站平台的运营和实现进行了深入的研究和详细的论述。

论文首先从国内外婚恋网站的发展和现状进行探讨，阐释了对爱驿站网站的

研究目的和意义。其次从四个用户群体的角度全面分析了爱驿站网站的业务逻辑。然后从数据库设计到程序模块划分，详细的论述了爱驿站网站系统的实现，最后部分是论述了网站发布上线以及之后的性能调优和安全防护工作。

2 爱驿站业务分析

2.1 普通用户相关

普通用户是指访问爱驿站网站的访客和在爱驿站网站上注册的用户，他们使用爱驿站提供的相关搜索等业务，满足他们自身的相亲需求。他们是爱驿站平台最广泛的组成部分，也将是未来凡几科技长远规划中的重要受众。

普通用户享受免费自由搜索相亲对象的服务，可以在爱驿站中查阅关于爱驿站的相关资料，获取使用方法和流程，可以获得最新的快报信息。同时，注册之后，普通用户也可以称之为普通会员，他们另外还享有参加和报名线下活动的权利，将会拥有自己的个人信息管理空间。但是他们的注册资料并不会被公开，爱驿站和凡几科技也不会不经授权就使用其他方式收集他们的个人隐私信息。如果用户想要联络心仪对象，则可以在驿站会员详情页面里面查看其所属店铺的联系方式，通过店铺介绍获得该会员的联系方式和其他介绍。

普通用户的主要相关事务如图 2-1。

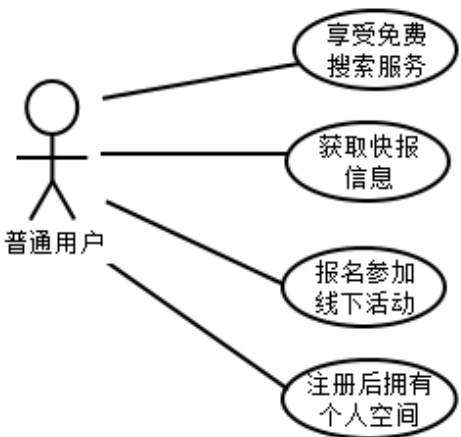


图 2-1 普通用户用例图

普通用户是网站规划中预期人数最多的一种角色，他们中的部分人更是爱驿站平台的驿站会员。他们是使用网站最频繁的人群，是整个爱驿站平台的发展动力。他们在凡几科技长远规划中，还将会成为未来爱驿站电子商城的主要消费者，为整个平台生态的不断发展注入源源不断的生机。

2.2 驿站会员相关

驿站会员是指在凡几科技指定的线下实体店铺实名验证后登记注册的会员，区别于仅仅只在爱驿站网站线上注册的普通会员和普通用户。多数情况下，驿站会员同时也是爱驿站平台的普通用户，甚至同时会注册成为普通会员。普通会员和驿站会员的信息会被分开登记和存储，以确保用户信息的安全。优质会员指的是在驿站会员中的部分付费会员，优质会员会被展示在爱驿站网站的首页上面。。

驿站会员选择登记之后，他们的非隐私资料会被部分公开展示在爱驿站平台上，这使得他们很容易地被普通用户搜索到，进而可以获得更多相亲机会。而他们的个人隐私资料中，联系方式、身份证号码、真实姓名等将会被隐藏。

驿站会员的主要相关事务如图 2-2。

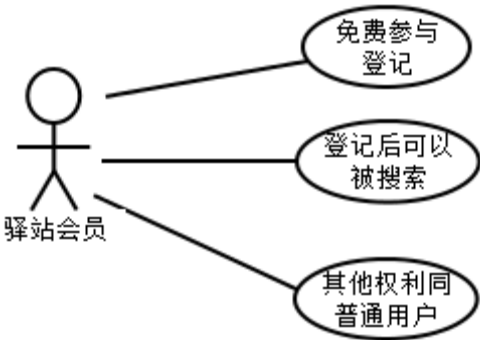


图 2-2 驿站会员用例图

驿站会员是爱驿站平台的基石，他们是网站最基础和最重要的信息来源。驿站会员的数量增长，意味着爱驿站平台的发展良好和前景广阔。更多的优质的驿站会员是爱驿站能够吸引更多普通用户使用的根本原因。有了大量的驿站会员吸引，加之庞大的普通用户群体支撑，才能构建起良好的爱驿站平台生态，才能为线下免费登记的店铺提供回报，才能支撑起一个大型的电子商城和产生广告流量利润。

2.3 店铺相关

店铺在初期规划中主要是当地的一家或几家婚纱摄影店。凡几科技选择他们作为合作店铺。他们将负责为前来登记的驿站会员验证身份和信息真实性，把信息录入系统，完成驿站会员的注册。同时，后面还会核对前来寻找自己店铺所属驿站会员的普通用户的真实身份，并给两者提供见面场所或给普通用户提供经过驿站会员授权的个人信息。但是并没有设置优质会员的权限。

店铺一旦合作即享有一个独立的店铺编号和宣传空间，拥有一个独立的管理后台。店铺主要收利是可以参与分享爱驿站中庞大的会员线下资源，通过店铺结识的相亲对象，无形中将会免费给店铺带来一次深刻的广告宣传；同时，爱驿站平台也给店铺提供了线上的宣传途径。这将会极大提升店铺的知名度和业绩。

店铺的主要相关事务如图 2-3。

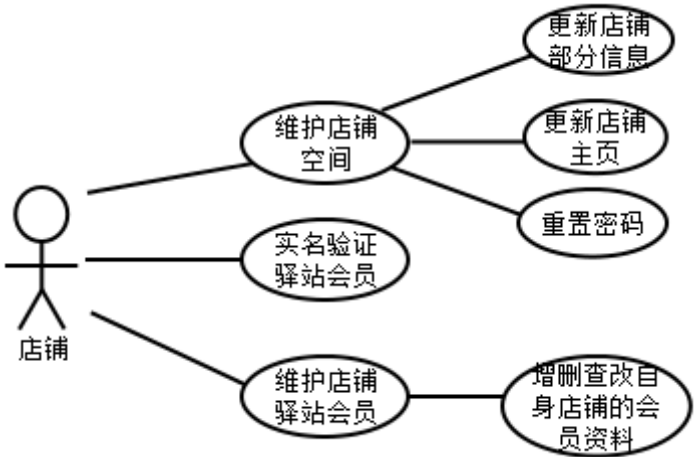


图 2-3 店铺用例图

店铺是用户和凡几科技的连接点，是区别普通用户和驿站会员的重要中转站，店铺验证用户信息后将信息转移到凡几科技的爱驿站平台中。同时，凡几科技通过联合店铺将边缘繁琐的实名制验证业务剥离出去，简化了自身工作。店铺由此成为爱驿站平台的重要组成部分，支撑着爱驿站平台的快速推广和发展。

2.4 管理员相关

管理员的角色主要是由凡几科技的公司人员负责。管理员的权限是最大的，可以管理所有的普通会员、驿站会员、店铺、活动和快报等信息。

管理员的权限较大，主要管理所有注册的普通用户、所有店铺和所有店铺的驿站会员。同时，还能对付费的驿站会员设置优质会员属性，以显示在首页，增加其相亲机会。另外，还可以对爱驿站平台的活动和快报进行发布、更新、设置推荐榜等。

管理员的主要相关事务如图 2-4。

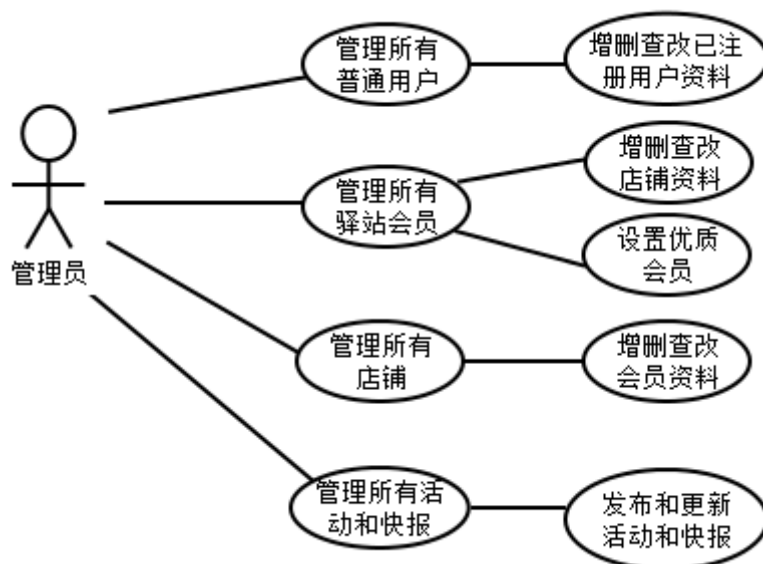


图 2-4 管理员用例图

2.5 本章小结

本章从爱驿站平台的四个主体组成元素角度，概述了爱驿站平台的业务运营情况。凡几科技与线下实体店铺合作组成运营主体，通过店铺实现对驿站会员的实名验证和保护业务，以爱驿站网站作为店铺、驿站会员、普通用户的沟通纽带和信息沟通平台，提供给广大单身青年男女真实可靠广阔的相亲介绍服务，同时帮助提升店铺影响力和形象、拓展客户。当用户资源积累到一定程度，爱驿站平台运行良好，凡几科技将会以此为基础打造以婚恋为主题的电子商城，并陆续开展广告等业务获取收益。

3 数据库设计

3.1 数据库概念分析

数据库是动态网页技术的基础，区别于静态网页技术，是 Web2.0 技术里不可或缺的一部分。数据库管理系统可以将数据持久地保存在硬盘上的数据库中，方便程序读取和更新。现在常见的数据库主要分为关系型数据库和非关系型数据库。前者适合存储一致性较强的结构化数据，而后者则更容易拓展和存储非结构化数据。

爱驿站平台的设计采用了传统的关系型数据库。爱驿站平台中涉及到的实体主要有普通会员、驿站会员、店铺、管理员、快报和活动。

普通用户实体指的是在爱驿站中注册的用户，也可以称之为普通会员。普通

用户的属性包括：用户系统编号、用户姓名、登录密码、用户性别、年龄、出生日期、手机号码、家庭住址、头像保存路径、创建时间、收藏会员编号、点赞次数、是否允许登陆。

驿站会员实体的属性如包括：系统编号、用户名称、登录密码、性别、年龄、出生日期、身份证号、手机号码、家庭住址、详细地址、婚姻状态、等级、厨艺水平、是否有房子、是否有车、身高、体重、月薪、生肖、备用手机号码、固定电话号码、QQ 号、家庭成员情况、职业、民族、毕业学校、受教育水平、血型、对做家务态度、是否介意和父母一起居住、是否愿意做上门女婿、约会时间安排、是否吸烟、是否喝酒、是否有驾照、个人简介、头像保存路径、创建时间、所属店铺编号、是否允许登录、相册保存路径、计划结婚时间、被举报信息、被点赞次数、详情页面被查看次数、是否是优质会员。

店铺实体的属性有：店铺系统编号、登录密码、店铺主人名称、店铺名称、店铺地址、店铺联系电话、店铺备注、店铺头像保存路径、创建时间、店铺主页信息、店铺背景图保存路径。

管理员实体的属性包括：系统编号、管理员名称、登录密码。

快报实体的属性有：快报系统编号、快报创建人名称、快报题目、快报发布时间、快报简介、快报主题图片保存路径、创建时间、快报内容、被点击次数。

活动实体的属性有：系统编号、活动创建人名称、活动名称、活动组织人名称、活动开始时间、活动报名截至日期、活动举办地点、活动限制人数、联系电话、活动主题图片保存路径、创建时间、报名人的系统编号集合、被点击次数、报名人数中男性人数、报名人数中女性人数、活动内容。

使用 Pencil 软件画的简略 ER 图如下图 3-1。

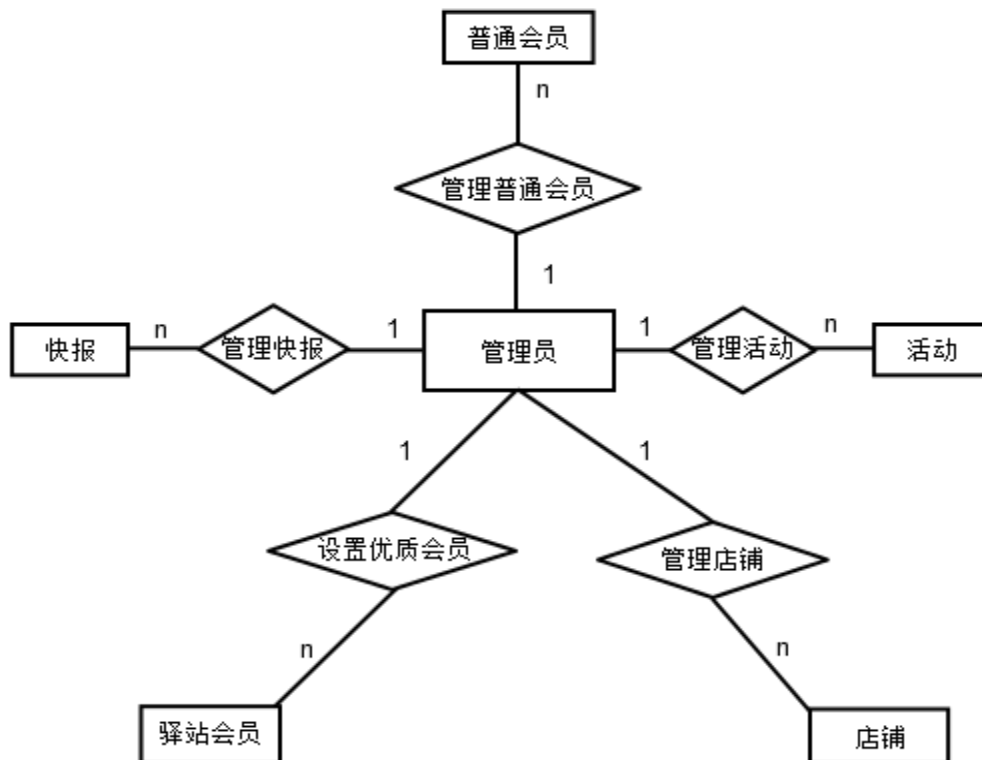


图 3-1ER 图

3.2 数据库设计和生成

数据库设计中，不仅需要对数据库需求有详细的分析后，而且在后期设计物理模型中需要符合很多规范要求，以减少数据冗余和降低发生操作异常的可能性，达到优化数据库的目的。这些有着一定第次顺序的规范要求一般被称为范式。范式越高，数据冗余越少，但是会造成更多的联合查询，可能会带来查询效率的大幅降低。所以，实际企业应用中一般会综合考虑性能表现和业务需求。爱驿站中的数据库设计，考虑到实体的联系比较简单并且要求满足查询性能需要，所以范式只是应用到第二范式。

在 3.1 章节中实际上已经对数据库进行了一次优化，按照 ER 图设计的物理模型已经基本符合了第一范式的要求。这是对关系数据库设计的基本要求。第二范式是在第一范式基础上建立起来的，要求实体的属性必须对主键完全依赖。爱驿站的数据库设计也是基本符合第二范式的。另外，我们为了简化逻辑和提升程序性能，特别把快报和活动中的推荐榜信息剥离出来，另外组建两个表。

根据范式优化过数据库设计后，就可以根据 ER 图和资料建立数据库物理模型。同时，物理模型图 3-2 中还标注了所有表的字段信息，展示了每个表的详细

结构。

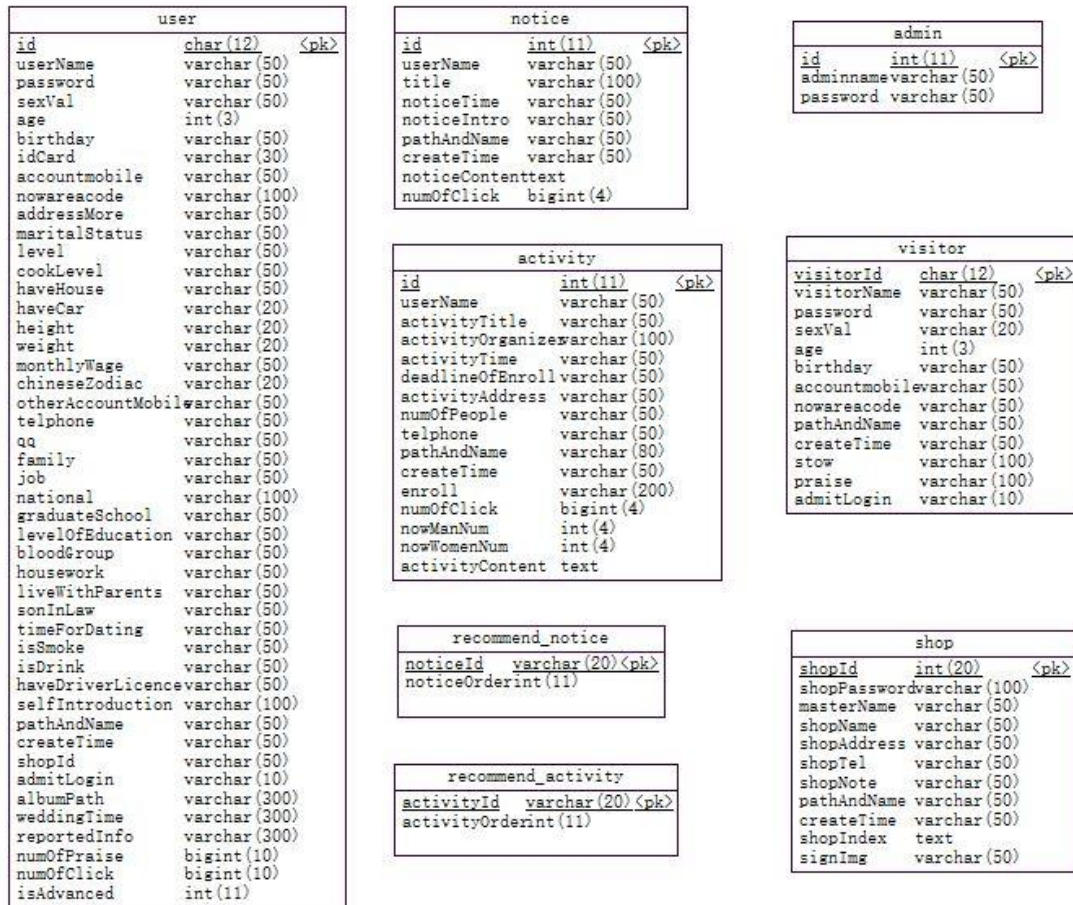


图 3-2 物理模型图

物理模型的设计完成标志着数据库设计工作的结束。根据物理模型可以写出 SQL 语句，建立相应的数据库和表。立表的 SQL 语句如下。

```
CREATE TABLE visitor(
visitorId char(12) primary key,
visitorName varchar(50),
password varchar(50),
sexVal varchar(20),
age int(3),
birthday varchar(50),
accountmobile varchar(50),
nowareacode varchar(50),
admitLogin varchar(10) default 'admit',
pathAndName varchar(50),
createTime varchar(50),
stow varchar(100),
praise varchar(100)
);
```

```

CREATE TABLE user(
  id char(12) primary key,
  userName varchar(50),
  password varchar(50),
  sexVal varchar(50) ,
  age int(3),
  birthday varchar(50) ,
  idCard varchar(30),
  accountmobile varchar(50) ,
  nowareacode varchar(100),
  addressMore varchar(50),
  maritalStatus varchar(50),|
  level varchar(50),
  cookLevel varchar(50),
  haveHouse varchar(50),
  haveCar varchar(20),
  height varchar(20),
  weight varchar(20),
  monthlyWage varchar(50),
  chineseZodiac varchar(20),
  otherAccountMobile varchar(50) ,
  telephone varchar(50) ,
  qq varchar(50) ,
  family varchar(50) ,
  job varchar(50) ,
  national varchar(100),
  graduatesSchool varchar(50),
  levelOfEducation varchar(50),
  bloodGroup varchar(50),
  housework varchar(50),
  liveWithParents varchar(50),
  sonInLaw varchar(50),
  timeForDating varchar(50),
  isSmoke varchar(50),
  isDrink varchar(50),
  haveDriverLicence varchar(50),
  selfIntroduction varchar(100),
  pathAndName varchar(50),
  createTime varchar(50),
  numOfClick bigint(10) default 0,
  numOfPraise bigint(10) default 0,
  albumPath varchar(300),
  reportedInfo varchar(300),
  weddingTime varchar(300),
  shopId varchar(50),
  isAdvanced int,
  admitLogin varchar(10)
);

```

```

CREATE TABLE shop(
  id int primary key not null auto_increment,
  shopId varchar(50) not null,
  shopPassword varchar(100) not null,
  shopName varchar(50) not null,
  masterName varchar(50) not null,
  shopAddress varchar(50) not null,
  shopTel varchar(50) not null,
  shopNote varchar(50) not null,
  pathAndName varchar(50) ,
  createTime varchar(50),
  signImg varchar(50),
  shopIndex text
);

```



```

CREATE TABLE notice(
id int primary key,
userName varchar(50),
title varchar(100),
noticeTime varchar(50) ,
noticeIntro varchar(50),
pathAndName varchar(50),
createTime varchar(50),
numOfClick bigint(4) default 0,
noticeContent varchar(50)
);

CREATE TABLE recommend_notice(
noticeId varchar(20) primary key,
noticeOrder int
);

CREATE TABLE activity(
id int primary key ,
userName varchar(50),
activityTitle varchar(50),
activityOrganizer varchar(100),
activityTime varchar(50),
deadlineOfEnroll varchar(50) ,
activityAddress varchar(50),
numOfPeople varchar(50),
telephone varchar(50),
activityContent text,
pathAndName varchar(80),
numOfClick bigint(4) default 0,
nowManNum int(4),
nowWomenNum int(4),
enroll varchar(200),
createTime varchar(50)
);

CREATE TABLE recommend_activity(
activityId varchar(20) primary key not null,
activityOrder int
);

CREATE TABLE admin(
id int primary key,
adminname varchar(50) not null,
password varchar(50) not null
);

```

4 爱驿站平台实现

4.1 开发环境

开发环境主要围绕后台开发语言 PHP 而选定的。选择 PHP 是因为它的开源免费特性，而且 PHP 语言的开源第三方库很多，方便程序后期学习使用和功能拓展。数据库则选择了开源的社区版 MySQL。MySQL 是近年来最流行的开源关系型数据库，其性能和可拓展性无需质疑。与这两者最搭配的 Web 服务器选择的是开源的 Apache 服务器，这是全球目前应用范围最广和最流行的 Web 服务器。具体

实施上，挑选了中文的 PHP 框架 Thinkphp，它在目前国内 PHP 开发社区中占据了重要的地位。详细的开发环境如表 4-1。

表 4-1 详细开发环境表

软硬件	细分领域	选用项	版本等参数
硬件	所有	所有	当前的主流配置
软件	操作系统	Windows10	64 位
	前端	HTML、CSS、Javascript	常规
	后台编程语言	PHP	5.6
	数据库	MySQL	5.6
	Web 服务器	Apache	2.4
	前端应用框架	Jquery	1.4.2
	后台应用框架	Thinkphp	3.2.2

4.2 系统设计概述

爱驿站平台的设计中主要采用了 MVC 的设计思想，划分 Model、View、Controller 三个层次进行开发。Model 层里面主要是 PHP 代码，负责数据库建模、连接和传递程序对数据库的操作；View 层也就是数据显示层，混合了少量 PHP 代码和前端语言代码，它负责将程序读取出来并把数据展示给用户。在这里，View 层也可以认为等价于网页；Controller 层主要负责控制和转发，但是在本次实际开发过程中，它还会被加入具体处理业务的功能。实际编写程序过程中，这些思想将会在 Thinkphp 框架中体现。三层原理结构如图 4-1。

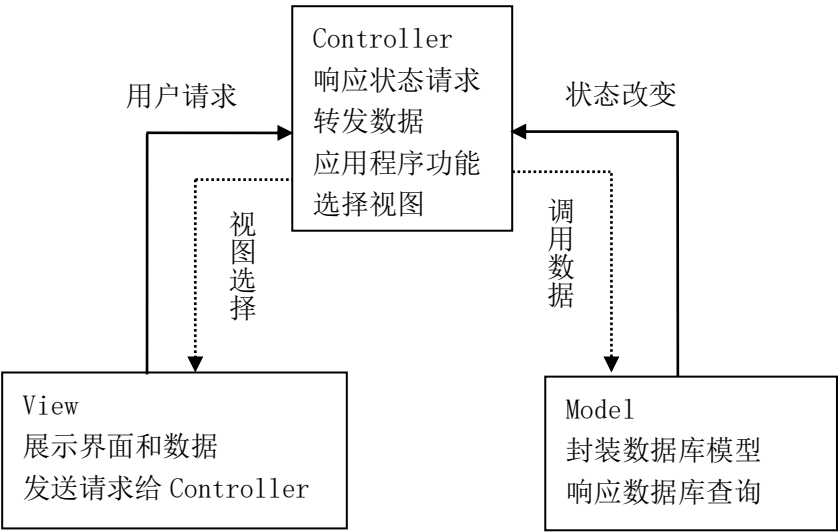


图 4-1 原理架构图

其次根据逐步求精和自顶向下的软件开发原则，按照功能和实体为区分标准，将爱驿站划分成 13 个模块。每个模块都是 MVC 三层结构，但是由于程序中经常出现跨模块调用 Model 层程序，所以把所有的 Model 程序整合放在了 Common 文件夹下，作为公共部分使用。代码结构如图 4-2。

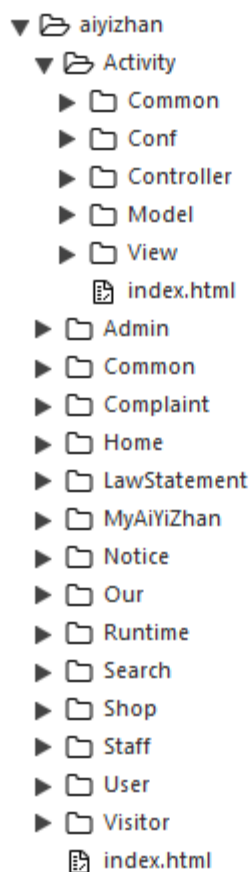


图 4-2 代码结构图

4.3 关键问题解决方案

4.3.1 时区配置

PHP 在 5.2.4 版本之前的版本无需设置时区。在本次开发中，使用的是 5.6 版本的 PHP，所以查看日志会发现类似“PHP Warning: date() [function.date]”的警告错误。这会导致日志文件的快速增长。

解决方法是打开 php.ini 文件，去掉“;date.timezone =”这一行前面的分号，改成“date.timezone = Asia/Shanghai”。

4.3.2 图片轮播

在产品设计中，要求实现一个图片轮播的视觉效果。图片轮播的原理一般采用将图片嵌入 ul 标签中，然后按照一定的时间规律，显示某一张图片，隐藏其

他图片的方式。

在程序编写过程中，采用了一个 nivo-slider 的插件来实现快速开发。主要 Javascript 代码如下。

```
//对动态轮播的图片做一些设置
//这个函数一定要放在最后，要不然以后的function不会执行
$(window).load(function() {
    $('#slider').nivoSlider({
        effect: 'sliceUpDown', // 过渡效果
        slices: 1, //effect为切片效果时的数量
        boxCols: 0, //effect为格子效果时的列
        boxRows: 0, //effect为格子效果时的行
        animSpeed: 1500, //动画速度
        pauseTime: 5000, //图片停留时间
        startSlide: 0, //从第几张开始
        directionNav: false, //是否显示图片切换按钮(上/下页)
        directionNavHide: false, //是否鼠标经过才显示
        controlNav: false, // 显示序列导航
        controlNavThumbs: false, // 显示图片导航
        controlNavThumbsFromRel: false, // 使用img的rel属性作为缩略图地址
        controlNavThumbsSearch: '.jpg', // 查找特定字符串(controlNavThumbs必须为true)
        controlNavThumbsReplace: '_thumb.jpg', // 替换成这个字符(controlNavThumbs必须为true)
        keyboardNav: false, // 键盘控制(左右箭头) PS:建议把代码中的keypress换成keydown,因为在Chrome下有兼容问题。
        pauseOnHover: false, // 鼠标经过时暂停播放
        manualAdvance: false, // 是否手动播放(false为自动播放幻灯片)
        captionOpacity: 0.8, // 字幕透明度
        // prevText: 'Prev',
        // nextText: 'Next',
        randomStart: false, //是否随机图片开始
        // beforeChange: setImgWidth(), //动画开始前触发
        // afterChange: setImgWidth(), //动画结束后触发
        // slideshowEnd: function(){}, // 本轮循环结束触发
        // lastSlide: function(){}, // 最后一张图片播放结束触发
        // afterLoad: setImgWidth() // 加载完毕时触发
    });
});
```

4.3.3 上传固定格式的图片

在普通用户注册以及管理员设置活动、快报等业务中，都需要一个可以裁剪图片的功能。具体实现中，采用了一个 flash 插件。前端方面，由 flash 根据用户选择的虚线边框确定图片需要裁剪保存的部分，然后剪裁完成后生成新的图片并根据 PHP 代码存放到后台指定的位置，最后把存放路径和 uniqid() 函数生成的文件名存放到数据库中。

前台代码和后台部分 PHP 代码如下。

```
<!-- flash插件, html5技术 -->
<div style="margin: 30px auto 60px; width: 100%; height: 100%;">
<OBJECT classid="clsid:D27CD86E-AE6D-11cf-96B8-444553540000" codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/
swflash.cab#version=6,0,40,0" WIDTH="650" HEIGHT="450" id="myMovieName">
<PARAM NAME=movie VALUE="/aiyizhan/Common/flash/threePhoto.swf">
<PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#FFFFFF>
<!-- <param name="flashvars" value="imgUrl=../Public/flash/default.jpg&uploadUrl=/aiyizhan/visitor.php/
uploadImg&uploadSrc=false"/> -->
<EMBED src="/aiyizhan/Common/flash/threePhoto.swf" quality=high bgcolor=#FFFFFF" WIDTH="500" HEIGHT="470" wmode="transparent"
flashVars="imgUrl=/aiyizhan/Common/flash/default.jpg&uploadUrl=/user.php/uploadImg&uploadSrc=false" NAME="myMovieName" ALIGN=""
TYPE="application/x-shockwave-flash" allowScriptAccess="always" PLUGINS PAGE="http://www.macromedia.com/go/getflashplayer">
<!-- </EMBED> -->
</OBJECT>
</div>
```

```

// 单独上传头像
public function uploadImg(){
    $destination_folder = $COMMON_PATH.'aiyizhan/Common/UserUploadImg/';

    try {
        $filename270 = substr(md5(uniqid()),2,10);
        $filename200 = substr(md5(uniqid()),1,10);
        $filename120 = substr(md5(uniqid()),0,10);

        $pic1=base64_decode($_POST['pic1']);
        $pic2=base64_decode($_POST['pic2']);
        $pic3=base64_decode($_POST['pic3']);

        // 保存270x320格式
        if(file_exists($destination_folder.$filename270."270.jpg")) {
            $filename270 = substr(md5(uniqid()),0,10);
            if(file_exists($destination_folder.$filename270."270.jpg")) {
                $rs['status'] = '1,';
            }
            else{
                file_put_contents($destination_folder.$filename270."270.jpg",$pic1);
                $rs['status'] = '0,'.$filename270;
            }
        }
        else{
            file_put_contents($destination_folder.$filename270."270.jpg",$pic1);
            $rs['status'] = '0,'.$filename270;
        }

        // 保存200x200格式
        if(file_exists($destination_folder.$filename200."200.jpg")) {
            $filename200 = substr(md5(uniqid()),1,10);
            if(file_exists($destination_folder.$filename200."200.jpg")) {
                $rs['status'] = '1,';
            }
            else{
                file_put_contents($destination_folder.$filename200."200.jpg",$pic2);
                $rs['status'] = $rs['status'].','.$filename200;
            }
        }
        else{
            file_put_contents($destination_folder.$filename200."200.jpg",$pic2);
            $rs['status'] = $rs['status'].','.$filename200;
        }

        // 保存120x120格式
        if(file_exists($destination_folder.$filename120."120.jpg")) {
            $filename120 = substr(md5(uniqid()),2,10);
            if(file_exists($destination_folder.$filename120."120.jpg")) {
                $rs['status'] = '1,';
            }
            else{
                file_put_contents($destination_folder.$filename120."120.jpg",$pic3);
                $rs['status'] = $rs['status'].','.$filename120;
            }
        }
        else{
            file_put_contents($destination_folder.$filename120."120.jpg",$pic3);
            $rs['status'] = $rs['status'].','.$filename120;
        }

        echo json_encode($rs);
    }
    catch (Exception $e) {
        // 异常
        $rs['status'] = '3,';
        echo json_encode($rs);
    }
}

```

4.3.4 分页技术

数据分页技术是一个企业级应用中常见的关键问题。分页这一个功能做好，可以提供给用户良好的体验，大大增强应用的实用性。本次开发中，主要依赖于 Thinkphp 提供的 Page 类， 加上对 SQL 查询语句的 limit 条件的应用，从而实现数据分页显示。爱驿站网站中开发中多处都用到该技术，下面仅以活动模块中

作为样例。

后台关键部分代码如下。

```
// 访问活动主页
public function index(){
    $Activity = new \Common\Model\ActivityModel();
    // 修改page的数字参数, 可以改变数据库查询的每页多少, 请保持和下面创建Page类对象时的数字一致
    $list = $Activity->order('createTime desc')->page($_GET['p'],10)->select();
    $this->assign('list',$list);// 赋值数据集

    // 查询满足要求的总记录数
    $count = $Activity->order('createTime desc')->count();
    // 实例化分页类 传入总记录数和每页显示的记录数
    $Page = new \Think\Page($count,10);

    $Page->setConfig('prev','上页');
    $Page->setConfig('next','下页');
    $Page->setConfig('theme','%FIRST% %UP_PAGE% %LINK_PAGE% %DOWN_PAGE% %END%');
    $show = $Page->show();// 分页显示输出
    $this->assign('page',$show);// 赋值分页输出
    $this->display('activity');
}
```

4.4 模块划分

4.4.1 主页模块

主页模块主要是用以展示首页内容和控制登录入口。访问首页会先检查 session 中是否已经存储了该用户信息。

登录检查中, 首先接收客户端使用 ajax 技术传送给服务端的登录用户名和密码, 然后将密码和数据库中存储的该用户密码对比检查。如果相同则通过登录验证, 存储用户信息到 session 中并返回表示正确的返回值, 否则返回固定的表示错误的值。首页接收到返回值, 根据返回值更改首页登录框的内容。下面的代码展示了登录入口控制的详细机制。

```
//登录时验证密码是否对应相等
public function vericate($accountMobile,$password){
    //这样实例化可以清楚看到model位置。
    $Visitor = new \Common\Model\VisitorModel();
    $result = $Visitor->field("password")->where("accountmobile=".$accountMobile)->limit(1)->select();
    $passwordDb = $result[0]['password'];

    // echo "passwordDb : ".$result."<br>";
    if (strcmp($password,$passwordDb)==0) {
        return true;
    }
    else{
        return false;
    }
}
```

```

// 登录验证
public function login()
{
    $accountMobile = $_POST['visitorName'];
    $password = $_POST['password'];

    try {
        // 验证是否允许登录
        $Visitor = new \Common\Model\VisitorModel();
        // $result = $Visitor->where("accountmobile=".$accountMobile)->limit(1)->select();
        $result = $Visitor->where("accountmobile=".$accountMobile)->limit(1)->select();

        if (count($result) > 0) {
            if (strcmp($result[0]['admitLogin'], "admit") == 0) {
                // 验证密码是否正确
                if ($this->vericate($accountMobile, $password)) {
                    session_start();
                    if (!isset($_SESSION['accountMobile'])) {
                        $_SESSION['visitorId'] = $result[0]["visitorId"];
                        $_SESSION['visitorName'] = $result[0]["visitorName"];
                        $_SESSION['accountMobile'] = $result[0]["accountmobile"];
                        $_SESSION['sexVal'] = $result[0]["sexVal"];
                    }
                    echo 0;
                }
            } else {
                echo 1;
            }
        } else {
            // 表示被禁止登录
            echo 4;
        }
    } catch (Exception $e) {
        echo 3;
    }
}

```

退出登录时首先销毁该用户的 session 信息，然后重新访问首页，这时首页检查到 session 里不存在该用户的有关信息，则显示未登录状态。代码如下。

```

// 退出登录,返回登录页面
public function logout(){
    unset($_SESSION['visitorId']);
    unset($_SESSION['visitorName']);
    unset($_SESSION['accountMobile']);
    unset($_SESSION['sexVal']);

    echo "<script>
        location.href='/myAiYiZhan.php/index';
    </script>";
    // alert('成功退出登录!\n\n即将返回登录页面! ');
}

```

4.4.2 搜索模块

搜索模块主要功能是提供给任何使用爱驿站网站的人，搜索任何已经被登记并被允许展示的驿站会员，并且可以通过点击头像进入他们的详情资料展示页面。

搜索模块，首先会把客户端的选择条件在后台组合起来，然后用组合条件去

查询数据库，结果将采用分页技术进行分页展示。关键代码如下。

```
// 搜索驿站会员
public function searchUser(){
    $User = new \Common\Model\UserModel();

    if ($_GET['userId'] != "" && $_GET['userId'] != null) {
        $map['id'] = $_GET['userId'];
    }
    // else{
        //对厨艺进行判断
        if (strcmp($_GET['sexinfo'], "性别") != 0) {
            $map['sexVal'] = $_GET['sexinfo'];
        }

        if (strcmp($_GET['huiyuanleixing'], "会员类型") != 0) {
            $map['level'] = $_GET['huiyuanleixing'];
        }

        $data['startAge'] = (int)$_GET['startAge'];
        $data['endAge'] = (int)$_GET['endAge'];
        $data['maritalStatus'] = $_GET['marriage'];
        $data['startHeight'] = (int)$_GET['startHeight'];
        $data['endHeight'] = (int)$_GET['endHeight'];
        $data['weight'] = (int)$_GET['tizhong'];
        $data['monthlyWage'] = $_GET['yuexin'];

        //对厨艺进行判断
        if (strcmp($_GET['chuyi'], "厨艺") != 0) {
            $map['cookLevel'] = $_GET['chuyi'];
        }
        //对住房进行判断
        if (strcmp($_GET['fang'], "住房") != 0) {
            $map['haveHouse'] = $_GET['fang'];
        }
        //对车子进行判断
        if (strcmp($_GET['car'], "车子") != 0) {
            $map['haveCar'] = $_GET['car'];
        }

        //对地名进行组合，以便进行sql查询
        if (strcmp($_GET['sheng'], "地区") == 0) {
            // $map['nowareacode'] = array('like', "%".$_GET['nowareacode']."%");
        }
        else if (strcmp($_GET['shi'], "市") == 0 || $_GET['shi'] == "") {
            $data['nowareacode'] = $_GET['sheng'];
            $map['nowareacode'] = array('like', "%".$_GET['nowareacode']."%");
        }
        else if (strcmp($_GET['xian'], "县\区") == 0 || $_GET['xian'] == "") {
            $data['nowareacode'] = $_GET['sheng']."_".$_GET['shi'];
            $map['nowareacode'] = array('like', "%".$_GET['nowareacode']."%");
        }
        else if (strcmp($_GET['xiang'], "乡镇\街道") == 0 || $_GET['xiang'] == "") {
            $data['nowareacode'] = $_GET['sheng']."_".$_GET['shi']."_".$_GET['xian'];
            $map['nowareacode'] = array('like', "%".$_GET['nowareacode']."%");
        }
        else{
            $data['nowareacode'] = $_GET['sheng']."_".$_GET['shi']."_".$_GET['xian']."_".$_GET['xiang'];
            $map['nowareacode'] = array('like', "%".$_GET['nowareacode']."%");
        }

        //对年龄范围判断，并判断大小，组合查询条件
        if ($data['startAge'] > 0 && $data['endAge'] > 0) {
            if ($data['startAge'] < $data['endAge']) {
                $map['age'] = array(array('egt', $data['startAge']), array('elt', $data['endAge']));
            }
            else if ($data['startAge'] > $data['endAge']) {
                $map['age'] = array(array('egt', $data['endAge']), array('elt', $data['startAge']));
            }
            else {
                $map['age'] = $data['startAge'];
            }
        }
    }
}
```



```

//对身高范围判断,并判断大小,组合查询条件
if ($data['startHeight'] > 0 && $data['endHeight'] > 0) {
    if ($data['startHeight'] < $data['endHeight']) {
        $map['height'] = array(array('egt',$data['startHeight']),array('elt',$data['endHeight']));
    }
    else if($data['startHeight'] > $data['endHeight']){
        $map['height'] = array(array('egt',$data['endHeight']),array('elt',$data['startHeight']));
    }
    else {
        $map['height'] = $data['startHeight'];
    }
}
//留下以后备用 //对体重进行判断是否大于200,分两种情况
// if (strcmp($data['monthlyWage'],'-1') == 0) {

// }
// else if ($data['weight'] == 200) {
//     $map['weight'] = array('egt',$data['weight']);
// }
// else{
//     $map['weight'] = array('elt',$data['weight']);
// }

if (strcmp($data['monthlyWage'],'月薪') != 0) {
    $map['monthlyWage'] = $data['monthlyWage'];
}

if (strcmp($data['maritalStatus'],'婚姻状况') != 0) {
    $map['maritalStatus'] = $data['maritalStatus'];
}

if (strcmp($_GET['levelOfEducation'],'学历') != 0) {
    $map['levelOfEducation'] = $data['levelOfEducation'];
    $m = "12134";
}

if (strcmp($_GET['sonForLaw'],'上门女婿') != 0) {
    $map['sonInLaw'] = $data['sonForLaw'];
}

// }

```

// 修改page的数字参数,可以改变数据库查询的每页多少,请保持和下面创建Page类对象时的数字一致
\$list = \$User->where(\$map)->page(\$_GET['p'],18)->order('createTime desc')->select();

```

if (count($list) > 0) {
    // 查询满足要求的总记录数
    $count = $User->where($map)->count();
    // 实例化分页类 传入总记录数和每页显示的记录数
    $Page = new \Think\Page($count,18);
    // 分页跳转的时候保证查询条件
    foreach($map as $key=>$val) {
        $Page->parameter[$key] = urlencode($val);
    }

    $Page->setConfig('prev','上一页');
    $Page->setConfig('next','下一页');
    $Page->setConfig('theme','%FIRST% %UP_PAGE% %LINK_PAGE% %DOWN_PAGE% %END%');
    $show = $Page->show();// 分页显示输出
    $this->assign('page',$show);// 赋值分页输出
    $this->assign('userList',$list);// 赋值数据集
    if ($_GET['p'] == "" && !isset($_GET['p'])) {
        $this->display('searchResult');
    }
    else {
        $this->display('searchIndex');
    }
}
else{
    echo '1';
}
// echo $User->getLastSql()."<br>".strcmp($data['levelOfEducation'],'学历')."<br>".$m.$_GET['levelOfEducation'];
}

```

4.4.3 我的驿站模块

我的驿站模块是给已经注册的普通用户使用的模块。这里给普通会员提供了修改密码、查看收藏会员、管理我的活动等功能。

模块实现效果如图 4-3。



图 4-3 我的驿站模块效果图

4.4.4 活动模块

活动模块主要有展示活动和推荐榜功能。同时管理员的管理活动功能代码也写在这里。

添加删除活动的代码逻辑都是通过对数据库的数据进行操作实现的。报名活动中需要牵涉到 `activity` 表的操作，需要将当前报名会员的编号存储到 `activity` 表中，这样就通过检查 `activity` 表中是否存在某一会员的编号，就可以知道该会员是否已经报名和该活动的报名人数等信息。同时在存储编号到 `activity` 表时，更新 `nowManNum` 和 `nowWomenNum` 两个字段，从而可以方便的展示活动报名人数变化和分布。关键部分 PHP 代码如下。

```

// 添加活动信息到数据库
public function addActivityToDb(){
    $data['userName'] = $_POST['userName'];
    $data['id'] = $_POST['activityId'];
    $data['activityTitle'] = $_POST['activityTitle'];
    $data['activityTime'] = $_POST['activityTime'];
    $data['numOfPeople'] = $_POST['numOfPeople'];
    $data['activityAddress'] = $_POST['activityAddress'];
    $data['telephone'] = $_POST['telephone'];
    $data['deadlineOfEnroll'] = $_POST['deadlineOfEnroll'];
    $data['activityOrganizer'] = $_POST['activityOrganizer'];
    $data['activityContent'] = $_POST['activityContent'];
    // $data['isRecommend'] = $_POST['isRecommend'];

    $destination_folder = $COMMON_PATH.'aiyizhan/Common/ActivityUploadImg/';

    // 检查是否存在存储路径, 若不存在则创建该目录
    if(!is_dir($destination_folder))
    {
        mkdir($destination_folder);
    }

    // 为上传图片做设置
    $upload = new \Think\Upload();// 实例化上传类
    $upload->maxSize = 1048576 ;// 设置附件上传大小为1M
    $upload->exts = array('jpg', 'gif', 'png', 'jpeg');// 设置附件上传类型
    $upload->rootPath = $destination_folder; // 设置附件上传根目录
    $upload->savePath = '';
    $upload->saveName = 'uniqid'; // 默认的命名规则设置是采用uniqid函数生成一个唯一的字符串序列。
    // 上传文件
    $info = $upload->upload();
    if(!$info) { // 上传错误提示错误信息
        $this->error($upload->getError());
    }else{ // 上传成功 获取上传文件信息
        $data['pathAndName'] = $info['activityImg']['savepath'].$info['activityImg']['savename'];
        $data['createTime'] = date('Y-m-d H:i:s',time());

        try {
            $Activity = new \Common\Model\ActivityModel();
            $insertId = $Activity->add($data);
            if ($insertId) {
                echo "<script>location.href='/activity.php/addActivity';
                    alert('添加活动成功! ');</script>";
            }
            else {
                echo "<script>alert('添加活动失败! 请重新添加或联系管理员! ');
                    location.href='/activity.php/addActivity';</script>";
            }
            // echo $Activity->getLastSql();
        }
        catch (Exception $e) {
            echo "<script>alert('添加活动失败! ');
                location.href='/activity.php/addActivity';</script>";
        }
    }
}
}

```

```

// 添加到推荐榜
public function addRecommendToDb(){
    for ($i=1; $i <= 10; $i++) {
        if ($_POST['activityId'].$i != "") {
            $dataList[$i] = array('activityId' => $_POST['activityId'].$i, 'activityOrder' => $i);
        }
    }

    if (count($dataList) > 0) {
        $RecommendActivity = new \Common\Model\RecommendActivityModel();
        // 先清除所有之前的数据
        $RecommendActivity->where('1')->delete();
        foreach ($dataList as $key => $value) {
            $result = $RecommendActivity->add($dataList[$key]);
        }

        if ($result > 0) {
            echo "<script>location.href='/activity.php/activityManage';
                alert('添加活动到推荐榜成功!');</script>";
        }
        else{
            echo "<script>location.href='/activity.php/recommendActivity';
                alert('添加活动到推荐榜失败! ');</script>";
        }
    }
    else{
        echo "<script>location.href='/activity.php/recommendActivity';
            alert('没有数据需要添加! ');</script>";
    }
}

// 删除活动
public function deleteActivity(){
    $destination_folder = $COMMON_PATH.'aiyizhan/Common/ActivityUploadImg/';
    $deleteIndexId = $_GET['id'];

    try{
        $Activity = new \Common\Model\ActivityModel();
        $RecommendActivity = new \Common\Model\RecommendActivityModel();

        // 先删除图片文件
        $map['id'] = array("eq",$deleteIndexId);
        $pathAndNameRes = $Activity->field('pathAndName')->where($map)->select();
        $filename = $destination_folder.$pathAndNameRes[0]['pathAndName'];

        if(file_exists($filename)){
            // 从数据库中查出路径为空时，将直接进行删除操作。因为路径是存在的，所以执行这里。
            unlink($filename);
            // 再删除数据库中两张表中的活动记录
            $RecommendActivity->delete($deleteIndexId);
            $resultNum = $Activity->delete($deleteIndexId);
            if ($resultNum < 0) {
                echo "<script>alert('删除失败! ');</script>";
            }
            echo "<script> location.href='/activity.php/activityManage'; </script>";
        }
        else{
            echo "<script>alert('路径错误! 删除失败! ');</script>";
            echo "<script> location.href='/activity.php/activityManage'; </script>";
            //fortest: echo $filename;
        }
    }
    catch(Exception $e){
        echo "<script>alert('删除失败! ');</script>";
    }
}

```

```

// 报名活动
// 登录检查已经在前端php页面做了
public function enrollActivity(){
    $id = $_POST['activityId'];
    $enroll = $_POST['enroll'];

    $Activity = new \Common\Model\ActivityModel();
    $SoldEnroll = $Activity->field('enroll,nowManNum,nowWomenNum')->where('id='.$id)->limit(1)->select();

    $SoldEnrollArray = explode(",", $SoldEnroll[0]['enroll']);
    $isEnrolled = false;
    foreach ($SoldEnrollArray as $key => $value) {
        if (strcmp($value, $enroll) === 0) {
            $isEnrolled = true;
        }
    }

    // isEnrolled为真说明已经报名了
    if ($isEnrolled) {
        echo -1;
    }
    else{
        $data['enroll'] = $SoldEnroll[0]['enroll'].$enroll.",";

        if ($_SESSION['sexVal'] == "男") {
            $data['nowManNum'] = $SoldEnroll[0]['nowManNum'] + 1;
            $resultNum = $Activity->field('enroll,nowManNum')->where('id='.$id)->save($data);
        }
        elseif($_SESSION['sexVal'] == "女"){
            $data['nowWomenNum'] = $SoldEnroll[0]['nowWomenNum'] + 1;
            $resultNum = $Activity->field('enroll,nowWomenNum')->where('id='.$id)->save($data);
        }
        else{
            $resultNum = $Activity->field('enroll')->where('id='.$id)->save($data);
        }

        // 返回更新数为0说明未更新
        if ($resultNum === 0) {
            echo 1;
        }
        // 返回更新数为1说明更新成功
        else if($resultNum === 1){
            echo 0;
        }
        // 返回false说明更新失败
        else if($resultNum === false){
            echo 1;
        }
        else {
            echo 3;
        }
    }
}
}

```

4.4.5 快报模块

快报模块的也是主要展示快报。管理员的添加删除等管理功能同样写在该模块下。基本原理和代码机制同活动模块。

4.4.6 法律模块

法律模块主要展示相关法律声明和关于公司的资料。实现效果如图 4-4。



图 4-4 法律模块效果图

4.4.7 店铺模块

店铺模块主要功能包括：管理店铺的驿站会员、管理自身的部分资料、搜索店铺内的驿站会员资料。通过对 shop 表以及 user 表的数据库操作，实现搜索和管理驿站会员资料、更新自身资料的功能。

更新店铺主页的 HTML 内容，是通过 session 变量传递。搜索驿站会员是根据驿站会员系统编号、手机号码、身份证号查询驿站会员，并且加入 user 表中 shopId 字段作必须的查询条件，限制店铺的查询范围，防止店铺恶意查询其他店铺的会员资料。功能代码如下。

```
// 保存店铺主页的html内容
public function saveShopIndex(){
    // $htmlData = '';
    if (!empty($_POST['content1'])) {
        if (get_magic_quotes_gpc()) {
            // $htmlData = stripslashes($_POST['content1']);
            $_SESSION["shopIndex"] = stripslashes($_POST['content1']);
        } else {
            $_SESSION["shopIndex"] = $_POST['content1'];
        }

        echo '<script type="text/javascript">
            window.history.go(-1);
            hiddenShopIndex();
        </script>';
    }
    else{
        echo '<script type="text/javascript">
            window.history.go(-1);
            alert("上传内容为空");
        </script>';
    }
}
```

```
// 店铺搜索会员功能
public function searchUser(){
    $User = new \Common\Model\UserModel();

    $data['id'] = $_POST['userId'];
    $data['accountmobile'] = $_POST['accountmobile'];
    $data['idCard'] = $_POST['idCard'];

    foreach ($data as $key => $value) {
        if ($value != "") {
            $map[$key] = $value;
        }
    }

    // 店铺搜索会员的条件：属于该店铺
    $map["shopId"] = $_POST["shopId"];

    $listRes = $User->where($map)->select();

    if (count($listRes) > 0) {
        echo json_encode($listRes);
    }
    else{
        echo '1';
    }
}
```

4.4.8 管理员模块

管理员的主要功能是管理驿站会员、普通用户、店铺、活动、快报等内容。这些功能的具体实现都分散在各个模块中，管理员界面中只是做函数的调用工作。这样可以大大简化管理员模块的代码复杂性，增强整体网站程序的可读性。

管理员的管理界面如图 4-5。



图 4-5 管理员模块界面

5 网站测试和优化

5.1 功能测试

对于软件工程，在完成初步开发后，接下来的重要工作之一就是功能测试。只有通过完整系统的功能测试，软件才可以正式进行发布。

对于爱驿站网站，规划的功能测试步骤是从四种用户的角度分别进行，这样可以最大程度减少无用的重复测试、增加对重要功能的测试机会。四种用户分别是普通用户、驿站会员、店铺、管理员。

普通用户主要可能涉及的功能包含有注册、登录、查看爱驿站公开页面和信息、报名活动、搜索驿站会员、收藏和举报驿站会员等。驿站会员的功能基本同普通用户相同。店铺的功能包含有登录、管理自身信息、更新主页、修改密码、管理店铺内部驿站会员信息、搜索内部会员资料。管理员则涉及到管理普通用户、管理所有店铺、管理所有驿站会员、管理举报信息、管理快报和活动、设置各种推荐榜等内容。

5.2 网站部署

完成功能测试后，可以进行网站服务器的部署和发布。网站在服务器的部署过程基本类似于本地部署，但是后期还要根据实际运行效果和业务需求做性能优化。

当前网站部署在 Windows Server 系统的服务器上，首先要补全新系统的 `msvcr100.dll` 文件，下载后放在指定位置，否则会影响 Apache Server 安装。然后把 Apache Server、MySQL、PHP 环境安装配置好。之后就可以把程序从 Apache 的 `htdocs` 文件夹拷贝到服务器对应的位置，开启相关服务后就可以运行了。

当然还需要在阿里云上，给域名配置指定的服务器地址，给 Apache 配置 80 端口访问，这样才可以实现域名访问网站的效果。

5.3 性能与安全优化

5.3.1 性能优化

作为一个企业级应用，网站部署完成后，还针对网站实际运行效果和业务需求进行一定的性能优化。否则在存在大量访问的情况下，网站系统有可能会因为程序崩溃而无法使用。

性能优化主要分为程序优化、网页服务器优化、数据库优化三个方面。本文中主要做了前两点。程序优化方面，通过减少循环次数和循环层数优化程序运行等方法，减少复杂业务的运行时间。在图片方面，还通过专业的图片压缩软件，将占用空间几百 KB 和几兆的高清图片压缩到几十 KB，大幅减少静态资源载入的时间。服务器优化方面主要在程序入口文件中通过下方的代码启用了 gzip 压缩功能，这样可以大幅减小网页的占用空间，提高网页的显示速度和减少等待时间，适应大量访问和网站长期运行的情况。如图 5-2。

```
// 如果服务器支持，则开启gzip压缩模式
define ( "GZIP_ENABLE", function_exists ( 'ob_gzhandler' ) );
ob_start ( GZIP_ENABLE ? 'ob_gzhandler' : null );
```

5.3.2 网站安全优化

网站部署完成不代表网站可以一直稳定运行下去。网站运行的稳定不仅需要对性能进行优化，还需要对其安全性进行考虑。否则很容易就会被 DDOS 攻击瘫痪或者被盗取服务器账号。

安全工作上，一方面针对 Apache 配置进行修改，例如禁止访问目录等。另一方面尽可能设置高复杂度的登录密码。最后还给服务器安装了安全狗软件。这款软件可以较好的针对网站服务器进行特定安全优化和防护。实际应用效果良好。图 5-2 和图 5-3 是安全狗在服务器运行的截图画面。



图 5-2 安全狗运行截图



图 5-3 安全狗运行截图

结论

本次毕业设计是基于 Thinkphp 框架的爱驿站网站研究与实现。首先从婚恋网站的起源和发展进行探讨，阐释了对爱驿站网站的研究目的和意义，然后从不同角度分析了爱驿站平台业务逻辑，从而为业务逻辑实现编写了程序，即爱驿站网站系统，从而能够为单身青年提供一个真实良性运作的婚恋网站平台，并给店铺和凡几科技带来新的商机。

项目主要分配时间如下：

需求分析和业务逻辑分析花费了整个项目时间的 20%。具体编程实践和系统实现时间占据了 70%的时间。最后的系统部署和优化占据了项目剩余的 10%时间。

爱驿站平台的成功实现主要依赖于以下几个方面：

1. 便捷流畅的沟通，和企业导师之间良性的互动。
2. 前期需求分析和业务分析做的比较到位，逻辑清晰，开发任务顺序分配合理，这样很大程度上避免了返工现象的出现。

3. 和企业统一了着重于实现功能需求的开发思想，不拘泥于界面和文字细节的处理。将工作重点放在功能实现方面，这使得开发进度大大加快。

4. 把大量口头表述整理成文档，结构化的文档资料有效保障了本项目的有序进展。

爱驿站平台在以后的实际应用中，可能会发现一些 Bug，今后随着公司发展，功能需求也有可能发生改变，网站仍然需要不断完善和改进。

致谢

本次毕业论文，是在学校论文导师孙仁鹏老师和企业导师陆中浩先生的悉心指导下，才得以顺利完成。从论文的选题、论文的结构安排等方面，孙仁鹏老师都给予了大量的指导意见和帮助，使得本人在论文的撰写思路有了明确的方向。在爱驿站平台的开发过程中，企业导师陆中浩先生在业务逻辑分析等方面也给予了很多指导意见和帮助。两位老师在繁忙工作之余给我提出了许多宝贵意见，使我论文工作有了不少进展。在此论文完成之际，特别向孙仁鹏老师和陆中浩先生表示最衷心的感谢！

参考文献

- 1 谭华孚. 中国婚恋网站发展研究: 学位论文, 福建师范大学, 2011
- 2 李金玉. 婚恋网站: 一种新型的择偶方式: 黑龙江大学政府管理学院
- 3 孙炜路. 基于 Thinkphp 框架的软件学院教务管理系统的设计与实现: 吉林大学计算机科学与技术学院, 2015
- 4 王亮. 基于 Thinkphp 的 RSS 阅读系统的设计与实现: 华中科技大学, 2012
- 5 王浩百. 基于 PHP 和 MySQL 的办公室网站设计与实现: 吉林大学, 2010
- 6 人民网. 中国大陆总人口达 13.6 亿人 男性比女性多 3376 万. <http://politics.people.com.cn/n/2015/0120/c1001-26417578.html>, 2015-01-20
- 7 腾讯科技. 评论婚恋网站瓶颈. <http://tech.qq.com/a/20101206/000116.htm>, 2010-12-06