|  |  |
| --- | --- |
|  | 学校代码：10246 |
|  | 学 号： |



|  |
| --- |
| 硕 士 学 位 论 文 |

（专 业 学 位）

|  |
| --- |
| **互动零售商业社区系统的设计与实现**  **Study on Interactive Retail Business System** |

|  |  |
| --- | --- |
| 院 系： | 软件学院 |
| 专 业： | 软件工程 |
| 姓 名： |  |
| 指 导 教 师： |  |
| 完 成 日 期： | 2013年9月30日 |

# 目 录

[目 录 I](#_Toc398928642)

[摘 要 II](#_Toc398928643)

[ABSTRACT III](#_Toc398928644)

[第一章 绪 论 1](#_Toc398928645)

[1.1 课题研究宏观背景 1](#_Toc398928646)

[1.2 课题研究目的和意义 4](#_Toc398928647)

[1.3 论文主要内容和框架结构 4](#_Toc398928648)

[第二章 关键技术介绍 5](#_Toc398928649)

[2.1 中间件技术介绍 5](#_Toc398928650)

[2.2 移动APP开发技术介绍 6](#_Toc398928651)

[2.3 与既有软硬件平台的接口 7](#_Toc398928652)

[第三章 系统总体设计 11](#_Toc398928653)

[3.1 系统背景 11](#_Toc398928654)

[3.2 系统需求分析 12](#_Toc398928655)

[3.3 系统总体设计 15](#_Toc398928656)

[3.4 软硬件选型 17](#_Toc398928657)

[第四章 系统详细设计 20](#_Toc398928658)

[4.1 系统模块设计 20](#_Toc398928659)

[4.2 互动触摸模块设计 22](#_Toc398928660)

[4.3 移动APP应用设计 31](#_Toc398928661)

[4.4 整合社会化媒体模块设计 40](#_Toc398928662)

[4.5 系统用例分析 48](#_Toc398928663)

[第五章 总结与展望 53](#_Toc398928664)

[5.1 本文内容总结 53](#_Toc398928665)

[5.2 未来工作展望 54](#_Toc398928666)

[参考文献 55](#_Toc398928667)

[致 谢 57](#_Toc398928668)

# 摘 要

随着中国的消费市场不断地发生着变化，无论是商品还是消费渠道的丰富程度相较以往都有了巨大的发展。同时，消费者的关注重点也从集中在商品本身，而转移到了消费的过程体验中来。但是，传统的零售企业往往还停留在落后的经营理念和陈旧的管理模式中，同时在对于新技术、新观念的接受和应用上的表现并不积极，从而在激烈的市场竞争中逐渐处于劣势。因此，在当今的市场环境下，如何满足消费者日益增长的对消费过程体验的需求，已经成为国内零售企业寻求长期发展的重要使命。为了能够改善传统企业在经营及管理上的不足之处，使零售企业能够向现代化的方向发展，就必须开发出一套互动零售商业社区系统。

本文在了解中国消费市场的宏观背景后，对于零售行业的现状作了细致的分析，确定以中间件技术为核心来进行系统开发以解决现存的问题。在此基础上，论文详细分析了大悦城购物中心的实际需求，确定了系统的业务流程及功能划分，给出总体设计及详细的模块设计。并重点对系统中的互动触摸模块、移动APP应用以及整合社会化媒体模块的设计与实现进行了深入的分析。最后，针对系统的功能特点进行了讨论，并对系统的不足之处进行了总结，同时就未来的发展方向提出了展望。

关键词 零售信息系统，互动交互系统，移动App开发

# ABSTRACT

With the constant change of the consumer market in China, both consumer goods and consumer channels changed a lot compares to the past. Meantime, the focus of the consumers also transferred from only the consumer goods itself to the consuming experience. However, most retailers still stay in the old business idea and outdated management model, do not positively accept and utilize the new technology and concept, and became at the disadvantage in the market competition. Under this situation, to fulfill the growing demands of the consumers has already been an important mission for the domestic retailers to seek for long-term development. In order to improve the operation and management level of the traditional retailers and make them have further development of modernization. The development of an interactive retail community system becomes indispensable.

After the analysis of the macro-background of the Chinese consumer market and the detail investigation of the current situation of the retail industry, Middleware was selected to be the kernel of the system development to solve the current issues. The thesis analyzed the actual needs of Joy City shopping center, determined the business procedure and function divisions, and confirmed the system architecture diagram and the detailed module design. Moreover, an in-depth analysis of the design and the implementation was conducted for the Interactive Touch Screen, Mobile App and Social Media integration. At the last part, the function characteristics of the system are discussed, the inadequacies of the system are summarized, and some imperfections are raised for the future improvement.

Keywords Retail information system, Interactive system, Mobile APP development

# 第一章 绪 论

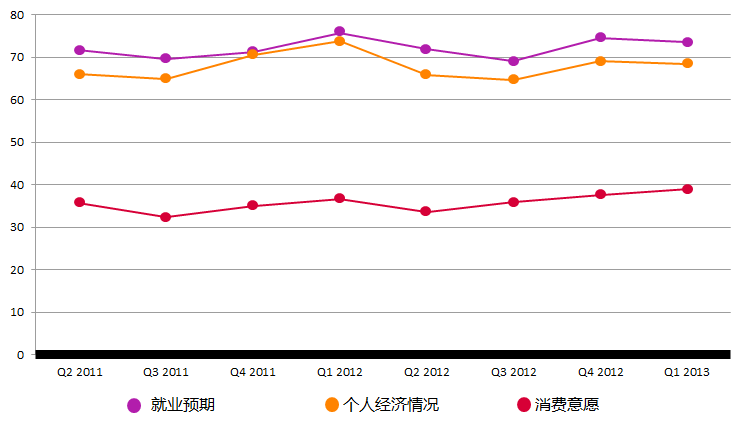
# 1.1 课题研究宏观背景

在过去的几年内，中国的消费市场正在悄然地发生变化。首先，中国的消费者信心指数有了显著的上升，我们从尼尔森最新的公开研究报告中可以得知，中国的消费者信心指数在2012年末上升至108（根据研究，指数达到或超过100属于乐观，低于100为相对悲观），并在2013年第一季度稳定于这个水平。2012年第二和第三季度由于全球金融危机，导致中国经济增长首次出现放缓，国内生产总值增长低于8%，消费者信心指数比年初回落了约五个百分点，而目前中国消费者信心指数则明显高于全球平均指数和美国指数93%[1]。如下图所示：



**图 1-1 2009年第一季度到2013年第一季度的消费者信心指数**

同时，在2013年的第一季度，中国消费者的消费意愿达到39%[2]，这是2010年第三季度以来的最高值，这体现了在未来的六个月中，消费者信心将呈现健康的增长势头，拉动消费者信心上扬的主要原因是他们对未来数月就业机会和个人经济状况的乐观预期。从下图我们可以看到这种成长趋势：



**图 1-2 中国消费者的消费意愿（百分比计算）**

消费者信心指数的上扬和消费意愿的提高带来了机遇，也带来了挑战。越来越多的消费者在愿意将更多的个人收入投入到消费行为中来的同时，也开始变得越来越成熟，他们不仅仅对所购买的商品本身（包括质量、外观、价格），同时将对购买方式、消费渠道、支付便利性、过程体验等等各方面提出更高的要求。

通过研究我们可以得知，超过半数消费者倾向于在靠近住址的、经常光顾的商店进行购物。同时，也有约五分之一的消费者表示：亲朋好友的口碑，以及报纸及宣传海报上吸引眼球的促销信息也能对其购物地点的选择产生相当的影响力。

另外，经济环境变化增强了购物者对促销和价格的敏感性。事实上，三分之一的购物者会积极主动地在经常购物的零售店铺内寻找促销信息；四分之一的人会因为促销而买其他品牌的产品，更有至少五分之一的消费者（根据城市不同该比例会有不同变化）表示，为了找到最物美价廉的商品，他们很有可能会转换店铺。

同时，消费者的购买决定和消费方式也在发生着变化。2012年，中国网络用户超过5.64亿，这一数量是美国网民数量的两倍，是印度网民数量的近四倍。（数据来源：基于全世界网民的公开记录列表）去年有2.42亿中国网民进行了网上购物，这一数量大约是英国网络购物用户的六倍，日本网络购物用户的两倍。（数据来源：由阿里巴巴制作的关于中国电子商务的图表）在中国，有2.2亿用户使用电子银行或电子支付系统，其中有41%的消费者使用移动设备进行网上购物。这一切只是电子商务发展的开端，随着中国政府继续大力投入、扶持数字化基础设施建设和创新，这些数字仍会继续上升[3]。

随着互联网和移动设备的普及，从国际最新流行趋势来看，传统零售行业的互动与智能化已势不可挡。例如PERCH Interactive所推出的虚拟展示，其将传统的标签牌转化成动感展示系统。当消费者拿起柜台上所陈列的商品时，系统将会通过投影技术来展示一段蕴含商品的使用说明，模特试穿等信息的视频。从而将现实购物的直观性与网上购物的信息丰富性有机地结合了起来。又例如微软推出的Surface系统，Surface通过一个内置动作捕捉系统的互动触摸屏，将信息的输入输出变得极为简单且充满乐趣，消费者将一件商品放在Surface上时，内置的摄像头读出商品上的二维码，然后直接在商品旁边显示各种相关信息，最为有趣的是，消费者还可以直接点击这些信息进行更为深入地互动，而同时，零售商也可以通过消费者的这些行为而收集更多的消费者偏好等信息，从而为消费者提供更好的服务。然后，我们还可以看到越来越多的移动APP应用也在零售行业中扮演着重要的角色，如UrbanDaddy这个iOS应用，它的口号就是“重新发现你的城市”，用户通过一系列简单的选择输入自己的基本状态，APP就会根据这些信息来推算用户可能的消费需求，结合用户所处的地理位置而推荐临近的，最为合适的商业场所。

所以，当面对着快速发展并变化着的市场环境，为了更为牢固地抓住消费者，更为有效地抓住市场发展的契机，零售商必须从传统的，简单的纯商铺运营模式中解放思路，投身到电子商务的大潮中来，并且做好准备应对以下挑战：

赢得消费者的信任。消费者很重视对于整个购物场所的感觉，而不仅限于单纯的购物。在一个令人感觉舒适的环境中购物，可以让消费者了解到自己所受到的重视程度。一旦消费者找到他们喜欢的商店，就会成为常客，而客户忠诚度的作用至关重要，因为忠诚顾客的光顾频率会比平均水平要高出不少。

保持和消费者的联系。零售商需要成为购物者的“最好的朋友和邻居”。在一个每天都能接受到无数广告信息的世界里面，作为“朋友”，所发出的信息才有可能得到重视 。2012年有超过3.09亿中国人使用微博。各级城市中，97%的网民是社交媒体用户。这些用户每人每月至少花42小时使用微博，与他人互动，不断接受新信息。零售商需要跳出传统媒体的定势，在社会化媒体上增加投入，这样才能拉近与消费者的距离，在信息战中脱颖而出。

有效地与消费者交流。如今，传统的硬广告已经越来越不得人心。零售商要多使用软文广告，以及一些更为感性的沟通方式来与消费者进行交流。此外，同时整合线上/线下的活动的促销手段往往能够事半功倍。更为重要的是，零售商应该邀请消费者亲自参与到互动中来，让消费者从被动地接受信息，转变为主动地追逐信息。

# 1.2 课题研究目的和意义

随着时代的发展，越来越多身处传统领域的企业开始意识到，紧紧抓住时代的潮流，主动融合新概念，应用新技术才是在激烈竞争中制胜的关键，这其中，互联网技术的应用被越来越广泛地接受。一开始，企业只是通过建设简单的静态HTML网站来展示企业的一些基本信息，随着流行趋势的不断发展，企业的行为也开始变得多样化：有的企业发布了自己的会员卡，并且与网站绑定进行各种互动；有的企业开始与社会化媒体合作，包括微博、微信来更主动地向消费者发布信息并获得反馈；更有企业开发了自己的移动APP，试图从移动互联网终端来吸引消费者的关注。面对着层出不穷的新应用，新软硬件平台，它们之间应该如何互相有效地进行通讯，交互信息并且凝聚成一个有机的整体从而最大化其所能产生的影响力，成为了一个值得研究的课题。而利用中间件技术不但可以对既有的软硬件平台进行整合，更多更新的互动应用的开发也将在这个通用的平台上成为可能，这样企业从中获得的投资回报将会变得更为有效且作用长久。所以本文将初步研究基于中间件技术的数据传递及功能整合的主要技术方式和工具，来仔细分析其在应用和开发上的优势，从而进一步优化互动型应用的开发流程，更好地服务于最终用户。

# 1.3 论文主要内容和框架结构

论文根据于大悦城购物中心实际的市场推广需求，是一个以中间件技术为基础，整合多个不同功能模块的大型互动零售商业社区系统，着重研究了在开发系统时的关键技术、整体设计方案、如何实现与消费者的无缝连接、如何与消费者进行实时的互动、如何利用社会化媒体来最大化市场推广的效果等。

本文的后文主要做如下安排，第二章首先介绍了作为架构基础的中间件技术以及本系统中选用的Microsoft BizTalk Server、移动APP的开发技术以及与既有软硬件平台的接口，并对各功能模块的关键技术背景进行了一定的阐述。第三章中主要是分析大悦城购物中心试图通过构建该互动零售商业社区系统以提升其市场推广效果的总体背景、目的及具体业务功能需求，在此基础上进行了系统总体设计及软硬件选型。第四章中论文详细分析了如何设计与实现几大核心功能模块包括互动触摸模块、移动APP应用以及整合社会化媒体模块，并对系统开发过程中所遇到的若干难点问题及解决方案进行了介绍。在第五章的总结与展望中，总结了本文所完成的工作，对系统存在的问题与不足做出了总结，并对将来的改进方向提出了展望。

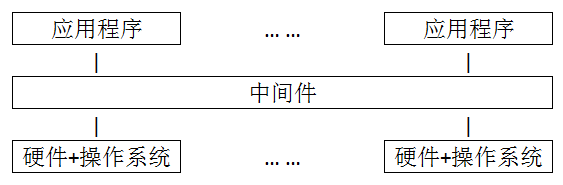
# 第二章 业务背景和关键技术介绍

# 2.1 中间件技术简介

本系统的运行需要连接多个基于不同软硬件平台的功能模块，为了提高信息传递的效率和可靠性，同时也为了系统将来可能的功能扩展，因此我们选择中间件来进行信息的集中处理及传输。

从一般意义来说，中间件（Middleware）是一种为应用程序提供基于平台（硬件和操作系统）之上的服务的计算机软件。对于软件开发者来说，中间件让通讯及输入/输出的过程变得简单而透明，因此他们可以将更多的精力投入于实现应用程序功能本身。

中间件所提供的服务有着标准的程序接口和协议。针对不同的操作系统和硬件平台，它们可以有符合接口和协议规范的多种实现。下图说明了不同的软硬件和操作系统平台通过中间件交互信息的基本流程：



**图 2-1 通过中间件交互信息的基本流程**

由于标准接口对于可移植性和标准协议对于互操作性的重要性，中间件已经成为许多标准化工作的主要部分。对于应用程序开发，中间件远比操作系统和网络服务更为重要，中间件提供的程序接口定义了一个相对稳定的高层应用环境，不管底层的计算机硬件和系统软件怎样更新换代，只需要将中间件升级更新，并保持中间件对外的接口定义不变，应用软件几乎不需要任何修改，从而保护了企业在应用软件开发和维护中的重大投资。

目前主流的中间件产品有很多，在本系统中，我们对成本和性能进行了综合的考虑，决定采用了Microsoft BizTalk Server作为中间件，并在其基础上开发了针对各功能模块的API。

Microsoft BizTalk Server常被简称为“BizTalk”，是由微软公司出品的一种中间件（企业服务总线）。其对于Web Services提供了广泛的支持[4]，并为企业量身定制了“适配器（Adapter）”，可连接不同的协议、信息格式及特定软件，使得企业经营中所使用的不同软件系统之间的通讯成为可能，此举使企业有可能大幅提高业务流程的自动化水平。

# 2.2 移动APP开发技术简介

移动APP是本系统中一个非常重要的组成部分。它承担起了与会员进行无缝连接及互动的责任。会员可以通过其实现注册及登录、绑定会员卡、查询及修改会员信息、接收促销信息推送、签到、参与网上活动等功能。

APP源自Application的简称，是一种专门设计在智能手机、平板电脑或其他移动设备上运行的软件应用程序。APP这一词汇目前已经非常流行，已在2010年被美国方言学会评为年度词汇[5]。APP通常通过移动操作系统开发商所运营的应用程序发布平台（如Apple App Store, Google Play, Windows Phone Marketplace等）发布。

目前，移动APP的普及率在持续地上升。2012年5月comScore的研究报告指出，更多的移动设备用户通过移动APP而不是传统的网页浏览方式访问其所订阅的信息，其比值分别是51.1%和49.8%[6]。ABI Research的研究报告更是指出，在2013年，整个移动APP的市场价值将会达到270亿美元[7]。

因此我们可以看到，企业通过移动APP来进行营销推广有着很大的意义。首先可以留住客户，通过用户对于移动设备的依赖来增加用户黏性，使普通用户转化为忠诚用户。其次是进行用户拓展，企业可以通过网络营销手段，以移动APP为渠道而吸引用户，从而扩大用户规模。

在本系统中，我们对于目前主流的两大平台iOS和Android进行了分析，认为iOS用户的特质更加符合大悦城的目标用户 – 年轻、时尚、动感。从而选择了支持iOS平台，而使用的开发语言则为Objective-C。

Objective-C通常写作ObjC，是一种通用的、面向对象的编程语言，其在C语言的基础上加入了Smalltalk风格的消息系统。在Objective-C中，对象不是简单地调用方法，而是传递信息，这一点与C++有着非常大的不同，其差异主要在于调用方法/传递模型这个动作。C++里类型与方法的关系严格清楚，一个方法必定属于一个类型，而且在编译时（compile time）就已经紧密绑定，不可能调用一个不存在类型里的方法。但在Objective-C，类型与信息的关系比较松散，调用方法视为对对象发送信息，所有方法都被视为对信息的回应。所有信息处理直到运行时（runtime）才会动态决定，并交由类型自行决定如何处理收到的信息。也就是说，一个类型不保证一定会回应收到的信息，如果类型收到了一个无法处理的信息，程序只会抛出异常，不会出错或崩溃[8]。

以下举简例说明Objective-C和C++的区别。

向一个被obj指针所指向的对象传递一个消息的方法，在C++中需要以下代码：

|  |
| --- |
| obj->method(argument); |

而在Objective-C里面，我们会采用：

|  |
| --- |
| [obj method:argument]; |

这两种语言各有其优势及劣势，在本文中我们将不多做阐述。

# 2.3 与周边软硬件平台接口简介

在本系统中，主要的功能模块包括互动触摸模块，移动APP应用以及整合社会化媒体模块。这些功能模块运行在截然不同的软硬件平台上，采用不同的程序编码，数据库结构也大相径庭，更采用了不同的通讯方法。如何将其有机地、无缝地整合起来，使其能够高效地运作，从而为消费者带来流畅及充满乐趣的体验，成为了一个颇具挑战的课题。所幸，Microsoft BizTalk Server为Web Services提供内置支持，而通过Web Services我们则可以顺利地解决这些问题。

在详细介绍Web Services之前，有必要简单介绍一下关于互动触摸模块以及社会化媒体的基础知识以帮助更为深入地理解整个系统的运行，由于移动APP应用在上文已有介绍，在这里就不再重复。

作为互动触摸模块的核心设备 - 触摸屏，其实是用户可以通过简单点击，或者多点触摸手势来进行控制的一种电子显示屏。通过使用触摸屏，使得用户可以直接地对其显示的内容进行控制并产生互动（例如对字体大小进行缩放），这比使用鼠标、手写板以及其他的输入设备更为直接和有效。

从技术角度来讲，触摸屏有很多种类，其中主要包括电阻式和电容式两大类：

电阻式触摸屏的工作原理主要是通过压力感应原理来实现对屏幕内容的操作和控制。这种触摸屏是一个多层结构，其中最重要的是两层非常薄而且透明的，相互面对但是被分隔开的电阻层。表面板（直接触摸的那一层）的下侧有一个电阻层，在它的正下方是另一个位于基板上的类似的电阻层。一层电阻层的上下两侧由导线连接，另一层的左右两侧由导线连接。当在表面板上施加足够的压力时，如用指尖或者手写笔进行划写，上层的电阻层产生形变并与下层电阻层发生接触。控制器同时从四个角读出相称的电流并计算接触点的位置[9]。

在此结构特性的基础上，电阻式触摸屏有着很多优点，例如其对液体和污物有着良好的绝缘性，因此电阻式触摸屏被广泛应用于餐厅、工厂、医院等环境中。同时，由于电阻式触摸屏仅需要施加压力就可以工作，所以可以用任何物体来触摸，即使戴着手套也可以进行操作。另外，电阻式触摸屏技术已经相当成熟，所以其成本相当低廉。

电容式触摸屏主要是通过人体的电流感应而进行工作的。电容式触摸屏的主体是一个其上涂有诸如铟锡氧化物（ITO）的透明导电层的绝缘体（例如玻璃），并在四个角上引出四个电极。由于人体也是一个导体，所以当手指触摸到屏幕表面时会形成一个耦合电容，而四个角的电极中的电流就会流向这个点，而电流的强弱与触点到电极的距离成正比，控制器就可以通过计算电流的比例及强弱，准确测出触摸点的位置[10]。

相对于电阻式触摸屏来说，电容式触摸屏的主要优势在于：首先电容式触摸屏只需要触摸，而不需要施加压力来进行操作，因此其灵敏度相当高；其次电容式触摸屏可以支持多点触摸，也就是说，用户可以用多个手指，甚至是双手，来对触摸屏进行直观地操作，这样不但摆脱了与传统网页雷同的“点击”动作，同时也使得一些在视觉上更为华丽的技术成为可能。基于这两个主要的优点，为了给消费者提供更为流畅的使用体验，也为了能够在触摸屏上展示更为绚丽的效果，我们在本系统中选择了电容式触摸屏。

大悦城作为一个全新的品牌登陆上海，其在品牌传播上有着相对本土企业更为迫切的需求，因此，时下最为流行的社会化媒体也进入了品牌运营者的视野而被选择作为整个互动零售商业社区系统的一个关键组成部分。

社会化媒体（Social Media）也称为社交媒体，社会性媒体。其指的是大量网络用户在由自己所创造，共享并交互信息及想法的虚拟社区和网络中进行的互动行为。所以社会化媒体有两个关键特征，一是人数众多，另一个是自发的传播，缺乏这两点中的任何一点都不能构成社会化媒体的范畴[11]。

同时，社会化媒体还有一个更为技术的定义就是“基于Web2.0的思想以及技术所开发的，允许用户自行创造并交流信息的互联网应用”[12]。Web2.0的发展赋予了互联网用户更多的主动权，激发了互联网用户对于互动，自我表达的强烈愿望，这种愿望又催生了大量眼花缭乱的技术，而发展的技术演化出更为绚丽的互动模式和应用，进一步释放互联网用户的需求。因此，社会化媒体正是基于用户支持和技术发展才得以发扬光大。

从尼尔森的研究报告来看，相对于其他类型网站，互联网用户正在社会化媒体上消耗越来越多的时间，2012年7月美国互联网用户从电脑及移动设备访问社会化媒体的总时间相对于去年同期上升了37个百分点，达到了1210亿分钟[13]。因此，对于企业来说，主动参与社会化媒体所能获得的益处已经不仅仅是共享社交信息，而在于建立企业声誉，维护品牌形象，甚至从中直接获得利润。正是在这样一种背景下，整合社会化媒体，成了企业与消费者进行有效沟通不可或缺的重要渠道。

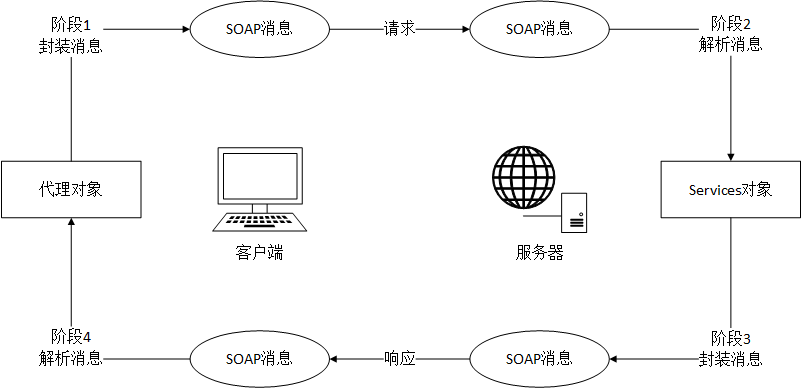
以上大致介绍了在本系统中所选择的软硬件平台，那么接下来要介绍的就是中间件如何通过Web Services与这些软硬件平台进行通讯。

Web Services扩展了互联网的基础架构，以提供让不同应用程序之间互通互联的解决方案。应用程序使用通用的网络接口和数据格式例如HTTP、XML、SOAP来访问Web Services，无需考虑各个Web Service到底是如何实现的。Web Services作为微软.NET编程模型的基础，将网络和基于组件开发的优点整合在了一起，实现了“基于Web的无缝集成”[14]。

在这里，我们看到两个非常相似的词“Web Services”和“Web Service”，很多人会认为，前者无非就是后者的复数形式。然而，这两者其实根本指的就是两样东西，Web Services是指用于架构Web Service的整体技术框架，而Web Service则是使用Web Services技术而创建的应用实例。

Web Services是描述了一些操作的接口，通过标准化的XML消息传递机制，可以通过网络访问这些操作。Web Services是用标准的、规范的基于XML的WSDL语言描述的，这称为Web Services的服务描述。这一描述囊括了与服务交互所需要的全部细节，包括消息格式（详细描述操作的输入输出消息格式）、传输协议和位置。该接口隐藏了服务实现的细节，允许通过独立于服务实现、独立于硬件或软件平台、独立于编写服务所用的编程语言的方式使用该服务。这使得基于Web Services的应用程序具备松散耦合、面向组件和跨技术实现的特点[15]。

Web Services的调用过程如下图所示：



**图 2-2 Web Services的调用过程**

在这个过程中，首先，客户端取得服务器端的服务描述文件WSDL，解析该文件的内容，了解服务器端的服务信息以及调用方式。然后根据需要生成SOAP请求信息（指定调用的方法以及调用的参数）并发往服务器端。服务器端接受到客户端发来的SOAP请求信息以后，对其中的方法调用和参数格式进行解析，根据WSDL和WSML的描述，调用相应的COM对象来完成指定功能，并把返回值封装入SOAP消息返回给客户端。客户端在接受到服务器端返回的SOAP回应消息以后，通过解析得到返回值[16]。

# 第三章 系统总体设计

# 3.1 项目背景介绍

“大悦城”是中粮集团精心打造的购物中心品牌。自2007年北京西单大悦城开业以来，这座西单商圈唯一的Shopping Mall迅速成为时尚达人、流行先锋、潮流新贵休闲购物的首选之地，同时也成为2008年北京商业地产的最大亮点。

“上海大悦城”作为这一成功的商业地产品牌向上海 – 中国东部最大的经济中心 - 进军的前哨站，对于中粮集团来说有着非常重要的意义。在整体规划中，上海大悦城将屹立在上海的苏河湾地区，并成为其中最为瞩目的大型城市综合体。项目整体开发超过40万平米，总投资额超过120亿，预计在2017年将全面竣工。在这个城市综合体中，包括大型时尚购物中心、豪华公寓、高品质酒店、甲级写字楼以及步行商业街，其意图在于打造海派文化与国际文化的兼容并蓄，创造动静结合的高品质生活。

但是同时，大悦城这一外来品牌也面临着很多现实的问题。例如其是否能够适应当地的文化及消费环境？稍嫌不足的选址（其座落于西藏路曲阜路口，距离南京路购物街约一公里）是否会严重影响客流？相比在本地非常强势的已有商业地产品牌（如正大、恒隆、港汇、久光等）能否尽快建立其知名度？其定位是否能有效接触到目标消费群体？等等现实问题都促使中粮集团必须在战略和战术上进行创新以取得更大的竞争优势。

作为走在时尚前沿的现代都市，上海有着与北京截然不同的商业环境。相较而言，上海的消费者对新事物以及新技术有着更为开放和积极的心态。时下流行的移动APP，微博等营销手段，在上海的渗透率也非常高。为了更好地适应本土文化，更好地服务本地消费群体，上海大悦城决定独辟蹊径，同时开发一系列时尚而又新奇的概念为其目标消费群体 - 年轻消费者 – 构建一个全球化的体验空间。

同时，上海大悦城采用最为先进的互动理念与IT技术来打造一个互动零售商业社区，从各角度全方位地去接触消费者并与其发生互动，争取以最快地速度扩大其影响力和知名度，得到消费者的认同，增加客户粘性，稳定客流，从而支持其业务的发展。

建设这样一个互动零售商业社区系统的特殊之处在于，它不再仅仅是一个向消费者展示和销售商品的“网上商店”，也不仅仅是一个信息传达及媒体发布的“门户网站”，而是一个通过整合Web2.0及各种前沿技术而实现的虚拟与现实边界模糊的电子商业社区。消费者所获得的体验不仅仅是传统电子商务网站的信息查询，在线购物等已经被普遍应用的陈旧功能。而是通过将虚拟世界与现实世界的有机结合，使消费者真实地全过程参与到这种体验式的购物行为中去，全方位地接受信息并为其所影响，让消费者理解并接受大悦城这一全新的品牌，让大悦城这三个字不仅仅代表一个购物中心的名称，同时也代表一种全新的生活方式和流行趋势。

# 3.2 系统需求分析

为实现与消费者的互动，大悦城互动零售商业社区系统有着如下的主要功能需求：

品牌管理：

系统运行的数据基础。由系统管理员向各商户分发商户用户帐号，商户用户可在系统中实现下列品牌管理功能：

**表3-1 品牌及商品管理需求**

|  |  |
| --- | --- |
| 功能名称 | 功能描述 |
| 新增品牌 | 管理账户可点击新建品牌按钮进入页面，修改品牌属性字段后保存；品牌的描述或介绍需使用所见即所得在线文本编辑器录入（+图片、视频）；品牌中的品牌图片需定义并上传到附件表中，在附件表中维护图片的相关属性 |
| 删除品牌 | 不删除实体数据，仅将标识位写为无效 |
| 查看品牌 | 需分配读取权限的后台管理账户才可以查看品牌列表，管理员账户在后台能够根据品牌的部分关键字或字段进行搜索 |
| 修改品牌 | 大悦城管理员可在后台用户查看品牌列表，点击需要修改的品牌进入品牌信息维护页面，维护品牌的信息；商家可在店铺设置页查看并修改自己具有的品牌信息 |

会员中心：

包括会员个人资料、安全选项、个人设置、消息管理、订单管理等。功能清单如下表所示：

**表3-2 会员中心**

|  |  |
| --- | --- |
| 功能名称 | 功能描述 |
| 会员信息 | 基本信息，昵称、姓名、所在地、生日、性别、介绍，显示在首页的设置，用户可查看并随时修改保存；  有关绑定的会员卡号、手机号、身份证号，需要有机制保证用户填写内容的真实性，用户Email作为登录凭证具有唯一性是不允许修改的；  相关数据（例如邮编、身份证号等）的修改等都需验证填写是否合法，所在国家、所在省、所在市都应为下拉菜单形式并显示对应数据 |
| 安全选项 | 密码修改功能，用户输入原密码、新密码，新密码确认，修改密码成功时需保存用户的修改历史；  输入身份证信息保存，所有的身份证修改历史，可作为在客服处取回账号的依据 |
| 个人设置 | 通知设置：可对所有收到的消息/评论通知提醒，用户具有在团购、特卖、预定、等应用各项设置，默认打开；RFID相关通知考虑直接读取显示；  隐私设置：可对评论回复、消息接收、首页个人信息显示、地理位置、勋章、个人活动信息显示等隐私选项进行设置（参考新浪微博）。  同步设置：评论时的同步到微博选项，团购、特卖时是否同步活动到微博的选项，以及其他考虑同步选项，默认都为打开；黑名单：可管理用户拉黑的用户，黑名单用户将无法关注、评论当前用户，一旦当前用户关注某个黑名单用户，将会自动解除拉黑 |
| 消息管理 | 用户可接收的系统应用消息：团购支付成功、团购失败；特卖品支付成功；预定服务成功、取消预定、变更预定成功；  用户可删除本地站内消息，系统消息由大悦城发出，用户必须接收系统消息，而系统应用消息具有配置选项，用户可在个人中心配置的通知设置中，关闭某项应用发出的系统应用消息；  微博消息由于数据来源的不同，应当与系统消息明显区分，放在不同的标签页下：   * 我的首页：用户的所有关注账户发言（类似新浪微博用户页面）； * 我的微博：用户自己的发言，包括评论时同时转发到自己的微博内容； * 私信对话：参考新浪微博私信，消息分为已读和未读消息，未读消息高亮显示并显示未读消息数目，要有消息删除和回复功能； * 我的评论：对应微博账户发出的评论显示评论的主体，时间，和所评论的原文链接；   所有的积分消费使用数量直接在使用时进行页面提示，不放入消息提醒中 |

新浪微博管理：

整合了新浪微博的大悦城官方网站，可以通过系统后台对微博内容进行管理和调整，所需功能如下：

**表3-3 新浪微博管理**

|  |  |
| --- | --- |
| 功能名称 | 功能描述 |
| 预览微博 | 博文维护列表在后台管理以缩略形式展示，单击博文封面或标题链接可预览当前博文 |
| 添加微博 | 在所见即所得超文本编辑器下编辑，以便管理员能控制内容色彩和排版，并嵌入图片、视频、音频 |
| 编辑和发布 | 保存后不会立即显示在前台页面上，后台具有“发布”按钮，点击发布后前台才能访问到该微博；  微博“发布”按钮点击后，自动同步到大悦城新浪官方认证微博，管理员也可使用“同步到微博”的按钮重新同步一条微博；  一旦此微博被发布，后台的“发布”按钮内容变成“取消发布”，点击取消发布此条微博在前台就不能被访问到 |
| 删除微博 | 微博的删除是标识记录为无效数据，放入回收站，在回收站中才能真正删除数据 |

移动APP应用：

针对系统设计所定义的目标消费者群体，移动APP应用的初始体验应该为：页面一目了然、不需要花时间研究功能、并且有相关的联想或搜索功能。其具体功能需求为：

表3-4 移动APP应用

|  |  |
| --- | --- |
| 功能名称 | 功能描述 |
| 用户信息 | 可查看会员积分、会员级别、签到计数、徽章、订单等信息 |
| 用户设置 | 通用的设置项：地理位置提醒设置、账户提醒设置；配置应用的主题风格。在日志中保存应用修改记录。对tag扩展功能，新增和修改 |
| 微博登录 | 与新浪微博的iPhone应用类似 |
| 促销活动 | 将优惠信息/促销信息按列表排序展示，可按地址、店铺名、品牌进行搜索，推荐的所有商品可能是促销品，也可能是优惠券/卡 |
| 商户导航 | 当用户具有签到信息，在商户导航中增加条件搜索精选优惠、折扣；根据条件列出当前位置附近品牌店 |

互动触摸系统：

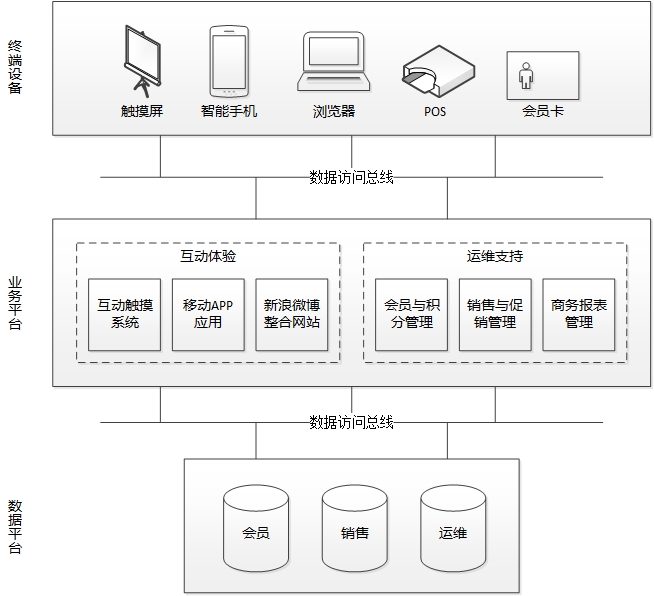
作为在商场内的主要互动终端，互动触摸屏需要展示清晰无闪烁、操作流畅、响应迅速，其主要功能需求如下表所示：

**表3-5 互动触摸系统**

|  |  |
| --- | --- |
| 功能名称 | 功能描述 |
| 用户登录 | 通过输入会员卡号、手机号作为用户名进行登录。登录后可查看会员基本信息：用户名、积分，上一次的签到记录 |
| 用户签到 | 登录后显示相应楼层的商户列表，用户可选择一个或多个进行签到工作，签到完成后累积签到积分 |
| 促销活动 | 将优惠信息/促销信息按列表排序展示，可按地址、店铺名、品牌进行搜索，推荐的所有商品可能是促销品，也可能是优惠券/卡 |
| 商户导航 | 当用户完成签到，在商户导航中增加条件搜索精选优惠、折扣；根据条件列出当前位置附近品牌店 |

# 3.3 系统总体架构设计

大悦城互动零售商业社区系统总体上采用面向服务（Service-Oriented Architecture，SOA）的架构。系统在逻辑上可以分为三个部分，即终端设备、业务平台以及数据平台。如下图所示：



**图 3-1 系统总体设计**

在终端设备部分，系统支持触摸屏、智能手机、浏览器、POS以及会员卡等多种设备。用户可以在这些设备上获取系统发布的信息，同时输入信息与系统发生互动。而这些设备则通过Web Service与业务平台进行通讯，为业务平台的运行提供数据。

业务平台是处理所有信息的中枢。在这里，系统按照功能的不同将其分为两大逻辑部分 – 互动体验和运维支持。在互动体验部分包含互动触摸模块，移动APP应用以及新浪微博整合网站三个模块。其中，互动触摸模块的功能主要是在互动触摸屏终端上向用户展示包括广告、促销等信息，并且同时提供商户信息查询、会员信息查询、会员签到、折扣券查询等功能。移动APP应用则主要提供了会员信息查询、积分查询、会员卡查询、折扣券查询、商户推荐，并整合了大悦城的官方新浪微博。新浪微博整合网站的功能主要是实现了在大悦城的官方网站中整合了新浪微博，会员和系统管理员都可以通过此功能实现大悦城官方网站与新浪微博的无缝连接。这三个模块是互动体验的核心模块，所有与互动体验相关的业务逻辑都在这里实现。

运维支持部分主要提供一系列与商务运营有关的功能。会员与积分管理部分处理所有的会员信息，并对会员通过消费、签到、互动活动等行为产生的积分进行管理。销售与促销管理则负责处理所有的商品信息，包括商品基本信息、价格信息、库存信息以及通过POS系统产生的销售数据，同时管理所有的促销行为，包括折让、买赠、试用等。商务报表管理则为企业的运营者提供了一个多样化的报表系统，其内容包括商户业绩分析、消费行为分析、精准促销分析等报表。

业务平台同时也通过Web Service数据访问总线实现所有与数据平台的通讯，即完成数据读取和持久化的功能。

数据平台负责对于所有的运营数据、会员数据、媒体内容进行存储。系统按逻辑和性能将数据平台划分为会员、销售和运维三个数据库。“会员”数据库负责存储所有与会员相关的信息，主要包括：会员的基本信息如姓名、电话、身份证号码、车牌号码等；会员的消费记录、会员的积分信息、会员的签到记录等。“销售”数据库存储所有与销售行为相关的信息，主要包括：商品品类、品名、价格、进销存、销售记录等。“运维”数据库则存储所有有关企业运维的信息，主要包括物业信息、合同、日常运营等数据。

# 3.4 系统软硬件选型

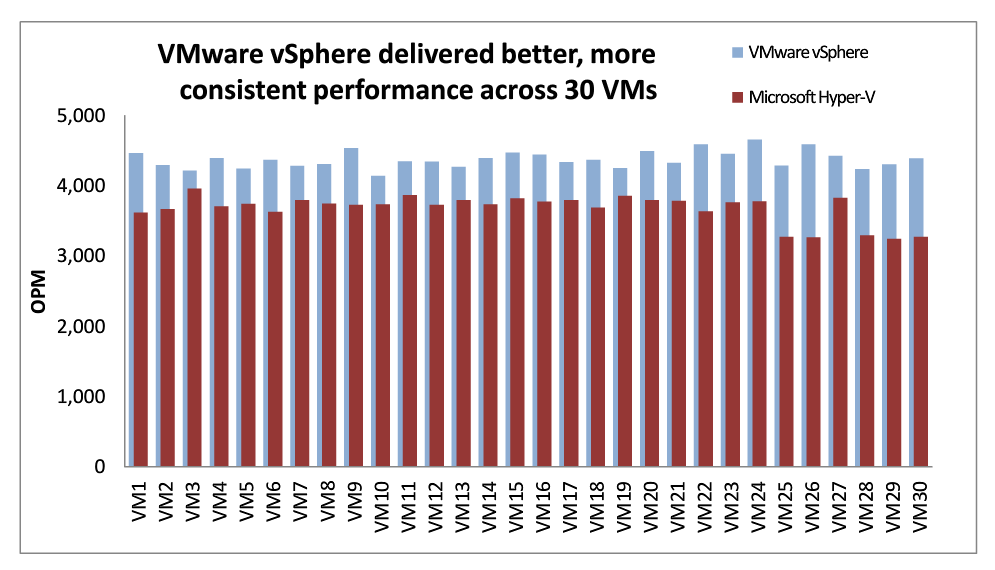
本系统中涉及到大规模的数据运算、传输与存储，为构建一个稳定而高效的运行环境，我们在软硬件选型上做了以下安排。

首先我们来看硬件部分，由于当前的服务器硬件性能已经达到一个非常高的水平，所以从成本优化的角度考虑，我们决定采用虚拟化技术。硬件部分的主要选型列表如下表所示：

**表 3-6 硬件配置清单**

|  |  |  |
| --- | --- | --- |
| 服务器及存储 | 配置及说明 | 用途 |
| Dell(TM) PowerEdge(TM) R710  机架安装服务器 | 2颗 Intel(R)四核E5506 Xeon(R) CPU，2.13GHz，4M高速缓存，QPI达4.86 GT/秒  Broadcom 5716双端口千兆以太网  64GB Memory (8x8GB), 1333MHz, Dual Ranked RDIMMs for 2 Processors  2个 450GB, 3.5", 15Krpm, SAS 硬盘,支持热插拔  SAS6i/R RAID 控制卡, 支持RAID0/1 | 数据库 |
| Dell(TM) PowerEdge(TM) R910  机架安装服务器 | 2颗 Intel(R) 4-Core E7520 Xeon(R) CPU,1.86GHz,18M Cache,4.80 GT/s QPI,Turbo  64GB Memory (16x4GB) 1066MHz Quad Rank BIC RDIMMs for 1/2/4CPU  PERC H200 Intergrated RAID Controller  2个 300GB 10K RPM,6Gbps SAS 2.5英寸热插拔硬盘  Internal Dual SD Module with 1x 1GB SD Card | 虚拟化 |
| NETSTICK FS-2404SR 8GB/S SAS-FC  双控磁盘阵列 | 提供双冗余双总线PowerPC ASIC 677 RISC架构RAID热插拔控制器  双控制器配置4GB内存,具备内存保护技术  提供主机通道：4条8GB/S Fibre Channel for Hosts光纤通道  可达到2,800MB/s读性能 与1,300 MB/s写性能  自动提供路径冗余及故障切换功能  主柜支持磁盘通道：24个SAS/SATAⅡ混插热插拨磁盘接口，还可再扩充SAS/SATA JBOD扩展柜。主柜结合专用扩展柜最多可提供104个磁盘接口  提供RAID JBOD, 0, 1, 3, 5, 5+Spare,6, 10, 30, 50, 60, NRAID | 存储 |
| NETSTICK 路径冗余模块 | NETSTICK Active-Active/Active-Standby Path Redundancy、Intelligent Load-Balancing 、I/O traffic monitoring 、Multiple paths to RAID volumes | 存储 |
| NETSTICKHD-1000SATA | 1000G SATA 7200转 磁盘 (配置16TB存储使用容量) | 存储 |
| 8G-GBIC-SFP(short ave) | 多模光纤模块，用于光纤阵列的级联及与HBA卡的连接接口 | 存储 |
| CB-FC-LC-5M | 8Gb LC-LC接口 5m 光纤线 | 存储 |
| Qlogic SB5800 光纤交换机 | 8GB/S | 存储 |
| Qlogic QLE2560 HBA 适配器 | 8GB/S | 存储 |

在软件部分，我们主要采用了Microsoft服务器操作系统和数据库软件，但是虚拟化技术则选用了VMware vSphere （以下简称VMware）。在这里，并不是没有考虑过Microsoft Hyper-V Server (以下简称Hyper-V)，而是因为相对于当时在业界应用尚不是非常广泛的Hyper-V而言，VMware有着更多的成功案例，证实其有着更高的性能和可靠性，并且也更容易获得技术支持。同时，从Principled Technologies®所提供的测试报告中可以看出，在从加载1到30个虚拟机的情况下，VMware的性能均明显优于Hyper-V[17]。



**图 3-2 VMware与Hyper-V性能对比**

因此，综合考虑了系统安全及性能等各方面，最终，我们决定在本系统中使用VMware来实现虚拟化。本系统所选用软件的主要清单如下表所示：

**表 3-7 软件配置清单**

|  |  |  |
| --- | --- | --- |
| 名称 | 版本及说明 | 用途 |
| VMware Enterprise Plus Acceleration Kit  For 8 processors | Includes vSphere Enterprise Plus for 8 Processors, 1 vCenter Server Standard | 虚拟化 |
| VMware vSphere 4 Enterprise Plus  For 1 processor | Max 12 cores per processor | 虚拟化 |
| Microsoft BizTalk Server | BztlkSvrEnt 2009 CHNS OLP NL 1Proc | 中间件 |
| Microsoft Windows Server | WinSvrEnt 2008R2 CHNS OLP NL | 操作系统 |
| Microsoft Windows Server | SysCtrCnfgMgrSvrMLEnt 2007R2 CHNS OLP NL | 操作系统 |
| Microsoft Windows Server | SysCtrCnfgMgrSvrwSQL 2007R2 CHNS OLP NL | 操作系统 |
| Microsoft Windows Server | SQLSvrEnt 2008R2 CHNS OLP NL 1Proc | 数据库 |

# 第四章 系统详细设计

# 4.1 系统模块设计

在本系统中，从业务角度来看，可以分为七个功能模块，分别是互动触摸模块、移动APP应用、整合社会化媒体模块、会员与积分管理模块、结算与销售管理模块、商户运维管理模块以及智能决策模块，如下图所示：



**图4-1 系统模块概览**

互动触摸模块通过设置在购物中心各楼层的触摸屏为消费者提供了最为直接的交流平台。本系统中互动触摸模块的功能主要为会员签到、会员积分奖励、商场公告和活动广告推送、店铺活动信息推送等。消费者在购物过程中可以通过触摸屏查看楼层向导和商场近期活动广告，通过输入会员信息进行签到可以获取会员积分，同时商户的店铺推介短信息会被发送至消费者的移动设备上。消费者可以选择自己感兴趣的商户进行了解，还可以利用签到功能集中获取一些消费优惠券，更有机会参与商场组织的抽奖或其他活动，让消费者购物、休闲两不误。

移动APP应用可以作为消费者的电子助手来寻找各种优惠促销信息、查看通知、发布微博，并且能够用来管理折扣券、积分、排号等会员信息，给消费者提供全新的、完美的购物体验。同时移动APP应用也充当了互联网站的精华提炼和引导入口。传统门户网站最大的不便之处在于需要消费者主动访问。网站内容再吸引人，消费者不访问也是徒劳。本系统可以将网站的精华内容在移动APP中进行发布，利用程序主动推送信息的优势和移动终端随时随地随身的特点让消费者进行查看，进而吸引用户主动访问互联网站查看更多精彩内容。

整合社会化媒体模块通过对新浪微博的整合以实现大悦城官方网站上所发布的促销、活动等内容能够实时地同步到新浪微博，同时也将大悦城的会员帐号和会员所注册的新浪微博帐号进行绑定，使会员也可以将对大悦城官方网站内容的评价同步到自己的微博帐号中去。本功能模块提供了信息在两个系统中的无缝切换，大大提高了大悦城购物中心的媒体发布效率，从而更为亲密地与消费者保持随时随地的接触。

会员与积分管理模块包含了会员信息管理及会员积分管理。会员信息管理主要针对会员的身份资料如用户名、密码、电话号码、地址等基本信息。会员积分管理记录会员的消费历史、签到记录、活动参与、积分累计与消费，并提供了积分规则。这一功能模块记录了会员的所有活动，为整个互动系统提供分析数据。

结算与销售管理模块关系到企业的盈利与流动资金的管理，在购物中心中发生的所有消费行为都由该模块进行管理。由于购物中心的业态特点，商户的销售必须通过购物中心提供的结算系统进行结算，这也就是为什么在购物中心中往往设置有结帐台的原因。一般结算付款主要采用月结方式，也可采用按商品或活动批次进行结算。因此有时候也可能会出现商户只结算一部分费用，并不完全结算的情况，所以系统也可按具体的实际情况，结算商户想要结算的那部分费用。本功能模块可以让购物中心及商户的管理人员及时了解营业帐款结算的情况，从而掌握企业的经营状况。

商户运维管理模块为商户提供了运营及维护管理的工具。本功能模块一方面可以实现进销存管理的功能，对商户运营中商品、现金的流动进行跟踪，实现了管理的即时性，让商户随时了解自己的运营状况。同时也提供了商户信息管理、促销信息管理、消费者互动信息管理等支持功能。商户可以随时通过该功能模块发布或更新自己的促销信息，并通过互动社区传递给消费者，最大化促销活动对销售的促进作用。

智能决策模块结合了人工智能和决策支持系统，同时应用了专家系统，从而更为充分地应用人类的推理性、过程性和描述性知识，通过逻辑推理来帮助解决复杂的决策问题。该模块基于会员信息、商户信息、会员行为记录、销售记录、互动信息等数据，充分利用各层次的资源，通过商务智能进行数据挖掘、统计和分析，为商户和购物中心提供销售分析、份额分析、促销效果分析、消费行为分析、潜在需求分析、精准促销分析等管理决策建议。

# 4.2 互动触摸模块设计

互动触摸屏以其易于使用、坚固耐用、反应速度快、节省空间等诸多优点，极大地简化了计算机的使用。即使是对计算机研究较少的女性及中老年用户，也能够迅速熟悉触摸屏所提供的这些既实现了实时交互，又提高了趣味性的功能，从而在零售商与消费者之间架设一座新颖、生动、实用、高效而直接的沟通桥梁。

在这一节中，我们主要讨论一下互动触摸系统的运行是一个怎么样的流程，然后针对在系统设计过程涉及的主要代码：签到功能和积分奖励进行详细的阐述和说明，以求获得更加细致而深入的理解。

在一开始，我们有必要了解一下关于签到的基本流程：1.输入会员卡号；2.输入手机后四位；3.选择相应的楼层；4.选择提交。当完成了上述流程以后，消费者就可以在触摸屏上看到系统已经显示出了自己在当前所处位置，以及在当前位置可以进行签到的商户。这时候可以对想要签到的商户进行选择，在完成了选择以后进行提交，然后整个输入流程宣告结束。这时候就完成了对一家（或多家）商户的签到操作。

从代码的分析来看，首先，我们在一开始引用了System、System.web等命名空间（Namespace），如下所示：

|  |
| --- |
| using System;  using System.Web;  using System.Data;  using System.Data.SqlClient; |

**图 4-2 所引用的.NET命名空间**

命名空间是表达多个变量和多个函数组合成一个组的方法，其主要作用是解决了命名（用户定义的类型名、变量名、函数名）冲突的问题，可以使得整个系统更为模块化。

以上所引用的命名空间的作用在于其创建了代码运行的基础环境，我们在这里使用了using指令。通过“using + 命名空间”这样的方式，我们就可以在程序中直接使用命名空间中的类型，而不必指定类型的详细命名空间，类似于Java的import，这个功能非常常用，几乎在每个.NET的程序中都会被使用到。

其中System 命名空间包含基本类和基类，这些类定义常用的值和引用数据类型、事件和事件处理程序、接口、属性和异常处理[18]。System.Web 命名空间提供使得可以进行浏览器与服务器通信的类和接口。此命名空间包括 HttpRequest 类（用于提供有关当前 HTTP 请求的广泛信息）、 HttpResponse 类（用于管理输出到客户端的 HTTP 输出）以及 HttpServerUtility 类（用于提供对服务器端实用工具与进程的访问）。System.Web 还包括用于 Cookie 操作、文件传输、异常信息和输出缓存控制的类[19]。限于篇幅，我们在这里就不一一解释所有的命名空间的功能及作用了。

在日常的网页访问中，用户会输入想要访问的URL并点击Enter，然后浏览器向服务器发送请求，服务器收到请求后将请求的页面发送回浏览器，在这个时候我们就会发现，由于网络速度，服务器繁忙等等各种原因，很有可能出现页面无法加载，或者仅加载了一部分的情况。

而触摸屏系统的有一个非常与众不同的特点，用户始终是直接在屏幕上进行操作，如果在使用过程中发生屏幕闪烁或者重新载入导致的白屏，将严重影响消费者的使用体验。所以我们在刷新页面时，不能使用常规的跳转页面方式，而需要在同一个页面上进行刷新，所有的内容都需要在同一个页面上进行相应的显示。也就是异步加载，这种方式不会影响已加载完毕的页面浏览。所以触摸屏系统的数据交互是通过Ajax调用来进行处理的。在本系统中，我们通过javascript与后台HttpHandler（一般处理程序）类来实现。

以发送会员身份验证请求为例，相关代码如下所示：

|  |
| --- |
| function PostMsg(){  //return 0; //成功返回0，重复签到返回1，信息错误返回2  var cardid=document.getElementById("input1").value;  var lastnumber=document.getElementById("input2").value;  var louceng=escape(document.getElementById("input3").value);  if (window.ActiveXObject) xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");  else if (window.XMLHttpRequest) xmlHttp = new XMLHttpRequest;  var ret  xmlHttp.open("POST", "../Handlers/Sign.ashx?membercardid="+cardid+"&phonelast="+lastnumber+"&floor="+louceng, true);  xmlHttp.onreadystatechange = function(){  if (xmlHttp.readyState == 4 && xmlHttp.status == 200){  ret= xmlHttp.responseText;  }  }  xmlHttp.send(null);  } |

**图 4-3 javascript调用HttpHandler来发送会员身份验证请求**

在上述代码中，我们可以看到Ajax与服务器异步交互的核心，也就是XMLHttpRequest指令。XMLHttpRequest可以在不重新加载页面的情况下更新网页，并且从后台向服务器发送数据。

首先我们定义三个变量，分别为cardid，lastnumber以及louceng，分别代表会员卡号，手机号码后四位以及楼层。

然后我们用XMLHttpRequest指令来初始化一个对象“xmlHttp”，通过“open”指令建立与服务器的URL的连接，并采用“POST”来指定发送请求的方法，以及以“ture”来表明采用异步方式。下一步我们设定监听XMLHttoRequest状态改变的事件处理函数“onreadystatechange()”，最后使用“send()”来发送请求。当完成了上述步骤以后，我们就实现了会员身份验证的请求发送。

然后，我们来研究一下后台如何接受并响应由前端发出的请求，相关的代码如下：

|  |
| --- |
| public void ProcessRequest(HttpContext context)  {  context.Response.ContentType = "text/plain";  string floor = context.Request["floor"].ToString();//楼层  string phoneLast = context.Request["phonelast"].ToString();//手机后四位  string userCard = context.Request["membercardid"].ToString();//会员卡号  //通过WebServices调用第三方会员积分api(会员卡签到加积分)  WebSign.Request req = new WebSign.Request();  req.PosVIPCode = userCard;//触摸屏签到输入的会员卡号  WebSign.Service ws = new WebSign.Service();  WebSign.Response rs = new WebSign.Response();  rs = ws.VIPQuery(req);//调用会员查询api，通过会员卡号返回会员信息  if (rs.Result.ToLower().Equals("true"))//返回string类型true表示成功，返回其他信息则表示调用会员查询api失败  {  string tempPhoneLast = "";  if (!String.IsNullOrEmpty(rs.Telephone))  {  tempPhoneLast = rs.Telephone.Substring(7, 4);  }  if (phonelast == tempPhoneLast)//手机号匹配，如果触摸屏输入的手机后4位与api返回的手机后4位匹配则进行下一步操作  {  string isExist = GetSignInfo(userCard); //查询会员卡是否在数据库里面有签到记录  if (isExist == "0")  {  //重复签到并返回结果  context.Response.Write("1" + "," + returnscore.ToString() + "," + signcount.ToString() + "," + telphone + "," + cardNum);  }  else  {  //未签到并返回结果  context.Response.Write("0" + "," + returnscore.ToString() + "," + signcount.ToString() + "," + telphone + "," + cardNum);  }  }  }  } |

**图 4-4 HttpHandler验证会员身份逻辑模块**

在这里我们有必要解释一下什么是HttpHandler。作为一个底层的请求及响应API，HttpHandler被用于处理所接受到的Http请求。HttoHandler通过处理程序（Handler）实现了IHttpHandler接口，在使用中我们没有必要去引用任何额外的命名空间来使用HttpHandler，因为它已经被包含在System.Web命名空间中了。而所谓的处理程序（Handler）则有点类似于Internet服务器应用程序编程接口（ISAPI）的扩展。

上述代码中，首先实现了IHttpHandler接口的方法“ProcessRequest()”（该方法将HttpContext用作参数，这使它能够访问Request和Response内部对象），该方法将用来处理会员身份验证的请求。

然后定义了三个字符类型的变量，用来接收请求中所接收到的一系列数据包括floor（楼层），phoneLast（手机号码后四位）及userCard（会员卡号），如下图所示：

|  |
| --- |
| string floor = context.Request["floor"].ToString();//楼层  string phoneLast = context.Request["phonelast"].ToString();//手机后四位  string userCard = context.Request["membercardid"].ToString();//会员卡号 |

**图 4-5 会员身份信息变量**

接下来将通过Web Services调用中间件会员信息的查询接口。在这个过程中，首先使用WebSign.Request来实例化一个对象“req”，然后对req对象中的PosVIPCode属性进行赋值，该值为“userCard”，代表通过触摸屏签到而输入的会员卡号。接下来，使用WebSign.Service继续实例化一个对象“ws”，该对象将被用来调用接口方法“VIPQuery()”。第三步，通过WebSign.Response再实例化一个对象“rs”来接受响应的结果，其使用方式为rs = ws.VIPQuery(req)。至此，就完成了对接口初始化的操作，接下去的代码将对输入的信息进行验证及匹配。

由于Web Services是使用XML来传输数据的，所以在这里也必须介绍一下VIPQuery()方法请求和响应的XML格式，限于篇幅，在这里仅对请求的格式做一些简单的介绍，如下图所示：

|  |
| --- |
| <Request xmlns="http://JoyCity.Common.VIPQuery">  <UserID xmlns="">**string**</UserID>  <PosVIPCode xmlns="">**string**</PosVIPCode>  <VIPCardNo xmlns="">**string**</VIPCardNo>  <RequestTime xmlns="">**dateTime**</RequestTime>  </Request> |

**图 4-6 会员身份验证请求的XML格式**

在上述的XML代码中，可以看到有三个变量，其中UserID代表用户ID，PosVIPCode代表POSVIP编号，VIPCardNo代表会员卡卡号，因为这三个变量都具有唯一性，所以任意输入一个都可以完成对数据的查询。

接下来，让我们回到系统中。当我们通过触摸屏输入一个会员卡号及一个手机号码的后四位数字后。首先，系统会根据会员卡号来验证会员信息，即判断该会员是否存在，如果存在，系统将返回一个值“ture”，相对应的代码是“rs.Result.ToLower().Equals("true")”。然后我们定义一个临时变量“tempPhoneLast”用来接受从中间件会员信息查询中返回的该会员的手机号后四位。接下来，我们将临时变量“tempPhoneLast”与会员在现场输入的手机号“rs.Telephone.Substring(7, 4)”进行匹配，如果匹配成功，系统将继续执行下一步的操作。

在下一步的操作中，首先还是定义一个变量“isExist”，该变量通过“string isExist = GetSignInfo(userCard)”来获取赋值，用来判断所查询的会员卡卡号在数据库中是否有签到的记录。判断的方法如下所示：

|  |
| --- |
| private string GetSignInfo(string membercardid)  {  string sql = string.Format(@"select top 1 datediff(day,Lasttime,getdate()) flag from Jc\_Ucount where Utype=2 and UserId={0} order by LatTime", membercardid);//Utype=2,会员类型，3-认证会员，1-网站会员  object obj = Maticsoft.DBUtility.DbHelperSQL.GetSingle(sql);  if (obj!=null)  {  if (obj.ToString() == "0")  {  return "0";  }  }  return "1";  } |

**图 4-7 查询会员签到记录**

在以上代码中我们根据参数memberCardId（会员卡号）查询该会员之前最后一次签到的记录，并使用“SQL Server datediff()”函数返回最后一次签到日期与当前日期之间的天数，若返回结果为“0”，说明当天有签到记录，当结果大于0时，说明当天没有签到记录，同时会输出相应结果信息返回给前端请求页面，并显示处理的界面。至此整个会员信息验证流程全部走完。接下根据验证信息输出的结果，在前端页面即触摸屏上加载店铺列表显示页面。用户可以选择不同楼层的店铺，最多选10个，点击确定按钮提交，该部分实现效果如下图所示：

|  |  |
| --- | --- |
| C:\Users\mm\Desktop\火狐截图20130727152336958.png |  |

**图 4-8 签到提交实现效果**

这时候，系统将创建指向ShopSendMessage.ashx处理页面异步请求。由于该部分代码含有多个逻辑，下面将会将其分成几个阶段进行解释。

首先我们定义几个公共变量，这些变量将在后续代码中被普遍使用。同时，我们使用了“int signResult=AddSignInfo(userCard,allShopId)”方法进行签到记录的保存，如下所示：

|  |
| --- |
| public void ProcessRequest(HttpContext context)  {  context.Response.ContentType = "text/plain";  string telPhone = context.Request["phone"];  string allShopId = "";  if (context.Request["allid"] != null)  {  allShopId = context.Request["allid"].ToString().TrimEnd(',');  }  string userCard = context.Request["card"];  string floor = context.Request["louceng"];  string ipCard = context.Request["cardNum"];  string shopName = "";  //保存签到记录  int signResult=AddSignInfo(userCard,allShopId);  } |

**图 4-9 定义公共变量及保存签到记录**

以上的变量分别为telPhone（手机号码），allShopId（店铺ID），userCard（会员卡号），floor（楼层）以及ipCard（ip卡号）。

“int signResult=AddSignInfo(userCard,allShopId)”方法的实现逻辑如下：

|  |
| --- |
| public int AddSignInfo(string usercard,string shopid)  {  //往数据库里插入数据  int score = 5;  string strSql = string.Format(@"insert into Jc\_Ucount(UserCard,Score,ShopId,Lasttime)values({0},{1},{2},getdate())", usercard, score, shopid);  int rows = Maticsoft.DBUtility.DbHelperSQL.ExecuteSql(strSql);  return rows;  } |

**图 4-10 保存会员签到记录的方法**

在以上代码中，我们首先接受到了两个参数，一个是usercard（会员卡号），另一个是shopid（店铺ID）。同时我们给签到积分一个5分的初始值“int score = 5”。然后进行了数据库语句的构造并把这个语句赋给我们所定义的另外一个变量“strSql”。最后我们调用“int rows = Maticsoft.DBUtility.DbHelperSQL.ExecuteSql(strSql)”方法来执行strSql变量中的SQL语句。如果该语句执行成功，它会返回一个大于“0”的数据库记录行数，代表签到记录已经被成功地保存在数据库中。

与此同时，系统也对会员的积分进行了更新，执行的代码如下所示：

|  |
| --- |
| public void CardPointUpdate(string UserCardNo)  {  int uscore = 0;  string strSql = string.Format(@"select sum(Score) from Jc\_Ucount where UserCard={0}", UserCardNo);  uscore = Maticsoft.DBUtility.DbHelperSQL.GetSingle(strSql);  //签到成功，会员卡加积分  WebRefMemberCardPoint.Service ws = new WebRefMemberCardPoint.Service();  WebRefMemberCardPoint.Response rs = new WebRefMemberCardPoint.Response();  WebRefMemberCardPoint.Request rq = new WebRefMemberCardPoint.Request();  rq.VIPCardNo = UserCardNo;  rq.Point = uscore;  rq.Type = "A";  rq.TXTime = DateTime.Now;  rq.OrderNo = DateTime.Now.ToString("yymmddhhss");  rs = ws.PointUpdate(rq);  } |

**图 4-11 会员积分的更新**

首先我们接受一个会员卡号的参数“UserCardNo”，同时定义一个会员积分的变量“uscore”且赋初始值“0”。并构造一个数据库语句“string.Format(@"select sum(Score) from Jc\_Ucount where UserCard={0}", UserCardNo)”，该语句用于统计该会员所有历史签到的总积分。接下来，我们调用“Maticsoft.DBUtility.DbHelperSQL.GetSingle(strSql)”语句来执行，并将结果返回给已定义的uscore。至此会员积分的计算过程已经全部完成，我们以“ws.PointUpdate(rq)”方法调用中间件接口将积分写入数据库。关于如何调用中间件接口的前文已有描述，在此不再赘述。

执行结果如下图所示：



**图 4-12 签到成功**

在本系统中我们大量采用了Web Services技术，这样做的优势在于，我们可以从用户界面直接调用中间件组件，不仅缩短了开发周期，还减少了代码复杂度，并能够增强应用程序的可维护性。

# 4.3 移动APP应用设计

无论购物中心或百货商场在互联网领域进展如何，在移动互联网领域，它们大多处在同一起跑线上。因此，移动互联网技术给予落后者一次反超的机遇，同时也可以让领先者进一步扩大优势。

接下来，我们将对本系统中移动APP所提供的“我猜我猜”功能就如何实现其设计功能进行进一步的阐述。

“我猜我猜” 功能基于大悦城购物中心实际需求而开发，主要用于商铺的推荐和导航。该功能借用游戏的形式，根据会员选择的不同时间、地点、愿望等条件的组合，给出相应的商户推荐并提供地图。

“我猜我猜”主要设计逻辑是将商铺按照不同的属性进行分类，如商铺的类别，营业时间，主要产品，目标客户群，环境，特色，人均消费等，然后根据不同的查询条件的组合，过滤出符合条件的商铺。对于不同条件的组合，主要是影响查询的结果和输出顺序。对输出顺序进行限定，主要是为了使查询结果看上去有所不同。如选择“独自一人”时，会要求系统随机的输出推荐结果，以区别于其他条件。这一做法主要可以有许多的联想，如，按照字母顺序输出，按照楼层逆向输出等，都可以增加这一应用的神秘感。

这一功能成功的关键不在于推荐的结果是否准确，而是由于结果的不可预知性所带给用户的好奇、新鲜和乐趣。

我们来看一下该应用的主菜单页面，如下图所示：



**图 4-13 “我猜我猜”的主菜单页面**

我们可以看到，从1到5一系列的选择项，给了消费者无穷发挥的空间，也使得应用本身变得有趣，让输出的结果具有强烈的神秘感和不可预估性。

当我们选定了时间和地点以后，将会进入“你想做什么？”，如下图所示:

|  |  |
| --- | --- |
|  |  |

图 4-14 “你想做什么”页面

在这里，消费者就有了很多的发挥空间，比如可以选择购物、用餐、喝点东西、商务约会、朋友聚会、休闲娱乐、看电影等，由于商场的目标消费群体为年轻女性，在这里还特地加入了“找女生感兴趣的事”的选项，以迎合目标消费者的喜好。

同时，我们在这里对二级查询条件的显示也做了一定的限制和过滤，例如：



图 4-15 二级查询条件的显示

在这里可以看到，如果我们在一级查询中选择“商务约会”，那“和谁在一起”的选项中只会显示“同事，客户”，显而易见，我们不可能和自己的亲人进行商务约会。如果我们在一级查询中选择“女生感兴趣的事”，那么“和谁在一起”的选项中将只有“同事，朋友，独自一人”，同时，“并且”选项则不会出现，即无需进一步的细分。

同时，查询结果和过滤条件之间也有着紧密的关联，例如：



图 4-16 查询结果与过滤条件的关联

当我们在“你想做什么”中选择了“看场电影”，那么其查询结果显然必须和“时间”以及“并且（即更为细化的动作）”关联。而如果选择了“喝点东西”，那么“时间”就显得并不是那么重要，也就不会被关联到查询结果中去。

当查询结果并不为一个的时候，我们就必须要考虑到一个先后顺序的问题，所以输出顺序和过滤条件，也必须有着明确的关联定义，例如：



图 4-17 输出顺序与过滤条件的关联

当选择了想做什么以后，系统会根据之前输入的“和谁在一起”等条件来决定输出结果的排序，例如当选择了“购物”时，如果选择“独自一人”则会随机输出一系列结果。但是有些“想做什么”的选项也不适用这些规则例如“看场电影”，无论和谁在一起，大悦城中只有一个电影院，且电影的排片表早已确认，只能按照时间顺序输出。

从上面的一些逻辑我们可以看到，“和谁在一起”是一个非常重要的选项，它将决定系统会返回什么样的查询结果。例如当在过滤“用餐”推荐，且用户选择“吃什么都行”时，根据不同对象与商铺的“氛围”标签的关联加以匹配，同时这一过滤条件还可以用在“休闲娱乐”推荐上，如下图所示：

|  |  |
| --- | --- |
|  |  |

图 4-18 “和谁在一起”对推荐结果的影响

“并且”是对于“你想做什么？”这个选项的细分，当用户选择“休闲娱乐”、“女生感兴趣的事”以及“随便逛逛”时不显示“并且”的按钮，即无需进一步细分，以下列出了一些“并且”中的清单以帮助理解：

|  |  |
| --- | --- |
|  |  |

图 4-19 “并且”中的清单

以上阐述了“我猜我猜”的业务背景，和一些基本的逻辑，接下来将详细介绍下该功能算法。“我猜我猜”功能主要分为购物、用餐、喝点东西、商务约会、朋友聚会、休闲娱乐、女生感兴趣的事、找些特别的东东、随便逛逛等项，不同的功能项，其算法和排序也各有不同，限于篇幅原因，这里只取一例，以“用餐”的算法为例来进行阐述。如下图所示：



图 4-20 “用餐”所使用的算法

在这部分算法中，首先在店铺中过滤“品类”=“餐饮”的品牌，这样大悦城中所有的用餐场所就被过滤出来。然后，对这些品牌按照不同的菜系以“细类”关键字进行二次过滤。最后，如果用户选择“吃什么都行”，根据“和谁一起”对搜索结果进行三次过滤。这样就可以得出最终的查询结果。而对于查询结果的排序逻辑在于：首先将过滤结果按照楼层由低向高顺序排列，其次如果用户选择了“独自一人”时，将输出结果随机排列。

有了算法了，接下就是为这个算法提供数据来源。由于“我猜我猜” 是APP应用，而支撑这个应用的数据是来自于大悦城官方网站数据库平台，无法直接进行数据调用的，只能通过一个中间技术进行桥连实现数据对接调用。而Web Services则是一个很好的解决方法案。

基于上述的算法的部分，我们接下来仔细分析一下Web Services数据调用的代码片段。在代码的一开始，我们还是要引用一些命名空间，如下所示：

|  |
| --- |
| using System;  using System.ComponentModel;  using System.Web;  using System.Web.Services;  using System.Web.Services.Protocols;  using System.Collections.Generic; |

图 4-21 所引用的.NET命名空间

相信大家已经发现，和互动触摸系统相比，这里所引用的命名空间有所不同，例如“using System.Web.Services”，顾名思义，System.Web.Services命名空间包含使我们可以利用 ASP.NET 和 XML Web 服务客户端来创建 XML Web 服务的类。XML Web services 是一些应用程序，它们提供了在使用标准协议（如 HTTP、XML、XSD、SOAP 和 WSDL）的松耦合环境下进行消息交换的能力。利用 XML Web services，可以在异质环境下的公司内部和公司之间生成模块化应用程序，使它们与各种各样的实现、平台和设备交互操作。这些应用程序的基于 SOAP 的 XML 消息可以具有定义完善的部分（结构化和类型化的部分）或者定义比较松散的部分（使用任意 XML）[20]。

在这里大家就会有疑问，同样整个系统中，我们都使用到了Web Services，为什么在互动触摸系统里面我们不引用这个命名空间呢？其实理由很简单，在互动触摸系统中，由于我们是中间件的使用者，而中间件的接口逻辑里面已经包含了这些引用，所以我们不需要再额外地去加载一次，只需要通过实例化相应的接口对象调用它所提供的接口方法以实现我们所需要的功能，如我们曾经分析过的“VIPQuery()”方法。而在移动APP系统中，我们需要将来自JoyCity官方网站数据库平台中的数据提供给移动APP使用，而它们分处于不同的软件平台，编程方式也不同，所以我们需要通过引用“System.Web.Services”这个命名空间，来为移动APP提供Web Services的接口方法。在下面的代码解析中我们会做更为详细的分析。

|  |
| --- |
| [WebService(Namespace = "http://localhost")]  [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]  [Description("本服务数据来Joycitizon主站。作者:xx")]  public class JoyShopKeyword : System.Web.Services.WebService  {  public MySoapHeader joycitizenHeader = new MySoapHeader();  [System.Web.Services.Protocols.SoapHeader("joycitizenHeader")]  [WebMethod(Description = "返回我猜我猜中店铺列表。参数:keyWord为店铺关键字，如果是多个关键字，请以','分隔，PageSize为每页获取多少条，PageCurrent为当前第几页，返回：集合List(ShopNameKey)", EnableSession = true)]  public List<ShopNameKey> Get\_ShopKeyword(string keyWord, int PageSize, int CurrentPage)  { //判断是否正确授权  if (!joycitizenHeader. IsValid ())  {  return null;  }  else  {  int r = 0;  ShopNameKey keyword = new ShopNameKey();  List<ShopNameKey> list = keyword.GetList(keyWord, PageSize, ref r, CurrentPage);  return list;  }  }  } |

图 4-22 Web Services的格式和接口方法

在这里，系统会自动创建一系列的Web Services特有的属性，其中“Namespace”用于获取或设置用于 XML Web services 的默认 XML 命名空间。“ConformsTo”用于获取或设置绑定声称所符合的 Web 服务互操作性 (WSI) 规范。而接下来要建立一个公共类型的类“JoyShopKeyword”，这个类继承了“System.Web.Services”中的“WebService”这个基类，其为 XML Web services 定义可选的基类，并提供对公共 ASP.NET 对象（如应用程序和会话状态）的直接访问。这一步的操作非常重要，如果 XML Web services 没有从该类继承，XML Web services 的组件设计器将不可用。而且实现 XML Web services 的类必须是公共的，并且必须具有公共的默认构造函数（不带参数的构造函数）。这使 ASP.NET 可以创建 XML Web services 类的实例以处理传入的 XML Web services 请求。

接下去我们定义了一个“MySoapHeader”类的对象“joycitizenHeader”用来处理Web Services的身份认证。同时我们定义了一个SOAP的标头“SoapHeader”，以方便在后续的身份认证中进行识别。关于身份认证的部分我们会在后面的代码过程中进行更为详细的分析。

当我们写了一个函数后，希望此函数成为外部可调用的接口函数，我们必须在函数上面添上一行代码“ [WebMethod(Description="函数的描述信息")]”，如果函数没有这个申明，它将不能被用户引用。

接下来的步骤是编写接口函数的处理逻辑。首先我们定义了一个公共类型的函数“Get\_ShopKeyword”，该函数用以返回“我猜我猜”的店铺列表，其含有三个主要的参数：keyWord为店铺关键字；PageSize为每页获取多少条；PageCurrent为当前第几页。其返回的数据“ShopNameKey”为一个泛型的数据集合。使用泛型的优点在于，其对内存的开销相对较小，且性能比非泛型（ArrayList）要高出不少。

然后就进入了身份认证的阶段，因为Web Services是开放在互联网上运行的，所只要知道Web Services的地址就可以调用它，这样是很不安全的。所以调用Web Services时我们需要进行身份验证，只有通过身份验证的用户才能调用相应的Web Services资源。

在这里我们使用了一个封装了的方法“IsValid()”来解决这个问题，封装部分的代码片段如下：

|  |
| --- |
| public class MySoapHeader : SoapHeader  {  public string UserName;  public string PassWord;  public bool IsValid()  { // 验证使用的用户名和密码  if (UserName.ToLower() == "joycitizen".ToLower() && PassWord.ToLower() == "D8E0F2CF3501DE307AE0278CEAC01428".ToLower())  {  return true; ;  }  else { return false; }  }  } |

图 4-23 MySoapHeader自定义身份认证代码

首先我们定义两个公共的变量属性“UserName”和“PassWord”提供给APP客户端使用，用来设置APP调用中间件接口时所使用的用户名和密码。当APP发送请求时，程序会判断SOAP协议中所提供的用户认证信息和系统中预设的信息是否一致，如验证正确，便可以正常调用，否则将无法获得权限。

最后介绍下Web Services的身份认证成功后“Get\_ShopKeyWord()”方法是如何返回算法中所需的数据的。

首先系统从移动APP客户端接受到根据算法所最后得出的关键字，然后将这些关键字作为参数通过调用上述接口方法向Web Services发送请求，当Web Services接受到了查询请求则会进入官方网站数据库进行查询，如店铺信息中包含相应的关键字，则将结果列出。查询语句如下：

|  |
| --- |
| select s.AscOrDesc,s.shopid,s.shopname,s.brandId,s.ShopNum,s.ShopImg,st.ShopTag from Jc\_shop s inner join Jc\_ShopInTag st on(s.Shopid=st.Shopid) where 1=1 and s.Status=1 or st.ShopTag like '%西餐%' |

图 4-24 关键字查询语句

至此，系统就完成了对于“我猜我猜”APP调用中间件对官方网站数据库进行查询并返回结果的全过程。系统运行的效果如下：

|  |  |
| --- | --- |
|  |  |
|  |  |

图 4-25 系统运行的效果

当用户在“你想做什么”中选择了“吃些甜品”，系统会提问“和谁在一起？”，用户做出了回答 - “朋友”。系统综合以上的信息，自动给出了恰当的推荐。

# 4.4 整合社会化媒体模块设计

本小节将主要介绍如何在大悦城官方网站中整合新浪微博API。与大悦城官方网站整合主要有几点功能：管理员在官方网站发表一篇文章，就会同步发送至新浪微博。同时，用户可以用微博帐号在官方网站进行登录对文章进行评论，并且可以选择把对文章的评论与新浪微博进行同步。至于如何进行整合，首先我们有必要来了解一下新浪微博API的授权机制。

新浪微博使用OAuth（开放授权）来进行身份认证。OAuth提供了一种以资源的所有者身份从客户端来访问服务器资源的方法，也就是说允许第三方应用访问某用户在某一服务器上存储的资源，而无需将用该用户的用户名及密码提供给第三方使用。

为了使用新浪微博开放平台提供的API（应用程序接口），首先需要先注册一个应用。新浪会给每一个应用一个专属的App Key和App Secret。Key跟Secret的使用方式跟其他一些协议中的公钥私钥的方案相类似，用户可以使用任意所熟悉的编程语言将key和Secret结合，为所发出的每个请求添加签名，以此来向新浪微博开放平台表明自己身份的合法性[21]。

下图清晰表述了OAuth请求循环的全过程：

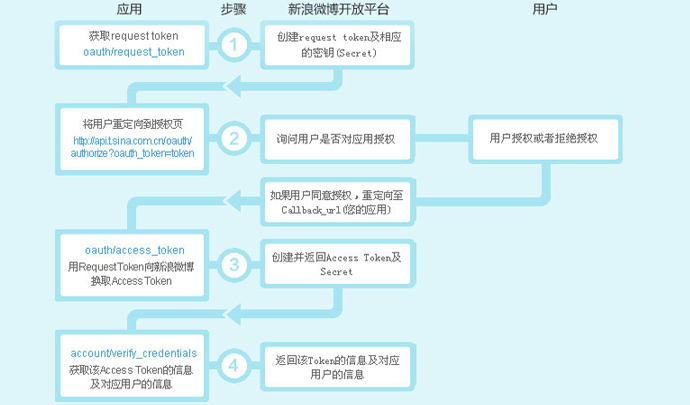


图 4-26 OAuth请求循环的全过程

从以上的流程中可以看到，首先由应用向新浪微博开放平台发出请求，获得未授权的Request Token及相应的Secret（密钥）。接着将上一步获得的Request Token作为参数，引导用户重定向到新浪微博的授权页面，如果用户没有登录新浪微博，在这个页面会要求用户登录。否则将会出现一个页面，用户可以在此页面上同意或者拒绝对此应用授权。如果用户同意授权，web应用页面将会重定向至用户指定的Callback\_url，如果是桌面应用，将会显示PIN码，用户需要将PIN码输入应用中来完成授权过程。

到了OAuth认证过程的最后一步，应用将之前得到的Request Token和Secret向新浪换取Access Token，服务器将会创建返回真正的Access Token与Secret，同时还会返回在新浪微博中对应的用户信息。至此就完成了一个OAuth请求的完整循环。

在本系统中，我们通过OAuth使用新浪微博账号登录并进行绑定，其实现过程如下：

首先，我们通过在网站页面上点击“与新浪微博连接”这个按钮来触发一个事件，如下图所示：



图 4-27 “与新浪微博连接”按钮

该按钮事件会执行一个跳转动作，将页面跳转至“Connect.aspx”并附加一个参数“action=login”。如下图所示：

|  |
| --- |
| public partial class UserCenter\_Connect : System.Web.UI.Page  {  protected void Page\_Load(object sender, EventArgs e)  {  oAuthSina sinaWeibo = new oAuthSina();  string requesttoken = sinaWeibo.AuthorizationSinaGet();  Session["TokenSecret"] = sinaWeibo.TokenSecret;  string callback = "";  string action = "";  string rooturl = "http://www.shjoycity.com";  if (Request["action"] != null)  {  action = Request["action"].ToString();  }  switch (action)  {  case "login":  callback = rooturl + "/UserCenter/SinaLogin.aspx?action=login";  break;  case "bind":  callback = rooturl + "/UserCenter/SinaLogin.aspx?action=bind";  break;  }  string url = requesttoken + "&oauth\_callback=" + callback;  Response.Redirect(url);  }  } |

图 4-28 Token请求的处理

在页面跳转以后，会立即执行到页面加载程序的函数“Page\_Load()”，该函数中我们首先定义了一个“oAuthSina”类的对象“sinaWeibo”以及一个变量“requesttoken”，通过“sinaWeibo”这个对象调用了“AuthorizationSinaGet()”方法，顾名思义，这个方法将向新浪请求授权。当新浪服务器接受到授权请求以后，会返回包含oauth\_token，oauth\_token\_secret等内容。接下来，我们定义了一个名为“TokenSecret”的session来保存刚才返回的“oauth\_token\_secret”。

然后就准备开始判断action参数中定义的操作类型是否为login（登录），如果是login，我们就通过先前定义的callback变量构建一个跳转语句。接下来，我们再定义一个名为url的变量，该变量将先前接受到从新浪服务器返回的“requesttoken”作为参数，并加上“callback”进行再次组合成一个新的跳转地址，该地址是新浪微博开放平台的验证地址，其格式如下：

|  |
| --- |
| http://api.t.sina.com.cn/oauth/authorize?oauth\_token=8ldIZyxQeVrFZXFOZH5tAwj6vzJYuLQpl0WUEYtWc&oauth\_callback=http://www.shjoycity.com/UserCenter/SinaLogin.aspx?action=login |

图 4-29 新浪微博开放平台的验证地址格式

最后我们通过“Response.Redirect(url)”进行跳转，发送刚才获取的“oauth\_token，oauth\_token\_secret”参数并获得用户授权。用户授权后，web应用页面将会重定向至刚才指定的oauth\_callback地址，这时oauth\_callback应该已经接到返回的信息，其中包含oauth\_token和oauth\_verifier两个参数。样例如下：

|  |
| --- |
| oauth\_token=8ldIZyxQeVrFZXFOZH5tAwj6vzJYuLQpl0WUEYtWc&oauth\_verifier=pDNg57prOHapMbhv25RNf75lVRd6JDsni1AJJIDYoTY |

图 4-30 oauth\_token和oauth\_verifier参数样例

接下来就是向新浪微博开放平台请求Access Token了，该段代码是在“SinaLogin.aspx”这个页面进行处理的。其核心代码如下：

|  |
| --- |
| oAuthSina sinaWeibo = new oAuthSina();  sinaWeibo.Verifier = Request["oauth\_verifier"].Trim(); //获取授权码  sinaWeibo.TokenSecret = Session["TokenSecret"].ToString();  sinaWeibo.AccessTokenGet(Request["oauth\_token"].Trim()); //授权通过后的设置 token |

图 4-31 换取Access Token

先定义一个“oAuthSina”类的对象“sinaWeibo”，然后给这个对象的Verifier属性进行初始化赋值，该值即为之前获取的“oauth\_verifier”参数，同时，对TokenSecret属性也进行初始化赋值，该值即为之前定义的“TokenSecret”session，其内容为先前请求时返回的“oauth\_token\_secret”。最后我们通过该对象调用“AccessTokenGet()”方法并传入参数“oauth\_token”向新浪微博开放平台的Access Token请求地址http://api.t.sina.com.cn/oauth/access\_token发送请求并换取Access Token。并通过“SinaService”类中的静态函数“Verify\_credentials()”获得用户信息，如下所示：

|  |
| --- |
| User u = SinaService.Verify\_credentials(sinaWeibo, "xml");//获取通过验证的用户的基本信息 |

图 4-32 获取用户信息

到这里为止，我们完成了向新浪微博请求授权并验证的全部过程。其运行结果如下，页面显示：



图 4-33 与新浪微博进行连接



图 4-34 与新浪微博关联成功

同时，大悦城也在新浪注册了官方微博，大悦城官方网站可以随时调用其新浪微博认证账户所发布的所有信息。当网站管理员在主站管理后台操作、新建博文、团购、特卖活动时，也会通过新浪微博认证账户发送一条微博（自定义内容 + 摘要 + 短链接）；用户在大悦城官方网站上对这些博文、团购或活动的可以进行评论，如果该用户也在大悦城官方网站上进行了新浪微博账户的绑定，那么他所发布的信息，即会发布在大悦城官方网站上，也会同步到新浪微博该用户的账户下。限于篇幅原因，本文将只介绍如何从新浪微博网站获取微博信息。

用以获取新浪微博的核心代码如下：

|  |
| --- |
| oAuthSina sinaWeibo = new oAuthSina();  sinaWeibo.Token = "d2d9a4a06d4777f1b86abefd7857cb6a";  sinaWeibo.TokenSecret = "d964ec17b2f1dab23ad0f67a20e98897";  DataTable list = SinaService.User\_timeline(sinaWeibo); |

图 4-35 从大悦城网站调用新浪微博

此代码中，首先实例化“oAuthSina”类型的对象“sinaWeibo”，并设置新浪微博认证账号的Token和TokenSecret。接下来调用“SinaService.User\_timeline()”方法获取微博列表信息，返回数据为DataTable数据集合。实现方法代码如下：

|  |
| --- |
| public static DataTable User\_timeline(oAuthSina oa)  {  DataTable lst =null;  string url = string.Format("http://api.t.sina.com.cn/statuses/user\_timeline.json");  string strList = oa.oAuthWebRequest(oAuthSina.Method.GET, url, String.Empty);  lst = GetStatusList( strList );  return lst;  } |

图 4-36 获取新浪微博

此代码接收参数是“oAuthSina”类对象“oa”，首先定义“DataTable”类型的集合对象“lst”，用于接收返回的数据。接着准备API的url，然后通过调用“oAuthWebRequest()”方法来向新浪微博的API发送获取新浪微博列表的HTTP请求。新浪微博开放平台在接受到请求以后，返回当前授权账户的最新微博列表，返回的数据格式为JSON（JavaScript Object Notation，是一种轻量级的数据交换语言，以文字为基础，且易于让人阅读）格式。在取得了JSON数据后，需要对其进行格式转换，此时用到了“GetStatusList()”方法，该方法基于Jayrock.Json（第三方控件）实现数据转换。以下为将JSON数据格式化并实现DataTable的代码片段：

|  |
| --- |
| using Jayrock.Json;  JsonReader jsonRead = new JsonTextReader(new StringReader(content));  JsonObject jsonobj = new JsonObject();  jsonobj.Import(jsonRead);  JsonArray wiboArr = jsonobj["statuses"] as JsonArray;  DataTable wibo = new DataTable();  wibo.Columns.Add("id", typeof(Int64));  wibo.Columns.Add("text", typeof(string));  foreach (JsonObject wibolist in wiboArr){  DataRow dr = wibo.NewRow();  dr["id"] = Convert.ToInt64(wibolist["id"]);  JsonObject user = wibolist["user"] as JsonObject;  dr["text"] = wibolist["text"].ToString();  wibo.Rows.Add(dr);  }  return wibo; |

图 4-37 JSON数据格式转换

上述代码是以JsonArray和JsonObject的转换来实现以键值方式读取JSON数据。首先我们使用“Import()”方法将文本的content数据转变成一个对象，然后获取“statuses”内容并转化成JsonArray对象。接着构造DataTable类型wibo的数据结构，遍历读取JsonArray集合的JSON数据，循环创建DataTable数据行的数据并添加到DataTable中。最后得到DataTable类型的wibo。至此，我们完成了从大悦城官方网站向新浪微博请求、接受、转换数据的全过程。其实现的效果如下：

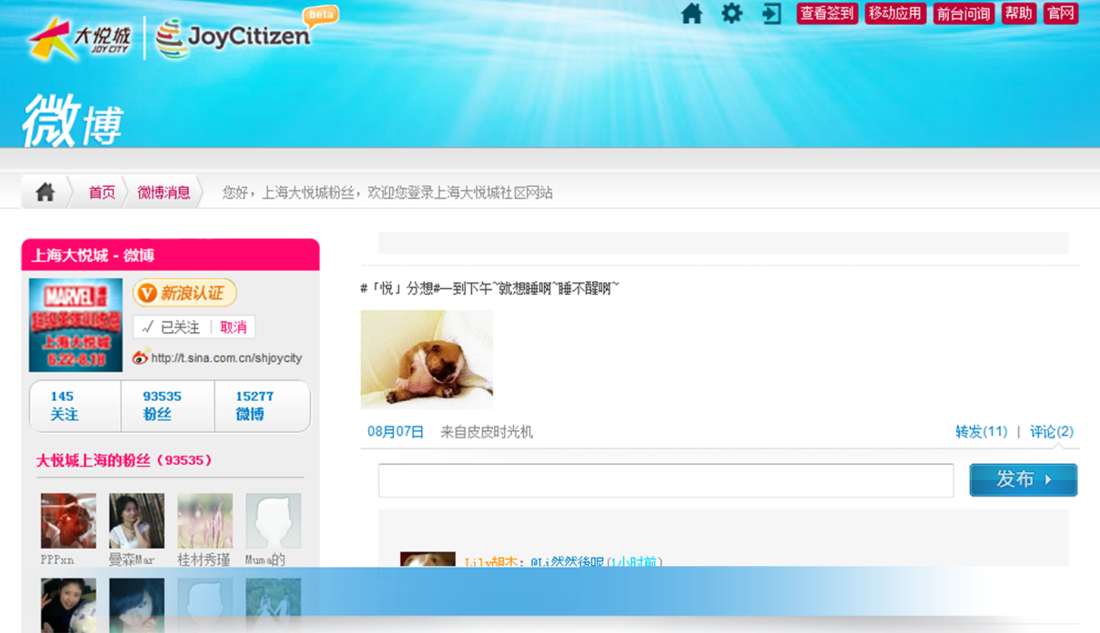


图 4-38 在大悦城网站上显示新浪微博

通过与新浪微博的整合，无论是消费者，还是系统管理员，都无需在大悦城官方网站和新浪微博之间来回切换，手动转发消息。只需要在一处进行登录，即可完成信息的同步。消费者可以轻松地将自己愉快的购物体验发布在微博上供朋友们阅读，系统管理员也可以简单地将促销信息等同步地发布在社会化媒体上，以获取更高的媒体接触率。

# 4.5 系统用例介绍

以上各部分内容分别就系统的几个重要功能模块做了较为详细的介绍，接下来，本节将以在大悦城移动APP应用“Joy City”中对大悦城官方新浪微博进行查看和关注为例，给出相应的操作流程和界面。大悦城移动APP基于iOS系统开发，其需要在运行iOS操作系统的移动设备（在这里我们以iPhone为例）上实现一系列的功能，以下为在该APP中实现对大悦城官方新浪微博进行关注所需要的的主要功能列表：

表4-1 功能列表

|  |  |
| --- | --- |
| 名称 | 功能描述 |
| 登陆 | 会员输入用户名、密码，在经过验证后登录系统 |
| 自动登录 | 会员登录过一次以后，下次登录无需再次输入用户名和密码 |
| 密码找回 | 会员忘记密码时，通过卡号/手机/身份证号/邮箱来取回密码 |
| 查看官方微博 | 会员在大悦城移动APP中查看大悦城官方新浪微博 |
| 关注官方微博 | 会员在大悦城移动APP中对大悦城官方新浪微博进行关注 |
| 系统退出 | 退出软件系统 |

以下就以一个完整的在大悦城移动APP中关注大悦城官方新浪微博的实例，结合APP运行截屏对以上功能的实现作出介绍。

用例：会员通过大悦城移动APP关注大悦城官方新浪微博

目标：会员登录系统并实现微博关注

前提：服务器、移动客户端、新浪微博网站运行良好，网络状况良好

触发：会员下载、安装、启动APP

场景：会员从Apple APP Store中查找并下载、安装APP。在iPhone桌面界面上点击APP图标。然后，会员在登录界面中输入用户名，密码进行登录。如忘记密码，则在登录界面中进行找回操作。登录成功后，会员进入会员中心界面中打开大悦城官方微博，查看最新信息，并对其进行关注。

首先，会员进入Apple APP Store中对APP进行下载和安装。并在iPhone桌面上打开APP，APP启动并进入初始化界面。

|  |  |
| --- | --- |
| IMG_6463 | IMG_6475 |

图4-39 下载、安装及启动APP

进入初始化界面后，系统首先会全屏推送最新广告，经过三秒钟自动跳转进入主界面。

|  |  |
| --- | --- |
| IMG_6464 | IMG_6465 |

图 4-40 全屏广告推送及主界面

会员在主界面中可以点击右上角的会员按钮（未登录时显示为灰色）来进入登录界面。在登录界面中，会员输入用户名和密码来进行登录操作，在这里用户名可以是会员的卡号、手机、身份证或者注册邮箱。如果会员忘记了密码，则可在此界面进行找回，在输入了上述任何一种用户名信息后，系统会将密码自动发送给会员注册时所使用的手机号码。在登录完成后，系统会自动记录登录信息，会员下一次使用移动APP的时候，不需要再次进行登录工作。

|  |  |
| --- | --- |
| IMG_6474 | IMG_6472  IMG_6480 |

图4-41 会员登录及密码找回

登录完成后，系统自动跳转回到主界面，在主界面上会员可以通过再次点击右上角的“会员”按钮进入会员中心，这时候我们留意到，“会员”按钮因为登录成功而已经变成了彩色。进入会员中心的动画动作是主界面整体向左移动并留下部分显示，而会员可以通过手指的左右划动动作来关闭会员中心。在会员中心中，最上方显示了会员基本信息（姓名，会员卡号）以及积分余额；中部显示了会员卡样，可以通过卡样右上角的翻转按钮翻转卡样，查看会员卡的正反面内容。同时会员还可以在其中完成更多的功能操作，例如进行积分记录查询、积分兑换、电子券管理，二维码扫描，查看商户推荐，大悦城官方微博的查看关注，以及会员账户信息管理。在会员帐户信息管理中，会员可以登出账户、选择是否接送推送信息（Push Message）、清除缓存以及修改密码。同时也可以查看到开发者信息以及系统的版本号。

|  |  |
| --- | --- |
| IMG_6470 | IMG_6479 |

图 4-42 会员中心及会员帐户信息管理

我们在这里点击“微博”，就可以在大悦城移动APP中查看大悦城官方微博所发布的最新消息，同时也可以通过点击“+加关注”按钮对其进行关注。

|  |  |
| --- | --- |
| IMG_6481 | IMG_6471 |

图4-43 在大悦城移动APP中查看官方微博及关注

点击“+加关注”按钮后，我们进入到新浪微博官方APP中（图4-43 右）进行查看，可以发现“大悦城上海”已经处于用户的关注列表中，且在这两个APP中的微博内容是同步更新的。

本小节通过一个实际案例，介绍了如何通过大悦城移动APP对大悦城官方新浪微博进行查看和关注，APP在实际运行时与上述过程完全一致。

# 第五章 总结与展望

# 5.1 本文内容总结

随着市场竞争的日益激烈，依靠互动零售商业社区系统主动地与消费者进行互动、为消费者提供更好的消费体验、从而获得消费者的认可，进一步在竞争环境中取得优势，已经成为零售企业的一项至为关键的战略举措。

本文依托大悦城互动零售商业社区系统，详细分析了建立一个为消费者提供互动体验的商业社区系统所需要的关键技术、总体架构及软硬件选型。重点针对其中关于互动触摸模块、移动APP应用以及整合社会化媒体模块的详细设计进行了详细的论述和深入的研究。

目前，该系统的设计与开发已经完成，并已投入使用。从大悦城用户的反馈来看，该系统运行表现良好，基本实现了预期的设计目标。其主要优势和商业价值主要体现在以下几个方面：

扩大了互动社区的参与人群。通过设置于购物中心各个角落的互动触摸屏上，消费者可以对感兴趣的内容进行直观而直接的操作。对于消费者来说，操作触摸屏完全不需要掌握任何专业知识，从而使那些对于电脑、智能手机等传统的信息终端并不熟悉的人群，如中老年消费者，也能够轻易地获得咨询，并参与互动。对于大悦城来说，互动社区的参与者越多，其所能收集的诸如消费者行为、偏好、消费记录等运营数据也就越丰富。

提高了消费者对大悦城的关注度。随着移动APP在iTunes Store的正式发布，越来越多的会员开始在自己的手机上安装了大悦城移动APP。会员不需要身处于购物中心中，甚至无需身处电脑前，无论其是在地铁上、逛街时，还是在电梯里，大悦城都可以通过这些碎片化时间以信息推送的方式来向会员发布促销、活动等信息，会员也可以打开APP查阅感兴趣的内容，整理自己的关注、优惠券等，以及通过使用APP中的一些趣味性功能来参与到与系统的互动中来。无时无刻的随身相伴，让大悦城这一品牌完全成为消费者生活的一部分，也让消费者对大悦城的关注达到了一个前所未有的高度。

最大化了媒体接触率。通过对于新浪微博的整合这一举措，完全颠覆了由用户主动向系统索取信息的传统模式，而转为由系统向用户进行进行被动的发布。用户无需再做出任何请求动作，而是随时随地置身于互动信息的接受中。无论用户是在使用移动APP、浏览大悦城官网还是查看自己的新浪微博账户，系统都可以在第一时间将促销、活动等最能吸引眼球的信息传递到用户身边，大悦城互动零售商业社区系统最大化了媒体接触率，从而提高了广告商的ROI（投入产出比，Return of Investment）。

# 5.2 未来工作展望

虽然大悦城互动零售商业社区系统很好地实现了零售商在为消费者打造现代化的购物环境和改善消费体验上的基本预期，在系统运行的前期也取得了一致的好评。但是随着时间的推移，从大悦城用户及消费者各方的反馈来看，对系统的功能仍然存在着更高的期望，主要体现在：

将移动APP扩展到Android和Windows Phone平台。大悦城移动APP在iPhone和iPad都有着优异的表现。但是由于在系统设计阶段主要考虑了iOS系统，目前并不能支持Android、Windows Phone等其他主流移动平台。因此在将来将会进一步对同样应用广泛的Android以及Windows Phone进行进一步的开发，更好地服务于多层次的消费者群体。

进一步改进会员卡材质和功能。本系统中的会员卡在系统设计时出于成本的考虑而选择了磁条卡。磁条卡存在着信息存储量较小、磁条较为脆弱易损，防伪及保密性能较弱等缺点。而随着二维码以及RFID技术的发展，其在耐用性、非接触读写和空间定位等方面的功能优势已经逐渐为大众所认知，并且其成本已经降低到可接受的程度，因此在将来会在会员卡的材质和功能上将朝着这些方向进行进一步的改进。

增强对新兴社会化媒体的支持。系统在设计时主要考虑了在微博类社会化媒体中份额最高的新浪微博。随着互联网技术的进一步发展，更多其他形态的社会化媒体如微信、优酷等也开始蓬勃发展并各自发展了大量的用户群体，成为了移动互联网用户的主要选择之一。因此系统在将来必须增强针对这些新兴社会化媒体的支持以满足更多不同消费者的需求。

目前，我们正在搜集和整理用户对于系统的意见和建议，并准备在近期对系统进行进一步的改版和升级。

# 参考文献

[1] Nielsen.China's Evolving Shopper Landscape[EB/OL].http://www.consumer360.com/chinas-evolving-shopper-landscape，24th May 2013.

[2] Nielsen.Tapping Opportunity Among China’s Increasingly Sophisticated Consumers[EB/OL].http://www.consumer360.com/tapping-opportunity-among-chinas-increasingly-sophisticated-consumers，24th May 2013.

[3] Nielsen.In China, Premium Sells: From Toothpaste to Cars to Banking[EB/OL].http://www.consumer360.com/in-china-premium-sells-from-toothpaste-to-cars-to-banking，24th May 2013.

[4] Moustafa Refaat.BizTalk: The Practical Course[M]. US: Lulu.com, 6th Apr 2009: 125, 126.

[5] "App" voted 2010 word of the year by the American Dialect Society (UPDATED)[EB/OL].http://www.americandialect.org/app-voted-2010-word-of-the-year-by-the-american-dialect-society-updated, 8th Jan 2011.

[6]CcomScore.In U.S. Mobile Market, Samsung, Android Top The Charts; Apps Overtake Web Browsing[EB/OL].http://techcrunch.com/2012/07/02/comscore-in-u-s-mobile-market-samsung-android-top-the-charts-apps-overtake-web-browsing, 2nd Jul 2012.

[7] ABI Research.The Mobile App Market will be Worth $27 Billion in 2013 as Tablet Revenue Grows[EB/OL].http://www.abiresearch.com/press/the-mobile-app-market-will-be-worth-27-billion-in-, 25th Jun 2013.

[8] Biancuzzi Federico, Warden Shane.Masterminds of Programming[M]. O'Reilly Media, Inc., 14th Apr 2009: 242, 246.

[9] Gareth Finn.New Touch-Screen Controllers Offer Robust Sensing for Portable Displays[EB/OL].http://www.analog.com/library/analogdialogue/archives/44-02/touch\_screen.html, 1st Feb 2010.

[10] How Does A TouchScreen Work?[EB/OL].http://ultratechtalk.com/how-does-a-touchscreen-work, 29th Mar 2013.

[11] Toni Ahlqvist, Asta Bäck, Minna Halonen & Sirkka Heinonen.Social Media Roadmaps[EB/OL].http://ec.europa.eu/digital-agenda/futurium/en/content/social-media-roadmaps-exploring-futures-triggered-social-media, 1st Jan 2008.

[12] Andreas M. Kaplan, Michael Haenlein.Users of the world, unite! The challenges and opportunities of Social Media[J].Business Horizons, 2010(53): 59, 68.

[13] Nielsen.Three Ways to Maximize Online Advertising[EB/OL].http://www.consumer360.com/3-ways-to-maximize-online-ads, 24th May 2013.

[14] MSDN.Web Services[EB/OL].http://msdn.microsoft.com/zh-cn/library/ms950421.aspx

[15] 柴晓路，梁宇奇.Web Services技术架构和应用[M].北京：电子工业出版社，2003：10，10.

[16] MSDN. Web Services Infrastructure[EB/OL]. http://msdn.microsoft.com/en-us/library/sd5s0c6d(v=vs.100).aspx

[17] Principled Technologies.Virtualization Performance: VMware Vsphere VS. Microsoft Hyper-V[EB/OL].http://www.principledtechnologies.com/featured.php, Aug 2011.

[18] MSDN.System Namespace[EB/OL].http://msdn.microsoft.com/zh-cn/library/system(v=VS.80).aspx

[19] MSDN.System.web Namespace[EB/OL].http://msdn.microsoft.com/zh-CN/library/system.web.aspx

[20] MSDN.System.Web.Services Namespace[EB/OL].http://msdn.microsoft.com/zh-cn/library/system.web.services.aspx

[21] 新浪微博.OAuth基本流程简介[EB/OL].http://open.weibo.com/wiki/OAuth#.E6.A6.82.E8.BF.B0, 30th Mar 2012

# 致 谢

本论文的写作从开题、搜集资料、起草到最后成文、修订。诸多环节无一不得到导师、同学、同事和亲友的帮助。在此表示衷心地感谢！

首先，也是最重要的，要感谢我的指导老师，XXX老师。在整个论文的写作过程中，XXX老师对我进行了无私的指导和帮助，不厌其烦、逐字逐句地帮助我进行论文的修正和改进。从每一个细节中都体现出XXX老师严谨的治学态度、渊博的学术知识、诲人不倦的敬业精神以及宽容的待人风范。

其次，感谢我的同事在我搜集整理资料中给予的无私的支持。

同时，感谢这篇文论所涉及的各位学者。本文引用数位学者的研究文献，各位学者的研究成果给了我很大的帮助和启发。

最后，感谢复旦大学所提供的一流的学习环境和氛围，使我得以顺利地完成这次论文的写作。

**论文独创性声明**

本论文是我个人在导师指导下进行的研究工作及取得的研究成果。论文中除了特别加以标注和致谢的地方外，不包含其他人或其它机构已经发表或撰写过的研究成果。其他同志对本研究的启发和所做的贡献均已在论文中作了明确的声明并表示了谢意。

作者签名： 日期： 2013.9

**论文使用授权声明**

本人完全了解复旦大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其它复制手段保存论文。保密的论文在解密后遵守此规定。

作者签名： 导师签名： 日期： 2013.9