蚂蚁热帮(Ant Help)

——订单、消息、活动管理模块的设计与实现

摘 要

通过绪论、需求分析、概要设计、详细设计与实现、测试和总结等各章对订单、消息、活动管理模块进行了详细的分析与设计。针对各个部分的需要，绘制了相关的图示和表格，用以达到最优的项目设计文档说明效果。

本系统是针对第三方生活服务平台的管理分析而设计。主要包括用户权限管理、订单管理和社区管理三大模块。同时该系统还可以作为生产制造执行系统中的一个有机组成部分，协助生产制造执行系统建立集成的生产现场控制与品质管理平台和完善的生产过程数据库。

产品质量管理工作对提升[企业竞争力](http://baike.baidu.com/view/540265.htm)有重要意义。过程检验可以防止出现大批不合格品，避免不合格品流入下道工序去继续进行加工。实施统计过程控制则可以帮助企业在质量控制上真正作到"事前"预防和控制。

本系统采用的C#进行编码，同时采用微软公司的Visual studio 2008和微软公司的Microsoft SQL Server 2008 开发环境进行开发。系统的结构组成主要有客户端、web服务和数据库访问层以及数据库持久化层。是系统层次明确，逻辑清晰，便于编码和后期维护。

**关键词**：统计过程 控制 质量 C#

Small Product Quality Management System For Manufacturing Enterprise

——Design and Implementation of Statistical Process Control Management Module

**ABSTRACT**

This paper through the introduction, demand analysis, outline design, detailed design and implementation, testing and summary of the statistical process management module for the statistical process of the detailed analysis and design. For each part of the need to draw the relevant diagrams and tables, in order to achieve the best results of the project design documentation.

This system is designed for the management of manufacturing enterprises to the product quality inspection. Mainly including the production process inspection management and statistical process control and management of the two modules. At the same time the system can also as a production manufacturing execution is an integral part of the system, assist the manufacturing execution system to establish the integration of production control and quality management platform and improve the production process database.

Product quality management is of great significance to enhance the competitiveness of enterprises. Process inspection can prevent the emergence of a large number of substandard products, to avoid the flow of substandard goods into the next process to continue processing. The implementation of statistical process control can help enterprises in the quality control of truly "advance" prevention and control.

This system uses the C# to carry on the code, simultaneously uses the Microsoft Corp studio Visual 2008 and the Microsoft Corp SQL Server Microsoft 2008 development environment to carry on the development. The structure of the system consists of client, web service, database access layer and database persistence layer. System level is clear, clear logic, easy to encode and post maintenance.

**Key words:** Statistical Process Control Quality C#

目 录

[第一章 绪论 1](#_Toc16644)

[1.1 项目背景与现状 1](#_Toc13737)

[1.1.1 项目背景简介 1](#_Toc11469)

[1.1.2 当前国内外现状 1](#_Toc20185)

[1.2 选题意义与系统主要工作 2](#_Toc32311)

[1.2.1 选题目的及意义 2](#_Toc27595)

[1.2.2 本系统主要工作 3](#_Toc15227)

[1.3 统计过程控制原理（SPC） 3](#_Toc8408)

[1.3.1 控制图简要介绍 3](#_Toc23145)

[1.3.2 控制图的应用方法 4](#_Toc31723)

[1.4 开发工具简介 5](#_Toc19254)

[1.4.1 Microsoft Visual Studio 2008 5](#_Toc9872)

[1.4.2 Microsoft SQL Server 2008 5](#_Toc19135)

[1.5 本章小结 5](#_Toc17836)

[第二章 需求分析 6](#_Toc16361)

[2.1 背景简介 6](#_Toc17671)

[2.2 可行性分析 6](#_Toc23476)

[2.2.1 社会可行性 6](#_Toc17775)

[2.2.2 经济可行性 6](#_Toc428)

[2.2.3 技术可行性 7](#_Toc6300)

[2.3 功能需求分析 7](#_Toc2306)

[2.3.1 功能划分 7](#_Toc3994)

[2.3.2 系统使用角色分析 8](#_Toc20002)

[2.5 本章小结 9](#_Toc20106)

[第三章 概要设计 10](#_Toc16514)

[3.1 系统结构设计 10](#_Toc23419)

[3.1.1 系统整体结构 10](#_Toc6304)

[3.1.2 质量管理系统结构 11](#_Toc21747)

[3.1.3 电芯检验过程数据检查系统结构 11](#_Toc10975)

[3.2 系统实体设计 12](#_Toc5597)

[3.2.1 系统实体 12](#_Toc9805)

[3.2.2 系统整体实体联系 14](#_Toc31154)

[3.3 数据库表结构设计 15](#_Toc18073)

[3.3.1 数据库表结构设计 15](#_Toc19219)

[3.3.2 数据库表间关系 19](#_Toc31429)

[3.4 数据流需求分析 20](#_Toc30781)

[3.4.1 数据流 20](#_Toc20055)

[3.4.2 数据流分层分析 20](#_Toc17396)

[3.4.3 数据字典 21](#_Toc4369)

[3.5 本章小结 24](#_Toc7888)

[第四章 详细设计及实现 25](#_Toc22910)

[4.1 背景 25](#_Toc29731)

[4.2 设计及实现 25](#_Toc8406)

[4.2.1 控制图数据查询导出至EXCEL设计与实现 25](#_Toc27394)

[4.2.2 预控图数据查询与导出到EXCEL设计与实现 30](#_Toc7572)

[4.2.3 控制项目添加设计与实现 33](#_Toc17356)

[4.2.4 控制项目与BOM号绑定设计与实现 37](#_Toc23629)

[4.2.5 控制图数据录入的设计与实现 40](#_Toc25213)

[4.2.6 预控图数据录入的设计与实现 42](#_Toc19787)

[4.3 本章小结 44](#_Toc3954)

[第五章 结论 45](#_Toc4403)

[5.1项目归纳与总结 45](#_Toc894)

[5.1.1 项目归纳 45](#_Toc5479)

[5.1.2 项目结论 45](#_Toc30465)

[5.2 新发展趋势 45](#_Toc22306)

[5.3 课题尚待改进的建议 46](#_Toc1827)

[5.3.1 课题尚存在的问题 46](#_Toc10336)

[5.3.2 课题改进的建议 46](#_Toc2462)

[5.4 本章小结 46](#_Toc19701)

[参考文献](#_Toc27626)

[致 谢](#_Toc14733)

# 绪论

## 1.1 项目背景与现状

### 1.1.1 项目背景简介

随着团购网站和移动端应用软件的迅速发展，O2O模式的电子商务正在被人们认识和接受，O2O的发展改变了传统的电子商务模式与格局。O2O的概念非常广泛，只要产业链中既可涉及到线上，又可涉及到线下，就可以统称为O2O。网上购买服务，我们司空见惯的是打车、家政服务等，但在以个人为单位进行其他方面的生活服务则少很多。因此国内外在相关方面的研究还比较缺乏，也不是非常成熟和完善。国内外学者对消费者消费行为的研究大约有十年多的时间，其研究方向和内容也较为广泛。

在人们越来越注重生活质量的当下，个人需求更是加快了互联网对生活服务类产品的开发和投入。国内电子商务和团购网站的全面普及，以及近两年来网约车软件以及共享单车等移动端应用软件的迅速发展，都体现了O2O电子商务模式在生活服务领域发展的重要性。这种电子商务模式的优势也显而易见，对于商家和企业而言，它能够带来真实的客户流和数据统计，能够更好的实现线上虚拟经济与线下实体经济的完美融合；对于客户而言，O2O也提供了更丰富、更全面的商家服务信息，能够让用户更加方便快捷的体验到相应的产品和服务；而对于服务提供商来说，这种电子商务模式可以为他们带来更加大规模、高粘度的用户，帮助他们获得商家资源和充沛的现金流。所以O2O实现的“三方共赢”的商务模式。所以，O2O生活服务以强大的生命力和商业价值，正在发展成为电子商务领域的新的主干发展线。

### 1.1.2 当前国内外现状

近年来，随着Internet的迅速崛起，我们从最初的信息发布阶段，极速的迈入了电子商务阶段，成为了我们这个商业信息社会的神经系统。随着互联网信息碎片以及云计算技术的愈发成熟，互联网营销模式顺势而出，O2O式的生活服务已经成为了人们生活中不可忽视的一部分。在国内，随着网络的飞速发展，网络营销已经广泛的映射到我们的日常生活中。O2O的概念从诞生到爆发，仅仅用了不到两年的时间。2010年王兴推出的美团网更是拉动了国内O2O的迅速发展。如今美团网作为一家本地服务类的电子商务企业，以超低折扣的优质服务和广泛销售和推广渠道，使得消费者与商家也越来越信赖于它。王兴本人指出，美团网与其他转型传统B2C的团购网站大不相同，在支付方面，中国团购更习惯于第三方支付。2016年，美团已成为国内最受大众喜欢的团购网站。陈浩然在[《美团网：服务电商的逆袭》](http://www.iyiou.com/p/13148)一文中提到中国的团购网站不是简单模仿国外Groupon的模式，而是一种改进与创新的模式。极大展示了O2O模式下的生活服务的发展潜力。

Martin Pasquier在《中国为何掀起了O2O革命》(2015.5)中提到了在中国掀起O2O热潮的三大重要因素，一是高质量和高可靠性的代码对现实生活和线上产品的桥接；二是人们对语音服务和远程服务信任的快速增长；三是手机支付，手机支付是促进O2O模式在中国取得巨大实施的关键因素。

国内的生活服务平台短时间内的迅速崛起，极大展示了O2O模式下的生活服务的发展潜力。从餐饮订购的饿了么，到团购服务的美团，到便利出行的滴滴，无一不采用O2O的电子商务模式以提供三方生活服务平台，提供更便捷、有效的生活方式，提高生活质量。

## 1.2 选题意义与系统主要工作

### 1.2.1 选题目的及意义

随着社会的发展与进步，人们更加注重生活质量，这样的趋势注定互联网也更加投入和开发生活服务类的产品。蚂蚁热帮是致力于更快捷、更有效的提高生活质量的一个第三方生活服务平台。短短5年内，国内的生活服务平台如雨后春笋般迅速崛起，从餐饮订购的饿了么，到团购服务的美团，到便利出行的滴滴，无一不采用O2O的电子商务模式。蚂蚁热帮同时也采用了这种O2O模式，力求改善人们的同城生活服务，提供多元化的专人速递速成服务，快速响应，解决急需等任何个人化的生活需求，例如：想找人帮忙购物、取件、送件，或是找代驾、帮排队等。

蚂蚁热帮的意义在于提供更方便、更便捷、更有效的生活方式，提高办事效率，改变人们的生活节奏，因而提高生活质量。同时，也拓宽人们的日常生活圈，提升人与人之间的交际和信任。

其应用范围有：代购代帮、代理销售、家政服务、宠物之家、汽车服务、休闲服务、二手处理等等一系列生活服务。

### 1.2.2 本系统主要工作

做好三方服务平台对提升[商业竞争力](http://baike.baidu.com/view/540265.htm)和个人生活质量有重要意义。平台发布过程可以解决个人所需，提供方便快捷、高响应的发单接单。平台管理可以使用户更方便的查看个人的发单接单详情等信息。

1.)系统相关用户角色以及对各种角色的用户信息进行管理。

2.)在用户的注册登录功能中，添加系统的Token认证，保障用户信息的安全。

3.)系统各项功能的访问权限的管理，对不同的用户开放不同的权限进行操作。

4.)发布者可以进行下单请求，并且由系统分发订单信息。

5.)其他用户抢单接单进行对订单的后续流程的操作。不同的用户角色对订单有不同的操作策略。

6.)管理员权限可以对所有用户、订单、消息及投诉等信息进行查看和管理。

7.)管理员可以对用户发送系统消息。

8.)用户可以对系统提出意见或反馈，也可以对其他用户非法操作提出的投诉信息。

9.)系统管理员发起优惠活动，用户可以在规定期间内进行活动的参与。

10.)社区模块类似于微信朋友圈等社交媒体的沟通平台，用户可以在平台内进行交流分享。

# 系统开发环境及相关技术简介

## 2.1 系统开发环境简介

### 2.1.1 Microsoft Visual Studio 2015简介

Visual Studio作为一个基于组件的软件开发工具，有强大的集成环境，可用于构建功能强大、性能出众的各类应用程序。Visual Studio 2015于2014年11月13日正式投入市场使用，它可以帮助各类开发人员打造跨平台的应用程序，从Windows到Linux,甚至iOS和Android。Visual Studio 2015发布以来具有以下新增功能：

1.与云集成，内置工具可让所有的NET Core、Azure应用程序、服务、Docker容器等全面集成。无缝体验让你感觉如同在Azure数据中心工作一样。

2.有效协作，直接管理任意提供程序托管的团队项目，包括Visual Studio Team Services、Team Foundation Server或GitHub。或者，使用新的“打开任意文件夹”功能，无需通过正式项目或解决方案即可快速打开并处理几乎所有代码文件。

3.交付优质移动应用，借助Xamarin的高级调试、分析工具以及单元测试生成功能，与以往相比你可以更快、更轻松地构建、连接和调整适用于Android、iOS和Windows的本机移动应用。你还可以选择使用Apache Cordova开发移动应用，或构建C++跨平台库。

4.提高语言水平，Visual Studio 继续加强对最新编程语言功能的支持。无论使用C#、Visual Basic、C++、TypeScript、F# 还是使用第三方语言（例如JavaScript），在整个开发体验中你都能获得一流的功能支持。

5.性能得以优化，Visual Studio包含了对日常使用的核心功能的大量性能改进。你还会发现在整个开发生命周期中，启动速度显著加快，内存占用大大降低，并且响应能力明显提高。

### 2.1.2 Microsoft SQL Server 2014简介

Microsoft SQL Server 2014在延续了历史版本的各项优势以外，更是添加了诸多的新特性。这些新的特性使我们的使用更加的方便快捷，提高了我们的开发效率。下面为Microsoft SQL Server 2014的新特性：

1.全程加密技术(Always Encrypted)，全程加密技术(Always Encrypted)支持在SQL Server中保持数据加密，只有调用SQL Server的应用才能访问加密数据。

2.动态数据屏蔽(Dynamic Data Masking)，利用动态数据屏蔽功能，你可以将SQL Server数据库表中待加密数据列混淆，那些未授权用户看不到这部分数据。利用动态数据屏蔽功能，你还可以定义数据的混淆方式。

3.JSON支持，在SQL Server 2008中，你现在可以在应用和SQL Server数据库引擎之间用JSON格式交互。微软公司在SQL Server中增加了对JSON的支持，可以解析JSON格式数据然后以关系格式存储。

4.SQL SERVER支持R语言，SQL Server支持R语言处理以后，数据科学家们可以直接利用现有的R代码并在SQL Server数据库引擎上运行。这样我们就不用为了执行R语言处理数据而把SQL Server数据导出来处理。该功能把R语言处理带给了数据。

5.Query Store，如果你经常使用执行计划，你就会喜欢新版的Query Store功能。在2016之前的版本中，你可以使用动态管理试图(DMV)来查看现有执行计划。

6.行级安全(Row Level Security)，SQL数据库引擎具备了行级安全特性以后，就可以根据SQL Server登录权限限制对行数据的访问。限制行是通过内联表值函数过滤谓词定义实现的。安全策略将确保过滤器谓词获取每次“SELECT”或者“DELETE”操作的执行。

## 2.2 系统实现技术简介

### 2.2.1 AngularJS 2简介

在一个互联网迅速膨胀的时代，“前端”在短短的几年内已经发展成为每一个程序员耳熟能详的单词，“PHP虽然是最好的语言”，但它不是最流行的语言。近年来，Javascript凭借它自身的优势，受到了广泛开发者的喜爱和青睐。因此，前端圈子近年来蓬勃发展，各种社区和前端框架也应用而生。如：在数据可视化方面，Processing被带入到了Web领域产生了Processing.js,还出现了D3.js,通过使用HTML、SVG和CSS把数据鲜活形象的展现出来；在移动应用方面，Cordova和React Native实现了一次开发，多平台发布；在服务端，V8的性能将JavaScript带到了一个新的高度，Node.js的诞生，迎来了JS的全栈时代。

AngularJS在前端膨胀发展的当下，成为了最受瞩目和广泛应用的前端框架之一，它作为一个MVVM(Model-View-ViewModel)框架，将视图和模型层分离，使得软件的UI层更加细节化、可定制化，有低耦合、可重用性、独立开发、可测试等特性。

AngularJS自身的优势也显而易见：

1. 模板功能强大丰富，自带丰富的Angular指令，并且是声明式的；
2. 包含模板、数据双向绑定、路由、模块化、服务、过滤器、依赖注入等特性功能；
3. 自定义的Directive,比jQuery插件使用更加灵活，重用性高；
4. Ng模块化大胆的引入了Java的一些东西，比如依赖注入等，加强了代码的可复用性；
5. 支持单元测试和e2e-testing。

Angular2在Angular1的基础上，形成了一个统一的、系统性的架构，在语言层面上，它使用Typescript，在架构层面上，它使用Component的概念统摄全局，大大地简化了概念模型并降低了学习的成本。它还提供了高度集成化的开发工具angular/cli，它以webpack为内核，内置了Karma/Jasmine/Protractor/TS compiler。CLI目前已经发展的相当完善，并推出正式版，可在正式项目中使用。通过CLI，开发者可以创建、开发、测试、构建项目，而不需要额外的依赖。

Angular是一个面向未来的科技，它要求浏览器支持ES6+，所以在开发过程中需要一个工具链进行解析编译等一系列自动化操作。如图2.1所示：

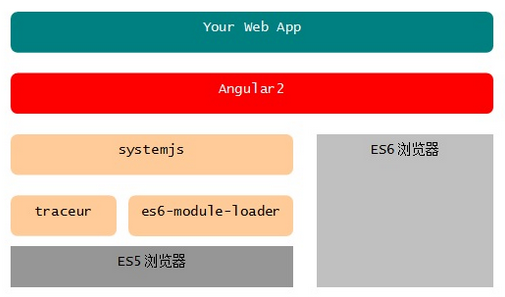


图2.1ES6工具链

Systemjs(通用模块加载器)，支持AMD、CommonJS、ES6等各种格式的JS模块加载；

ES6-module-loader(ES6模块加载器)，systemjs会自动加载这个模块；

Traceur(ES6转码器)，将ES6代码转换为当前浏览器支持的ES5代码。

Angular2中，通过重新设计和引入新技术，摆脱了旧的技术框架束缚，在开发上达到了高整合性、高复用的组件化的开发。在用户体验方面，对速度也有了极致追求，将之前的“多轮检查、直到稳定”策略改成了“一轮检查、直接稳定”策略。

### 2.2.2 Typescript简介

Typescript是微软开发的自由的，开源的编程语言，它是Javascript的强类型版本，语言中包含了后端开发人员熟悉的静态类型和基于类的面向对象编程特点，使得前后端的开发人员在语法的使用上更近了一步。TypeScript是JavaScript类型的超集，它在编译期去掉类型和特有的语法，生成纯粹的JavaScript代码，最终在浏览器中运行。TypeScript可以在任何浏览器、任何计算机和任何操作系统上运行，并且是开源的。TypeScript扩展了JavaScript的语法，所以任何现有的JavaScript程序可以不加改变的在TypeScript下工作。TypeScript是为大型应用之开发而设计，而编译时它产生 JavaScript 以确保兼容性。

由于Typescript是ECMAScript6的超集，因此它具有以下几大特性：

1. 类型批注和编译时类型检查
2. 支持类：继承了可选的类型批注支持的ES6的类
3. 支持接口
4. 支持模块
5. 支持lambda函数
6. Generator

除了它较之Javascript的特性外，Typescript有它先天的优势：

1. 高度兼容原生脚本语法；
2. 对语法的破坏性较少，容易上手；
3. 它讲究基于接口的强类型，因此也适用于原本的服务器开发者；
4. 提供了大量编辑器的集成，现已有大量库的.ts元文件可用。

2.2.3 Git简介

Git是一款免费、开源的分布式版本控制系统，可以有效、高速的处理从小到大的项目版本管理。版本控制是一种记录若干文件内容变化，以便将来查阅特定版本修订情况的系统，在没有自动化的版本控制之前，开发者们习惯复制整个项目目录的方式来保存不同的版本，这种本地版本控制方式经常会导致开发者混淆所在的工作目录，一旦弄错文件丢了数据就没有办法撤销恢复。

为了让在不同系统上的开发者协同工作，集中化的版本控制系统(Centralized Version Control Systems,简称CVCS)应运而生。比较常见的有CVS，Subversion以及Perforce等，都有一个单一的集中管理的服务器，来保存所有的文件修订版本。这需要协同工作的人员连接到该服务器，并取出最新的文件或提交更新。这种版本控制很显而易见的缺点就是中央服务器的单点故障，会造成开发过程中莫大的损失。

不久后分布式版本控制系统的面世又开启了开发者的一场新旅程，较常见的有Git,Mercurial,Bazaar以及Darcs等等。这种版本控制使得客户端并不只是提取最新版本的文件快照，而是把原始的代码仓库完整的镜像下来。所以，在协同开发的过程中，万一服务器发生故障，事后也都可以用镜像来进行本地仓库的恢复，所以得到了广大开发者的青睐。

Git作为如今最常用的协同开发版本控制系统之一，具有以下功能：

1. 从服务器上克隆完整的Git仓库到单机上，包括代码和版本信息等；
2. 在自己的机器上根据不同的开发目的，创建分支，修改代码；
3. 在单机上自己创建的分支上提交代码；
4. 在单机上合并分支；
5. 将服务器上最新版的代码拉取下来，跟自己的主分支合并；
6. 生成补丁(patch)，把补丁发送给主开发者；
7. 主开发者反馈，合理解决冲突；

使用Git进行版本控制的开发模式还具有以下优势：

1. 公共服务器压力和数据量都不会太大；
2. 采用分布式版本库，不需服务器端软件支持；
3. 适合分布式开发，强调个体；
4. 速度快、灵活；
5. 任意两个开发者之间都可以很容易的解决冲突；
6. 离线工作；
7. 协作开发，高效、便捷。

因而，在本项目中，为了方便小组成员之间的协作开发，我们采用了Git的分布式版本控制。

## 2.3 本章小结

本章对本课题的国内外发展情况和选题意义进行了调查及分析，明确了项目的背景信息，并对本系统所涉及的核心技术进行说明。本项目在开发过程中采用前后端的基本方式，后端主要使用以C#语言为根基的个人研发框架作为主开发模式，前端采用了最新较为流行的Web应用框架——基于Typescript语言的Angular2进行开发。本章节也对该项目的开发环境进行了简要的设定，项目环境主要使用了Visual studio 2015和微软公司的Microsoft SQL Server 2014。

# 系统定位与需求分析

## 3.1 背景简介

本题目基于近年来O2O商务模式的迅速发展，致力于更快捷、更有效的提高生活质量的第三方生活服务平台。力求改善人们的同城生活质量，提供多元化的专人速递速成服务，快速响应，解决急需等任何个人化的生活需求，例如：想找人帮忙购物、取件、送件，或是找代驾、帮排队、失物招领、宠物照看、二手处理等等。

## 3.2 网站设计目标

### 3.2.1 总目标

在平台上整合所有个人或团体大大小小的需求，让个人需求能够在平台被快速响应，无论是想找人帮忙购物、取件、送件，还是找代驾、帮排队、失物招领、宠物照看、二手处理等等，在平台上都能找到最快、最佳、最可靠的帮达人，最终达到客户的双方所需。

### 3.2.2 功能目标

1.系统相关用户角色以及对各种角色的用户信息进行管理。

2.在用户的注册登录功能中，添加系统的Token认证，保障用户信息的安全。

3.系统各项功能的访问权限的管理，对不同的用户开放不同的权限进行操作。

4.发布者可以进行下单请求，并且由系统分发订单信息。

5.其他用户抢单接单进行对订单的后续流程的操作。不同的用户角色对订单有不同的操作策略。

6.管理员权限可以对所有用户、订单、消息及投诉等信息进行查看和管理。

7.管理员可以对用户发送系统消息。

8.用户可以对系统提出意见或反馈，也可以对其他用户非法操作提出的投诉信息。

9.系统管理员发起优惠活动，用户可以在规定期间内进行活动的参与。

10.社区模块类似于微信朋友圈等社交媒体的沟通平台，用户可以在平台内进行交流分享。

### 3.2.3 性能目标

1.）友好性：实现网站的快速访问，减少用户等待时间，优化服务性能，提供便捷高响应的用户体验；

2.）安全性：项目所有数据传输都经过加密，同时个人信息访问需权限认证，增加安全系数，防止用户个人基本信息被窃取；

3.）数据库优化，数据库设计需合理，做到高效快速查询。

## 3.3 系统的需求结构

如图3.1所示，为系统需求结构图。

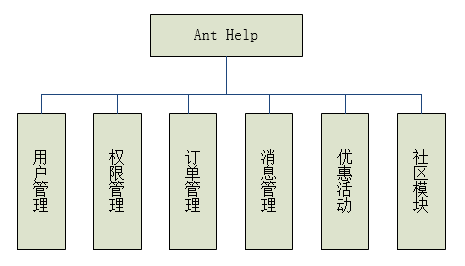


图3.1 系统需求结构图

需求结构说明：

1.用户管理：在用户管理模块中，设计系统相关用户的角色以及对各种角色的用户信息进行管理。在对用户信息进行管理的同时还涉及到了用户的注册登录等功能，该模块有涉及到系统的Token认证相关概念。

2.权限管理：该模块主要用于系统的各项功能的访问权限的管理，对不同的用户开放不同的权限进行操作，使得系统的稳定性和安全性得到了提高，加强了管理员对人员的管理以及系统的维护。

3.订单管理：订单管理模块用户接收用户的下单请求，并且由系统分发订单信息，其他用户抢单接单进行对订单的后续流程的操作。不同的用户角色对订单有不同的操作策略。

4.消息管理：消息管理模块主要用于管理员对用户发送系统消息，或者是用户对系统的意见或建议，以及用户对其他用户的投诉信息，可以向系统投诉信箱发送信息，系统管理员登录后可以查看用户发送来的投诉信息，并进行对投诉信息的处理。

5.优惠活动：优惠活动模块，由系统发起的优惠，用户可以在规定期间内进行活动的参与。

6.社区模块：社区模块类似于微信朋友圈等社交媒体的沟通平台，用户可以在平台内进行交流分享。

## 3.4 用例分析

### 3.4.1 参与者

在订单管理模块，参与者主要是发布者和接收者；在消息管理和活动管理模块，参与者主要是系统管理员。

### 3.4.2 用例图

涉及到发布者和接收者的用例图如图3.2所示。

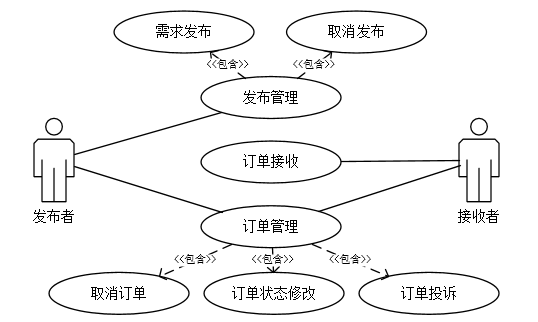


图3.1 发布者和接收者用例图

Fig.3.1 Publisher and Receiver use case diagram

关键用例描述如表3.1与表3.2所示：

表3.1 发布管理用例描述

Table3.1 Publish management use case description

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 发布管理 |
| 用例描述 | 发布者发布个人需求生成新订单，并且可以进行查询以及删除 |
| 参与者 | 发布者 |
| 前置条件 | 已经登录系统的用户，且身份验证合格 |
| 后置条件 | 如果这个用例成功，在系统中建立新的订单记录 |
| 基本操作流程 | 1, 发布者输入发布信息  2，系统验证数据合法性  3，发布者处理异常数据  4，需求发布完成 |
| 可选操作流程 | 1，取消相关操作  2，数据输入不合法，操作失败 |
| 被泛化的用例 | 无 |
| 被包含的用例 | 需求发布，取消发布 |
| 被拓展的用例 | 无 |

表3.2 订单管理用例描述

Table3.2 Order management use case description

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 订单管理 |
| 用例描述 | 发布者对已发布订单、接收者对已接收订单的管理 |
| 参与者 | 发布者、接收者 |
| 前置条件 | 已经登录系统的用户，且身份验证合格 |
| 后置条件 | 如果这个用例成功，订单据相应操作而更新 |
| 基本操作流程 | 1，发布者/接收者登录系统  2，订单相关操作  3，系统进行合法性验证  4，操作成功 |
| 可选操作流程 | 1，取消相关操作  2，操作不合法时，操作失败 |
| 被泛化的用例 | 无 |
| 被包含的用例 | 取消订单、订单状态修改、订单投诉 |
| 被拓展的用例 | 无 |

涉及到系统管理员的用例图如图3.3所示。

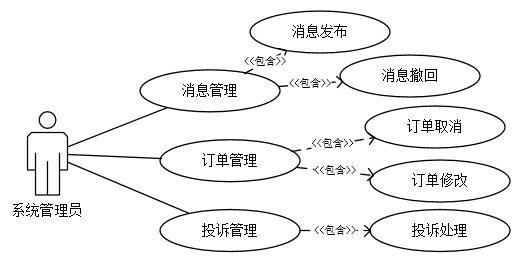


图3.3 系统管理员用例图

Fig.3.3 System manager use case diagram

关键用例描述如表3.3与表3.4所示：

表3.3 消息管理用例描述

Table3.3 Message management use case description

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 消息管理 |
| 用例描述 | 系统管理员对全站消息以及对用户推送消息的管理 |
| 参与者 | 系统管理员 |
| 前置条件 | 已经登录系统且具有管理员权限的用户，并且身份验证合格 |
| 后置条件 | 如果这个用例成功，系统中建立新的消息记录或进行相关消息操作 |
| 基本操作流程 | 1, 管理员输入要推送的消息及所要推送的用户  2，系统验证数据合法性  3，管理员处理异常数据  4，消息推送完成 |
| 可选操作流程 | 1，取消相关操作  2，数据输入不合法，操作失败 |
| 被泛化的用例 | 无 |
| 被包含的用例 | 消息发布，消息撤回 |
| 被拓展的用例 | 无 |

表3.4 订单管理用例描述

Table3.4 Order management use case description

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 订单管理 |
| 用例描述 | 管理员对全站所有订单的管理 |
| 参与者 | 系统管理员 |
| 前置条件 | 已经登录系统且具有管理员权限的用户，并且身份验证合格 |
| 后置条件 | 如果这个用例成功，订单数据据相应操作而更新 |
| 基本操作流程 | 1，管理员登录系统  2，订单相关操作  3，系统进行合法性验证  4，操作成功 |
| 可选操作流程 | 1，取消相关操作  2，操作不合法时，操作失败 |
| 被泛化的用例 | 无 |
| 被包含的用例 | 订单取消、订单修改 |
| 被拓展的用例 | 无 |

## 3.5 可行性分析

### 3.5.1 社会可行性

本项目专门针对现代人们生活所需和生活服务类平台分析而设计。主要包括用户权限管理、订单管理和社区管理三大模块。以生活服务的电子商务的模式，突出客户间随时随地便捷式的交流，充分体现以消费者为中心，满足客户个性化、多样化服务的指导思想。为客户带来方便的、灵活的、高质量的服务，同时产生极大的社会经济价值。

法律因素方面。开发中所有商业软件都采用购买的方式使用正版软件。所有源代码为自主开发，自有知识产权。涉及到第三方控件和支付接口，也通过正规渠道进行商务购买。开发中使用的开源技术不存在法律纠纷。在用户使用可行性方面，用户对于电子采购交易平台的理解和接受需要一定时间周期。具体操作和维护人员需要进行专项培训。

在项目功能实施方面，我们力求做到以下几点：

1. 安全

1.）数据安全：网站所有数据的传输都经过了安全加密传输，避免数据被非法盗取，同时，个人数据访问需要分配权限，无权限不能访问。

2.）交易安全：所有资金先支付到平台管理，消费者确认满意后才能关闭交易。

1. 完善

1.）全面的内容：该服务网站不只是提供简单的商品交易信息，我们更注重生活中的任何个人所需，例如：找代驾、帮排队、找人帮购物、取件、送件，失物招领、宠物照看、二手处理等等。

2.）完善的功能：网站同时提供即时通信交流、评价、优惠活动等。

1. 便捷

在该网站平台上，可以发布或找到自己的任何所需，更加方便快捷，无需成本，更实惠有效。

综上所述，本项目没有违背法律和道德，力求达成一个更安全、更完善、更便捷的生活服务平台，所以在社会上是可行的。

### 3.5.2 经济可行性

本项目专门针对现代人们生活所需和生活服务类平台分析而设计。主要包括用户权限管理、订单管理和社区管理三大模块。该项目采用生活服务类电子商务的模式，突出客户间随时随地便捷式的交流，充分体现以消费者为中心，满足客户个性化、多样化服务的指导思想。为客户带来方便的、灵活的、高质量的服务，同时产生极大的社会经济价值。

在支出方面，搭建该系统需要组建开发团队，需要项目经理一名，架构师一名，分析员一名，测试员两名，程序员若干位，预计开发过程需要6个月。除了需要支出组建团队的费用外，硬件设备支出方面，需要内网数据库服务器2台， 存储系统，备份系统，Web服务器2台，应用服务器2台，网络设备，还有安全设备。软件支出方面，需要服务器操作系统（应用内网和外网服务器操作系统Unix、Linux 、Windows10），SQL server数据库，备份软件，虚拟化软件，Web 中间件，杀毒软件等等。

其他费用除了项目中所需的固定资产投资，如人工费、材料费，其他支出流动资金还包括、差旅费用、调研费用等，最后还有系统的运行维护费用等等。

### 3.5.3 技术可行性

在功能目标能够如期达到的情况下，按对项目影响程序由大到小的依次排序为：资金、人员配置、软硬件设备。在技术可行性上具体体现在以下几个方面：

1. 开发技术：

本项目开发平台基于适用于Android、iOS、Windows、Web和云的应用等集于一体的Visual Studio 2015，可以快速编码、轻松调试和整断、方便测试与发布。同时，为了方便团队的协作开发，采用Git版本控制器，可以更有效的解决代码冲突，提高团队开发的效率等。

1. 系统功能分析与设计：

项目中采用面向对象的软件开发方法和思路，将抽象的事物以静态的、结构化的系统数据性质表示出来，提高了程序的灵活性、重用性和可维护性。

1. UML模型分析与设计：

UML作为一种建模标准，是面向对象开发过程中必不可少的，它为软件开发的所有阶段提供模型化和可视化支持，便于开发和理解。

### 3.5.4 可行性分析结论

通过对系统的经济、技术、社会等可行性分析，可以确定该系统具有开发的完全必要性和可行性，予以立项开发。

## 3.6 系统的风险分析与处理策略

对于以O2O模式开发的第三方生活服务平台而言，其订单完成的过程有可能是非面对面的方式完成，所以，诚信与互联网科技的安全与否变得十分的重要。作为一种新的电子上午的交易方式，相关法律法规的不健全等问题也给系统的开发带来了诸多不确定的风险因素。而对于平台客户而言，主要也包括了以下几点风险分析：

1. 信息风险：主要指网站上所发布的商品或其他信息的可靠性；
2. 隐私风险：指的是平台用户的个人隐私可能存在潜在的泄露风险；
3. 付款风险：指在互联网上使用信用卡等支付方式带来的风险；
4. 产品质量风险：主要包括平台发布需求信息的功能、描述是否与实际所需相一致；
5. 时间风险：主要指个人在平台发布的需求是否能够被快速响应，并在预期时间内完成。

对于开发人员来说，系统的开发目标必须以满足用户需求为前提，而在系统开发的第一阶段中，用户的需求往往是不完整的、不明确的，需要开发者协助用户对需求做进一步的理解乃至创造出较为详细的需求说明方法的特点是可以在整个开发过程中回头补充调整各阶段的工作，因此，用户需求分析在最初并不显得像结构化方法那么重要。

然而，一个相对完整、准确的用户信息系统开发带来良好的开端，也会使开发过程少走许多弯路。所以，需要很长时间进行详细的用户需求调查。通过对三方服务系统的分析，我们所采取的处理策略有：

1. 开发过程使用更安全、更有效的算法，尽量消除和减少编程错误；
2. 不断回溯检查文档设计，及时发现问题、解决问题；
3. 使用专业测试工具，消除和减少程序BUG；
4. 提高网站安全性能和响应速率；
5. 收集反馈意见并改正，加强后期维护。

# 概要设计

概要设计的主要任务是把[需求分析](http://baike.baidu.com/view/111493.htm)得到的系统模型转换为数据结构和[软件结构](http://baike.baidu.com/view/600142.htm)。数据[结构设计](http://baike.baidu.com/view/411272.htm)包括数据特征的描述、确定数据的结构特性、以及[数据库](http://baike.baidu.com/view/1088.htm)的设计。设计软件结构的具体任务是：确定一个复杂系统的功能模块划分、建立模块的[层次结构](http://baike.baidu.com/view/420833.htm)及调用关系、确定模块间的接口及人机界面等的初步设计。

## 4.1 系统功能模块划分

### 4.1.1功能模块图

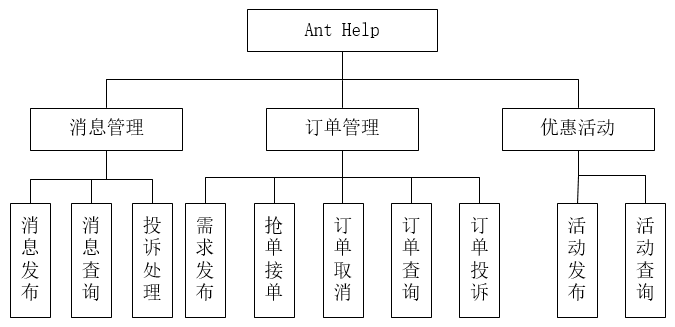


图4.1 消息、订单、活动管理功能模块图

消息管理、订单管理以及活动管理的功能模块图如图4.1所示：

1. 消息管理：
2. 消息发布：用于管理员对用户发送系统消息，发送后在用户的消息栏内进行通知，用户可进行查看；
3. 消息查询：方便管理员进行消息的统一管理；
4. 投诉处理：投诉信息来源于网站投入使用后的各个流程，用户对系统的建议或对其他用户的投诉信息，可以向系统投诉信箱发送信息，系统管理员可以查看用户发送来的投诉信息，并进行对投诉信息的处理。
5. 订单管理：
6. 需求发布：指用户在该平台上发布个人需求所生成的发布订单；
7. 抢单接单：用户之间具有抢单竞争，同一发布订单，只能被另一个用户所接单，同时接单时由系统自动分配，其他用户接单情况显示操作失败；
8. 订单取消：发布的订单在未接单的情况下，可以进行取消操作；
9. 订单查询：用户可对自己发布的订单及自己的接单进行条件帅选式的查询；
10. 订单投诉：在任何订单状态(待接单、待支付、待完成、待确认、已结束等)中，都可以进行投诉，投诉信息反馈给系统，有管理员权限即可可进行后续处理。
11. 优惠活动：
12. 活动发布：活动的发布只能由具有管理员权限的用户发起，发布完成后其他用户可见并参与优惠活动中；
13. 活动查询：方便管理员对优惠活动的统一管理。

## 4.2 数据库设计

数据库设计是应用系统及软件设计中必不可少的步骤之一，是应用开发系统化的核心技术。它用来满足用户的应用需求，能够有效地存储数据，进行方便快捷式的管理。好的数据库设计是软件设计好坏的评估标准之一，因此构造良好的数据模式具有重大的意义。

### 4.2.1 概念结构分析

数据库结构分析阶段是在需求分析的基础上，从属于概念结构设计阶段，将需求分析得到的用户需求 (如订单、用户、消息、投诉信息等)抽象为信息结构，即概念结构。

描述概念模型的直接有效表现途径是E-R图(Entity-Relationship,实体-联系模型)，设计出能够满足用户需求的各种实体，以及实体之间的关系，为以后的逻辑结构打下基础。根据上一章的需求分析，可以得出如图4.1所示的本人负责模块的实体及实体关系图。

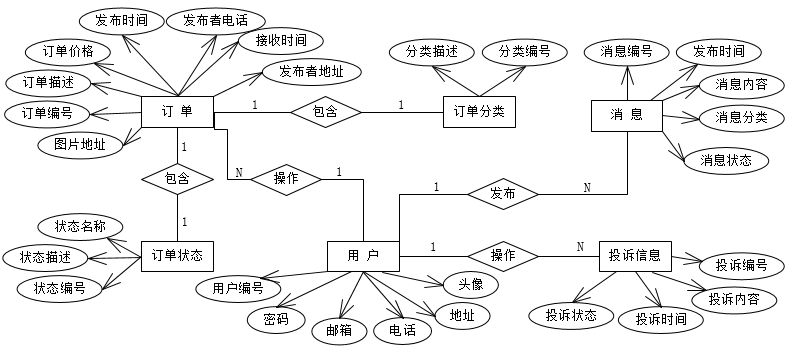


图4.4 订单、消息管理ER图

Fig.4.4 Order and message management ER diagram

各个实体与实体联系的描述具体如下：

1. 实体： 用户实体(用户编号，密码，邮箱，电话，地址，头像)；

订单状态实体(状态编号，状态名称，状态描述)；

订单分类实体(分类编号，分类描述)；

订单实体(订单编号，订单描述，订单价格，发布时间，发布者电话，接收时间，发布者地址，图片地址)；

消息实体(消息编号，发布时间，消息内容，消息分类，消息状态)；

投诉信息实体(投诉编号，投诉内容，投诉时间，投诉状态)；

2.实体关系：

1.）订单与订单状态之间是包含关系，一对一的关系，一个订单包含一个订单状态；

2.）订单与订单分类之间也是包含关系，一对一的关系，一个订单包含一个订单分类；

3.）用户与订单详情之间是操作关系，一对多的关系，一个用户可以拥有多个订单，并对其进行相关操作；

4.）用户与消息之间是发布关系，一对多的关系，一个用户可以发布多个消息；

5.）用户与投诉信息之间是操作关系，一对多的关系，一个用户可以发起多个投诉信息；

### 4.2.2 物理结构设计

在数据库设计的过程中，依据以上的概念结构分析，设计出E-R模型之后，充分考虑了数据表的结构实现及数据表之间的联系，同时综合考虑数据流向和查询效率的情况下，设计了如下的物理结构设计，即数据表结构：

1.订单表(Order)

表3.1 订单基本数据表

Table 3.1 Order basic data table

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否为NULL | 含义 |
| orderUid | uniqueidentifier | NOT NULL | 订单编号 |
| orderDescription | nvarchar(200) | NOT NULL | 订单描述 |
| orderCategoryId | int | NOT NULL | 订单分类编号 |
| publisherUid | uniqueidentifier | NOT NULL | 发布者编号 |
| publishTime | datetime | NOT NULL | 发布时间 |
| receiverUid | uniqueidentifier | NOT NULL | 接收者编号 |
| receiveTime | datetime | NOT NULL | 接收时间 |
| orderStatusId | int | NOT NULL | 订单状态编号 |
| orderValue | int | NOT NULL | 订单价格 |
| address | nvarchar(50) | NOT NULL | 地址 |
| phone | int | NOT NULL | 电话 |

2.订单分类表(OrderCategory)

表3.2 订单分类数据表

Table 3.2 Order Category data table

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否为NULL | 含义 |
| orderStatusId | int | NOT NULL | 订单状态编号 |
| orderStatusName | nvarchar(50) | NOT NULL | 订单状态名字 |
| orderStatusDescription | nvarchar(50) | NOT NULL | 订单描述 |

3.用户信息表(UserAccountInfo)

表3.3 用户信息数据表

Table 3.3 User accout Information data table

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否为NULL | 含义 |
| uid | [numeric](18, 0) | NOT NULL | 控制项目标示 |
| CONTROL\_TYPE | [varchar](20) | NOT NULL | 类别 |
| CONTROL\_NAME | [nvarchar](50) | NOT NULL | 控制项目 |
| TEMPLATEID | [numeric](18, 0) | NULL | 模板ID |
| TEMPLATENAME | [varchar](50) | NULL | 模板名称 |
| PROCESSESID | [numeric](18, 0) | NOT NULL | 工序ID |
| PROCESSNAME | [varchar](50) | NOT NULL | 工序名称 |
| SAMPLE\_CAPACITY | [nvarchar](20) | NOT NULL | 样本容量 |
| FREQUENCY | [nvarchar](20) | NOT NULL | 频率 |
| STANDARD | [nvarchar](20) | NULL | 工程规范 |
| CREATER | [varchar](20) | NOT NULL | 创建人 |
| CREATTIME | datetime | NOT NULL | 创建时间 |
| UPDATER | [varchar](20)] | NOT NULL | 更新人 |
| UPDATETIME | [datetime] | NOT NULL | 更新时间 |
| PIPELINE | [nvarchar](20) | NULL | 管道 |
| PROCESS\_TYPE | [nvarchar](20) | NULL | 类型 |
| CPTYPE | [int] | NULL | 0-电芯, 1-PACK |

4.BOM号表

用于存放物料清单相关信息。如表3.4所示：

表3.4 批次表（T\_REAL\_FLOW）

Table 3.4 Batch list (T\_REAL\_FLOW)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否为NULL | 含义 |
| INSTANCEID | [numeric](18, 0) | NOT NULL | 控制项目标示 |
| BOMNO | [varchar](255) | NOT NULL | 类别 |
| SIZENO | [varchar](255) | NOT NULL | 控制项目 |
| SIZENO\_TYPE | [int] | NOT NULL | 批号类型 |
| PRODUCE\_TYPE | [varchar](20) | NULL | 生产类型 |
| C\_SIZENO | [varchar](255) | NULL | 卷绕批号 |
| A\_SIZENO | [varchar](255) | NULL | 装配批号 |
| F\_SIZENO | [varchar](255) | NULL | 化成合批批号 |
| OLD\_INSTANCEID | [numeric](18, 0) | NULL | 插入编号 |
| PIPELINE | [nvarchar](20) | NULL | 管道 |
| PROCESSDEFINITIONID | [numeric](18, 0) | NOT NULL | 工序 |
| FLOW\_NAME | [varchar](100) | NOT NULL | 线体 |
| BOMNO\_SIZENO | ([BOMNO]+[SIZENO]) | NOT NULL | BOM号和批次号 |
| NEXT\_NODEID | [numeric](18, 0) | NOT NULL | 下次标注编号 |
| NEXT\_NODENAME | [varchar](50) | NOT NULL | 下次标注名 |
| FRONT\_NODEID | [numeric](18, 0) | NOT NULL | 前置标注编号 |
| FRONT\_NODENAME | [varchar](50) | NOT NULL | 前置标注名称 |
| NOW\_NODEID | [numeric](18, 0) | NOT NULL | 当前标注编号 |
| NOW\_NODENAME | [varchar](50) | NOT NULL | 当前标注名 |
| FIRST\_NODEID | [numeric](18, 0) | NULL | 第一次标注编号 |
| FIRST\_NODENAME | [varchar](50) | NULL | 第一次标注名称 |
| LAST\_NODEID | [numeric](18, 0) | NULL | 最后标注编号 |
| LAST\_NODENAME | [varchar](50) | NULL | 最后标注时间 |
| STATE | [varchar](10) | NULL | 状态 |
| LOCKFLAG | [varchar](5) | NULL | 是否锁定 |
| CREATER | [varchar](20) | NOT NULL | 创建者 |
| CREATTIME | datetime | NOT NULL | 创建时间 |
| UPDATER | [varchar](20) | NULL | 更新人 |
| UPDATETIME | datetime | NOT NULL | 更新时间 |
| ISCLOSED | int | NULL | 是否封闭 |

5.预控图与控制图数据表

用来存储预控图或控制图相关信息。如表3.5所示：

表3.5 预控图与控制图数据表（T\_XR\_DATA）

Table 3.5 Re control chart and the control chart data table (T\_XR\_DATA）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 是否为NULL | 含义 |
| vseq | [numeric](18, 0) | NOT NULL | 主键ID |
| TYPE | [varchar](50) | NOT NULL | 类型 |
| PROJECT | [varchar](50) | NOT NULL | 控制项目 |
| PROCESS | [varchar](50) | NOT NULL | 工序 |
| BOM | [varchar](50) | NOT NULL | BOM号 |
| LINE | [varchar](50) | NOT NULL | 流水线 |
| WORKSHOP | [varchar](50) | NOT NULL | 车间 |
| LINECODE | [varchar](50) | NOT NULL | 线体代码 |
| DEVICE | [varchar](50) | NULL | 设备 |
| UCL | [numeric](9, 4) |  | UCL |
| CL | [numeric](9, 4) | NULL | CL |
| LCL | [numeric](9, 4) | NULL | LCL |
| UCLR | [numeric](9, 4) | NULL | UCLR |
| CLR | [numeric](9, 4) | NULL | CLR |
| LCLR | [numeric](9, 4) | NULL | LCLR |
| UPPER\_LIMIT | [numeric](9, 4) | NULL | 上限 |
| LOWER\_LIMIT | [numeric](9, 4) | NULL | 下限 |
| TEST\_DATE | datetime | NULL | 测试日期 |
| WORKNAMEID | [nchar](20) | NULL | 工作编号 |
| CLASSNAME | [varchar](50) | NOT NULL | 班次 |
| WORKNAME | [varchar](50) | NOT NULL | 操作人 |
| CONFIRMNAME | [varchar](50) | NOT NULL | 确认人 |
| SIZENO | [varchar](50) | NOT NULL | 批号 |
| SAMPLETIME | [datetime] | NOT NULL | 样本时间 |
| SAMPLEGROUPNUMBER | [varchar](50) | NOT NULL | 样本组号 |
| SAMPLENUMBER | [varchar](50) | NOT NULL | 样本号 |
| SAMPLEVALUE | [numeric](9, 4) | NOT NULL | 样本值 |
| PROGRAM\_TYPE | [varchar](50) | NOT NULL | 类型 |
| NOCOLREASON | [varchar](255) | NULL | 失控原因 |
| RESOLVENT | [varchar](255) | NULL | 纠正措施 |

### 4.2.2 数据库表间关系

控制项目数据表与其他数据表间是主外键的关系，控制项目表是主键表，其他表中有相应的外键与其产生主外键约束关系。同时物料清单表中的BOM号作为主键也与其他表有主外键约束关系。如图3.10所示：

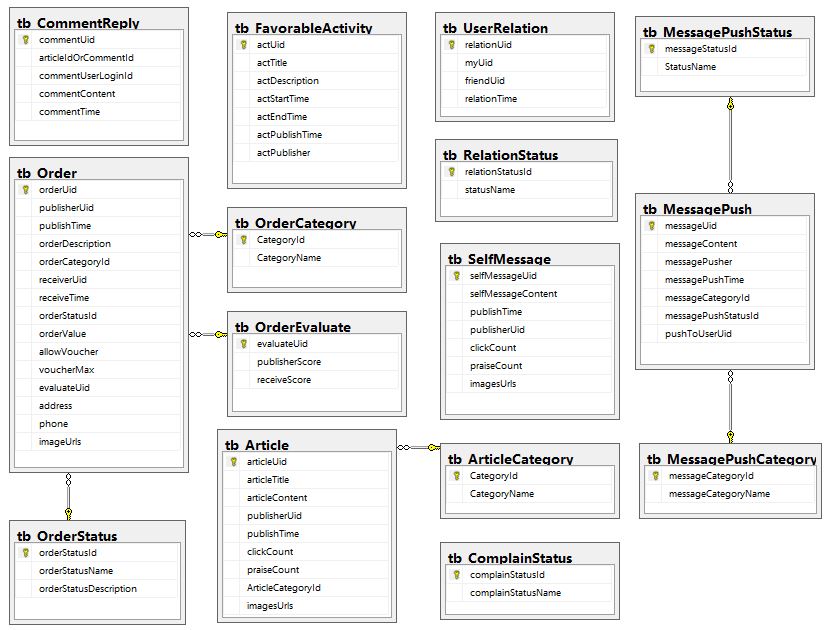


图3.10 数据库表间关系图

Fig .3.10 Relationship between database tables

## 4.3 本章小结

本章对统计过程管理进行了概要设计。本部分分析并绘制了该项目的系统结构图，明确了该系统的整体框架。然后，对该系统中所涉及的实体进行了分析，通过对项目中所涉及的实体进行分析，绘制了各实体的图例，之后绘制了实体之间的联系图（即全局实体联系图）和数据库表结构。而后通过对系统结构和实体联系的分析，抽象出了系统的数据库表关系图。紧接着，本章对系统的数据流向进行了分析，明确了该系统的数据流向后，绘制了数据流图，包括顶层数据流图、二层数据流图和三层数据流图。

# 详细设计

概要设计的主要任务是把[需求分析](http://baike.baidu.com/view/111493.htm)得到的系统模型转换为数据结构和[软件结构](http://baike.baidu.com/view/600142.htm)。数据[结构设计](http://baike.baidu.com/view/411272.htm)包括数据特征的描述、确定数据的结构特性、以及[数据库](http://baike.baidu.com/view/1088.htm)的设计。设计软件结构的具体任务是：确定一个复杂系统的功能模块划分、建立模块的[层次结构](http://baike.baidu.com/view/420833.htm)及调用关系、确定模块间的接口及人机界面等的初步设计。

## 5.1 业务逻辑类设计

### 5.1.1功能模块图

消息管理、订单管理以及活动管理的功能模块图如图4.1所示：

1. 消息管理：
2. 消息发布：用于管理员对用户发送系统消息，发送后在用户的消息栏内进行通知，用户可进行查看；
3. 消息查询：方便管理员进行消息的统一管理；

## 5.2 业务逻辑类设计

### 5.2.1功能模块图

消息管理、订单管理以及活动管理的功能模块图如图4.1所示：

1. 消息管理：
2. 消息发布：用于管理员对用户发送系统消息，发送后在用户的消息栏内进行通知，用户可进行查看；
3. 消息查询：方便管理员进行消息的统一管理；

# 系统实现与代码编写

概要设计的主要任务是把[需求分析](http://baike.baidu.com/view/111493.htm)得到的系统模型转换为数据结构和[软件结构](http://baike.baidu.com/view/600142.htm)。数据[结构设计](http://baike.baidu.com/view/411272.htm)包括数据特征的描述、确定数据的结构特性、以及[数据库](http://baike.baidu.com/view/1088.htm)的设计。设计软件结构的具体任务是：确定一个复杂系统的功能模块划分、建立模块的[层次结构](http://baike.baidu.com/view/420833.htm)及调用关系、确定模块间的接口及人机界面等的初步设计。

## 6.1 系统编码架构

消息管理、订单管理以及活动管理的功能模块图如图4.1所示：

1. 消息管理：

消息发布：用于管理员对用户发送系统消息，发送后在用户的消息栏内进行通知，

## 6.2 系统业务实现

### 6.2.1订单业务

订单业务是整个系统业务实现的核心部分，它包含了订单分类、订单发布、已发布订单的管理、接单抢单操作、已接单订单的管理等相关业务的实现。以下为订单相关业务的详细说明。

**1.订单分类**

如图6.1为首页显示的订单分类部分。



图6.1 首页订单分类

Fig.6.1 System Index interface

首页的订单分类主要展示了订单的各个分类及分类下的各个订单项，按照用户的需求分析，将订单主要划分为以下9类：代理销售、维修装修、汽车服务、免费专场、宠物之家、代帮代购、休闲服务、家政服务和其他服务。发布者在发布时可根据自己的需求选择相应的分类，便捷有效的进行用户订单的查看。

**2.订单发布**

订单发布界面如图6.1所示。



图6.1 订单发布界面

Fig.6.1 Order publish interface

订单发布是通过Form表单实现用户的输入，兼容图片上传功能，上传成功后返回服务器图片地址，在表单中进行预览显示。同时，要求对用户输入的数据进行合法性验证：价格输入不能为非数字，电话号码输入格式是否正确等。

**3.已发布订单的管理**

在个人中心，用户可以对自己发布所生成的订单进行管理，管理界面和查看订单详情界面如图6.1、6.2所示。



图6.1 已发布订单管理界面

Fig.6.1 Published order management interface



图6.1 查看订单详情界面

Fig.6.1 Order detail interface

在用户对已发布的订单(即‘我的发布记录’)进行管理时，按照订单状态将订单划分为以下5类：待支付订单、待接单订单、待确认订单、已结束订单和已取消订单。

在订单发布成功后，将订单状态修改为待支付，用户只有在该订单被支付后才能在平台上对其他用户可见，进而被其他用户接单抢单；在支付过程中，用户账户余额不足时，提醒用户先去充值，充值成功后，且余额充足的情况的可进行支付，支付成功后的订单状态为待接单，会在平台首页或分类页上加载该订单，等待其他用户的接单抢单响应。当其他用户接单抢单成功，并反馈已经完成了该订单的需求，订单状态为待确认，此时需要发布者确认订单的完成。发布者确认接单者完成无误后，订单状态修改为已结束，该交易结束。已取消的订单生成在未支付与待支付的过程中，即在发布者未支付或发布者已支付、接单者未接单时，发布者可以进行取消该订单的操作。

**4.接单抢单操作**

如图6.1为订单详情界面，其中包含了用户的接单抢单操作。



图6.1 订单详情界面

Fig.6.1 System Index interface

订单详情界面展示了相关订单的主要信息，包括：发布者、价格、分类、简介、地址、图片描述等。该订单在没有被接单或抢单的状态下，有红色的‘立即抢单’按钮，已登录的用户可进行抢单操作，游客只能进行浏览，不能进行抢单操作，进行该操作时，提示用户先登录。需要注意的是，已登录的用户只能抢别人发布的需求单，本人发布的订单不允许本人抢单，避免借助平台刷单的情况。当订单被某一用户接单成功后，红色的‘立即抢单’按钮会置为灰色的‘手慢一步，试试其他订单吧~’按钮，并禁止用户的抢单操作，保持唯一性，即一个订单只能被一个用户所接单。

**5.已接单订单的管理**

在个人中心，用户可以对自己已经接收或抢到的订单进行管理，管理界面和已接单订单详情界面如图6.1、6.2所示。



图6.1 已接单订单管理界面

Fig.6.1 Received order management interface



图6.1 已接单订单详情界面

Fig.6.1 Received order detail interface

在用户对已经接收或抢到的订单(即‘我的接单记录’)进行管理时，按照订单状态将订单划分为以下3类：待完成订单、已完成订单和已结束订单。

在接单或抢单成功后，订单的为待完成，接单者按照发布者的发布需求，于线下完成该需求，即可在平台上将订单详情中的订单状态修改为已完成。当接单者标记为已完成订单后，需要发布者确认订单的完成，确认完成无误后，发布者将订单标记为已结束，该交易结束。

### 6.2.2活动业务

活动业务是指管理员发布平台的相关优惠活动或重点文案性消息通知等，要求用户参与其中或只进行了解的重要信息的展示，如图6.1所示为活动中心界面。



图6.1 活动中心界面

Fig.6.1 Activity center interface

### 6.2.3消息业务

消息业务是系统消息管理模块的重要体现部分，从用户角色出发可以将消息业务划分为管理员消息管理和普通用户消息管理两种。

**1.管理员消息管理**

如图6.2和6.3所示，为管理员消息管理界面，包括消息推送界面和消息管理界面两部分。

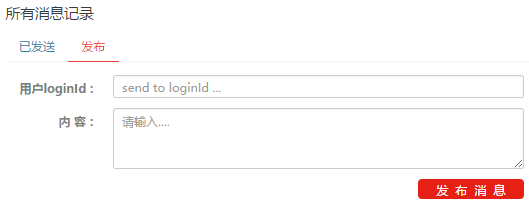


图6.1 消息推送界面

Fig.6.1 Publish message interface



图6.1 消息管理界面

Fig.6.1 Message management interface

管理员(即具有管理权限的用户)可对某用户进行消息推送，推送成功后用户可在自己的消息列表中进行查看。同时管理员对已推送的消息可进行撤回操作，撤回成功后从用户的消息列表中删除该条消息，用户不可见。

**2.普通用户消息管理**

如图6.2-6.3所示，为普通用户消息管理界面，包括未读消息列表界面、查看消息界面和已读消息列表界面。



图6.1 未读消息列表界面

Fig.6.1 Unread message list interface



图6.1 消息查看界面

Fig.6.1 View message interface



图6.1 已读消息列表界面

Fig.6.1 Readed message list interface

当管理员对某一用户进行消息推送后，消息会在用户的未读消息列表中显示，用户可对该条新消息进行查看操作，查看后，消息自动标记为已读消息。用户可在已读消息列表中对消息进行删除操作。为了方便用户查阅，每条消息通知罗列时都对其进行了相应的编号。

## 6.3 本章小结

本章对系统的编码结构及业务实现进行了分析，通过分析更加明确了系统的目标和整体结构。主要包括订单业务、活动业务和消息业务的实现结果分析，其中订单业务作为整个系统实现的核心业务，详细分析了订单分类、订单发布、已发布订单的管理、接单抢单操作、已接单订单的管理等相关订单业务的实现，使得用户行为分析更加明了，充分体现系统‘你需求，我实现’的总目标。具体代码实现见附录。

# 系统测试

软件测试是软件开发及维护阶段中质量控制的不可或缺的步骤之一，以下对本系统的实现进行了软件测试分析和测试实现。

## 7.1 测试目的

软件测试的目的是尽可能的发现软件中存在的错误，并进行改正，提高软件安全性和可靠性。软件测试的关键在于“分析”和“发现”，分析包含了项目的开发过程的分析和业务实现是否合理化的分析，发现指的是发现软件开发过程中不易发现或者容易被忽略部分的错误。

## 7.2 测试过程

### 7.2.1功能模块图

消息管理、订单管理以及活动管理的功能模块图如图4.1所示：

1. 消息管理：

消息发布：用于管理员对用户发送系统消息，发送后在用户的消息栏内进行通知，

## 7.3 测试结论

### 7.3.1功能模块图

消息管理、订单管理以及活动管理的功能模块图如图4.1所示：

1. 消息管理：

消息发布：用于管理员对用户发送系统消息，发送后在用户的消息栏内进行通知，

## 7.4 本章小结

本章对统计过程管理进行了概要设计。本部分分析并绘制了该项目的系统结构图，明确了该系统的整体框架。然后，对该系统中所涉及的实体进行了分析，通过对项目中所涉及的实体进行分析，绘制了各实体的图例，之后绘制了实体之间的联系图（即全局实体联系图）和数据库表结构。而后通过对系统结构和实体联系的分析，抽象出了系统的数据库表关系图。紧接着，本章对系统的数据流向进行了分析，明确了该系统的数据流向后，绘制了数据流图，包括顶层数据流图、二层数据流图和三层数据流图。

# 总结

## 8.1 结论

消息管理、订单管理以及活动管理的功能模块图如图4.1所示：

1. 消息管理：

消息发布：用于管理员对用户发送系统消息，发送后在用户的消息栏内进行通知，

## 8.2 展望

本章对统计过程管理进行了概要设计。本部分分析并绘制了该项目的系统结构图，明确了该系统的整体框架。然后，对该系统中所涉及的实体进行了分析，通过对项目中所涉及的实体进行分析，绘制了各实体的图例，之后绘制了实体之间的联系图（即全局实体联系图）和数据库表结构。而后通过对系统结构和实体联系的分析，抽象出了系统的数据库表关系图。紧接着，本章对系统的数据流向进行了分析，明确了该系统的数据流向后，绘制了数据流图，包括顶层数据流图、二层数据流图和三层数据流图。据的结构特性、以及[数据库](http://baike.baidu.com/view/1088.htm)的设计。设计软件结构的具体任务是：确定一个复杂系统的功能模块划分、建立模块的[层次结构](http://baike.baidu.com/view/420833.htm)及调用关系、确定模块间的接口及人机界面等的初步设计。

# 参考文献

[1] 张正礼，王坚宁. ASP.NET4.0从入门到精通. 清华大学出版社.P1

[2] 高宏，李俊民. ASP.NET典型模块与项目实战大全. 清华大学出版社.P1

[3] 戴宗友,张伍荣,杨辉.C#程序设计实训.清华大学出版社,2009.P1

[4] 甘志,C#高级应用开发篇.清华大学出版社,2008.P2

[5] 蒋金楠.ASP.NET MVC 4框架揭秘.电子工业出版社，2013.P2

[6] 高云.jQuery技术内幕：深入解析jQuery架构设计与实现原理.机械工业出版社，2014.P2

[7] 王玫丽.浅析OA 系统(网络办公自动化)的开发.科技信息(科学教研),2008.P2

[8] 雍珣. 基于WEB 的办公自动化系统的设计与实现. 山西广播电视大学学

报,2009.P2

[9] 周德荣.基于.NET 环境的OA 系统框架设计与实现.福建电脑,2007.P3

[10] 朱玉超,鞠艳,王代勇.ASP.NET 项目开发教程.北京：电子工业出版社,2008.P2

[11] 陈伟.SQL Server 2005 数据库应用与开发教程.北京：清华大学出版社,2007.P2

[12] 尹增明.ASP.NET Web 应用开发.机械工业出版社,2008.P5

[13] 李红日,宋俊兰.用SQL 语句实现关系代数运算探讨.科技信息(学术版)，2008.P5

[14] Nielsen Paul.SQL Server 2005 宝典.北京:人民邮电出版社,2008.P5

[15] 胥光辉.软件工程方法与实践.机械工业出版社,2009.P5

# 附 录

在本系统设计的过程中，老师为我们选定了该系统的方向，为我们分析了基本的客户需求。为我们的系统拟定了一个大致的方向，我们的团队就依着该方向进行领域分析，可行性分析和需求分析。我们为系统搭好了框架，并且老师对我们框架的细节进行完善，给予了细致的指导，给我们提出了具有建设性的建议，

# 致 谢

在本系统设计的过程中，老师为我们选定了该系统的方向，为我们分析了基本的客户需求。为我们的系统拟定了一个大致的方向，我们的团队就依着该方向进行领域分析，可行性分析和需求分析。我们为系统搭好了框架，并且老师对我们框架的细节进行完善，给予了细致的指导，给我们提出了具有建设性的建议，老师以他那渊博的知识、开阔的视野和那丰富的实战经验在写项目的时候给予了我们很大的帮助。

在这里我还要感谢我们团队里的每一个人，我们这个团队一共由留个人组成，因为我们的分工很明确，所以每个人都有各自的任务，我们团队里的每一个人都非常认真，负责的做着自己的那一份任务，没有以一个人一个人原因而耽误项目的进度。在做项目的过程中，我们虽然各自有着自己的工作，但是我们还互相帮助。并且一个团队里有多少人就有多少种思路，我们都会坐下来一起商量。就这样我们的团队从不稳定，慢慢的变的稳定。在这里我要感谢我的团队，感谢他们的兢兢业业，因为有了他们我们的项目才能够顺利，如期的完成。

在完成项目的时候，同学们给了我很多的意见和建议。同时我还参考了好多网上的资料，也给了我很大的启发。还有在这里还是要着重的感谢一下我的指导老师。感谢老师在我的专业思维及技能方面给予我的帮助，在老师的细心指导和鼎力的支持下让我变的非常有信心。就这样在这么多人的帮助下我顺利的完成了这个系统 。在这里感谢所有授我以业的老师，如果没有这么多年知识的积累，我也就没有有够的信心来完成这个系统。在这里，我非常诚恳的请老师对我的系统多加批评和指正，这样我就能够及时的修改我的系统，使我的系统更加的完善。