|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 北航校园购物网站软件开发计划   |  |  |  |  | | --- | --- | --- | --- | | **项目组成员信息** | | | | | **小组名称** | 北航校园购物网站软件开发小组 | | | | **学号** | **姓名** | **本文档中主要承担的工作内容** | **权重** | | 14061201 | 杨健 | 组长，撰写者，修改者，审核者 | 25% | | 14061125 | 于建勋 | 组员，撰写者 | 25% | | 14231012 | 李旭 | 组员，撰写者 | 25% | | 14061180 | 张绍钧 | 组员，撰写者 | 25% | |

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| 1.0 | 2016 10 10 | 张绍钧 | 杨健 |  |
| 2.0 | 201610 14 | 李旭 | 杨健 |  |
| 3.0 | 2016 10 15 | 杨健 | 于建勋 |  |
| 4.0 | 2016 10 15 | 于建勋 | 杨健 |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1. 引言 7](#_Toc464332160)

[1.1标识 7](#_Toc464332161)

[1.2系统概述 7](#_Toc464332162)

[1.3文档概述 8](#_Toc464332163)

[1.4与其他计划之间的关系 9](#_Toc464332164)

[1.5 术语和缩略词 9](#_Toc464332165)

[1.6基线 9](#_Toc464332166)

[2. 引用文件 10](#_Toc464332167)

[3. 交付产品 10](#_Toc464332168)

[3.1程序 10](#_Toc464332169)

[3.3服务 11](#_Toc464332170)

[3.4非移交产品 11](#_Toc464332171)

[3.5最后交付期限 12](#_Toc464332172)

[4. 所需工作概述 13](#_Toc464332173)

[4.1对所要开发系统、软件的需求和约束 13](#_Toc464332174)

[4.2对项目文档编制的需求和约束 13](#_Toc464332175)

[4.3该项目在系统生命周期中所处的地位 13](#_Toc464332176)

[4.4项目进度安排及资源的需求和约束 13](#_Toc464332177)

[4.5其他的需求和约束 14](#_Toc464332178)

[5. 实施整个软件开发活动的计划 14](#_Toc464332179)

[5.1软件开发过程 14](#_Toc464332180)

[5.2软件开发总体计划 15](#_Toc464332181)

[5.2.1软件开发方法 15](#_Toc464332182)

[5.2.2软件产品标准 15](#_Toc464332183)

[5.2.3可重用的软件产品 16](#_Toc464332184)

[5.2.4处理关键性需求 17](#_Toc464332185)

[5.2.5计算机硬件资源利用 18](#_Toc464332186)

[5.2.6记录原理 18](#_Toc464332187)

[5.2.7需方评审途径 18](#_Toc464332188)

[6.实施详细软件开发活动的计划 19](#_Toc464332189)

[6.1项目计划和监督 19](#_Toc464332190)

[6.1.1软件开发计划 19](#_Toc464332191)

[6.1.2 CSCI测试计划 19](#_Toc464332192)

[6.1.3系统测试计划 21](#_Toc464332193)

[6.1.4软件安装计划 21](#_Toc464332194)

[6.1.5软件移交计划 21](#_Toc464332195)

[6.1.6跟踪和更新计划，包括评审管理和时间间隔 21](#_Toc464332196)

[6.2建立软件开发环境 22](#_Toc464332197)

[6.2.1软件工程环境 22](#_Toc464332198)

[6.2.2软件测试环境 22](#_Toc464332199)

[6.2.3软件开发库 23](#_Toc464332200)

[6.2.4软件开发文档 23](#_Toc464332201)

[6.2.5非移交软件 24](#_Toc464332202)

[6.3系统需求分析 24](#_Toc464332203)

[6.3.1用户输入分析 24](#_Toc464332204)

[6.3.2运行概念 24](#_Toc464332205)

[6.3.3系统需求 25](#_Toc464332206)

[6.4系统设计 25](#_Toc464332207)

[6.4.1系统级设计决策 25](#_Toc464332208)

[6.4.2系统体系结构设计 26](#_Toc464332209)

[6.5软件需求分析 26](#_Toc464332210)

[6.6软件设计 27](#_Toc464332211)

[6.6.1 CSCI级设计决策 27](#_Toc464332212)

[6.6.2 CSCI体系结构设计 28](#_Toc464332213)

[6.6.3 CSCI详细设计 29](#_Toc464332214)

[6.7软件实现和配置项测试 29](#_Toc464332215)

[6.7.1软件实现 29](#_Toc464332216)

[6.7.2配置项测试准备 30](#_Toc464332217)

[6.7.3配置项测试执行 30](#_Toc464332218)

[6.7.4修改和再测试 31](#_Toc464332219)

[6.7.5配置项测试结果分析与记录 31](#_Toc464332220)

[文档审查：提前制作好文档审查单，并以此做参照。对不和审查单的地方进行统计和修正（包括代码）。 31](#_Toc464332221)

[6.8配置项集成和测试 31](#_Toc464332222)

[6.8.1配置项集成和测试准备 31](#_Toc464332223)

[6.8.2配置型集成和测试执行 31](#_Toc464332224)

[6.8.3修改和再测试 31](#_Toc464332225)

[6.8.4配置项集成和测试结果分析与记录 32](#_Toc464332226)

[6.9 CSCI合格性测试 32](#_Toc464332227)

[6.9.1 CSCI合格性测试的独立性 32](#_Toc464332228)

[6.9.2在目标计算机系统（或模拟环境）上的测试 32](#_Toc464332229)

[6.9.3 CSCI合格性测试准备 32](#_Toc464332230)

[6.9.4 CSCI合格性测试演练 32](#_Toc464332231)

[6.9.5 CSCI合格性测试执行 33](#_Toc464332232)

[6.9.6修改和再测试 33](#_Toc464332233)

[6.9.7 CSCI合格性测试结果分析与记录 33](#_Toc464332234)

[6.10 CSCI/HWCI集成和测试 33](#_Toc464332235)

[6.10.1 CSCI/HWCI集成和测试准备 33](#_Toc464332236)

[CSCI/HWCI集成和测试的测试用例（根据输入、预期输出和评价标准）、测试过程。测试用例应该覆盖系统范围设计和系统结构设计的所有方面。开发者应该将软件相关信息记录在软件开发文件中。 34](#_Toc464332237)

[6.10.2 CSCI/HWCI集成和测试执行 34](#_Toc464332238)

[6.10.3修改和再测试 34](#_Toc464332239)

[6.10.4 CSCI/HWCI集成和测试结果分析 34](#_Toc464332240)

[6.11实施详细软件开发活动的计划 34](#_Toc464332241)

[6.11.1系统合格性测试的独立性 34](#_Toc464332242)

[6.11.2在目标计算机系统（或模拟的环境）上的测试 35](#_Toc464332243)

[6.11.3系统合格性测试准备 35](#_Toc464332244)

[6.11.4系统合格性测试演练 35](#_Toc464332245)

[6.11.5系统合格性测试执行 35](#_Toc464332246)

[6.11.6修改和再测试 35](#_Toc464332247)

[6.11.7系统合格性测试结果分析与记录 36](#_Toc464332248)

[6.12软件使用准备 36](#_Toc464332249)

[6.12.1可执行软件的准备 36](#_Toc464332250)

[6.12.2用户现场的版本说明准备 36](#_Toc464332251)

[6.12.3用户手册的准备 36](#_Toc464332252)

[6.12.4在用户现场的安装 36](#_Toc464332253)

[6.13软件移交准备 36](#_Toc464332254)

[6.13.1可执行软件的准备 36](#_Toc464332255)

[6.13.2源文件准备 36](#_Toc464332256)

[6.13.3支持现场的版本说明的准备 37](#_Toc464332257)

[6.13.4 “已完成”的CSCI设计和其他的软件支持信息的准备 37](#_Toc464332258)

[6.13.5系统设计说明的更新 37](#_Toc464332259)

[6.13.6支持手册准备 37](#_Toc464332260)

[6.13.7到指定支持现场的移交 37](#_Toc464332261)

[6.13.8 非移交的产品 37](#_Toc464332262)

[6.14软件配置管理 37](#_Toc464332263)

[6.14.1配置标识 37](#_Toc464332264)

[6.14.2配置控制 38](#_Toc464332265)

[6.14.3配置状态统计 38](#_Toc464332266)

[6.14.4配置审核 39](#_Toc464332267)

[6.14.5发行管理和交付 39](#_Toc464332268)

[6.15软件产品评估 39](#_Toc464332269)

[6.15.1中间阶段的和最终的软件产品评估 39](#_Toc464332270)

[6.15.2软件产品评估记录（包括所记录的具体条目） 39](#_Toc464332271)

[6.15.3软件产品评估的独立性 40](#_Toc464332272)

[6.16软件质量保证 40](#_Toc464332273)

[6.16.1软件质量保证评估 40](#_Toc464332274)

[6.16.2软件质量保证记录、包括所记录的具体条目 40](#_Toc464332275)

[6.16.3软件质量保证的独立性 40](#_Toc464332276)

[6.17问题解决过程（更正活动） 41](#_Toc464332277)

[6.17.1问题/变更报告 41](#_Toc464332278)

[6.17.2更正活动系统 41](#_Toc464332279)

[6.18联合评审（联合技术评审和联合管理评审） 41](#_Toc464332280)

[6.18.1联合技术评审包括组建议的评审 41](#_Toc464332281)

[6.18.2联合管理评审包括组建议的评审 42](#_Toc464332282)

[6.19文档编制 42](#_Toc464332283)

[7.进度表和活动网络图 43](#_Toc464332284)

[7.1进度表 43](#_Toc464332285)

[7.2活动网络图 44](#_Toc464332286)

[8.项目组织和资源 44](#_Toc464332287)

[8.1项目组织 44](#_Toc464332288)

[8.2项目资源 44](#_Toc464332289)

[8.2.1人力资源 44](#_Toc464332290)

[8.2.2开发人员要使用的设施 44](#_Toc464332291)

[8.2.3需求资源进度表 46](#_Toc464332292)

[8.2.4其他所需的资源 46](#_Toc464332293)

[9.培训 46](#_Toc464332294)

[9.1项目的技术要求 46](#_Toc464332295)

[9.2培训计划 47](#_Toc464332296)

[10.项目估算 47](#_Toc464332297)

[10.1规模估算 47](#_Toc464332298)

[10.2工作量估算 47](#_Toc464332299)

[10.3成本估算 51](#_Toc464332300)

[11. 风险管理 51](#_Toc464332301)

[12.支持条件 52](#_Toc464332302)

[12.1计算机系统支持 52](#_Toc464332303)

[12.2需要需方承担的工作和提供的条件 52](#_Toc464332304)

[12.3需要分包商承担的工作和提供的条件 52](#_Toc464332305)

[12. 注解 52](#_Toc464332306)

# 引言

## 1.1标识

题目：北航新型多功能购物网站

## 1.2系统概述

该网站有多种功能，首先具备一般购物网站的功能。该网站的核心功能在于维护商品信息，内容包括各种商品的属性，特点和价格。每一个用户在该网站上会拥有一个账号，登录之后可以查询订购和退订商品，查询商品信息。其中订购业务流程包括在线浏览商品、购物车管理、提交及支付、收货确认、服务评价。

此外，由于该网站是针对北航师生所开设，所以用户在购买商品之后的取货地点默认为北航周围。因此该网站附带一张北航地图。当商品运到时，用户只要选择自己所在位置，该地图可以自动显示从用户位置到快递员位置的最合理路线。并且会在地图上显示取货地点。

网站的组成由以下几个子系统组成（按功能分类）：

1. 用户注册系统：用户可通过该模块注册账号，该系统保存所有账号的所有信息
2. 用户登录系统：用户输入用户名和密码，该模块判断输入者是否为用户
3. 商品查询系统：该模块保存所有商品信息，用户可查询
4. 订购系统：用户可利用该模块进行订货，该模块管理用户订货的信息
5. 退订系统：用户可利用该模块退订商品，相应退货信息及原因也会被保存
6. 服务评价系统：用户可利用该模块对所购商品进行评分和评价，该系统也会保留所有评价和评分的数据
7. 地图导航系统：该模块类似高德地图，可以为用户取货进行导航

随着时代的发展，网络距离我们的生活越来越近，通过网络来运行经营商店的技术早已实现，淘宝、京东等大型购物网站已经拥有一段时间的历史，如今是家喻户晓，与我们的生活密不可分。作为北航的学生，如果我们可以为北航的师生专门开发出一个购物网站，使得北航的师生在紧张学习和工作的同时能够很方便地购买商品，则不仅可以带来一定的商业利润，也会给予北航师生很大方便。

投资方：无

需方：北航师生

用户：北航师生

开发方：本软工小组4人

支持机构：北航软件工程基础课程负责教师

当前运行现场：无

计划运行现场：北航全体师生（含北航校内的中小学）

## 1.3文档概述

文档用于描述整个项目的开发计划，对于项目的实际开发起着指向标的作用。我们如果不能严格按照本开发计划进行本项目的开发，就有可能没有办法在规定时间之内完成本项目的开发。

文档的具体内容包括对该网站的基本功能的描述，具体交付产品的内容，开发该项目所需具体工作的概述，宏观大体计划以及详细计划、进度表和活动网络图、项目组织和资源、所需学习内容、价值估算、风险管理和支持条件等。

目前对于保密性和私密性的要求仅为该文档应仅允许北航软工基础的师生进行阅读。

## 1.4与其他计划之间的关系

本项目开发成功之后，可以与周边快递公司、相关购物平台进行合作，争取全面方便北航师生以及学院路各院校师生的日常购物。

## 1.5 术语和缩略词

无。

## 1.6基线

基线是软件工程活动从一个环节转入另外一个环节时对阶段产品或组件的标识。包括以下四部分：

需求基线：需求分析规格、功能清单： 用户登录、用户注册、忘记密码、用户须知、用户个人资料、热门商品、商品搜索、商品分类、用户购物车、订单、用户付账、用户评价、订单明细、订单完成、注销退出、地图导航等

设计基线：总体设计规格、数据库设计、详细设计规格等 

测试基线：手动测试、单元测试、集成测试、整体测试、自动化测试、撰写测试报告、修改问题、再测试等 

发布基线：用户手册、质量报告、配置报告等

# 引用文件

《计算机软件文档编制规范》

《华为开发文档》——深圳市华为技术有限公司

《软件开发计划样例》

《CSCI详细设计说明书模板》

《JSP网站开发入门教程-2012》

《购物网站毕业设计论文》——中北大学

《-购物网站毕业设计论文》——江西渝州科技职业学院

《软件工程-第8章-系统设计基础》——中国水利水电出版社

《软件可重用的问题与方法》——毛军，西北工业大学计算机科学与工程系

《网站测试流程&规范详解》

《软件工程：实践者的研究方法》——机械工业出版社

以及若干参考维基百科网页，不再一一赘述

# 交付产品

## 3.1程序

该网站源代码，分为前后端，前端部分页面采用（JSP+javascript+Jquery+html+css）模式，以及(Javascript+Jquery+html+css)模式，前端布局框架采用bootstrap框架，地图表格等开发框架采用百度echarts进行开发，与百度地图结合，关于百度地图方面参考百度地图API进行开发(已申请百度开发者账号及百度地图的连接密钥),后端采用SSH框架进行开发,SSH框架版本使用（Struts2+spring3+Hibernate4），Struts2用来管理网页的action，Spring3用于管理网页的业务逻辑（注:如果业务逻辑不是特别复杂的话，spring3也可能不用），Hibernate4用于连接数据库,数据库使用mysql数据库，数据库采用本地数据库或者使用服务器存储。

3.2文档

软件开发计划

软件需求规格说明

软件结构设计说明

概要设计说明书

详细设计说明书

软件测试说明

## 3.3服务

向用户提供的服务：

系统安装和保修、使用教程、疑问解答

## 3.4非移交产品

可行性分析报告：分析该项目开发的可行性

软件开发人员

软件问题报告

软件修改报告

## 3.5最后交付期限

2017年1月10日

# 所需工作概述

## 4.1对所要开发系统、软件的需求和约束

功能需求：该网站系统分为两大部分：用户的前台使用和管理员的后台管理。所需网页主要有首页、用户登录、用户注册、忘记密码、用户须知、用户个人资料、热门商品、商品搜索、商品分类、用户购物车、订单、用户付账、用户评价、订单明细、订单完成、注销退出地图导航。

## 4.2对项目文档编制的需求和约束

符合软件工程文档开发的基本规范，应当包括开发计划、需求规格说明书、结构设计说明书和测试说明书。

## 4.3该项目在系统生命周期中所处的地位

本文档包括可行性研究、需求分析、项目计划

## 4.4项目进度安排及资源的需求和约束

数据库设计比较复杂，我们需要有管理员的详细信息、用户信息、商品类别信息、商品详细信息、网店进货出货信息、收付款信息、客户订单信息、商品评论信息。

总体计划：

一周时间进行基础理论的学习，同时进行需求分析

一周时间进行网站页面设计，画图完成，同时对整个系统进行设计

一周时间收集素材，包括所需图片、网页模板等，同时对整个系统进行设计，十月底完成

两周之内将详细的数据库设计完毕，十一月中完成

一个月左右的时间将所需网页设计完成，十二月中完成

两周左右完成测试工作和地图，同时进行项目总结十二月底完成

## 4.5其他的需求和约束

需求：无

约束：较大的课业压力，中途可能遇到意料之外的困难

# 实施整个软件开发活动的计划

## 5.1软件开发过程

采用瀑布模型。瀑布模型（Waterfall Model） 是一个项目开发架构，开发过程是通过设计一系列阶段顺序展开的，从系统需求分析开始直到产品发布和维护，每个阶段都会产生循环[反馈](http://baike.baidu.com/view/80120.htm)，因此，如果有信息未被覆盖或者发现了问题，那么最好 “返回”上一个阶段并进行适当的修改。

我们的开发流程：

1可行性分析

2需求分析

3系统设计

4编码

5测试

6上线

7再测试

每进行到一个阶段发现问题就返回上一个阶段。

## 5.2软件开发总体计划

### 5.2.1软件开发方法

安装java、mysql数据库，下载配置tomcat7服务器

使用mysql设计数据库

使用javascript进行动态web开发

采用面向对象开发方法。

### 5.2.2软件产品标准

需求：尽可能满足京东淘宝等知名购物网站能满足的需求。

设计：尽量美观、向淘宝等网站学习

编码：待定

测试用例：覆盖我们能想到的各种可能发生的情况

测试过程：

1. 有无错别字、各种分辨率下页面有无错位
2. 各链接是否正确
3. 数据库连接是否正确
4. 填写评论、评价是否正确
5. 网站内容是否有常识性错误

测试结果：全面、系统、工整、真实

### 5.2.3可重用的软件产品

#### 5.2.3.1吸纳可重用的软件产品

搜寻范围：各种开源网站（比如github）

搜寻条件：

**①**[**软件系统**](http://baike.baidu.com/view/8343.htm)**应是模块化结构。**

模块化结构的软件层次性较好，我们可以较容易地添加一部分内容或者修改其中一部分内容而不至于过多破坏整个软件的架构和功能。

**②**[**软件系统**](http://baike.baidu.com/view/8343.htm)**应不依赖于具体的运行环境。**

依赖于特定环境的部分也可以集中在少数模块，这样将软件移到自己所用平台时可以只修改或者替换少数模块，既节省工作时间又不会破坏软件整体结构。

**③**[**软件系统**](http://baike.baidu.com/view/8343.htm)**应建立在标准的、统一的数据接口上，即软件系统在建立数据模块进行数据操作时，都要求以标准的数据模式为依据。**

这样可以只需修改部分与数据模块直接交互的部分，模块和模块之间的数据交互不会受到太大的影响。

**④**[**软件系统**](http://baike.baidu.com/view/8343.htm)**应有知识的帮助。**

这一要求不是必须的，但在[软件系统](http://baike.baidu.com/view/8343.htm" \t "http://baike.baidu.com/_blank)进行重构、扩充时，[知识库](http://baike.baidu.com/view/7976.htm)系统可以提供并学习系统组合、生成及重用方面的知识，从而提高工作效率、改进工作质量。

#### 5.2.3.2开发可重用的软件产品

准则：除上述准则外，软件功能也要与目标软件相近。

将符合上述规则中的软件中与目标需求能够对应的部分进行总结和标识。功能相近的比例和软件本身质量这两点是决定我们团队工作量和工作效果的重要关键，在各个软件中对这两点进行统计，选出最合适的，并把统计数据和所选软件的选择原因进行报告。

### 5.2.4处理关键性需求

#### 5.2.4.1安全性保证

测试的时候严格按照需求的每一个细节进行测试，保证需求实现的正确性。商品的信息会经过多次复查以保证正确、用户的购买信息和付款信息不会出现错误。

#### 5.2.4.2保密性保证

测试的数据和流程不泄露给无关人员，保证已经正确实现的需求不被无关人员进行破坏，因此对数据库的要求较高。

#### 5.2.4.3私密性保证

本网站只针对北航师生进行开放，无关人员不得入内，用户和管理员的信息不会泄露。

采用设置防火墙等方式保证数据的私密性。

### 5.2.5计算机硬件资源利用

软件大小要安排合理，无用部分及时删除，除此之外目前未发现需要考虑的相关方法。

### 5.2.6记录原理

应描述记录原理所遵循的方法，该原理在支持机构对项目作出关键决策时是有用的。应对项目的“关键决策”一词作出解释，并陈述原理记录在什么地方。描述应覆盖合同中论及它的所有条款。

### 5.2.7需方评审途径

评审方可先注册用户账号，再让非北航人注册看是否成功。注册成功之后，评审员可查看商品信息，看商品信息是否完整而准确，商品分类是否正确。评审员还可以编辑个人资料，编辑完毕之后查看资料保存是否正确。查看上述信息完毕之后，评审员可以购买若干商品，然后付账，看付账后结果是否正确。最后取货时可以现场试验地图导航信息是否正确合理。最后评审员可以对商品做出评价，看评价信息是否保存正确。

此外，评审员在评审的时候，应当注意将评审的层次划分好，对于评审员进行尽心挑选和严格培训。

# 6.实施详细软件开发活动的计划

## 6.1项目计划和监督

### 6.1.1软件开发计划

一周时间进行基础理论的学习，同时进行需求分析

一周时间进行网站页面设计，画图完成，同时对整个系统进行设计

一周时间收集素材，包括所需图片、网页模板等，同时对整个系统进行设计，十月底完成

两周之内将详细的数据库设计完毕，十一月中完成

一个月左右的时间将所需网页设计完成，十二月中完成

两周左右完成测试工作和地图，同时进行项目总结十二月底完成

### 6.1.2 CSCI测试计划

软件配置项包括：

①与合同、过程、计划和产品有关的文档和资料；

②[源代码](http://baike.baidu.com/view/60376.htm)、[目标代码](http://baike.baidu.com/view/1272000.htm)和[可执行代码](http://baike.baidu.com/view/1272001.htm)；

③相关产品，包括软件工具、库内的[可重用软件](http://baike.baidu.com/view/117771.htm" \t "http://baike.baidu.com/view/_blank)、外购软件及顾客提供的软件等。

遵循方法和原则：

1. 对于文档和资料，要反复检查看有无与实际作品相冲突的地方，有无本该说明却没有说明的地方，有无与现实客观事实明显冲突的地方，以及各种错别字、语病等等，力求所有内容完整、准确。
2. 对于各种代码，逐项检查其功能，看有没有错误，有没有与需求相背的地方。对于每项功能的检查在设计测试用例时要考虑到所有的情况。其中测试过程包括单元测试（针对软件设计的最小单元），需要保证控制结构中所有执行路径都能走一遍。此外，边界测试，错误输入的测试也是必要的。再有就是集成测试，这点我们在做CPU的时候深有体会，每一个模块实现正确并不意味着整个CPU能够正常工作。比较好的方法有自顶向下集成法，模块的集成顺序即从主控模块开始，沿着控制层次逐步向下，以深度优先搜索或广度优先搜索的方式将从属于主控模块的模块集成到结构中去。
3. 对于所用的较为流行的工具要确认其功能正确之后再使用，对于其他软件则要向上述所说那样准确测试其功能的正确性之后再投入使用。

功能测试：2天

性能测试：1天

边界测试：1天

交互界面测试：1天

### 6.1.3系统测试计划

系统测试是指是将已经确认的软件、[计算机硬件](http://baike.baidu.com/view/1024731.htm)、[外设](http://baike.baidu.com/view/1162088.htm)、网络等其他元素结合在一起，进行信息系统的各种[集成测试](http://baike.baidu.com/view/2303221.htm)和[确认测试](http://baike.baidu.com/view/106760.htm)，系统测试是针对整个产品系统进行的测试，目的是验证系统是否满足了需求规格的定义，找出与需求规格不符或与之矛盾的地方，从而提出更加完善的方案。其中包括一下几种方法

恢复测试：使用各种方式强行让系统发生故障，验证其是否能够恢复。

安全测试：测试者扮演攻击系统的黑客角色，通过外部手段获取密码等信息。

压力测试：以非正常的数量、频率或容量的方式执行系统，以这种方式破坏程序。

性能测试：测试软件在集成环境中的运行性能。

### 6.1.4软件安装计划

需要支持html5的浏览器，如chrome,IE11等等，安装是与编写同时进行的。

### 6.1.5软件移交计划

### 6.1.6跟踪和更新计划，包括评审管理和时间间隔

严格按照计划进行，如果意识到某些任务可以提前完成就及时更新计划。从而做好计划活动的跟踪。

如果某些任务出现问题，需要时间更长就及时更改计划，及时修复，从而做好缺陷跟踪。

## 6.2建立软件开发环境

### 6.2.1软件工程环境

系统：windows

软件：Java开发工具包JDK7

Tomcat7

Dreamweaver2008

MySQL5.5、Navicat

STS3.0

### 6.2.2软件测试环境

测试环境是指测试运行其上的软件和硬件环境的描述，以及任何其它与被测软件交互的软件，包括驱动和桩。测试环境=软件+硬件+网络+数据准备+测试工具

除上述工具之外还有我们设计的数据库、测试数据。工具不需要再多准备，数据库的设计需要详细、完整和准确，具体细节应多参考淘宝等网站。对硬件、网络目前还没发现需要特别注意的地方，测试数据和方法则在之前的测试计划中已经有较详细的描述，这里不再赘述。

### 6.2.3软件开发库

软件开发的过程是[程序员](http://baike.baidu.com/view/39175.htm)们开发出各种功能的函数（零件）然后加以整合的过程，而程序员们开发出来的各种功能的函数（零件）的集合便称之为软件开发库。

库中的各种函数需要成员们共同进行维护，我们认为在开发每一个函数之前首先应该将大问题化小，分为若干模块，再将每一个模块的功能明确并细分，确认需要哪些函数。每个组员不一定要清楚所有人要写的所有函数的具体实现细节，但对于整体的架构一定要充分熟悉。做好上述工作之后，每个组员在开发自己所要开发的函数之前应该明确该函数的规格，然后在写完函数后对其进行正确性证明。

### 6.2.4软件开发文档

[软件开发文档](http://baike.baidu.com/view/4546879.htm)是软件[开发](http://baike.baidu.com/view/522596.htm)使用和维护过程中的必备资料。它能提高软件开发的效率，保证[软件](http://baike.baidu.com/subview/37/6030295.htm)的质量，而且在软件的使用过程中有指导，帮助，解惑的作用，尤其在维护[工作](http://baike.baidu.com/subview/10614/11101789.htm)中，文档是不可或缺的资料。开发文档中包括需求分析、技术分析、系统分析、数据库设计、功能函数设计等。这些部分紧密相关，互相依赖，利润，如果数据库的设计和需求分析不符，技术分析出现错误，就会出现很多问题，既不利于文档自身的开发和维护，也不利于软件的开发和维护。

因此，文档的先后顺序非常重要，各文档内容的准确性，特别是技术分析文档，更要绝对保证。需求分析是首要的，其次是技术分析，我们认为这个顺序很重要。

### 6.2.5非移交软件

非移交产品是指需要交给公司但不必向用户提交的产品。

## 6.3系统需求分析

### 6.3.1用户输入分析

首先仔细分析用户需求，系统地总结出所有需求。开发者站在用户的角度，思考用户可能进行的输入与操作。具体到我们的作品可以多在淘宝等网站中进行体会。目前我们想到的用户可能有的输入有用户名、密码、个人资料、想要查询的商品、订单下达、退订信息及原因、用户付账、用户评分、用户评价、导航目的地。

### 6.3.2运行概念

指软件运行时的过程，需要明确以下三点。

1. 本系统的初始化过程
2. 运行控制——对系统施加不同的外界运行控制时所引起的各种不同的运行组件组合、每种运行所经历的内部组件和支持软件、每一种外界运行控制的方式方法和操作步骤、每种运行组件组合将占用各种资源的情况以及系统运行时的安全控制
3. 本系统运行的结束过程

本作品为一个网站，在服务器上运行。其他细节还有待学习。

### 6.3.3系统需求

1.服务器：Pentium 3以上或更高，内存4G以上； 硬盘至少80G以上；

2.网络适配器：10M/100M自适应，一台UPS；

3.工作站：Pentium 4以上微机；内存1G；硬盘至少80G以上；网络适配器10M/100M自适应；

## 6.4系统设计

### 6.4.1系统级设计决策

我们的作品由于是购物网站，所以需要一定数量的顾客，因此我们会设计一个用户注册和登录的前端，以及用户资料填写的页面，这些内容会和后台管理用户信息的数据库进行交互。

此外，该网站还要和商品介绍的网站相链接，每一个商品介绍的页面都有和保存自己信息的数据库的交互。

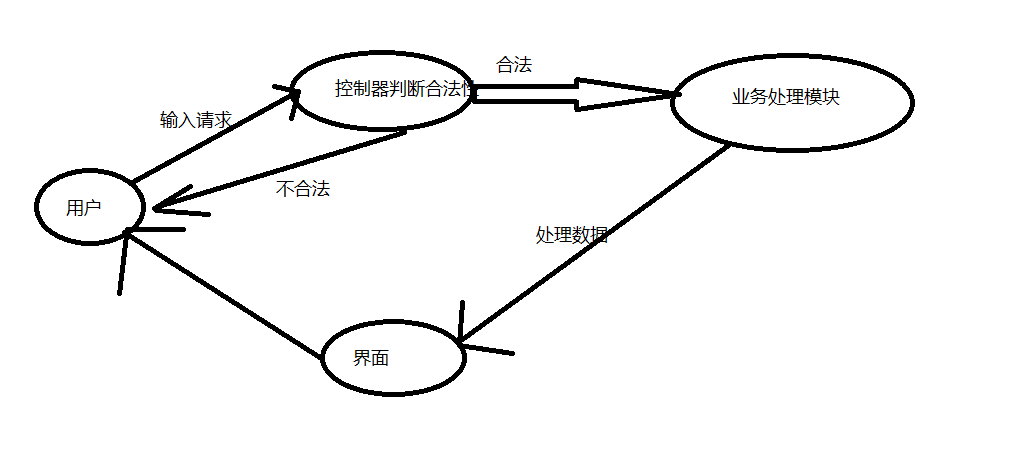
商品页面要和用户订购和退订页面相链接，订购页面和退订页面分别和保存订购和退订信息的数据库相交互。同时这两个数据库和网站管理网站收入支出信息的数据库也要交互。

订货页面有用户评价和评分的功能，这两个功能的实现意味着我们还需要设计用户评价，用户评分这两个数据库，用以和网站上该功能进行交互。

最后，地图导航模块算是独立性较大的一个模块，包含北航所有重要位置以及从每个位置到达其他任意位置的最短路线。

### 6.4.2系统体系结构设计

我们的作品采取B/S（Browser/Server）结构，即浏览器服务器结构。在这种结构下，用户工作界面通过浏览器来实现，极少部分[事务](http://baike.baidu.com/view/121511.htm)逻辑在前端（Browser）实现，主要事务逻辑在服务器端（Server）实现，形成所谓[三层结构](http://baike.baidu.com/view/280960.htm" \t "http://baike.baidu.com/_blank)。



## 6.5软件需求分析

软件[需求分析](http://baike.baidu.com/view/111493.htm)就是把软件计划期间建立的[软件可行性分析](http://baike.baidu.com/view/671807.htm)求精和细化，分析各种可能的解法，并且分配给各个软件元素。需求分析是软件定义阶段中的最后一步，是确定系统必须完成哪些工作，也就是对目标系统提出完整、准确、清晰、具体的要求。

进行需求分析时，应注意一切信息与需求都是站在用户的角度上。尽量避免分析员的主观想象，并尽量将分析进度提交给用户。在不进行直接指导的前提下，让用户进行检查与评价。从而达到需求分析的准确性。

分析员通过需求分析，逐步细化对软件的要求，描述软件要处理的数据域，并给软件开发提供一种可转化为数据设计、结构设计和过程设计的数据和功能表示。在软件完成后，制定的软件规格说明还要为评价软件质量提供依据。

对于我们的作品来说，由于目前并没有明确的客户，所以我们要做的，就是把老师的要求加上我们自己的设计进行分析、整合，把每一个需求的大致解法搞清楚。

## 6.6软件设计

### 6.6.1 CSCI级设计决策

软件配置项包括：

①与合同、过程、计划和产品有关的文档和资料；

②源代码、目标代码和可执行代码；

③相关产品，包括软件工具、库内的可重用软件、外购软件及顾客提供的软件等

我们认为大致相当于软件级设计决策，指软件行为的设计决策（忽略其内部实现，从用户角度出发，描述软件将怎样运转以满足需求）和其他影响组成该软件的软件配置项的选择与设计的决策。

大致内容：有关软件接收的输入和产生的输出的设计决策；对每个输入或条件进行响应的软件行为的设计决策；有关数据库/数据文件如何呈现给用户的设计决策；为满足安全性、保密性和私密性需求所选用的方法；为响应需求而作出的其他软件级设计决策。

在我们的网站中，我们认为输入相当于用户注册、登录、购买、评价等操作，对于这些操作软件给出怎样的行为，我们应当给出相应的合适的设计决策。网站用户和管理员资料的安全性和私密性，网站交易数据的安全性和私密性也应当考虑。此外、保密和私密性约束，安全性约束，语言约束前面已经提到过了。

### 6.6.2 CSCI体系结构设计

软件体系结构描述着本软件的基本特征，至关重要。在体系结构设计开始的时候，待开发的软件必须放在所处的环境中，与软件交互的外部实体和交互的特性也必须清楚。这些信息一般来自需求分析，一旦软件环境模型得到确定，体系结构原型集（类似于类）也就确定了。接下来只是对这些“类”的具体实现。总的来说分为以下五点：

程序结构设计

全局数据结构设计

软件配置项设计

动态交互设计

接口设计

我们的作品也基本应当按照上述所说进行软件体系结构设计，就不再多说了。

### 6.6.3 CSCI详细设计

详细设计是软件工程中软件开发的一个步骤，就是对概要设计的一个细化，就是详细设计每个模块实现算法，所需的局部结构。在详细设计阶段，主要是通过需求分析的结果，设计出满足用户需求的嵌入式系统产品。开发者应该开发和记录每个软件单元的设计描述。结果应该包括《软件设计说明书》模板的所有项目。

传统软件开发方法的详细设计主要是用结构化程序设计法。详细设计的表示工具有图形工具和语言工具。图形工具有业务流图、程序流程图、PAD图（Problem Analysis Diagram）、NS流程图（由 Nassi和 Shneidermen开发，简称 NS）。语言工具有伪码和PDL（Program Design Language）等。

## 6.7软件实现和配置项测试

### 6.7.1软件实现

在体系结构设计和详细设计已经完成的情况下进行软件的具体实现，我们认为应当严格遵守软件每一个模块，每一个函数的规格进行实现，这样如果出错，应该是规格的问题。因此规格的正确性至关重要。

### 6.7.2配置项测试准备

配置项的概念：配置项是相对于配置而言的，指纳入配置管理的产品集合，包括文档和程序，以及其它配件项。

准备：测试人员、文档、用户需求、测试数据、需求规格等等

### 6.7.3配置项测试执行

测试类型，通常包括很多种，例如文档审查，功能测试，安全性测试等等。

1. 文档审查：﻿着重查看需求文档中对功能和性能的描述，是遵循软件任务书的要求，需求的描述是否明确，是否有二义性和模棱两可的地方，是否与软件配置项的功能实现有分歧。﻿查看用户使手册的是否支持用户对软件的安装和使用，功能描述和使用步骤是否与软件实现保持一致，描述是否合理等。﻿﻿
2. 功能测试：以需求规格说明为依据，测试软件的功能是否满足需求规格说明的要求，所以，功能测试的前提是需求规格说明完全正确。
3. 安全性测试：有关验证应用程序的安全服务和识别潜在安全性缺陷的过程。此节包括一些重要的测试建议，用来验证已创建安全的应用程序。由于攻击者没有闯入的标准方法，因而也没有实施安全性测试的标准方法。

### 6.7.4修改和再测试

对发现的问题应当分析其原因，对错误和不合理之处进行修改。由于文档和代码是紧密相联的，因此对于每一个问题要检查文档和代码是否都已经进行了修改，然后再按之前所说重新测试。

### 6.7.5配置项测试结果分析与记录

## 文档审查：提前制作好文档审查单，并以此做参照。对不和审查单的地方进行统计和修正（包括代码）。

功能测试：对不合要求的结果进行记录，并分析原因。

安全性测试：对不安全的情况进行统计和制定解决方案。

## 6.8配置项集成和测试

### 6.8.1配置项集成和测试准备

配置项测试结果、待测试系统、系统测试预期效果、测试数据、测试人员、测试步骤，开发者应该将软件相关信息记录在软件开发文件中。

### 6.8.2配置型集成和测试执行

按照测试步骤对整个系统进行测试，逐项与预期效果进行对比。

### 6.8.3修改和再测试

对于发现的问题，首先检查整个配置项系统的交互部位，如果没问题，则是配置项测试出了问题，回到上一步重新检查，改正错误。更新相应的软件开发文件和其他软件产品。

### 6.8.4配置项集成和测试结果分析与记录

分析测试结果，将其写入相应的开发文件中。

## 6.9 CSCI合格性测试

### 6.9.1 CSCI合格性测试的独立性

负责CSCI合格性测试的人不应该是进行详细设计或软件实现的人。这并不排除负责详细设计或实现的人对这个过程作出贡献。

### 6.9.2在目标计算机系统（或模拟环境）上的测试

开发者的CSCI合格性测试应该包括在目标计算机（或其它用户同意的系统）上的测试。

### 6.9.3 CSCI合格性测试准备

测试用例、测试过程以及测试用例和系统需求之间的跟踪性。

### 6.9.4 CSCI合格性测试演练

如果CSCI合格性测试需要用户见证，开发者应该参加（自己动手）运行CSCI合格性测试用例和过程以保证其完整性和正确性。开发者应该将这些测试活动的结果记录在相应的软件开发文件中并根据需要对测试用例和过程进行更新。

### 6.9.5 CSCI合格性测试执行

测试应该根据测试用例和过程进行。

### 6.9.6修改和再测试

根据CSCI合格性测试的结果，开发者应该对软件做必要的修正，给用户提供重新测试的建议，参加所有需要的重新测试并更新软件开发文件和其他软件产品。

### 6.9.7 CSCI合格性测试结果分析与记录

开发者应该参加分析和记录CSCI合格性测试结果。

## 6.10 CSCI/HWCI集成和测试

### 6.10.1 CSCI/HWCI集成和测试准备

HWCI是指硬件配置项。

### CSCI/HWCI集成和测试的测试用例（根据输入、预期输出和评价标准）、测试过程。测试用例应该覆盖系统范围设计和系统结构设计的所有方面。开发者应该将软件相关信息记录在软件开发文件中。

### 6.10.2 CSCI/HWCI集成和测试执行

按照CSCI/HWCI集成测试用例和测试过程进行。

### 6.10.3修改和再测试

根据CSCI/HWCI集成和测试的结果，开发者应该相应的修正，参加所有需要的重新测试，更新相应的软件开发文件和其他软件产品。

### 6.10.4 CSCI/HWCI集成和测试结果分析

分析CSCI/HWCI集成测试的结果。软件相关的分析和测试结果应该记录在相应的软件开发文件中。

## 6.11实施详细软件开发活动的计划

### 6.11.1系统合格性测试的独立性

负责系统合格性测试的人不应该是进行详细设计或软件实现的人。这并不排除负责详细设计或实现的人对这个过程作出贡献，例如：提供需要了解系统内部实现的测试用例。测试时可以使用log日志进行检测。

### 6.11.2在目标计算机系统（或模拟的环境）上的测试

开发者的系统合格性测试应该包括在目标计算机（或其它用户同意的系统）上的测试。

### 6.11.3系统合格性测试准备

测试用例、测试过程以及测试用例和系统需求之间的跟踪性。

### 6.11.4系统合格性测试演练

如果系统合格性测试需要用户见证，开发者应该参加（自己动手）运行系统合格性测试用例和过程以保证其完整性和正确性，还应该将这些测试活动的结果记录在相应的软件开发文件中并根据需要对测试用例和过程进行更新。

### 6.11.5系统合格性测试执行

测试应该根据测试用例和过程进行。

### 6.11.6修改和再测试

根据系统合格性测试的结果，开发者应该对软件做必要的修正，给用户提供重新测试的建议，参加所有需要的重新测试并更新软件开发文件和其他软件产品。

### 6.11.7系统合格性测试结果分析与记录

开发者应该参加分析和记录系统测试结果。

## 6.12软件使用准备

### 6.12.1可执行软件的准备

一台安装有chrome或IE浏览器的计算机

### 6.12.2用户现场的版本说明准备

最新版本

### 6.12.3用户手册的准备

开发小组预先准备好的用户手册

### 6.12.4在用户现场的安装

遵循一般安装规则以及用户现场的特殊情况

## 6.13软件移交准备

### 6.13.1可执行软件的准备

一台安装有chrome或IE等主流浏览器的计算机

### 6.13.2源文件准备

打包好的源程序文件，报告可安装的文件。

### 6.13.3支持现场的版本说明的准备

现场提供最新的版本

### 6.13.4 “已完成”的CSCI设计和其他的软件支持信息的准备

### 6.13.5系统设计说明的更新

在开发和维护的过程中随时进行

### 6.13.6支持手册准备

小组自行准备用户手册

### 6.13.7到指定支持现场的移交

### 6.13.8 非移交的产品

## 6.14软件配置管理

是一组管理变更的活动

### 6.14.1配置标识

配置标识是定义各类配置项、建立各种基线、描述相关软件配置及其文档的过程。

配置标识分为三个步骤：

1、将软件分组成一系列软件配置项

2、定义对配置项命名规则

3、对配置项的描述文档（功能，性能，物理特性等）

配置标识是指为了方便对软件配置的各个片段进行管理，必须对每一个配置项进行标识。其原则为：

(1)用易于理解和推测的方式定义文件的标识；

(2)当需要修改时，提供进行修改和跟踪它们的方法；

(3)为了便于控制与管理，要隐含如下内容：配置项内容、版本、完成时间等。

### 6.14.2配置控制

配置控制是对配置项的变更申请进行初始化、评估、协调、实现，包括将通过和实现的变更加入到基线中的更改控制过程。包括成本/效益分析、决定是否进行变更、实施变更、审查、检入。

### 6.14.3配置状态统计

是跟踪对软件的更改的过程，它保证对正在进行和已完成的变更进行记录、监视并通报。用以跟踪对已建立基线的需求、源代码、数据，以及相关文档的更改。

1、配置状态记录

2、编制配置状态报告：软件配置项的状态、变更申请和已批准的变更实现情况

3、配置状态发布：通知相关管理人员和软件工程师

### 6.14.4配置审核

是验证一个可发布的软件基线是否包含了它应包括的所有内容。包括功能配置审核和物理配置审核。

1、功能配置审核：确认软件已通过测试并满足基线规定的需求说明，保证正确性。

2、物理配置审核：确认将发布的软件包含了所有必需的组成部分（代码、文档、数据），保证完整性

判断变更是否正确完成，需进行正式技术审核和软件配置审核

1、正式技术审核：检查已完成修改的软件配置对象的技术正确性

2、软件配置审核：各项产品在技术上和管理上的完整性。

### 6.14.5发行管理和交付

## 6.15软件产品评估

### 6.15.1中间阶段的和最终的软件产品评估

### 6.15.2软件产品评估记录（包括所记录的具体条目）

将评估人员的评估记录完整、详细、真实地记录在开发文档中

### 6.15.3软件产品评估的独立性

评估人员与不应该来自开发人员

## 6.16软件质量保证

### 6.16.1软件质量保证评估

评估和以下几点相关：

1. 标准：IEEE,ISO及其他标准化组织制定了一系列广泛的软工标准和相关文档
2. 评审和审核：评审由软件工程师执行，目的是发全错误。审核由软工质量保证人员执行，意在确保 软工工作遵循质量标准。
3. 测试：目标为发现错误
4. 错误、缺陷收集和分析：分析错误和缺陷如何引入，如何消除

### 6.16.2软件质量保证记录、包括所记录的具体条目

标准记录、测试记录、缺陷记录、错误记录、变更记录、安全和风险记录，保证真实、详细和完整

### 6.16.3软件质量保证的独立性

软件质量保证人员不应该来自软件开发人员

## 6.17问题解决过程（更正活动）

### 6.17.1问题/变更报告

问题包括质量上的脱离、成本上的脱离和进度上的脱离

变更主要来自新的业务或市场条件、新的客户需要、企业改组或扩大\缩小规模、预算或进度的限制

对以上问题和变更必须如实记录

### 6.17.2更正活动系统

更新的关键在于对问题的分析和解决。

分析：首先对于问题进行分类，其次要从原因、发生场所、带来影响等角度对问题进行分析。找出真正的原因之后，还要进一步确认是否还会发生类似的问题。

解决：对于重要性不高的问题，只需和当事人进行协商即可。对于很重要的问题，需要有项目管理者进行决策。明确问题解决的实施者、需要的时间等。

## 6.18联合评审（联合技术评审和联合管理评审）

### 6.18.1联合技术评审包括组建议的评审

评审目标：

1. 发现软件的任何一种表示形式中的功能、逻辑或实现上的错误
2. 验证评审中的软件是否满足其需求
3. 保证软件的表示符合预先制定的标准
4. 获得以统一方式开发的软件
5. 使项目更易于管理

为了确定质量控制活动是否起作用，应该收集一组度量。

### 6.18.2联合管理评审包括组建议的评审

软件项目管理是为了使软件项目能够按照预定的成本、进度、质量顺利完成，而对人员、产品、过程和项目进行分析和管理的活动。因此评审时应该着重于以下几点：

1. 软件开发成本是否合理，实际开发成本与预期差距有多大
2. 进度安排是否合理，实际进度与预期差距有多大
3. 质量方面的计划是否合理，是否真正实现

## 6.19文档编制

文档编制方法：在各个阶段应该包含以下内容：

1.可行性与计划研究阶段

可行性研究报告、项目开发计划

2.需求分析阶段

软件需求说明书、数据要求说明书、初步的用户手册

3.设计阶段

概要设计说明书、详细设计说明书、数据库设计说明书、测试计划初稿

4.实现阶段

模块开发卷宗、用户手册完工、操作手册、测试计划终稿：

5.测试阶段

模块开发卷宗、测试分析报告、项目开发总结报告

6.运行与维护阶段

开发进度月报

# 7.进度计划

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 任务名称 | 开始时间 | 完成时间 | 负责人员 | 标志事件 | 进度 |
| 1 | 项目开发计划 | 2016.10.01 | 2016.10.10 | 张绍钧 | 开发计划制定完成并通过 | 100% |
| 2 | 软件需求分析 | 2016.10.11 | 2016.10.17 | 杨健 | 需求分析完成并获得客户肯定 | 0 |
| 3 | 网站页面设计 | 2016.10.18 | 2016.10.25 | 于建勋 | 完成前端设计 | 0 |
| 4 | 后端系统设计 | 2016.10.26 | 2016.11.1 | 李旭 | 完成后端系统设计 | 0 |
| 5 | 数据库设计 | 2016.11.2 | 2016.11.16 | 杨健 | 完成数据库设计 | 0 |
| 6 | 测试 | 2016.12.18 | 2016.12.31 | 张绍钧 | 测试完成 | 0 |

# 8.项目组织和资源

## 8.1项目组织

组长：杨健

组员：于建勋、李旭、张绍钧（开发和测试）

## 8.2项目资源

### 8.2.1人力资源

杨健：java,javascript,html5,mysql,css,ajax，SSH框架

张绍钧：Java,html5（学习中），javascript（学习中）

### 8.2.2开发人员要使用的设施

已有：Win7及Win10计算机、java、mysql数据库

需要：

开发平台:

(1)eclipse设计后端以及JSP页面（当然Myeclipse更好但需要money）

(2)HBuilder 进行（javascript+html+css+jquery）纯页面设计

开发所需要的资源及安装包

1. Struts2安装包,在eclipse中搭建环境
2. Spring3安装包，在eclipse中搭建环境
3. Hibernate 4安装包,在eclipse中搭建环境
4. Jquery 使用<script>语句引入
5. Echarts使用<script>语句引入
6. 百度地图API使用<script>语句引入
7. Tomcat7服务器下载后，经由eclipse导入
8. 使用eclipse创建dynamic项目，搭建web开发环境
9. OpencStreetMap上获取本地地图数据,网址<http://www.openstreetmap.org/>
10. 将以上用到的资源搭建在eclipse上，并利用web.xml进行规范。

(11)mysql使用命令行或者用workbench（workbench使用最新版本）进行可视化管理。

### 8.2.3需求资源进度表

目前各种资源已经搭建在组长杨建的win7电脑上了

### 8.2.4其他所需的资源

(1)获取json格式的地图信息，

网址:<http://echarts.baidu.com/>

# 9.培训

## 9.1项目的技术要求

组员均具有java编程基础。以下是需要掌握的

(1)html和html5（先学html，再学html5）

(2)css和css3（先学css，再学css3）

(3)学习javascript。

(4)学习jquery。

(5)学习ajax技术，进行前后端交互

(6)学习struts2网址：<http://struts.apache.org/>

(7)学习hibernate4网址：<http://hibernate.org/orm/>

(8)学习spring,网址

(9)学习mysql

(10)预计采用MVC编程模式

(11)学习servlet

(12)学习xml可扩展语言，进行网页的后端SSH框架设计。

(13)学习echarts。

(14)学习前端bootstrap框架

## 9.2培训计划

(1)html+html5+css+css3:2天

(2)javascript+jquery+ajax：2天

(3)SSH框架:2天

(4)echarts+bootstrap1天

(5)mysql:1天

(6)json+xml:1天

# 10.项目估算

## 10.1规模估算

总共大约有50个左右的模块，预计代码量1到2万行

## 10.2工作量估算

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 估计值（人天） |
| **1** | **开发阶段** | **101** |
| **1.1** | **学习阶段** | **16** |
| 1.1.1 | Javascript，HTML学习 | 9 |
| 1.1.2 | 收集素材 | 2 |
| 1.1.3 | 撰写软件开发计划 | 2 |
| 1.1.4 | 撰写可行性分析报告 | 2 |
| **1.2** | **分析阶段** | **9** |
| 1.2.1 | 可行性分析 | 1 |
| 1.2.2 | 需求分析 | 2 |
| 1.2.3 | 系统结构分析 | 3 |
| 1.2.4 | 数据库设计 | 3 |
| **1.3** | **代码编写阶段** | **67** |
| 1.3.1 | 撰写软件需求规格说明 | 4 |
| 1.3.2 | 设计软件架构 | 4 |
| 1.3.3 | 用户登录 | 3 |
| 1.3.4 | 用户注册 | 3 |
| 1.3.5 | 忘记密码 | 3 |
| 1.3.6 | 用户须知 | 2 |
| 1.3.7 | 个人资料 | 5 |
| 1.3.8 | 商品搜索 | 3 |
| 1.3.9 | 热门商品 | 2 |
| 1.3.10 | 商品分类 | 3 |
| 1.3.11 | 用户购物车 | 2 |
| 1.3.12 | 订单 | 3 |
| 1.3.13 | 付账 | 3 |
| 1.3.14 | 用户评价 | 4 |
| 1.3.15 | 用户评分 | 3 |
| 1.3.16 | 网站财务信息 | 6 |
| 1.3.17 | 订单明细 | 3 |
| 1.3.18 | 注销退出 | 3 |
| 1.3.19 | 地图导航 | 8 |
| 1.4 | 初步测试阶段 | 26 |
| 1.4.1 | 规定测试要点 | 3 |
| 1.4.2 | 撰写软件测试说明 | 2 |
| 1.4.3 | 开发人员手动测试 | 4 |
| 1.4.4 | 单元测试 | 4 |
| 1.4.5 | 模块集成测试 | 2 |
| 1.4.6 | 整体测试 | 2 |
| 1.4.7 | 自动化测试 | 2 |
| 1.4.8 | 撰写测试报告 | 1 |
| 1.4.9 | 修改代码以解决测试中出现的问题 | 5 |
| 1.4.10 | 再次测试修改后的代码 | 1 |
| 2 | 上线阶段 | 67 |
| 2.1 | 上线宣传阶段 | 31 |
| 2.1.1 | 网站上线运行 | 1 |
| 2.1.2 | 宣传网站吸引用户使用 | 30 |
| 2.2 | 用户测试阶段 | 34 |
| 2.2.1 | 压力测试 | 10 |
| 2.2.2 | 安全性测试 | 10 |
| 2.2.3 | 收集测试中遇到的问题 | 5 |
| 2.2.4 | 补充测试报告 | 2 |
| 2.2.5 | 修改代码解决遇到的问题 | 7 |

## 10.3成本估算

学习：无

素材寻找：200

服务器租借：借用同学的租借的服务器，暂定为免费

代码编写：3000

代码维护：2000

上线宣传：1500

其他：1000

# 风险管理

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 描述 | 发生概率 | 危害程度 | 规避措施 | 整改策略 | 编号 |
| 1 | 用户无法清晰描述需求 | 高 | 高 | 和用户充分交流，获取用户需求的准确信息 | 根据需求的改变对开发计划进行相关的调整 | 1 |
| 2 | 开发环境出现问题 | 中 | 中 | 预先选择适合的开发环境 | 对出现问题的部分及时进行修改 | 2 |
| 3 | 预算不充分 | 高 | 高 | 细化预算开支 | 及时增加预算 | 3 |
| 4 | 进度赶不上预期计划 | 高 | 高 | 进行赶工，或者修改原有计划 | 调整计划文档 | 4 |
| 5 | 开发人员技术水平不达标 | 中 | 高 | 在进行项目之前开发人员进行自学 | 技术弱的人员由团队共同辅导 | 5 |
| 6 | 软件运行故障 | 高 | 高 | 对每个功能模块都进行详细的测试记录 | 迅速排查并解决问题 | 6 |
| 7 | 中途有组员退组 | 小 | 高 | 让组员之间互相理解 | 组员交接任务 | 7 |

# 12.支持条件

## 12.1计算机系统支持

Win10系统上执行，此外还有tomcat7服务器，支持html5的chrome等浏览器。

## 12.2需要需方承担的工作和提供的条件

无

## 12.3需要分包商承担的工作和提供的条件

无

# 注解

（1）本开发文档可能在具体使用中根据实际情况有部分修改。