**天津工业大学**

**毕业设计**

**海豚湾美甲店美业管理系统设计与开发**

**姓 名 刘 鑫**

**学 院 管理学院**

**专 业 信息管理与信息系统**

**学 号 1110650127**

**指导教师 王建正**

**职 称 讲 师**

2015年6月

1. 绪 论

1.1美甲店信息化的发展现状

20世纪90年代以来，信息化以及信息技术所推动的技术革命，其必要性和重要性已经被世人所认同，不论人们的主观意愿如何，信息化必将渗透到各行各业，用以提升这些行业的生产力，其急速发展的趋势对所有国家既是严峻的挑战，又是难得的机遇。

就近几年而言，我国的信息化得到了迅猛的发展，信息化水平越来越高，加速了很多传统行业的改造，呈现了传统企业若不实施信息化就有可能被淘汰的局面。并且信息化在国民经济中的地位越来越不可忽视。

企业信息化在我国信息化领域是起步较早的部分，尤其是大中型企业。包括金融、电信、汽车等行业的信息化已经取得了阶段性成果并逐渐步入了成熟阶段。主要表现在：这些企业大都设立了信息化管理机构，企业基础网络及市话建设基本完成，财务管理系统和办公自动化建设上了新台阶。中小企业是国民经济中最具活力的组成部分，近几年来，中小企业的信息化意识也有了一定提高，但发展水平参差不齐。

2014年中国线下的美业实体店已经超过300万家，线下的市场规模达到了。1.2万亿。然而整个行业的信息化、互联网化的水平却相当低，另外服务水平也一直被诟病，消费不透明、不诚信等是制约这个行业发展的关键因素。因此，美业行业可以选择实施信息化来帮助实体店升级转型，来解决上述两个痛点。

1.2美业管理系统的开发背景

其他行业一样，能跟得上市场发展趋势的企业，才会面临更多机会。2014年是互联网思维在传统企业的应用元年，业内当今谈论最多的话题也是基于互联网环境的信息化下的用户研究，在传统企业如何落地应用。对于美业行业来说，在互联网高度发达的信息化时代，对企业实施信息化已经成为在现代商战中占领制高点的必备武器。然而国内面对美业管理的系统不仅功能参差不齐，有些甚至极为昂贵，使很多的中小型美业门店还在使用手工记账或在使用一些破解的非行业专业性的系统，致使这个行业信息化任然较为落后，生产力得不到提高，不能对店铺进行精准、高效的管理，员工需要花大量的时间做一些琐碎的，与核心业务无关的工作，增加了店铺的经营成本

1.3开发本系统的目的和意义

为了更好的帮助美业的发展，让更多美业的门店拥有简单实用的信息化管理工具，提高他们的生产力，让员工能够将更多的精力放在核心业务上，并降低店铺的运营成本，做到精准、高效的管理，是开发本系统的目的。该系统主要针对美容、美发和美甲门店处于初创阶段或者门店经营业绩暂时不是很高的门店，需要使用系统化的工具管理实现门店的收银、会员管理、会员充值、打折、员工业绩提成、店面营业报表统计等功能，以便老板能及时掌握客户的充值情况、消费情况和门店经营情况。该系统不仅可以把会员信息录入并进行信息化的管理，还可以对会员信息进行分析筛选，能够把会员的消费服务项目的数据流水作为店铺业务分析的工具，从中可以发现好的营销点和卖点，并能对不好的项目做出改善。老板能够通过该平台追踪每一笔经销存信息，对每一个员工的报酬、每个会员交易记录进行统一管理。

1.4本论文内容的主要安排

本论文主要是围绕海豚湾美甲店美业管理系统设计与开发而展开的。

第一章是绪论。阐述了我国信息化的发展现状，和我国企业信息化的发展现状，以及美业行业信息化的发展现状。然后介绍了本系统的开发背景，以及本系统开发的目的及意义，对以后的分析和开发起到了指导性作用。

第二章主要根据海豚湾美甲店的实际业务，对海豚湾美甲店美业管理系统进行了需求分析和可行性分析，以及根据该店铺的实际组织结构和业务流程绘制了系统组织结构图、业务流程图和数据流程图等。

第三章是在第二章所进行的系统分析的前提下，对要开发的系统做出了体系结构设计、代码设计、数据库设计以及输入输出设计。

第四章进入了系统开发的实施阶段。主要介绍了系统所用到的开发工具以及数据库的连接方式和过程。

第五章主要是系统的测试和维护。主要介绍了系统测试所用到的方法以及系统维护的内容和所需要的一些管理工作。

1. 海豚湾美甲店美业管理系统的分析

2.1需求分析

2.2可行性分析

可行性分析的目的是用最小的代价在尽可能短的时间内确定问题是否能够解决。其实质上是要进行一次大大压缩和简化系统的分析和设计过程，也就是在较高层次上以比较抽象的方式进行的系统分析和设计的过程。我所设计的海豚湾美甲店美业管理系统是一个规模不太大的系统，但是目标明确，清晰的描述了对目标系统的一切限制和约束。

2.2．1技术可行性分析

本系统是一个典型的B/S架构的系统。后端部分所采用的是NodeJS作为服务器端技术，MySQL作为数据库。前端部分用到了HTML、CSS和JavaScript等Web前端技术。

NodeJS是服务器端的JavaScript运行环境，它具有无阻塞（non-blocking）和事件驱动（event-driven）等特色，NodeJS采用V8引擎，实现了类似Apache和Nginx的Web服务，让开发者可以通过它来搭建基于JavaScript的Web应用。NodeJS有着强大的生态系统，所有的主流关系型数据库（MySQL、MariaDB、Oracle、SQL Server）的接口都有，不仅有NPM(Node Package Manager)作为NodeJS的包管理和分发工具，还有一个精简、灵活的Web框架——Express，为构建单页、多页及混合的Web程序提供了一系列健壮的功能特性。

NodeJS技术的特点：

1、异步I/O。程序在执行过程中必然要进行很多I/O操作，如读写文件、输入与输出、请求与相应等。由于I/O操作通常都很费时，在传统的编程模式中，要读一个几G的文件，需要整个线程都暂停下来，等待文件读完后继续执行。换言之，I/O操作阻塞了代码的执行，极大的降低了程序效率。

2、事件循环与回调函数。事件循环是指NodeJS会把所有的异步操作使用事件机制来解决，有个线程在不断地循环检测事件队列。事件循环会检查事件队列中有没有未处理的事件，直到程序结束。事件的编程方式具有轻量级、松耦合、只关注事务等优点，并可通过NodeJS的async模块，使多个异步的任务同步执行。在NodeJS中，回调函数无处不在，回调函数是最好的接受异步调用返回数据的方式。

3、单线程。NodeJS保存了JavaScript在浏览器中单线程的特点。单线程的最大好处就是可以不像多线程那样处处在意状态的同步问题，没有死锁的存在，也没有线程上下文切换的开销。

4、跨平台。NodeJS是跨平台的，即同一套代码可以部署运行在Windows、Linux和OS X等平台。

MySQL是由Oracle公司运营的一款关系型数据库管理系统。它具有如下一些特点 ：

1. 支持多种字段类型。
2. 使用的核心线程是完全多线程，即支持多处理器。
3. MySQL可工作不同的平台上。支持Java、Python、PHP和NodeJS等。
4. 全面支持SQL的GROUP BY和ORDER BY子句。
5. 可以在同一查询中混用来自不同数据库的表。

综上所述，开发本系统所采用的技术都已比较成熟，可以完成系统的开发。

2.3系统组织结构图

根据海豚湾美甲店的日常运营及业务类型，可画出如图2-1所示的店铺的组织结构图。美甲店有一位店长，他主要负责店铺的运营工作，包括招聘美甲师，给美甲师发工资，顾客的收银、结算和会员卡的办理，对店铺的经营流水进行分析包括财务分析和业务分析，以及对店铺进行战略上的的规划等等。美甲师主要是为顾客提供服务，辅助店长对店铺的服务进行分析和优化。顾客主要是到店铺做服务的人，分为会员和散客，会员会拥有该店铺的会员卡，在收银结算的时候可以享受打折或减免的优惠，在会员生日的时候也可以享受相应地生日优惠活动，而散客什么也没有，只能按服务的原价支付。

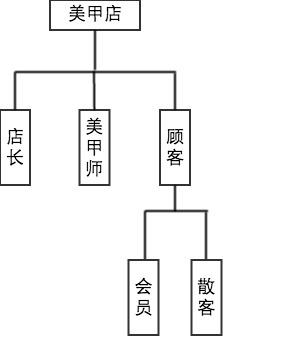


图 2-1海豚湾美甲店组织结构图

2.4系统业务流程图

业务流程分析可以帮助了解某项业务的具体处理过程，发现和处理系统调查工作中的错误和瑕疵，修改和删除原系统不合理的部分，在新系统的基础上优化处理流程。业务流程分析是对业务功能分析的进一步细化，从而得到业务流程图即TFD （Transaction Flow Diagram ）,业务流程图用一些尽可能少的规定的符号及连线来表示某个具体业务处理过程，业务流程图易于阅读和理解，是分析业务流程的重要步骤。

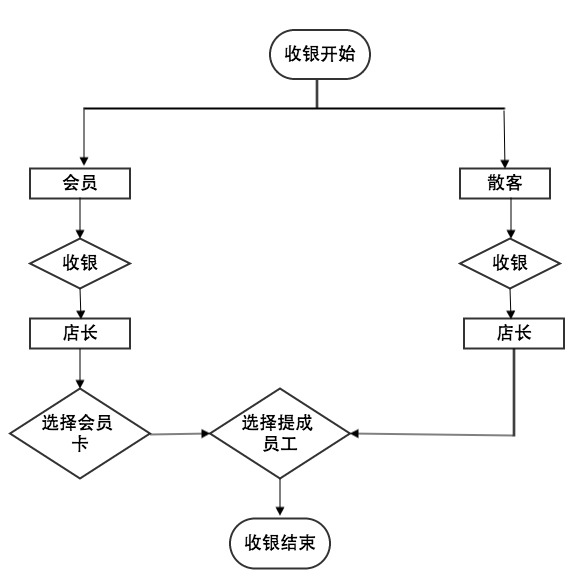


图 2-2店铺收银业务流程图

如图2-2所示为店铺的收银业务流程图，在店铺的收银业务流程中，如果结算的对象是会员则在结算的过程中可以选择该会员所拥有的会员卡给予相应地打折或减免，顾客再支付减免后的金额，然后再选择要提成的员工，就完成了该位顾客的收银。如果结算的顾客不是会员即散客，则在收银的过程中与会员收银的流程相比没有了选择会员卡的那一步，其余流程与会员收银的流程一样。

2.5系统数据流程图

数据流程图的主要用来反映功能需求和数据需求之间的联系，清晰的表达各模块之间数据的处理及加工，为之后的系统设计打下基础。

2.6数据字典

1. 海豚湾美甲店美业管理系统的设计

系统设计根据系统分析阶段所确定的新系统的逻辑模型、功能要求，在用户提供的环境条件下，设计出一个能在计算机网络环境上实施的方案，即建立新系统的物理模型。

3.1系统的体系结构

3.2系统的代码设计

3.3系统的数据库设计

数据库设计主要是进行数据库的逻辑设计，即将概念数据模型转换成特定的数据库管理系统（DBMS）支持的数据模型。

表3-1店铺会员信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项 | 含义说明 | 数据类型 | 长度 | 允许空 |
| id | 会员编号 | varchar | 64 | 否 |
| username | 用户名 | varchar | 64 | 否 |
| mobilephone | 手机号 | varchar | 64 | 否 |
| join\_date | 入会时间 | double | 16 | 否 |
| birthday | 会员生日 | double | 16 | 是 |
| enterprise\_id | 店铺编号 | varchar | 64 | 否 |

表3-2管理员信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项 | 含义说明 | 数据类型 | 长度 | 允许空 |
| id | 店铺编号 | varchar | 64 | 否 |
| username | 管理员用户名 | varchar | 64 | 否 |
| password | 登录密码 | varchar | 64 | 否 |
| signup\_time | 注册时间 | double | 16 | 是 |

表3-3店铺所开设的会员卡信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项 | 含义说明 | 数据类型 | 长度 | 允许空 |
| id | 会员卡编号 | varchar | 64 | 否 |
| card\_name | 会员卡名称 | varchar | 32 | 否 |
| card\_type | 会员卡类型 | varchar | 16 | 否 |
| discount | 折扣量 | int | 16 | 是 |
| count | 次数 | int | 16 | 是 |
| enterprise\_id | 店铺编号 | varchar | 64 | 否 |

表3-4店铺服务分类信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项 | 含义说明 | 数据类型 | 长度 | 允许空 |
| id | 服务编号 | varchar | 64 | 否 |
| service\_name | 服务名称 | varchar | 32 | 否 |
| enterprise\_id | 店铺编号 | varchar | 64 | 否 |

表3-5店铺经营项目信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项 | 含义说明 | 数据类型 | 长度 | 允许空 |
| id | 项目编号 | varchar | 64 | 否 |
| item\_name | 项目名称 | varchar | 16 | 否 |
| item\_price | 项目价格 | float | 16 | 否 |
| service\_id | 服务编号 | varchar | 64 | 否 |
| enterprise\_id | 店铺编号 | varchar | 64 | 否 |

表3-6店铺预约信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项 | 含义说明 | 数据类型 | 长度 | 允许空 |
| id | 预约编号 | varchar | 64 | 否 |
| customer\_name | 顾客姓名 | varchar | 32 | 否 |
| customer\_phone | 顾客电话 | varchar | 32 | 否 |
| start\_time | 开始时间 | double | 16 | 否 |
| end\_time | 结束时间 | double | 16 | 否 |
| comment | 备注 | varchar | 128 | 是 |
| staff\_id | 预约员工 | varchar | 64 | 是 |
| enterprise\_id | 店铺编号 | varchar | 64 | 否 |

表3-7店铺员工信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项 | 含义说明 | 数据类型 | 长度 | 允许空 |
| id | 员工编号 | varchar | 64 | 否 |
| staff\_name | 员工姓名 | varchar | 32 | 否 |
| staff\_phone | 员工电话 | varchar | 16 | 否 |
| entry\_time | 入职时间 | double | 16 | 否 |
| enterprise\_id | 店铺编号 | varchar | 64 | 否 |

3.4系统的输入输出设计

1. 海豚湾美甲店美业管理系统的实施

在信息系统分析与设计的阶段中，开发人员为新系统设计了它的逻辑模型和物理模型。系统实施阶段的目标就是把系统设计的物理模型转换成可实际运行的新系统。系统实施阶段既是系统的成功实现的最后一步，又是取得用户对新系统信任的关键阶段。

4.1系统开发工具的选择

4.1.1MySQL数据库的选择

MySQL数据库是当前数据库领域最为流行的关系型数据库之一，它具有如下特点：

* 可以同时处理几乎不限数量的用户。
* 处理多大50,000,000以上的记录。
* 命令执行速度快，也许是现今最快的。
* 简单有效的用户特权体系。
* 可以运行在不同的平台上。
* 可以在同一查询中混用来自不同数据库的表。

4.1.2NodeJS的选择

NodeJS是一个基于Chrome JavaScript运行时建立的平台，是一个JavaScript的运行环境(runtime)。实际上它是对Google V8引擎的封装。NodeJS是开发该系统的关键技术。

NodeJS具有如下特点：

* 它是一个JavaScript运行环境。
* 依赖于Chrome V8引擎进行代码解释。
* 事件驱动。
* 非阻塞I/O。
* 轻量、可伸缩，适用于数据交互应用。
* 单线程，单进程。

JavaScript是一个事件驱动语言，NodeJS利用了这个优点，编写出可扩展性高的服务器。NodeJS采用“事件循环”(event loop)的架构,使得编写可扩展性高的服务器变得既容易又安全。NodeJS采用一系列“非阻塞”库来支持事件循环的方式。本质上就是为文件系统、数据库之类的资源提供接口。向文件系统发送一个请求时，无需等待硬盘(寻址并检索文件)，硬盘准备好的时候非阻塞接口会通知NodeJS。该模型以可扩展的方式简化了对慢资源的访问。有利于开发高性能的应用。

4.2数据库的连接

在进行系统开发时，一个很重要的步骤就是建立程序到数据库的连接，用以访问数据库。连接数据库有DSN(Data Source Name)和非DSN两种方法。应用DSN访问数据库需要配置ODBC数据源（即系统DSN），该方法的优点是比较安全；而应用非DSN访问数据库不需要配置ODBC数据源，比较方便，但不够安全。本系统采用DSN的方式访问数据库。

本系统用到了连接池来处理程序对数据库的访问和操作。创建数据库连接主要分三步进行：一是创建数据库配置文件dbConnection.js;二是创建数据库连接文件dbHelper.js；三是在需要进行数据库操作的程序中引入dbHelper.js文件（var dbHelper = require(‘dbHelper.js’)）。下面进行详细介绍。

本系统的数据库的配置文件为dbConnection.js,对数据库操作是必不可少的。它指明了数据库的地址、名称、用户名和密码，并创建了连接池pool作为导出对象。代码如下：

var mysql = require('mysql');

var options = {

host: 'localhost',

port: '3306',

database: 'graduationProject',

user: 'root',

password: ''

};

exports.pool = mysql.createPool(options);

数据库连接文件dbHelper.js用于从连接池中取出一个连接并对数据库进行操作，然后再释放连接。代码如下：

var dbConnection = require('../config/dbConnection.js');

var pool = dbConnection.pool;

exports.execSql = execSql;

function execSql(sql, option, callback){

pool.getConnection(function(err, connection){

if(err){

console.log('与MySQL数据库建立连接失败。');

callback(err);

}

else{

connection.query(sql, option, function(err, rows){

if(err){

console.log('数据操作失败');

callback(err);

}

else{

callback(null,rows);

connection.release();

}

});

}

});

}

在需要进行数据库操作的程序中，必须引入dbHelper.js文件，当程序需要执行SQL语句时，即可调用dbHelper.js中的execSql()方法，对数据库进行增，删，查，改（CRUD）操作。

4.3系统的详细设计及实现

4.3.1系统的登录和注册模块的设计及实现

在用户登录界面如图4-1所示，用户需要在登录表单中填写在注册时所填写的用户名和密码才能进入系统。该登录表单有一定的验证机制，若用户表单中有某一字段未填写就提交表单，会有相应的提示信息告知用户填写该字段如图4-2所示。当用户的用户名或密码输错后也会有相应的提示信息如图4-3、4-4所示。如果用户没有账户，则需要先注册一个账户，就可以点击“注册一个账号”跳转到注册页面如图4-5所示。注册页面的表单也有和登录页面的表单一样的不允许提交空字段的验证。



图 4-2用户登录界面



图 4-3有字段未填写的提示信息



图 4-4密码错误的提示信息



图 5-4用户名填写错误的提示信息



图 4-6用户注册页面

4.3.2系统的收银模块的设计及实现

当顾客需要买单时，店员会进入收银界面，若该顾客是会员，选择该顾客的会员号及该顾客所做的服务会出现图4-6所示的界面，该界面会显示该顾客所做各个服务的价格及所有服务的总价格。

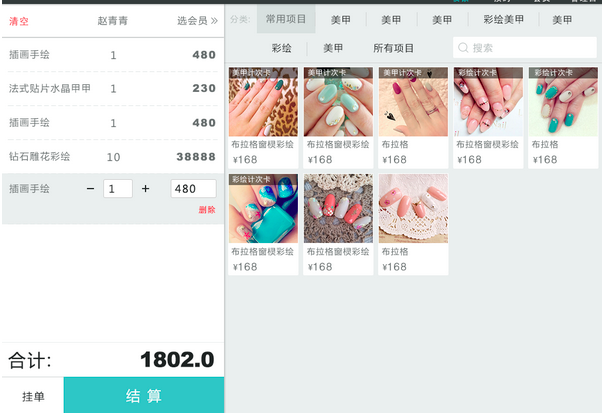


图 4-6会员收银界面

点击图4-6界面中的“结算”后，会出现图4-7所示的会员收银结算界面，在该界面中可以选择结算顾客所拥有的打折卡或其他卡券进行结算，然后再选择现金或银行卡对剩余的金额进行支付。



图 4-7会员收银的结算界面

点击图4-7界面中的“下一步”后，会出现图4-8所示的会员收银提成界面，在该界面中可以选择对该顾客做过服务的员工进行提成，点击“完成”之后即完成了本次收银。



图 4-8

散客收银的界面和设计与会员收银的界面和设计基本一样，只是没有收银最开始的会员选择过程和结算时选优惠卡的过程。

4.3.3系统的预约模块的设计及实现

如果有顾客需要提前预约某位美甲师在某个时间进行服务，则可进入会员预约界面如图4-9所示，该界面会显示那个时间那个美甲师被预约了，点击右上角的“添加”，就会呈现出入图4-10所示的添加预约的界面。



图 4-9预约界面

在添加预约界面，默认是散客进行预约，如果会员要进行预约则需要先选择会员。选择了要预约的员工及预约的时间点后，点击右上角的“完成”，即完成了本次预约。



图 4-10添加预约界面

成功预约后，预约的缩略信息就会出现在图4-9所示的预约界面的时间表中，点击该缩略信息就可以看到该预约的详细信息。

4.3.4系统的会员管理模块的设计及实现

一个美甲店铺，它会有很多会员。需要对这些会员进行高效的记录和管理，记录他们所拥有的卡券和消费明细，以便做到对店铺的精细化管理。比如能够知道什么样的会员卡最受会员欢迎，从他们的消费明细中可以知道他们最喜欢做哪些服务以及喜欢哪位美甲师为他们提供服务。

会员界面如图4-11所示，在该界面中的搜索框内，只需输入用户的手机号，即可找到该会员，可看到该会员的入会时间、姓名、生日以及所拥有的卡券等信息，点击右上角的“添加”即可添加一位会员。



图 4-11

当点击会员列表的某位会员后，就会进入该会员的详细资料界面如图4-12所示。在该界面用户不仅可以看到他所有卡券的详细信息，也可以看到他的消费明细，还可以通过点击添加卡项办理一张新的会员卡。



图 4-12

1. 海豚湾美甲店美业管理系统的测试与维护

5.1系统测试

系统测试是系统开发过程的重要组成部分，是用来确认一个系统的品质或性能是否符合开发之前所提出的一些要求。系统测试的目的，第一是要确认系统的质量，其一方面是确认系统做了我们所期望的事情（Do the right thing），另一方面是确认系统以正确的方式来做了这个事（Do it right）；第二是提供信息，比如提供给开发人员或产品经理的反馈信息，可作为风险评估的信息；第三系统测试不仅是在测试系统的本身，而且还包括系统开发的过程。如果一个系统开发完成之后出现了很多问题，这说明此系统开发过程很可能是有缺陷的。

5.1.1系统测试的原则

在进行正式测试之前，测试人员应该知道系统测试的一些基本原则。这里有一组测试原则：

1、测试应从“小规模”开始，逐步转向“大规模”。

2、穷尽测试是不可能的。

3、不充分测试是不负责任的；过分测试是一种资源的浪费，同样也是一种不负责任的表现。

4、为了达到最佳效果，应该又独立的第三方来测试。

5、所有的测试都应该追溯到用户需求。

6、应该在测试工作真正开始前的较长时间内就进行测试计划。

7、一定要注意测试中的错误集中发生的现象。

8、回归测试的关联性一定要引起充分的注意，修改一个错误而引起更多的错误是很常见的现象。

5.1.1系统测试的内容

1、验证（verification）

验证（verification）是保证系统正确的实现了一些特定功能的一系列活动，即保证系统做了我们所期望的事情。（Do the right thing）

1. 确定系统生存周期中的一个给定阶段的产品是否达到前阶段确立的需求的过程；
2. 程序正确性的形式证明。
3. 评审、审查、测试、检查、审计等各类活动，或对某些项处理、服务或文件等是否和规定的需求相一致进行判断和提出报告。

2、确认（validation）

确认（validation）是一系列活动和过程，目的是想证实在一个给定的外部环境中系统的逻辑正确性。即保证系统以正确的方式来做了这个事情。（Do it right）

1. 静态确认，不在计算机上实际执行程序，通过人工或程序分析来证明系统的正确性。
2. 动态确认，通过执行程序做分析，测试程序的动态行为，以证实系统是否存在问题。

系统测试的对象不仅仅是程序测试，系统测试应该包括整个系统开发期间各个阶段所产生的文档，如需求规格说明、概要设计文档、详细设计文档，当然系统测试的主要对象还是源程序。

5.2系统维护

系统运行后可能会暴露出一些错误、缺陷或者用户由于业务需要对系统有一些新的功能需求，这就要求系统在后续的使用中不断的得到完善和改进。因此，系统维护的主要任务就是保证系统的正常运转使系统的资源得到有效运用，并使系统的功能在后续的运行中不断的得到完善和扩充，以提高系统的工作效率和延长系统的生命周期。对系统的维护工作贯穿于系统的整个生命周期，维护工作的质量将直接影响到系统的使用效果和使用寿命。所有，系统维护是系统生存的重要保证。

5.2.1维护的内容

由于系统环境的变化或系统运行中发现了错误或问题以及随着用户的增多系统出现了性能问题，以及用户要求增加系统的某些功能等，提出了修改申请。系统维护工作包括以下三个方面：改正性维护、适应性维护、完善性维护。

改正性维护是在软件运行中发生异常或故障时进行。这些故障是由于开发过程中某些环节上的未被发现的错误造成的，在开发的末期所进行的测试也未能将其发现和修复。这些错误在某些特定情况下才能暴露出来。对已经发现的问题进行修改时，一定要十分谨慎。修改时一定要加上注释，如注明修改的日期、修改的原内容与修改后的内容以及修改人等信息。

适应性维护是要使运行的软件能适应外部环境的变动。由于计算机技术近年来发展得越来越快，摩尔定律越来越不适用，几乎6-12个月就出现一代新的系统和硬件，建立在硬件和操作系统之上的应用系统，其使用年限远远超过6-12个月，这要求应用软件能跟上发展的趋势而作出相应的修改。

完善性维护是为扩充软件的功能，提高软件的性能而进行的开发的维护工作。在系统使用一段时间之后，用户提出了新的要求，要求在现有软件基础上进行扩充开发新的功能；以及随着用户的增加，系统的并发量越来越高，需要对系统进行性能上的优化等所需的开发工作。

当前，系统维护工作还存在一定的困难，由于整个系统开发过程是由一个人完成的，难免会存在个人能力上的技术瓶颈，对后期维护经验不足，在人力物力上存在一些实际困难。这样一来，导致维护力量薄弱，常常由于考虑不周，在维护的过程中可能带来新的问题或引入新的差错。

5.2.1系统维护的管理

系统维护工作不仅是技术性的工作，还需要大量管理工作与之配合，才能保证维护工作的质量，一般维护过程如下：

1. 维护人员提出维护修改建议或要求。
2. 管理部门进行审查和分析，并对由于修改所带来的风险作充分的估计。
3. 根据审批的修改方案进行修改并作严格的测试。
4. 更新文档资料的相关部分。