**天津工业大学**

**毕业设计**

**XXX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

随着信息技术的发展以及互联网的普及

1.2美业管理系统的开发背景

1.3开发本系统的目的和意义

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。分析了我国信息化的发展现状，和我国传统行业的信息化的发展现状，以及美甲店信息化的发展现状。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. XXX管理系统的分析

2.1需求分析

2.2可行性分析

2.3系统组织结构图

2.4系统业务流程图

2.5系统数据流程图

2.6数据字典

1. XXX管理系统的设计

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

3.4系统的输入输出设计

1. XXX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系统的详细设计的实现

1. XXX管理系统的测试与维护

5.1系统测试

5.2系统维护

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软件维护的管理

软件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。