**项目名称**

**—— 项目开发文档**

**项目小组名称**

**组员：组员名称**

**修订表**

| 编号 | 生成版本 | 修订人 | 修订章节与内容 | 修订日期 |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |

**目录**

[1 系统概述 3](#_Toc298840971)

[1.1 构建目的 3](#_Toc298840972)

[1.2 小组成员及分工 3](#_Toc298840973)

[1.3 系统的部署与运行 3](#_Toc298840974)

[2 数据设计 3](#_Toc298840975)

[2.1 相关数据库表说明如下： 3](#_Toc298840976)

[3 系统功能设计 4](#_Toc298840977)

[3.1 用户账号管理 4](#_Toc298840978)

[3.1.1 用户注册 4](#_Toc298840979)

[3.1.2 用户登入 5](#_Toc298840980)

[3.1.3 用户退出 5](#_Toc298840981)

[3.1.4 找回密码 6](#_Toc298840982)

[3.2 用户信息管理 6](#_Toc298840983)

[3.2.1 用户修改资料界面 6](#_Toc298840984)

[3.2.2 用户关注对象 7](#_Toc298840985)

[3.2.3 用户好友 8](#_Toc298840986)

[3.2.4 用户粉丝 8](#_Toc298840987)

[3.2.5 用户取消关注 9](#_Toc298840988)

[3.3 微博管理 9](#_Toc298840989)

[3.3.1 微博发布 9](#_Toc298840990)

[3.3.2 微博转发 10](#_Toc298840991)

[3.3.3 微博收藏 10](#_Toc298840992)

[3.3.4 微博评论 11](#_Toc298840993)

[3.3.5 微博删除 11](#_Toc298840994)

[3.4 搜索业务 12](#_Toc298840995)

[3.4.1 搜索微博 12](#_Toc298840996)

[3.4.2 搜索关注对象 12](#_Toc298840997)

[3.4.3 搜索好友 13](#_Toc298840998)

[3.4.4 搜索收藏 13](#_Toc298840999)

# 系统概述

## 构建目的

请假系统的业务总体目标是建立一个先进、高效、安全、可靠的能被有效和应用于员工请假的信息化数据库管理系统。该系统基于B/S结构，即采用浏览器/服务器模式，服务器端由运行ASP.Net的Web应用程序及运行Microsoft SQL Server2005的数据库服务器组成，客户端可以Internet Explorer，FireFox访问系统。本信息系统体现了员工请假管理的业务逻辑行为、大大简化了相关工作的烦琐流程，同时为管理层提供有力的决策支持。

## 需要完成的软件和文档

列出需要完成的程序的名称、所用编程语言及存储程序的媒体形式。

|  |  |
| --- | --- |
| **程序名称** | **编程语言** |
|  |  |

列出项目阶段性所要提交的每种文档的名称，内容要点和存储形式。

|  |  |
| --- | --- |
| **文档名称** | **内容要点** |
|  |  |
|  |  |
|  |  |

## 项目的开发环境：

标识项目运行的所需要的软硬件环境和版本。

硬件环境：

1. CPU
2. 内存
3. 硬盘

软件环境：

1．操作系统：

2．开发工具：

3.数据库系统：

## 小组成员及分工

人员分工：

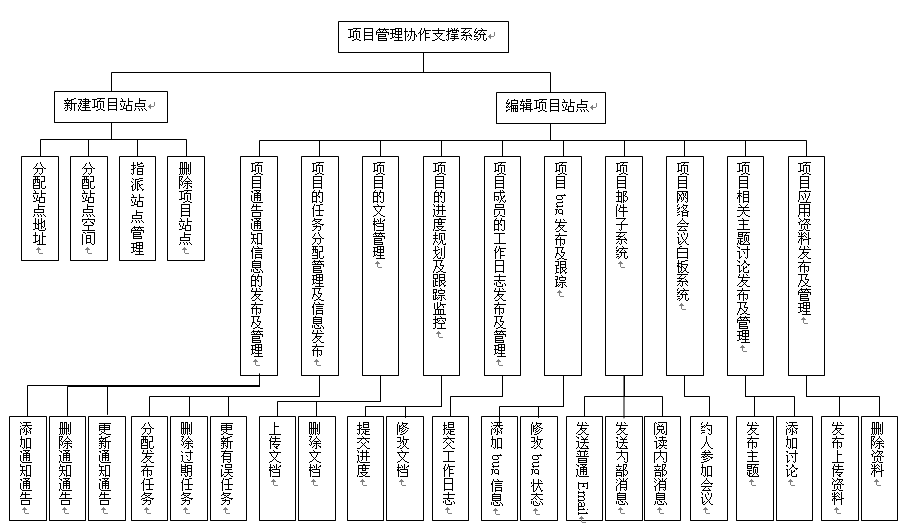
|  |  |  |
| --- | --- | --- |
| **序号** | **模块** | **姓名** |
| 1 | 用户登入功能模块 | 张俏 |
| 2 | 修改用户资料模块 | 陈飞 |
| 3 | 数据库设计 | 熊彦儒 |
| 4 | 添加关注，微博评论、转发、删除，分页 | 王文声 |
| 5 | 搜索、收藏 | 舒德华 |
| 6 | 关注、粉丝、好友信息，感兴趣的人功能模块 | 陈学航 |

任务分解和进度安排

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 任务名称 | 开始时间 | 结束时间 | 负责人 | 参与人员 | 任务成果 |
| 制定项目计划 | 2012/6/13 AM9:00 | 2012/6/13 AM12:00 | 陈飞 | 张翘， 熊彦儒，王文声，舒德华，陈学航 | 实训文档系统概述，需求分析和功能设计概要完成 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 系统需求分析

产品功能需求概要分析：



# 数据设计

## 相关数据库表说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **逻辑表名** | **物理表名** | **说明** |
| 1 | 用户信息表 | UserInfo | 记录用户的的基本信息 |
| 2 | 微博信息表 | Blog | 记录微博的基本信息 |
| 3 | 关系表 | Friends | 记录注册用户的相互关系 |
| 4 | 微博评论表 | Comment | 记录微博评论信息 |
| 5 | 微博收藏表 | Collect | 记录用户收藏的微博信息 |
| 6 | 微博转发表 | Transpond | 记录微博的转发信息 |

## 各表详细说明

用户信息表(UserInfo)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **逻辑字段** | **物理字段** | **类型（长度）** | **规则** | **字段说明** |
| 1 | 用户ID | u\_id | 自动编号 | 主键 | 自动生成的长整型编号，具有唯一值 |
| 2 | 用户名 | u\_email | 文本(45) | 非空 | 用户邮箱当用户名，非空且唯一数据 |
| 3 | 密码 | u\_password | 文本(20) | 非空 | 用户密码为非空且唯一数据 |
| 4 | 昵称信息 | u\_nickname | 文本(20) | 非空 |  |
| 5 | 性别 | u\_sex | 文本(10) | 非空 | 只能为男或女，非空 |
| 6 | 地址信息 | u\_addr | 文本(45) | 非空 |  |
| 7 | 生日信息 | u\_birth | 日期 |  |  |
| 8 | qq信息 | u\_qq | 整型 |  |  |
| 9 | msn信息 | u\_msn | 文本(20) |  |  |
| 10 | 格言、签名 | u\_info | 文本 |  |  |
| 11 | 真实姓名 | u\_name | 文本(45) |  |  |
| 12 | 照片信息 | u\_img | 文本(45) |  | 自己上传添加 |

发表微博表(weibo)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **逻辑字段** | **物理字段** | **类型（长度）** | **规则** | **字段说明** |
| 1 | 微博ID | w\_id | 自动编号 | 主键 | 自动生成的长整型编号，具有唯一值 |
| 2 | 用户ID | u\_id | 整型 |  | 用户ID为非空且唯一数据 |
| 3 | 微博发表时间 | w\_time | 日期时间 | 非空 | 发表的具体时间 |
| 4 | 微博内容 | m\_message | 文本 | 非空 |  |
| 5 | 微博种类 | category | 文本(60) | 非空 | 原创和非原创 |
| 6 | 转发数量 | z\_id | 整型 | 非空 | 自动计数 |

**……**

# 系统功能设计

整个系统说明：

整体的数据结构（包括全局变量），整体函数一个规划：.h文件的一个安排。

要实现用户对微博系统的正常使用，需要将用户的基本信息和微博的信息整合起来，并为所有已注册的用户建立一张关系网。

## 用户账号管理

用户账号管理处理用户的注册、登入与退出系统功能模块

### 用户注册

* + 1. 功能描述

用户在微博注册页面成功注册一个微博账号，账号注册通过邮箱注册。如果注册信息有误，系统会自动提示用户输入正确的信息。

* + 1. 功能实现
  + 用户填写注册信息。
  + 信息无误，点击提交，完成注册，将数据写到用户数据库中。
  + 进入新用户首页。

具体程序实现如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **函数名称** | **功能说明** | **备注** |
|  |  |  | 说明一下函数的参数，返回值，涉及到的全局变量。函数包含的.h的文件。 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

主要的界面图：

主要函数流程图：

### 用户登入

* + 1. 功能描述

已注册的用户可以根据注册邮箱和密码登入系统，如果信息有误，系统提示用户重新输入。登入之前用户可以根据需要选择是否下次自动登入。

* + 1. 功能实现
  + 用户填写登入信息。
  + 信息无误，点击提交，完成登入。
  + 进入用户首页。

具体程序实现如下：

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

### 用户退出

* + 1. 功能描述

登入成功的用户可在使用完系统后安全退出，浏览器会将保存的cookie删除，保证用户信息的安全。

* + 1. 功能实现
  + 用户点击退出按钮。
  + 退出系统，删除cookie。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.entity.UserInfo | 封装用户信息的实体 |
| C | com.codecore.servlet.RemoveServlet | 用户退出的servlet，方法setCookie用于置空cookie。 |

### 找回密码

* + 1. 功能描述

用户在登入时，如果忘记密码，可以在找回密码页面输入信息，系统会以邮件的形式将用户的密码发送到用户邮箱。

* + 1. 功能实现
  + 用户填写信息。
  + 信息无误，点击提交，将密码发到用户邮箱。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.entity.UserInfo | 封装用户信息的实体 |
| M | com.codecore.dao.UserDao | 方法findPassword用于查找信息匹配的密码 |
| C | com.codecore.servlet.FindPassword | 用户找回密码的servlet，给用户发邮件 |
| V | findpassword.jsp | 找回密码界面 |

## 用户信息管理

用户信息用于标示用户的特征，提供用户修改基本信息的接口。

### 用户修改资料界面

* + 1. 功能描述

资料修改界面方便用户对自己的个人信息进行改动，包括基本的用户信息、密码、头像修改。

* + 1. 功能实现

修改资料需要将当前用户的信息显示出来，在用户对信息进行修改提交后，更新数据库信息。用户更改密码需要进行前后验证和长度匹配，上传的头像以路径的形式存储在数据库中。

* + 用户信息通过请求servlet来获取，servlet调用DAO方法，返回一个UserInfo实体对象，包括所有的用户信息。
  + 将UserInfo转发到JSP页面，并相应的显示出来。
  + 根据用户的提交申请，及时更新数据库。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.entity.UserInfo | 封装用户信息的实体 |
| M | com.codecore.dao.UserDao | 获取用户信息的DAO类，通过getUserInfoById获取用户信息 |
| C | com.codecore.servlet.UpdateimgServlet  com.codecore.servlet.UpdatepasswordServlet  com.codecore.servlet.UpdateuserInfoServlet | 获取用户信息并进行相应更新的的三个servlet |
| V | mypassword.jsp  userinfo.jsp  myface.jsp | 显示用户信息 |

### 用户关注对象

* + 1. 功能描述

用户可以添加任何一位未关注的对象为关注

* + 1. 功能实现
  + 当前用户选择另一未关注的对象。
  + 用户点击+关注按钮触发servlet，servlet更改用户关系表

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.entity.Friends | 封装用户关系的实体 |
| M | com.codecore.dao.AttentionDao | 添加关注的DAO，方法addAttention实现添加关注 |
| C | com.codecore.servlet.AddAttentionServlet | 添加关注的servlet |

### 用户好友

* + 1. 功能描述

双方关注的用户即可成为好友，在数据库中状态用2表示。

* + 1. 功能实现
  + 用户选择另一用户为关注对象
  + 若被关注的用户以关注当前用户，则两个用户成为好友关系，直到其中一人取消关注。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.entity.Friends | 封装用户关系的实体 |
| M | com.codecore.dao.AttentionDao | 添加关注的DAO，方法isAttention判断是否已添加关注 |

### 用户粉丝

* + 1. 功能描述

当其他用户添加当前用户为关注时，这个用户就成为当前用户的粉丝，在数据库中状态用1表示。

* + 1. 功能实现
  + 通过servlet统计用户粉丝数量。
  + 用户的粉丝数量在每页显示。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.entity.Friends | 封装用户关系的实体 |
| M | com.codecore.dao.FansDao | 获取粉丝信息的DAO，方法getFansByUid查找粉丝信息 |
| V | friend.jsp | 显示粉丝信息 |

### 用户取消关注

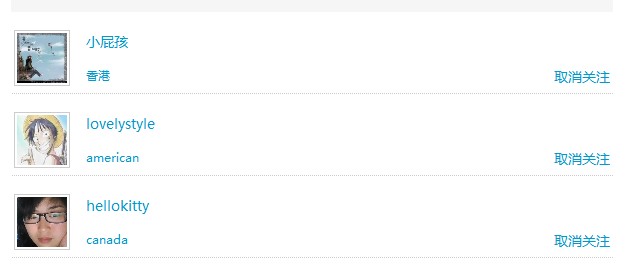
* + 1. 功能描述

用户可以取消对已关注对象的关注状态。对象包括关注和好友。

* + 1. 功能实现
  + 当前用户选择已关注的对象。
  + 用户点击取消按钮触发servlet，servlet更改用户关系表

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.entity.Friends | 封装用户关系的实体 |
| M | com.codecore.dao.CancelFollowingDao | 取消关注的DAO，方法cancelFollowing实现取消关注 |
| C | com.codecore.servlet.CancelFollowingServlet | 取消关注的servlet |



## 微博管理

微博管理模块主要实现用户发布新微博，转发微博，收藏微博，评论微博功能，以及当前用户对自己的微博进行管理，比如删除。

### 微博发布

* + 1. 功能描述

用户在登入账户后，可以发若干条微博，微博内容可以由文字和图片组成，用户成功发布之后，将在首页显示微博信息。

* + 1. 功能实现
  + 用户从输入框输入微博，上传图片。
  + 将获取的数据插入到数据库中，并通过servlet将最新微博显示出来

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.BlogPublish | 用来实现微博的发布，blogpublish向数据库写入数据 |
| M | com.codecore.entity.Blog | 表示微博信息的实体类 |
| C | com.codecore.servlet.BlogPublishServlet | 发布微博的servlet，并将最新信息转发 |
| V | home.jsp | 显示发布微博的界面 |

### 微博转发

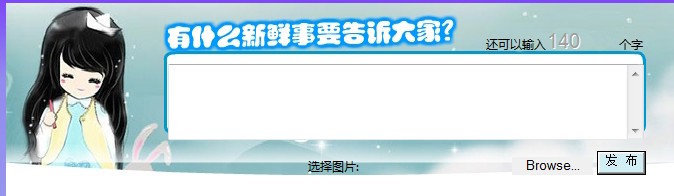
* + 1. 功能描述

用户可以转发首页的任意微博，被转发的微博转发次数加1。

* + 1. 功能实现
  + 选择一条微博作为转发对象。
  + 将微博转发到自己的微博列。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.BlogPublish | 转发微博的方法dispatchBlog |
| M | com.codecore.entity.Blog | 封装微博的实体类 |
| C | com.codecore.servlet.DispatchBlogServlet | 转发微博的servlet |
| V | home.jsp | 显示微博界面 |



### 微博收藏

* + 1. 功能描述

用户收藏任意可收藏的微博。

* + 1. 功能实现
  + 用户选择一条微博作为收藏对象
  + 用户收藏微博，并将数据插入到数据库

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.CollectDao | 添加收藏的方法addcollect |
| M | com.codecore.entity.Blog | 封装微博信息的实体类 |
| C | com.codecore.servlet.AddCollectServlet | 添加收藏的servlet |
| C | com.codecore.servlet.ShowCollectServlet | 显示收藏的servlet |
| V | collect.jsp | 显示收藏的界面 |

### 微博评论

* + 1. 功能描述

用户在选择一条微博后可以对该微博进行评论，该微博所有的评论信息会显示在评论框下方

* + 1. 功能实现
  + 选择一条微博作为评论对象。
  + 用户评论该微博，并将数据插入数据库。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.CommentDao | 添加微博评论的方法postComment |
| M | com.codecore.entity.Comment | 评论信息实体类 |
| C | com.codecore.servlet.DispatchBlogServlet | 评论微博的servlet，并将评论信息转发到前台 |
| V | home.jsp | 显示评论信息界面 |

****

### 微博删除

* + 1. 功能描述

用户选择一条自己微博列的微博，可对该微博进行删除操作。

* + 1. 功能实现
  + 用户选择一条微博作为删除对象。
  + 用户删除该条微博，从数据库中删除记录。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.AttentionDao | 删除微博的方法deleteBlog |
| M | com.codecore.entity.Blog | 微博信息实体类 |
| C | com.codecore.servlet.DeleteBlogServlet | 删除微博的servlet |
| V |  | 显示拥护微博界面 |

## 搜索业务

### 搜索微博

* + 1. 功能描述

用户在登入账户后，可对首页所有的微博按关键字进行搜索，搜索结果返回在搜索结果集页中显示。

* + 1. 功能实现
  + 用户在搜索框中输入关键字。
  + 搜索请求servlet，servlet将搜索结果返回在搜索页中。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.SearchDao | 用来实现微博的搜索，searchHome是实现搜索首页的方法。 |
| M | com.codecore.entity.Blog | 表示微博信息的实体类 |
| C | com.codecore.servlet.SearchHomeServlet | 搜索首页的servlet，将结果转发到其他页面 |
| C | com.codecore.servlet.SearchProfileServlet | 搜索我的微博页面的servlet，将结果返回到其他页面。 |
| V | home1.jsp | 显示首页搜索结果的界面 |
| V | profile1.jsp | 显示我的微博页面结果的界面 |

### 搜索关注对象

* + 1. 功能描述

用户在登入账户后，可在我关注的人页面对所有以关注对象进行搜索，返回结果是关注人的基本信息。

* + 1. 功能实现
  + 用户在搜索框中输入关键字。
  + 搜索请求servlet，servlet将搜索结果返回在搜索页中。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.SearchDao | 用来实现搜索，searchFollowing是实现搜索关注人的方法。 |
| M | com.codecore.entity.UserInfo | 表示用户信息的实体类 |
| C | com.codecore.servlet.SearchFriendServlet | 搜索关注的servlet，将结果转发到其他页面 |
| V | friend1.jsp | 显示关注对象搜索结果的界面 |

### 搜索好友

* + 1. 功能描述

用户在登入账户后，可在我的好友页面对所有好友进行搜索，返回结果是好友的基本信息。

* + 1. 功能实现
  + 用户在搜索框中输入关键字。
  + 搜索请求servlet，servlet将搜索结果返回在搜索页中。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.SearchDao | 用来实现搜索，searchFollowing是实现搜索好友的方法。 |
| M | com.codecore.entity.UserInfo | 表示用户信息的实体类 |
| C | com.codecore.servlet.SearchFriendServlet | 搜索关注的servlet，将结果转发到其他页面 |
| V | friend1.jsp | 显示好友搜索结果的界面 |

### 搜索收藏

* + 1. 功能描述

用户在登入账户后，可对自己收藏的所有微博按关键字进行搜索，结果返回在收藏结果叶中。

* + 1. 功能实现
  + 用户在搜索框中输入关键字。
  + 搜索请求servlet，servlet将搜索结果返回在搜索页中。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.SearchDao | 用来实现微博的搜索，searchCollect是实现搜索首页的方法。 |
| M | com.codecore.entity.Blog | 表示微博信息的实体类 |
| C | com.codecore.servlet.SearchCollectServlet | 搜索收藏的servlet，将结果转发到其他页面 |
| V | collect1.jsp | 显示收藏搜索结果的界面 |

# 系统测试及运行

## 系统测试

测试就是利用测试工具按照测试方案和[流程](http://baike.baidu.com/view/158951.htm)对产品进行功能和[性能测试](http://baike.baidu.com/view/106187.htm)，甚至根据需要编写不同的测试工具，设计和维护测试系统，对测试方案可能出现的问题进行分析和评估。执行测试用例后，需要跟踪故障，以确保开发的产品适合需求。

### 测试过程

测试过程按4个[步骤](http://baike.baidu.com/view/1121831.htm" \t "_blank)进行，即单元测试、集成测试、确认测试和系统测试及发版测试。

开始是单元测试，集中对用源代码实现的每一个[程序单元](http://baike.baidu.com/view/2610106.htm" \t "_blank)进行测试，检查各个[程序模块](http://baike.baidu.com/view/421086.htm)是否正确地实现了规定的功能。

集成测试把已测试过的模块组装起来，主要对与设计相关的[软件体系结构](http://baike.baidu.com/view/1317046.htm" \t "_blank)的构造进行测试。

确认测试则是要检查已实现的软件是否满足了需求规格说明中确定了的各种需求，以及[软件配置](http://baike.baidu.com/view/551674.htm" \t "_blank)是否完全、正确。

系统测试把已经经过确认的软件纳入实际运行环境中，与其它系统成份组合在一起进行测试。

### 测试目的

(1)测试并不仅仅是为了找出错误.通过分析错误产生的原因和错误的发生趋势,可以帮助[项目管理](http://baike.baidu.com/view/65955.htm" \t "_blank)者发现当前[软件开发](http://baike.baidu.com/view/190611.htm)过程中的[缺陷](http://baike.baidu.com/view/1122313.htm),以便及时改进。

(2)这种分析也能帮助测试人员设计出有针对性的测试方法,改善测试的[效率](http://baike.baidu.com/view/47610.htm" \t "_blank)和有效性。

(3)没有发现错误的测试也是有价值的,完整的测试是评定微博质量的一种方法。

### 测试方法和工具

本系统采用如下方法进行测试：

（1）后台打印相关数据：在系统测试出错误的时候，可以把相关重要的数据和方法用后台打印的方法打印出来，通过打印出的数据进行分析从而找出漏洞所在。后台打印的好处是简单易行，工作量比较小。

（2）借助编译工具的辅助功能：可以在测试中使用MyEclipse的DeBug功能设置程序断点，跟踪程序找出错误所在，这样测试的好处是可以准确的跟踪程序走向，从整个程序运行的流程上一步一步找出问题所在。

测试工具：MyEclipse，通过编译工具MyEclipse的debug功能有效的跟踪代码，找出出错原因。

## 系统运行

本系统为JAVA编写的B/S结构的WEB项目，运行步骤如下。

1.运行windows操作系统。

2.由于本系统是由JAVA编写所以需要安装JDK1.6JAVA虚拟机器，安装方法为：运行JDK1.6安装文件，选择路径进行安装。

3.运行tomcat安装文件，选择安装路径，选择第2步已经安装好的JRE路径，进行tomcat的安装。

4.部署程序至tomcat。

5.安装MySQL数据库，导入表结构的SQL脚本，启动数据库实例服务。

6.启动tomcat。

7.打开浏览器并且输入如下网址。

<http://localhost:8080/>mail/welcome.jsp

系统需也要一些日常管理：包括日志的查看，服务器的维护等。