码农聊天网站

**目录**

**[1、概述](#_Toc26072_WPSOffice_Level1)** **[1](#_Toc26072_WPSOffice_Level1)**

[1.1 系统简述](#_Toc23902_WPSOffice_Level2) [1](#_Toc23902_WPSOffice_Level2)

[1.2 软件设计目标](#_Toc10426_WPSOffice_Level2) [1](#_Toc10426_WPSOffice_Level2)

[1.3 参考资料](#_Toc24844_WPSOffice_Level2) [1](#_Toc24844_WPSOffice_Level2)

[1.4 修订版本记录](#_Toc6231_WPSOffice_Level2) [1](#_Toc6231_WPSOffice_Level2)

**[2、术语表](#_Toc868_WPSOffice_Level1)** **[1](#_Toc868_WPSOffice_Level1)**

**[3、设计概述](#_Toc14947_WPSOffice_Level1)** **[1](#_Toc14947_WPSOffice_Level1)**

[3.1 系统结构设计](#_Toc27683_WPSOffice_Level2) [1](#_Toc27683_WPSOffice_Level2)

[3.2 系统接口设计](#_Toc21787_WPSOffice_Level2) [5](#_Toc21787_WPSOffice_Level2)

[3.3 约束和假定](#_Toc22591_WPSOffice_Level2) [5](#_Toc22591_WPSOffice_Level2)

[3.4 非功能性设计](#_Toc10909_WPSOffice_Level2) [6](#_Toc10909_WPSOffice_Level2)

**概要设计**

**1、概述**

**1.1 系统简述**

码农聊天网站是一个基于BS架构的交流应用。该系统是用于建立一个码农交流平台。系统功能包括用户认证、管理小组、处理小组及日程管理。

系统应当是稳定的、用户友好、易于使用、响应快速的。

**1.2 软件设计目标**

码农聊天网站致力于让用户享有完善的功能，同时性能也能令用户满意。反应速度快，数据准确，功能丰富是我们的基本目标。用户能享受简单，快捷，方便的操作，简单的，舒适的交流，实用的处理与高效的日程管理。网站能够准确的执行用户的各种操作，错误率低。

**1.3 参考资料**

**1.4 修订版本记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 修订内容 | 修订人 | 修订时间 |
| 1.0 | 完成了概要设计文档初稿 | 谢烨斌 | 2018.10.22 |
| 2.0 | 补充和完善了1.0部分内容，完成了系统数据模型 | 何志城 | 2018.10.23 |
| 3.0 | 修改1.1数据模型中的E-R图和数据字典 | 黄嘉豪 | 2018.10.24 |
| 3.1 | 修改逻辑架构和物理结构 | 谢烨斌 | 2018.11.10 |

**2、术语表**

见《需求分析2.1》

**3、设计概述**

**3.1 系统结构设计**

**3.1.1 系统逻辑架构**

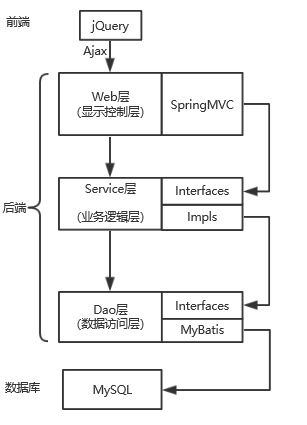
系统是基于BS架构的J2EE-web应用，前后台分离。

系统前端是基于jQuery 的web页面，是提供给用户的界面，及提供给用户的操作接口。

系统后台使用分层架构，采用springMVC + spring+mybatis框架 。

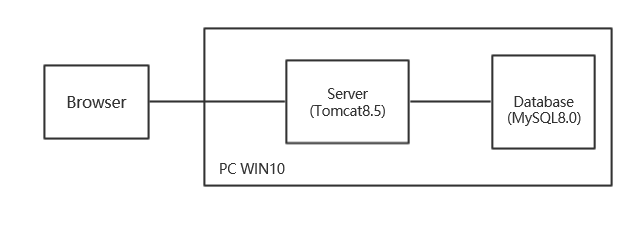
系统从上至下分为Web、Service及DAO层。Web层负责响应用户操作，通过SpringMVC框架完成用户的请求接收功能。SpringMVC的Controller作为整个应用的控制器，完成用户请求的转发及对用户的相应。Service层负责处理用户的业务逻辑，DAO层通过Mybatis框架实现JAVA Bean和数据库的交互,完成对数据库的增删查改。最后，使用Spring框架对系统进行管理。

数据库使用MySQL。



**3.1.2 系统物理结构**

系统采用BS架构



**3.1.3 系统数据模型**

**参见《系统数据模型》文档**

**3.2 系统接口设计**

**3.2.1 用户接口**

|  |  |  |
| --- | --- | --- |
| 页面 | 目标用户 | 提供接口 |
| Home | 普通用户 | 登陆、注册 |
| Group | 普通用户 | 浏览小组信息，加入/退出小组 |
| File | 普通用户 | 浏览小组文件记录，上传/下载文件 |
| Date | 普通用户 | 浏览个人日程，增加/修改/删除日程 |
| User | 普通用户 | 查看个人信息，编辑个人信息 |
| Ad-Member | 管理员用户 | 查看小组成员信息，邀请/允许/删除成员 |
| Announce | 小组管理员用户 | 发布/修改/删除小组公告 |
| Broadcast | 教师用户 | 往被选小组发布广播 |

**3.2.2 外部接口**

**3.3 约束和假定**

系统采用Java语言编写后台，html+css+jQuery编写前台页面。

**3.4 非功能性设计**

##### 3.4.1性能设计

1. 在PC机上搭一台服务器(双核4G 50GB 1MPs带宽以上) ，搭建环境（Linux + Apache + MySQL + JAVA），依靠云服务保证网站服务器性能方面达到最大保障。
2. 可以增加带宽大小，保证多用户并发访问，防止系统因访问量过大响应迟缓。
3. 根据需求租用更高性能的服务器，提升用户访问体验。

##### 3.4.2安全性设计

1. 服务器的托管选择在大的云服务器商处，以减少被攻击的可能性。

2. 服务器上系统和软件的版本尽可能选择稳定且新的版本，以减少软件或系统漏洞，关闭服务器不用的端口，及时更新安全补丁。

3. 后台开发时加强进行安全性测试，包括密码加密、使用验证码等。

4. 常做数据备份，防止数据丢失。

5. 管理员本身的密码尽可能复杂，并且通常访问数据库或后台的时候尽可能使用低权限的账户。

##### 3.4.3可靠性设计

1. 在云服务部署服务器，依靠腾讯云强大的服务器支持，可以保证系统7x24小时运行，全年持续运行故障停运时间累计不超过10小时。
2. 系统管理员不定期检查系统，当系统出现问题时及时维护。
3. 设置备份服务器，避免系统因服务器问题全面瘫痪。

##### 3.4.4数据保密设计

1. 监控数据库服务器的所有输出端口，控制其配置和信息，防止服务器被外部入侵。

2. 对传输的数据进行加密，防止数据在传输过程中被截取后泄露。

3. 严格把控数据库及后台管理员的账户及密码，定期更换复杂密码，防止数据库被内部破解。

##### 3.4.5可用性设计

1. 所有数据，包括用户数据，课程数据，留言信息数据，全部对应建立历史备份数据库，定期将当前数据库中的数据追加到历史数据库中。对历史数据库中的数据，定期备份，一般当前数据，每月一次自动复制到历史数据库中，历史数据库保留半年后，使用光盘刻录设备，进行2份以上备份，以后清除历史数据库这部分数据，以提高系统运行效率，释放硬件资源。
2. 主数据库服务器使用磁带备份系统，对数据库中的实时数据更新备份，可以采用每日，每周，每月系统自动备份到磁带机上，用跟踪带对系统的操作进行记录。
3. 如果当前数据库遭到破坏，不能保证完整性，可以考虑采用一下方法进行恢复：

①将历史数据库中的数据追加到数据库中；

②将磁带上的数据按月备份，周备份，日备份的顺序依次恢复到数据库中；

③还可以用光盘的数据导入数据库中。