**会议室预定系统**

**概要设计说明书\_**

**预定管理模块**

**文档描述**

|  |  |
| --- | --- |
| **文档名称** | 会议预定模块 |
| **文档编号** |  |
| **当前版本** | V1.0 |
| **内容简介** | 包括对会议预定的操作，并在预定成功后由会议创建者给会议参与者发送一份提醒会议邮件 |

**文档变更**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** | **审核** |
| 2018-01-16 | V1.0 | 创建 | 李春娣 |  |
| 2018-01-17 | V1.0 | 修改 | 李春娣 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

[第1章 引言 - 1 -](#_Toc14116)

[1.1 编写目的 - 1 -](#_Toc11202)

[1.2 项目背景 - 1 -](#_Toc19163)

[1.3 定义 - 1 -](#_Toc22116)

[1.3.1专门术语 - 1 -](#_Toc6718)

[1.3.2缩写 - 1 -](#_Toc30847)

[1.4 参考资料 - 2 -](#_Toc2238)

[第2章 任务概述 - 2 -](#_Toc7596)

[目标 - 2 -](#_Toc31186)

[2.1 运行环境 - 2 -](#_Toc4285)

[2.2 需求概述 - 3 -](#_Toc5463)

[2.3 条件与限制 - 3 -](#_Toc2343)

[条件 - 3 -](#_Toc20430)

[限制 - 3 -](#_Toc24767)

[第3章 总体设计 - 4 -](#_Toc23983)

[3.1 总体结构 - 4 -](#_Toc31817)

[3.2 子系统处理流程 - 5 -](#_Toc12840)

[- 5 -](#_Toc9654)

[3.2.2后台管理端处理流程 - 6 -](#_Toc27904)

[3.2.3图片服务 - 6 -](#_Toc19704)

[3.2.4缓存服务 - 6 -](#_Toc30844)

[3.2.5数据库系统 - 6 -](#_Toc3289)

[3.3 系统功能设计 - 7 -](#_Toc24436)

[3.3.1预定会议 - 7 -](#_Toc31551)

[3.3.2修改会议 - 9 -](#_Toc25847)

[3.3.3删除会议 - 11 -](#_Toc18615)

[3.3.4会议日程查看 - 13 -](#_Toc29232)

[3.3.5邮件发送 - 13 -](#_Toc4422)

[第4章 接口设计 - 14 -](#_Toc21619)

[4．1 外部接口 - 14 -](#_Toc4300)

[4．1．1 用户界面 - 14 -](#_Toc21649)

[4．1．2 软件接口 - 14 -](#_Toc16258)

[4．1．3 硬件接口 - 14 -](#_Toc14819)

[4．2 内部接口 - 15 -](#_Toc16211)

[第5章 数据库设计 - 15 -](#_Toc7861)

[5．1 数据字典 - 15 -](#_Toc2345)

[5.1.1 会议信息  - 15 -](#_Toc26866)

[5．2 数据库数据结构设计 - 15 -](#_Toc348)

[5.2.1.会议表的清单 - 15 -](#_Toc4414)

[第6章 运行设计 - 17 -](#_Toc5353)

[6．1 运行模块的组合 - 17 -](#_Toc29671)

[6．2 运行控制 - 17 -](#_Toc19742)

[6．3 运行时间 - 17 -](#_Toc594)

[第7章 出错处理设计 - 18 -](#_Toc30169)

[7．1 出错输出信息 - 18 -](#_Toc8396)

[7．2 出错处理对策 - 21 -](#_Toc4750)

[第8章 安全保密设计 - 22 -](#_Toc7142)

[第9章 维护设计 - 23 -](#_Toc176)

[第10章 附件 - 24 -](#_Toc14799)

[附录1:部署结构图 - 24 -](#_Toc23089)

[附录2:功能框图 - 26 -](#_Toc20285)

# 引言

## 编写目的

编写该会议室预定系统设计文档目的是为了解除各公司、企业、单位和个人需要受到地域限制，到达现场查询和预定会议室的局限，具体说明该系统的各个方面参数，具体功能、输入输出，以便向需要预定会议室进行规定范围内用途的客户提供全面的对系统的认识和使用。可以提高公司内部各会议室的利用效率；减少各部门因为会议室使用引起的冲突；增强软件工程师的设计、编程能力；

## 项目背景

* 项目名称：会议室预定系统(MMS)-会议预定模块
* 项目提出者：王勇
* 项目开发者：北京华钦

## 定义

### 1.3.1专门术语

* Mysql: 系统服务器所使用的数据库管理系统（DBMS）。
* SQL: 一种用于访问查询数据库的语言
* 主键：数据库表中的关键域。值互不相同。

### 1.3.2缩写

* 系统：若未特别指出，统指会议室预定系统。
* SQL: Structured Query Language（结构化查询语言）。
* DB:DataBase(数据库)的缩写
* AJAX: 即“Asynchronous Javascript And XML”（异步JavaScript和XML），是指一种创建交互式网页应用的网页开发技术。
* Nginx :("engine x") 是一个高性能的[HTTP](http://baike.baidu.com/view/9472.htm" \t "_blank)和 [反向代理](http://baike.baidu.com/view/1165595.htm" \t "_blank) 服务器，其特点是占有内存少，[并发](http://baike.baidu.com/view/684757.htm" \t "_blank)能力强.
* RESTFUL: （英文：Representational State Transfer，简称REST）描述了一个架构样式的网络系统，它提供了一组设计原则和约束条件。它主要用于客户端和服务器交互类的软件。基于这个风格设计的软件可以更简洁，更有层次，更易于实现缓存等机制。

## 参考资料

以下列出在概要设计过程中所使用到的有关资料：

1．  会议室预定系统项目计划任务书 华钦团队

2． 需求调研报告 华钦团队

3． 原型和需求规格说明书 华钦团队

# 任务概述

## 目标

我们的目标是将此会议室系统应用到人们的生活中去，为用户带来方便，

## 运行环境

系统将由七部分程序组成，包括：

1. 数据库。存储从外围系统获得的数据，目前已知会和ODS进行数据同步，并保持人网匹配系统自有数据。数据库需要做热备份或主从分离。
2. 缓存服务。存放程序从数据库加载来的，使用频繁的数据，以降低数据库IO压力，提高性能。
3. 图片服务。存取用户提交的图片，并且从主服务器分离。
4. 自动化同步程序。独立部署的应用程序，专门处理一些自动化任务
5. 移动端程序。调用api服务程序的接口完成业务功能（https方式）；这部分程序以weblogic为容器，独立部署，供邦邦社区远程加载。
6. Api服务程序。独立部署的应用程序，以weblogic为容器，对internet暴露接口，为移动端程序提供数据支持。
7. 后台管理系统。独立部署的内网程序，以weblogic为容器，对内网用户提供服务。

这7部分内容通过对内或对外暴露接口，实现子系统间的数据通讯。

所有功能全部运行于服务器之上，其基本要求CPU至少为XERO 2G，500G硬盘，8G内存；

## 需求概述

系统以会议室预定关系为核心：

我做的部分是会议预定功能，会议预定功能包括我的会议和全部会议。

我的会议查询的是当前用户的会议。将会议信息显示在日历日程上，当前用户是可以对会议进行预定，修改，删除。当预定成功和修改成功时向会议参与者发送邮件提醒，当会议结束时自动将会议的状态改为已结束，删除会议时向会议参与者发送取消会议的提醒且用户不可以删除状态为已结束的会议。所有会议是仅可查询所有的会议，并可以按多条件分页查询，方便用户对会议的管理。要求系统能有效、快速、安全、可靠和无误的完成上述操作。并要求用户的界面要简单明了，易于操作，服务器利于维护。

## 条件与限制

### 条件

本文基于以下条件的成立而成立

1. 预定管理系统(BM)总体存在.
2. 不同的权限;
3. 不同的会议显示前台;
4. 用户可对会议进行增删改查的操作;

### 限制

1. 本次的项目主要是涉及网页端考虑到客户端平台各自对html5的处理情况,尽量添加更多的主题来提供界面;
2. 数据的实时性限制:由于需要在载入前台操作界面时候需要访问后台数据层，所以前台访问会有一定延时;
3. 计算和处理能力限制:本系统本模块Me,因此对提前对要访问的数据库进行索引表的设计进行提高前台响应速度;
4. 语言和开发工具. Java语言,开发包版本号1.7+,推荐jdk1.8.开发工具使用eclipse;
5. 数据库有主从或者热备份.防止主数据库故障而导致程序中断

# 总体设计

## 总体结构

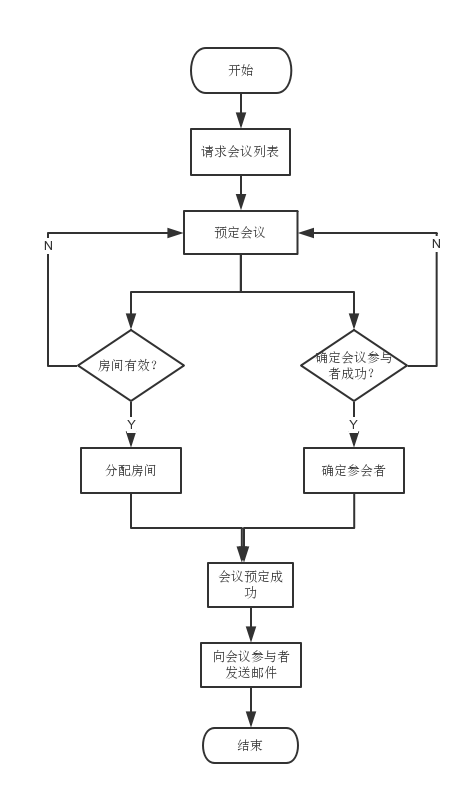
拓扑结构见[附录1](#_附录1:部署结构图)

本期项目包含的功能:

* 会议管理:包括对会议的增删改查。
* 会议信息管理:包括将会议显示在日程中，并可进行日期中对应会议 的查看及对会议信息进行增删改。;
* 邮件管理:包括当预定会议成功时，由创建人给每个会议参与者发送一份邮件提醒。

## 子系统处理流程

## 



从功能的菜单包含若8个子系统,包括:

* 会议预定管理
* 用户管理
* 会议室预定管理
* 邮件发送

基本协作流程如下:

前台用户进行登录后会根据某些客观原因，例如房间是否空闲等等来进行预定会议，并可以对会议及会议信息进行管理，当预定成功后，同时给会议参与者发送会议提醒邮件。

### 3.2.2后台管理端处理流程

1. 根据会议要求对会议进行管理。
2. 根据不同的url返回到不同的界面显示不同的界面效果;
3. 根据预定会议的情况为会议参与者发一份会议提醒邮件

### 3.2.3图片服务

独立应用程序,为系统提供图片的存储与读取服务.

1. 电脑客户端页面发起图片存储请求;
2. 图片服务校验token是否已经存在于缓存服务器;如果不存在;禁止用户存储图片;如果已经存在,用户合法,程序继续;
3. 存储图片,返回图片标识;
4. 读取图片操作与存储图片同理,返回的是图片的url;

### 3.2.4缓存服务

独立内网应用程序,为系统暂存周期短,操作频率高的数据(如用户信息等)或变化频度极低的数据(如组织机构,国家地区数据).

自动同步系统(详见下文)写入预处理的数据;

应用系统(api,后台管理系统)读出数据;

### 3.2.5数据库系统

这是整个系统的核心,所有的运行数据都是存储其中.

Api服务会对其进行读写操作;

缓存服务会从其中取出需要缓存的数据;

自动同步系统会对其进行读写操作.

后台管理系统也会对其进行读写操作.

外围系统(ODS)定时对数据库做同步,目前已知的是采用ETL方式.

这五者对数据库系统的操作在时序上并无固定的先后关系.

## 系统功能设计

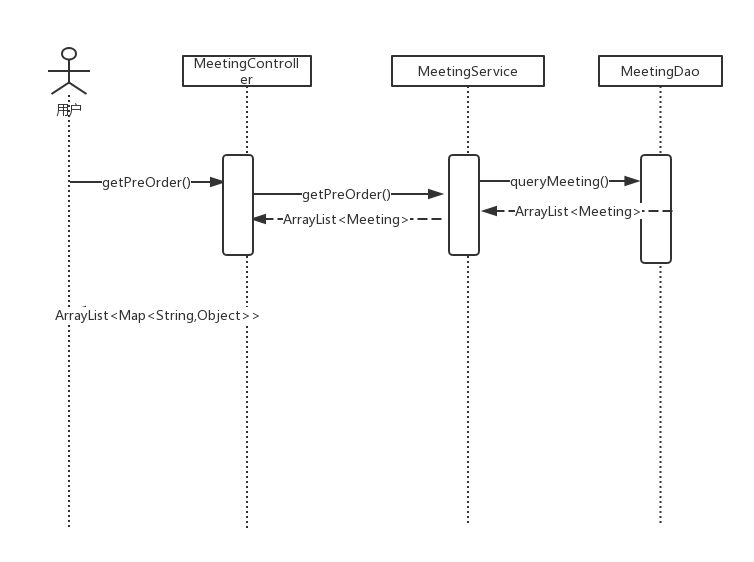
### 3.3.1预定会议

#### 功能描述

当用户进入查看会议页面时，首先显示会议列表在界面上。用户可以根据权限对会议进行增删改查。

#### 业务流程

* 1. 用户进入界面对已预订的会议信息进行查看。
  2. 根据工作分配添加预订会议。
  3. 为会议分配合适的会议室，如果会议室空闲且预订会议的所有都满足条件则预订成功。
  4. 预定成功后通过会议创建者向会议参与者发送一份会议提醒邮件。



#### 输入/输出内容

输入:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | | **类型** | **字段长度** | | **是否必输** | | **备注** | | |  |
| meeting\_id | | 整型 | 11 | | no | | 会议编号 | | |  |
| meeting\_name | | 文本 | 50 | | no | | 会议名称 | | |  |
| meeting\_user\_ids | | 文本 | 200 | | no | | 会议参与者(用户id的连接) | | |  |
| meeting\_title | | 文本 | 1000 | |  | | 会议标题 | | |  |
| meeting\_begintime | | 日期类型 | 50 | | no | | 会议开始时间 | | |  |
| meeting\_endtime | | 日期类型 | 50 | | no | | 会议结束时间 | | |  |
| meeting\_user\_id | | 整型 | 11 | | no | | 会议负责人 | | |  |
| meeting\_room\_id | | 整型 | 11 | | no | | 会议室 | | |  |
| meeting\_address | | 文本 | 50 | |  | | 会议所在地 | | |  |
| meeting\_issuccess | | 文本 | 2 | | no | | 是否预定成功 | | |  |
| meeting\_is\_enable | | 文本 | 2 | | no | | 是否删除 | | |  |
| meeting\_status | | 文本 | 2 | | no | | 会议状态0--结束 1--未结束 | | |  |
| 名称 | | 类型 | | | 是否必输 | | 长度 | | 备注 |
| 菜单ID | | 预订后自动增加 | | | 是 | | 11 | | 整形 |

输出:

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必输 | 备注 |
| 结果 | 文本 | 是 | 成功/失败 |

#### 业务规则

根据条件搜索会议。添加成功后Boolean返回true,反之返回false.

### 3.3.2修改会议

#### 功能描述

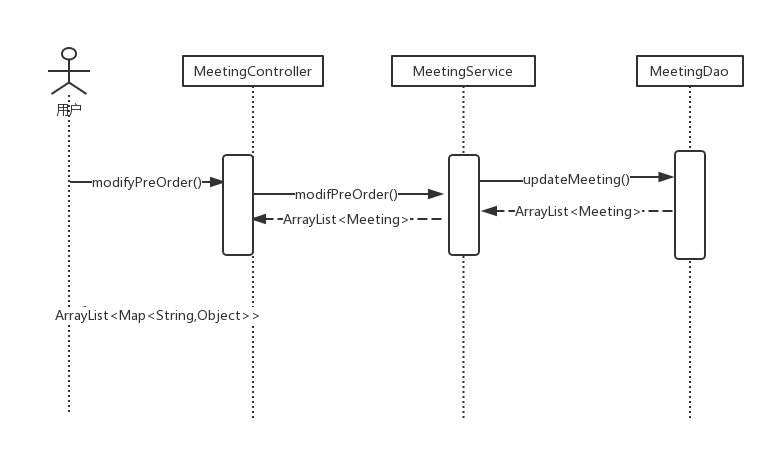
修改会议的具体信息。

#### 业务流程

1. 输入要修改的会议的信息

2. 输入要修改的会议信息

3. 修改成功后进行保存



#### 输入输出

1. 输入：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **字段长度** | **是否必输** | **备注** |  |
| meeting\_id | 整型 | 11 | no | 会议编号 |  |
| meeting\_name | 文本 | 50 | no | 会议名称 |  |
| meeting\_user\_ids | 文本 | 200 | no | 会议参与者(用户id的连接) |  |
| meeting\_title | 文本 | 1000 |  | 会议标题 |  |
| meeting\_begintime | 日期类型 | 50 | no | 会议开始时间 |  |
| meeting\_endtime | 日期类型 | 50 | no | 会议结束时间 |  |
| meeting\_user\_id | 整型 | 11 | no | 会议负责人 |  |
| meeting\_room\_id | 整型 | 11 | no | 会议室 |  |
| meeting\_address | 文本 | 50 |  | 会议所在地 |  |
| meeting\_issuccess | 文本 | 2 | no | 是否预定成功 |  |
| meeting\_is\_enable | 文本 | 2 | no | 是否删除 |  |
| meeting\_status | 文本 | 2 | no | 会议状态0--结束 1--未结束 |  |

1. 输出：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必输 | 备注 |
| 结果 | 文本 | 是 | 成功/失败 |

#### 业务规则

每个会议id对应着一条会议信息,根据条件搜索会议。添加成功后Boolean返回true,反之返回false。

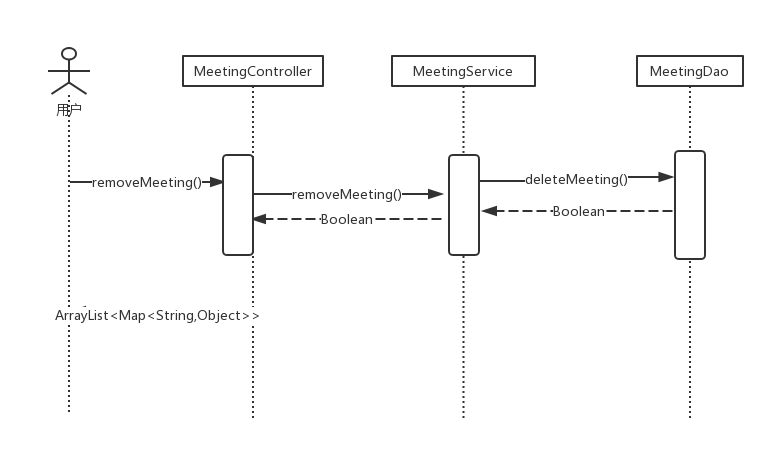
### 3.3.3删除会议

#### 功能描述

当会议被临时取消或是会议已过期时就可以对会议进行删除的操作。

#### 流程描述

1. 根据会议的id,执行将数据库中的表数据进行逻辑删除。
2. 删除成功后并通过会议创建人向会议参与者发送一份取消会议提醒。



**输入输出**

1.输入：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否必输 | 长度 | 备注 |
| Meeting\_id | 文本 | 是 | 11 | 会议ID |

2.输出：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必输 | 备注 |
| 结果 | 文本 | 是 | 成功/失败 |

#### 业务规则

Id唯一，且为逻辑删除

### 3.3.4会议日程查看

#### 功能描述

用户根据会议的时间将会议分别显示在大日历上，形成一个方便用户查看的会议日程。

#### 流程描述

在前台页面上根据日期显示会议。并可对会议进行增删改查管理。

#### 输入输出

1. 输入

用户传输URL地址。

1. 输出

用ArrayList<Meeting>接收的会议集合列表，并显示在前台页面上。

#### 业务规则

将通过对会议的查询LIst通过前台页面显示出来。

### 3.3.5邮件发送

### 功能描述

编辑邮件信息，通过会议创建人发送给每个会议参与者。

#### 流程描述

当用户预定会议成功后向每一个会议参与者发送一份会议提醒邮件。

#### 输入输出

无

#### 业务规则

无

# 接口设计

## 4．1 外部接口

### 4．1．1 用户界面

在用户界面部分，根据需求分析的结果，用户需要一个用户友善界面。在界面设计上，应做到简单明了，易于操作，并且要注意到界面的布局，应突出的显示重要以及出错信息。外观上也要做到合理化，考虑到用户已经对邦邦社区的操作已经非常熟悉，因此，整体风格和应尽量向这一方向靠拢。在设计语言上，已决定使用 Java8进行api端和后端管理系统编程，在界面上可使用，Javascripts和HTML5 编写界面。

总的来说，系统的用户界面应作到可靠性、简单性、易学习和使用。

### 4．1．2 软件接口

服务器程序可使用Java提供的对Oracle的接口，进行对数据库的所有访问。

API程序提供RESTFUL风格的接口,为html5移动端提供数据和各种类型的操作。

在网络软件接口方面，使用一种安全的无状态的传输协议。

### 4．1．3 硬件接口

在输入和输出方面，主要以智能手机（Android和iOS系统）的触控方式为主。目前这对绝大多数用户已经非常熟练。

## 4．2 内部接口

内部接口方面，各模块之间采用方法调用、参数传递、返回值的方式进行信息传递。接口传递的信息将是以数据结构封装了的数据，以参数传递或返回值的形式在各模块间传输。

# 数据库设计

## 5．1 数据字典

### 5.1.1 会议信息

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 会议ID | 会议标题 | 会议参与者 | 会议内容 | 会议结束时间 | 会议负责人 | 所在会议室 | 是否删除 | 是否预定成功 | 会议状态 | 创建时间 | 创建人名 | 修改时间 | 修改人名 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |

会议信息表保存了从预定系统同步而来的数据 ，是整个系统的基础数据之一。

## 5．2 数据库数据结构设计

### 5.2.1.会议表的清单

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **表字段名** | **类型** | **数据长度** | **是否必填** | **主键** | **唯一\*** | **外键\*** | **默认值** | **具体值** | **说明** |
| 会议编号 | MEETING\_ID | INTEGER | 11 | T | \* | \* |  | 1 | 1 | 自增 |
| 会议标题 | MEETING\_TITLE | VARCHAR | 50 | T |  |  |  |  |  | 会议的名称 |
| 会议参与者 | MEETING\_USER\_IDS | VARCHAR | 200 | T |  |  |  |  |  | user表中用户的id连接 |
| 会议内容 | MEETING\_CONTENT | VARCHAR | 1000 | F |  |  |  |  |  |  |
| 会议开始时间 | MEETING\_BEGINTIME | VARCHAR | 50 | T |  |  |  |  |  |  |
| 会议结束时间 | MEETING\_ENDTIME | VARCHAR | 50 | T |  |  |  |  |  |  |
| 会议负责人 | MEETING\_USER\_ID | INTEGER | 11 | T |  |  | \* |  |  | 外键 |
| 所在会议室 | MEETING\_ROOM\_ID | VARCHAR | 11 | T |  |  | \* |  |  | 外键 |
| 是否删除 | MEETING\_IS\_ENABLE | CHAR | 2 | T |  |  |  | 1 |  | 0-已删除 1-未删除 |
| 是否预定成功 | MEETING\_SUCCESS | CHAR | 2 | T |  |  |  | 0 |  | 0-已成功 1-未成功 |
| 会议状态 | MEETING\_STATUS | CHAR | 2 | T |  |  |  | 1 |  | 0-已结束 1-未结束 |
| 创建时间 | MEETTING\_CREATE\_DATETIME | VARCHAR | 50 | T |  |  |  | 当前日期 | 2018-1-2 20:20:20 |  |
| 创建人名 | MEETING\_CREATE\_NAME | VARCHAR | 50 | T |  |  |  |  |  | 用户姓名 |
| 修改时间 | MEETING\_UPDATE\_DATETIME | VARCHAR | 50 | T |  |  |  | 修改日期 | 2018-1-2 20:20:20 | 以上一次的修改时间为数据内容 |
| 修改人名 | MEETING\_UPDATE\_NAME | VARCHAR | 50 | T |  |  |  |  |  | 用户姓名 |
| 默认字段1 | MEETING\_DEFAULT1 | VARCHAR | 50 | F |  |  |  |  |  |  |
| 默认字段2 | MEETING\_DEFAULT2 | VARCHAR | 50 | F |  |  |  |  |  |  |

# 运行设计

## 6．1 运行模块的组合

系统的移动端以html5页面异步调api的方式完成各项功能.这其中会涉及诸如自动数据预处理,缓存的存取与更新,数据库的存取,图片的存取以及后台系统对参数的配置.通过这些模块功能的搭配使用,能够实现系统的快速响应,快速处理,数据安全等方面的需求.

## 6．2 运行控制

移动端应用程序的运行控制将严格按照各功能模块间对API调用关系来实现。

在网络传方面，手机端的邦邦社区通过远程的方式,加载部署的移动端首页,页面通过ajax方式调用系统api,验证用户的合法性,并计算角色,根据角色生成相应的页面.

成功进入首页后,用户可以使用各种所见到的系统功能.基本方式都是通过页面发https请求,调用各种预定义的api接口完成相应功能。

## 6．3 运行时间

在系统的需求分析中，对运行时间的要求为必须对作出的操作有较快的反应。网络硬件对运行时间有最大的影响，当网络负载量大时，对操作反应将受到很大的影响。所以将采用高速网络，实现各服务器或服务器集群之间的连接，以减少网络传输上的开销。其次是服务器的性能，这将影响对数据库访问时间即操作时间的长短，影响加大用户的等待时间，所以必须使用高性能的服务器，建议使用Xero 2G以上处理器,内存不低于8G。

# 出错处理设计

## 7．1 出错输出信息

程序在运行时主要会出现两种错误：

1、由于输入信息，或无法满足要求时产生的错误，称为软错误。

2、由于其他问题，如网络传输超时等，产生的问题，称为硬错误。

对于软错误，须在各业务功能中的数据验证模块由数据进行数据分析，判断错误类型，再生成相应的错误提示语句，送到输出模块中。

对与硬错误，可在出错的相应模块中输出简单的出错语句，并将程序重置。返回输入阶段。

出错信息必须给出相应的出错原因，下面是部分预定义的错误信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 系统错误码表 | | | | |
| 起始代号 | 级别 | 代码 | 描述 | 备注 |
| 9 | 系统 | 901 | 内部服务错误 | 内部程序错误 |
|  |  | 902 | 没有此选项(值),或选项(组)被禁用 | 没有此选项(值),或选项(组)被禁用 |
|  |  | 903 | 查询字典错误 | 没有此字典数据 |
|  |  | 904 | Api自定义权限出错 | Api自定义权限出错 |
|  |  | 905 | 校验token出错出错 | 校验token出错出错 |
|  |  | 906 | Api权限校验出错 | Api权限校验出错 |
|  |  | 907 | token不存在 | token不存在 |
|  |  | 908 | token不合法或者无效 | token不合法或者无效 |
|  |  | 909 | 没有指定调用方法 | 没有指定调用方法 |
|  |  | 910 | 指定方法没有默认的构造函数 | 指定方法没有默认的构造函数 |
|  |  | 911 | 接口处理类出错,权限校验成功 | 接口处理类出错,权限校验成功 |
|  |  | 912 | 指定业务处理方法没有默认的构造函数 | 指定业务处理方法没有默认的构造函数 |
|  |  | 913 | 获取规则类错误 | 获取规则类错误 |
| 1 | 应用-通用 | 100 | 未授权 | 不带token或使用错误的token访问api |
|  |  | 101 | 用户名或密码错误 | 登录失败 |
|  |  | 102 | 此员工不是十人长 | 此员工不是十人长 |
|  |  | 103 | 查询资源人资源的所属渠道失败 | 查询资源人资源的所属渠道失败 |
|  |  | 104 | 获取渠道长下属的三级机构列表(包括本级和4级,嵌套显示)失败 | 获取渠道长下属的三级机构列表(包括本级和4级,嵌套显示)失败 |
|  |  | 105 | 机构下属的资源人列表查询失败 | 机构下属的资源人列表查询失败 |
|  |  | 106 | 网点所属的渠道查询失败 | 网点所属的渠道查询失败 |
|  |  | 107 | 用户头像查询失败 | 用户头像查询失败 |
| 2 | 应用-外勤 |  |  |  |
| 3 | 应用-资源 | 301 | 上报信息删除失败 | 上报信息删除失败 |
|  |  | 302 | 上报信息查询错误 | 上报信息查询错误 |
|  |  | 303 | 上报信息保存失败 | 上报信息保存失败 |
|  |  | 304 | 网点信息查询错误 | 网点信息查询错误 |
|  |  | 305 | 查询联系人错误 | 查询联系人错误 |
| 4 | 应用-业绩 |  |  |  |
|  |  |  |  |  |
| 5 | 应用-客户 | 501 |  |  |
|  |  |  |  |  |
| 6 | 应用-组件 | 601 | 内部服务错误 | 内部程序错误 |
|  |  | 602 | 未注册的服务 | 没有这个方法 |
|  |  | 603 | 存储文件出错 | 存储文件出错 |
|  |  | 604 | 删除文件未成功 | 删除文件未成功 |
|  |  |  |  |  |
| 7 | API权限校验 | 701 | 非资源人身份 | 非资源人身份 |
|  |  | 702 | 非十人长身份 | 非十人长身份 |
|  |  | 703 | 非渠道长身份 | 非渠道长身份 |
|  |  | 704 | 非资源人所属的信息数据 | 非资源人所属的信息数据 |
|  |  | 705 | 非资源人所属的准客户信息 | 非资源人所属的准客户信息 |
|  |  | 706 | 非组员归属的事业部的十人长 | 非组员归属的事业部的十人长 |
|  |  | 707 | 非本事业部组长 | 非本事业部组长 |
|  |  | 708 | 非本资源人业绩数据 | 非本资源人业绩数据 |

## 7．2 出错处理对策

所有的服务器都必须安装不间断电源以防止停电或电压不稳造成的数据丢失的损失。

所有部件,包括移动端页面,api,数据库,缓存,图片服务,自动同步程序,后台管理系统分开部署,且重要部件,如移动端页面,api,缓存,图片服务必须做集群和负载均衡.

作为系统的核心部件,数据库必须有热备份获主从的设置,防止因数据库的原因导致系统宕机.

在网络传输方面，可考虑建立一条成本较低的后备网络，以保证当主网络断路时数据的通信。

在硬件方面要选择较可靠、稳定的服务器机种，保证系统运行时的可靠性。

# 安全保密设计

由于数据的传输上需要通过网络传输，为了确保信息的保密，需要在网络的传输过程中对数据进行加密。因此，对api接口的调用都采用https协议。

对于移动端功能，通过内置的权限系统首先按角色进行第一重权限划分。

对于暴露在Internet的公用api，首先使用token验证用户的第二重合法性。如果用户不带token或者伪造token，将无法获取到任何有效数据。

对暴露在外网的数据，还会做第三重用户验证。即进行数据表内的行级数据权限验证。

后台管理功能通过权限管理系统，为后台用户分配不同的权限，

在加密算法选择上将使用MD5 加密算法。

关键信息如密码在传输之前就进行加密.

用户登录时在服务器端产生的token等重要信息设置有效期,并且做定时清除.客户端不存储用户数据.

所有日志不能删除.

输入的数据会做字符长度的限制,并根据应用场景对特殊字符进行处理.

# 维护设计

维护方面主要为对服务器上的数据库数据进行维护和。可使用 Oracle 的数据库维护功能机制。例如，定期为数据库进行备份,ETL方式数据同步,数据库运行时的热备份.

# 附件

## 附录1:部署结构图



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能 | 建议服务器数量 | 配置要求 | 备注 |
| 1 | 静态html服务器集群 | 3 | CPU: XERO 2G  内存:8G  硬盘500G | 建议使用apahce服务器容器或者直接使用Nginx |
| 2 | Api服务器集群 | 3 | CPU: XERO 2G  内存:8G  硬盘500G | Weblogic |
| 3 | 后台管理系统集群 | 2 | CPU: XERO 2G  内存:8G  硬盘500G | Weblogic |
| 4 | 图片服务集群 | 2 | CPU: XERO 2G  内存:8G  硬盘500G | 与mongoDB搭配使用 |
| 5 | mongoDB集群 | 2 | CPU:XERO 2G  内存:8G  硬盘500G或更高 |  |
| 6 | 数据库服务器(主) | 1 | CPU: XERO 2G  内存:8G或更高  硬盘500G或更高 | Oracle |
| 7 | 数据库服务器(从) | 1 | CPU: XERO 2G  内存:8G或更高  硬盘500G或更高 | Oracle |
| 8 | 同步系统 | 1 | CPU: XERO 2G  内存:8G  硬盘500G | 做数据预处理 |
| 9 | 缓存服务器集群 | 2 | CPU:XERO 2G  内存:8G以上  硬盘500G | Redis |

以上的配置要求不包含图片备份和数据库备份需要的额外服务器.

## 附录2:功能框图

