 **南京航空航天大学**

计算机科学与技术学院/人工智能学院

软件工程综合课程设计

**概 要 设 计 报 告**

|  |  |
| --- | --- |
| 项目名称： | 企业会议通信系统 |
| 学号姓名： | 161830207柴华溢  161830129郭世祺  161830123蒋希伟  161830314肖家豪 |
| 所在专业： | 软件工程 |
| 手 机 号： | 18073173124 |
| 授课老师： | 张德平 |

二〇二一年十一月

目 录

[1范围 4](#_Toc87715991)

[1.1标识 4](#_Toc87715992)

[1.2系统概述 4](#_Toc87715993)

[1.3文档概述 4](#_Toc87715994)

[2引用文档 4](#_Toc87715995)

[3设计决策 4](#_Toc87715996)

[3.1假设 5](#_Toc87715997)

[3.2系统体系结构 5](#_Toc87715998)

[3.3软件体系结构 5](#_Toc87715999)

[3.4设计决策 7](#_Toc87716000)

[3.4.1输入/输出设计决策 7](#_Toc87716001)

[3.4.2行为设计决策 7](#_Toc87716002)

[3.4.3数据显示设计决策 7](#_Toc87716003)

[3.4.4安全性设计决策 7](#_Toc87716004)

[3.4.5保密性设计决策 8](#_Toc87716005)

[3.4.6其他CSCI级设计决策 8](#_Toc87716006)

[4 CSCI体系结构设计 9](#_Toc87716007)

[4.1 CSCI体系结构概述 9](#_Toc87716008)

[4.2 CSCI包汇总 9](#_Toc87716009)

[4.3 CSCI类汇总 9](#_Toc87716010)

[4.4 CSCI 时许图 10](#_Toc87716011)

[4.4.1 用户管理模块 10](#_Toc87716012)

[4.4.1.1 登录 10](#_Toc87716013)

[4.4.1.2 注册 11](#_Toc87716014)

[4.4.2 会议预定模块 12](#_Toc87716015)

[4.4.2.1 预约会议 12](#_Toc87716016)

[4.4.2.2 查看预约会议 12](#_Toc87716017)

[4.4.3 视频通话模块 12](#_Toc87716018)

[4.4.3.1 视频通话 12](#_Toc87716019)

[4.4.3.2 文字聊天 13](#_Toc87716020)

[4.4.3.3 发送文件 13](#_Toc87716021)

[4.4.3.4 白板演示 14](#_Toc87716022)

[4.4.3.5 分享桌面 14](#_Toc87716023)

[4.5 接口设计 15](#_Toc87716024)

[4.5.1 接口标识和接口图 15](#_Toc87716025)

[4.5.2 信息接口 16](#_Toc87716026)

[4.5.3 API接口： 16](#_Toc87716027)

[5 用户界面设计 16](#_Toc87716028)

[5.1应当遵循的界面设计规范 16](#_Toc87716029)

[5.2界面信息汇总 17](#_Toc87716030)

[6配置文件设计 20](#_Toc87716031)

[7性能设计 21](#_Toc87716032)

[7.1缓存性能： 21](#_Toc87716033)

[7.2多线程性能： 21](#_Toc87716034)

[7.3数据库优化： 21](#_Toc87716035)

[7.4代码性能： 21](#_Toc87716036)

[8系统出错处理设计 21](#_Toc87716037)

[8.1出错信息 22](#_Toc87716038)

[8.2补救措施 22](#_Toc87716039)

[8.3系统统维护设计 23](#_Toc87716040)

[8.4错误处理设计 23](#_Toc87716041)

[9需求可追踪性 23](#_Toc87716042)

[10注释 24](#_Toc87716043)

[10.1软件标识号 24](#_Toc87716044)

[10.2 CSCI 24](#_Toc87716045)

[10.3 ODBC 24](#_Toc87716046)

[10.4 ISO 24](#_Toc87716047)

[10.5 OS 24](#_Toc87716048)

[10.6 DBSM与DB 24](#_Toc87716049)

[10.7 DBDD 25](#_Toc87716050)

1范围

1.1标识

1. 软件标识号： Alpha（α）
2. 发布号：V01
3. 名称： 企业会议通信系统
4. 缩略名： 会议系统
5. 版本号：V1.0

1.2系统概述

企业会议通信系统就是企业基于市场部署和成本控制的考虑，在不同地点部署的满足企业的集中式管理，降低日益频繁的跨地域沟通交流和快速决策成本的多媒体通信视频会议系统。

目前已完成文档：《需求获取报告》 《开发手册》

1. 需方：需要改善现状，检测不良驾驶行为的人
2. 用户：全国需要对自己驾驶行为进行检测的机动车驾驶员
3. 开发方：软件工程课设小组

1.3文档概述

本文档主要描述企业会议通信系统检测软件配置项CSCI级的设计决策以及体系结构设计，包括CSCI从部件到软件单元的执行方案、接口说明，满足需求可追踪性，以确保它们完备、可行、可实现和可验证性。

在使用此文档时，需要首先阅读《需求获取报告》 和 《开发手册》，以保证获得准确的信息。

2引用文档

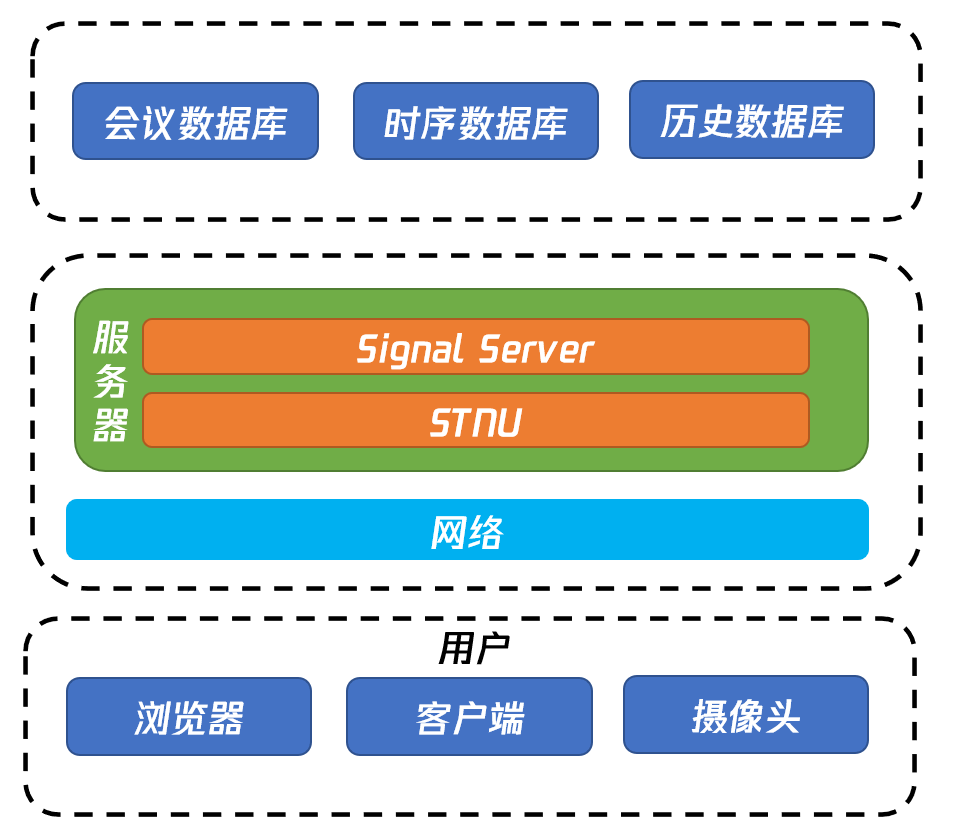
1. 国家软件1工程开发标准
2. 《需求获取报告》
3. 《开发手册》

3设计决策

3.1假设

在进行设计之前，首先假设网络通畅无阻，并且用户授权浏览器开启音频和视频权限，以确保会议系统能够获取视频流和音频流并实时交换传输信息、

3.2系统体系结构



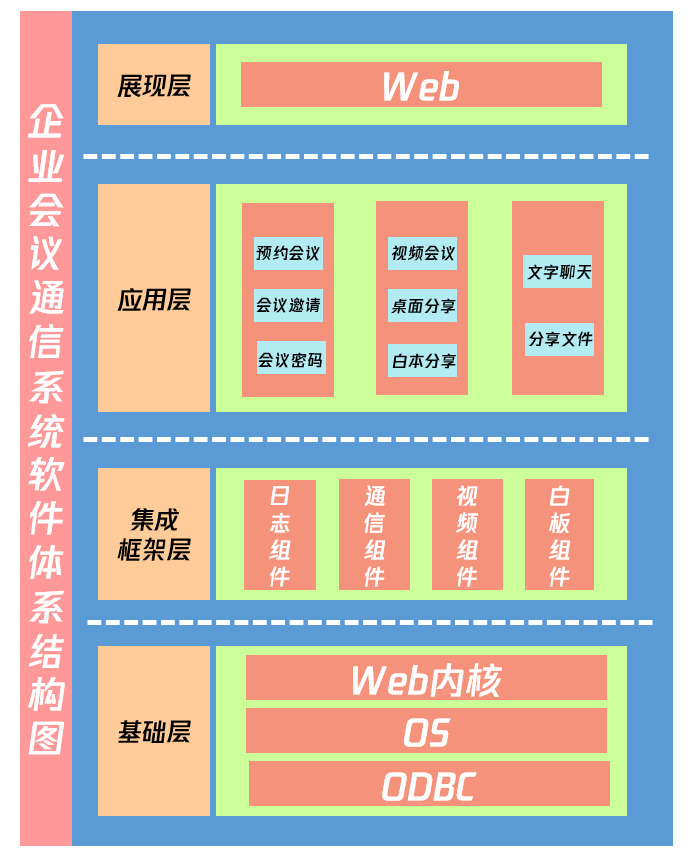
图【1】企业会议通信系统软件系统体系结构图

客户在开启摄像头以及麦克风后，系统会通过调用浏览器接口获取对应视频流和音频流，并及时在客户端显示出来。

在加入会议后，如果有其他用户再次入会，双方客户端在后端转发A\O模型，并转为P2P通信。

数据库中会记录会议预约信息，参会信息，主持人，聊天记录，文件记录等。

3.3软件体系结构



图【2】企业会议通信系统软件软件体系结构图

本次软件体系结构图采用了分层架构，也叫做N层架构，采用此模式是因为它是大多数应用的实际标准，与许多传统IT公司的组织构架十分相似。

分层架构模式中，组件被分成了就几个平行的层次，比如展示层、基础层、数据库层等。每一层都有其特定的职能。

展示层，也可以叫界面层，主要看预计的表现方式，比如我们这次使用的是WEB方式，表现层如何定义和修改，都不会从根本上决定一个软件的好坏，一切还是要看软件的设计逻辑。

应用层应用层包括ISO中的表示层，会话层，应用层。应用层也是工作在终端设备的，包括手机，计算机。应用层的主要功能是定义应用间的通信规则。

应用集成框架层，其实是对应用层的一个整合与集成，将各个零零散散的、联系薄弱的应用功能合并多个大型组件，方便以后在基础层上的连接与修改。最重要的还是可以去除应用冗余，给应用“瘦身”，节省更多的空间。

基础层，顾名思义，是应用的重中之重，做好基础层才能有更好的应用开发。他包括浏览器内核，用以将应用以WEB形式展现。它也包括操作系统，实现系统各行为的定义，管理计算机硬件与软件资源。操作系统同时也需要处理如管理与配置内存、决定系统资源供需的优先次序、控制输入设备与输出设备、操作网络与管理文件系统等基本事务。操作系统也提供一个让用户与系统交互的操作界面。最后基础层还包括ODBC，可以解决异构数据库间的数据共享。ODBC 为异构数据库访问提供统一接口，允许应用程序以SQL 为数据存取标准，存取不同DBMS管理的数据；使应用程序直接操纵DB中的数据，免除随DB的改变而改变。

数据层相对而言比较简单，主要是对非原始数据的操作与存取。除了对数据的操作，还有对数据库的操作，并为基础层提供数据服务。

3.4设计决策

3.4.1输入/输出设计决策

表【1】输入/输出设计决策

|  |  |  |  |
| --- | --- | --- | --- |
| **数据处理模块** | **原始输入数据** | **输出数据** | **特殊需求说明** |
| 用户账户信息 | 账号、密码 | 判断用户是否存在 | 用户权限界定 |
| 加入会议 | 会议号 | 判断会议是否存在 | 防止系统创建过多不存在的会议占用资源 |
| 视频流信息 | 用户视频流 | 对方和自己的视频流 | 核心需求 |
| 白板信息 | 输入笔画、颜色、粗细等信息 | 视频流 | 捕捉用户的绘画信息 |
| 文件信息 | 文件 | 文件 | 无 |

3.4.2行为设计决策

视频音频数据传输延迟不超过500ms。

白板数据传输延迟不超过100ms。

3.4.3数据显示设计决策

1. 视频数据显示：为便于视频传输过程中对目标信息进行直观分析，接收到的目标信息采用态势图的形式显示；
2. 文件导出：只有软件空闲时，才允许对数据文件的导出。
3. 违规提醒：出现违规时，违规提醒将被显示在客户端最中央；多种违规同时出现，则违规提醒将被排列显示与客户端中央。
4. 个人信息：只有账号仅在一处登录时才可进行信息的修改和查看，避免出现访问冲突。
5. 查看历史记录：将会显示近期所有的违规，并对违规种类进行统计。

3.4.4安全性设计决策

CSCI的安全性是指防止其合法用户使用该CSCI时对系统或其中的信息造成不良影响。

表【2】企业会议通信系统软件软件安全性设计决策表

| 序号 | 安全性需求 | 采取的防护措施 |
| --- | --- | --- |
|  | 输入信息的合法性 | * 所有用户录入信息入库前必须进行数据的合法性检查，对非法数据CSCI将提示用户修改，直至数据合法方可入库； * 对录入信息的必填项进行高亮或特殊颜色提示，入库前进行数据库主键检查； * 在各人机界面上进行信息录入说明。 |
|  | 误操作防护 | * 按照工作流程要求，在软件界面中对功能按钮的状态（有效/无效）进行控制； * 编制CSCI运行导航程序； * 增加操作撤消与恢复功能。 |
|  | 信息删除警示 | * CSCI中所有数据库数据的删除操作均给予用户以删除警示，并由用户进行删除确认后方执行信息删除操作； * 用户删除的所有数据文件均存放在系统的回收站中，用户可根据需要还原或清理回收站中的文件。 * CSCI中所有数据库数据的删除操作均给予用户以删除警示，并由用户进行删除确认后方执行信息删除操作； * 用户删除的所有数据文件均存放在系统的回收站中，用户可根据需要还原或清理回收站中的文件。 |
|  | 数据库访问权限控制 | * 利用数据库管理系统，对不同的用户分配不同的角色，并对其进行可访问信息的范围和访问权限进行设置； * 对用户访问数据库的操作进行审计。 |

3.4.5保密性设计决策

CSCI的保密性主要是防止非法用户对CSCI的攻击。

表【3】企业会议通信系统软件保密性设计决策表

| 序号 | 保密性需求 | 采取的措施 |
| --- | --- | --- |
|  | 身份验证 | 启动软件时，首先要求用户输入用户身份信息（用户名和口令），CSCI对其进行比对识别后，合法用户方能够启动运行该软件；对非法用户将给予登陆警示和记录，登陆警示超过X次将退出；  使用用户身份卡、读卡器和识别接口卡等设备对本CSCI运行的计算机设备的开关机及软件启动等操作进行控制。 |
|  | 数据库访问权限控制 | 利用数据库管理系统，对不同的用户分配不同的角色，并对其进行可访问信息的范围和访问权限进行设置；  对用户访问数据库的操作进行审计。 |

3.4.6其他CSCI级设计决策

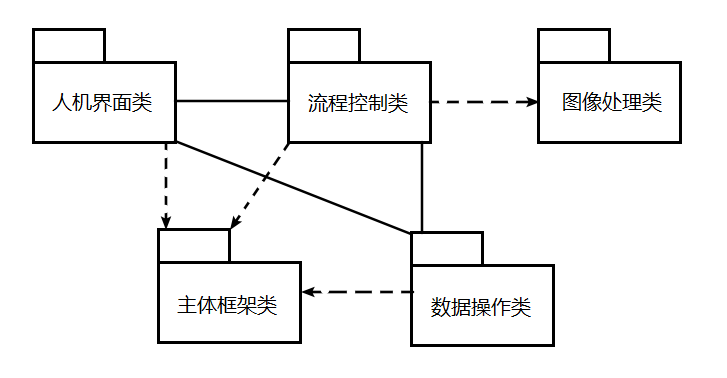
表【4】企业会议通信系统软件其它设计决策表

| 序号 | 灵活性/可维护性需求 | 采取的措施 |
| --- | --- | --- |
|  | 短时间大量数据涌入导致系统调转速度慢 | * 采用实时+历史双数据库，保证数据库的存储量； * 优化底层逻辑，提高系统抗压能力。 |
|  | 某项功能需要暂时关闭 | * 在应用开发时留下相应的接口。 |
|  | 系统遭到攻击 | * 切断网络 * 寻找攻击源 * 备份用户数据 * 修复程序或系统漏洞 * 恢复数据和连接网络 |

4 CSCI体系结构设计

4.1 CSCI体系结构概述

该软件的CSCI体系结构如下图所示，展示了包之间的依赖关系：



图【2】CSCI体系结构图

4.2 CSCI包汇总

确定包的命名规则，确保软件设计文档的风格与代码的风格保持一致。软件的包汇总表如下：

表【5】驾驶员违规驾驶行为检测软件包汇总表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 标识 | 名称 | 说明 | 开发类型 |
| 1 | PAK\_UI | 人机界面包 | 前端页面展示、向后端发送请求、渲染处理结果 | 新研 |
| 2 | PAK\_CRTL | 流程控制包 | 参数传递、功能逻辑 | 新研 |
| 3 | PAK\_IMGP | 图像处理包 | 视频用户行为检测的主要依赖 | 修改重用 |
| 4 | PAK\_DBOP | 数据操作包 | 数据库操作 | 新研 |
| 5 | PAK\_FRAME | 主体框架包 | 该软件所使用的Vue、NodeJS、MySQL等主要技术中的关键部分，起到整合和互联各组件的作用 | 修改重用 |

4.3 CSCI类汇总

确定类的命名规则，确保软件设计文档的风格与代码的风格保持一致。软件的类汇总表如下：

表【6】企业会议通信系统软件类汇总表

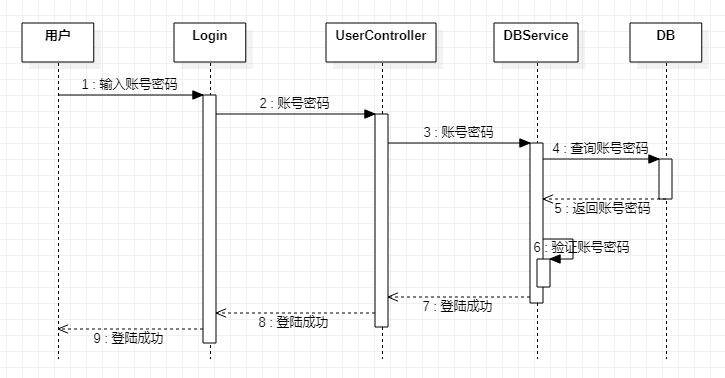
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAK\_UI（人机界面包） | | | | |
| 序号 | 标识 | 名称 | 说明 | 开发类型 |
| 1 | Home | 主页类 | 启动软件后展示的第一个页面 | 新研 |
| 2 | Login | 登录类 | 登录页面 | 新研 |
| 3 | Register | 注册类 | 注册页面 | 新研 |
| 4 | DetectCenter | 图像检测类 | 会议行为检测的页面 | 新研 |
| 5 | CommonShow | 普通展示类 | 违规历史展示页面 | 新研 |
| 6 | CommonDetail | 普通详情类 | 违规记录详情页面 | 新研 |
| 7 | CommonAnalyse | 普通分析类 | 违规行为分析页面 | 新研 |
| 8 | CommonAppeal | 普通申诉类 | 普通用户向管理员申诉的页面 | 新研 |
| 9 | AdminShow | 管理展示类 | 管理员查看和删除违规历史的页面 | 新研 |
| 10 | AdminAdd | 管理添加类 | 管理员添加违规记录的页面 | 新研 |
| 11 | AdminEdit | 管理修改类 | 管理员修改违规记录的页面 | 新研 |
| 12 | AdminAppeal | 管理申诉类 | 管理员回复申诉的页面 | 新研 |
| 13 | CommonVerify | 普通验证类 | 验证当前身份是否为普通用户的页面 | 新研 |
| 14 | AdminVerify | 管理验证类 | 验证当前身份是否为管理员的页面 | 新研 |
| 15 | Help | 帮助类 | 帮助页面 | 新研 |
| 16 | About | 关于类 | 关于页面 | 新研 |
| PAK\_CRTL（流程控制包） | | | | |
| 序号 | 标识 | 名称 | 说明 | 开发类型 |
| 1 | UserController | 用户管理控制类 | 控制用户管理模块 | 新研 |
| 2 | DetectController | 用户行为  检测控制类 | 用户行为检测模块 | 新研 |
| 3 | SignalController | 信令信息控制类 | 信令信息管理模块 | 新研 |
| 4 | MeetingController | 会议管理  控制类 | 会议管理模块 | 新研 |
| 5 | VerifyController | 身份验证控制类 | 控制身份验证模块 | 新研 |
| 6 | OtherController | 其他控制类 | 控制帮助模块 | 新研 |
| PAK\_DBOP（数据操作包） | | | | |
| 序号 | 标识 | 名称 | 说明 | 开发类型 |
| 1 | tUser | 用户类 | 用户身份信息数据库表的操作 | 新研 |
| 2 | tMeeting | 会议信息类 | 会议的数据库表的操作 | 新研 |
| PAK\_FRAME（主体框架包） | | | | |
| 序号 | 标识 | 名称 | 说明 | 开发类型 |
| 1 | settings | 框架配置类 | Django框架相关配置 | 修改重用 |
| 2 | backend\_urls | 后端url配置类 | 维护后端url | 修改重用 |
| 3 | index | 前端url配置类 | 维护前端url | 修改重用 |
| 4 | models | 数据模型类 | 维护数据库模型 | 修改重用 |
| 5 | serializers | 序列化类 | 数据序列化，促进前后端交互 | 修改重用 |
| 6 | views | 视图集类 | 支持前后端交互 | 修改重用 |

4.4 CSCI 时许图

4.4.1 用户管理模块

4.4.1.1 登录

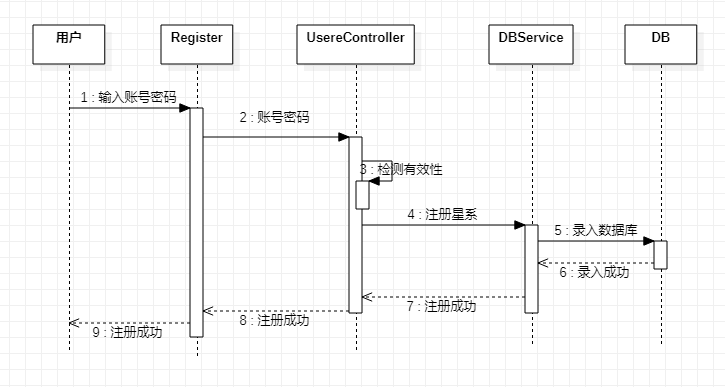
输入账号和密码，当账号已被注册，且密码正确时，登录成功。登录后的身份为普通用户身份或管理员身份，具体由注册时决定。账号未注册、密码错误时，均有相应提示。时序图如图：



图【8】登录时序图

4.4.1.2 注册

选择注册身份，若选择注册普通用户账户，则输入账号和密码、确认密码，若选择注册管理员账户，则还需要额外输入口令，口令的具体内容由开发该系统的人员决定，使用该系统的人员需要询问开发者得知口令。账号和密码均为自定义，当账号已被注册过、账号长度不符合规范、密码长度不符合规范、两次密码不一致、口令错误时均会有相应提示。时序图如图：

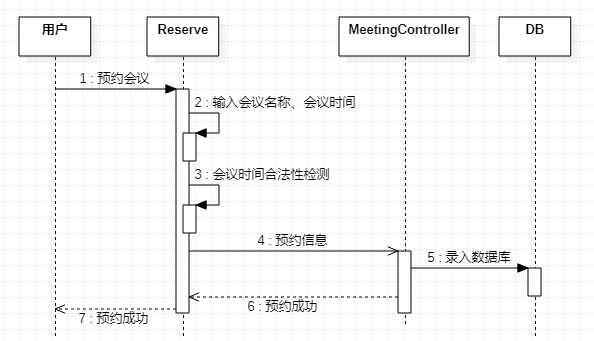


图【9】注册时序图

4.4.2 会议预定模块

4.4.2.1 预约会议

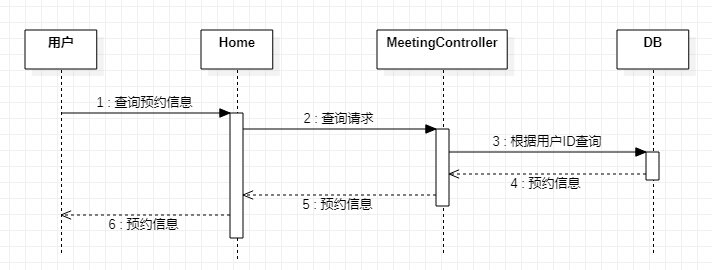
用户登陆后，可以点击预约会议按钮，填写会议名称、会议时间、会议预计时长进行预，会议的预约信息会录入数据库并展示在软件主页。时序图如图：



图【10】预约会议时序图

4.4.2.2 查看预约会议

用户可以在主页查看自己预约的会议。时序图如图：

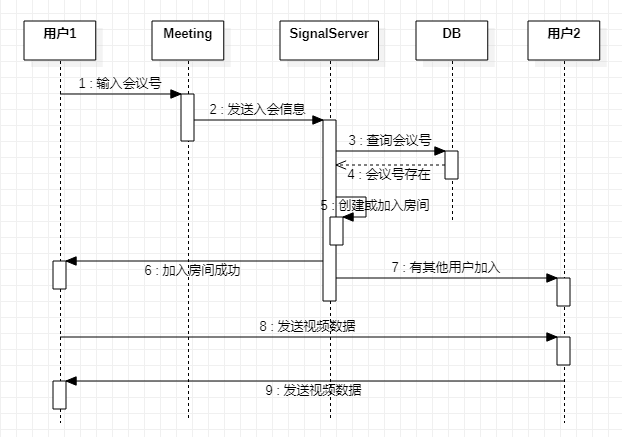


图【11】查看预约会议时序图

4.4.3 视频通话模块

4.4.3.1 视频通话

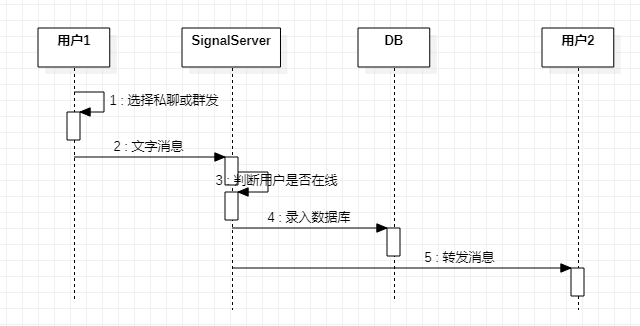
用户根据会议号加入会议，选择开启视频以及音频功能后，用户加入会议的消息会发送给后端的信令服务器，信令服务器判断是否会议已预约，若预约，则再判断该会议房间是否已创建，没创建则会创建。根据房间获得其他用户ip，之后用户将自己的视频数据传输给房价内所有人。时序图如图：



图【16】视频通话时序图

4.4.3.2 文字聊天

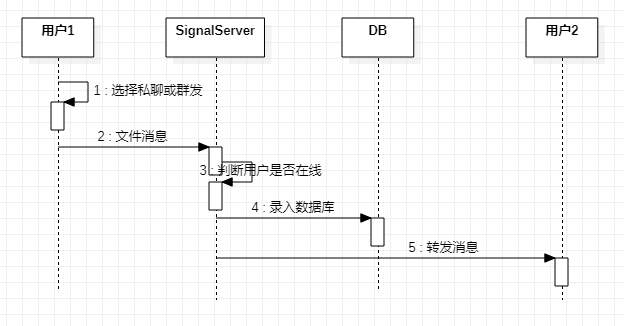
用户在会议内，可以发送文字消息。有两种选择，发送给全体成员或者私聊某人。时序图如图：



图【17】文字聊天时序图

4.4.3.3 发送文件

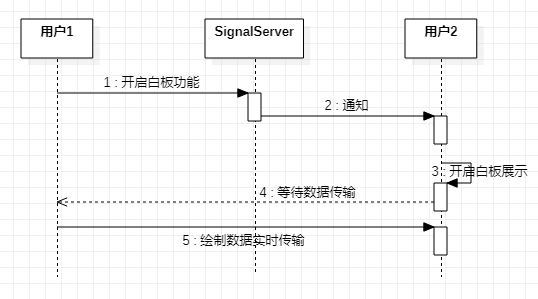
用户在会议内，可以上传文件。有两种选择，发送给全体成员或者传递给某人。时序图如图：



图【18】传递文件时序图

4.4.3.4 白板演示

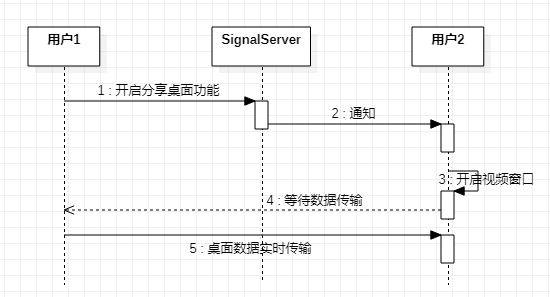
用户在会议室中，可以点击开启白板演示按钮，此时用户端会产生一个供用户绘制的绘画板，会议内其他用户会在视频窗口产生一个不可绘制的白板，用于实时展示绘制内容。时序图如图：



图【19】白板演示时序图

4.4.3.5 分享桌面

用户可同时开启视频以及桌面分享功能，打开桌面分享后，桌面的实时数据也会以视频流的形式传递给会议内其他用户。时序图如图：

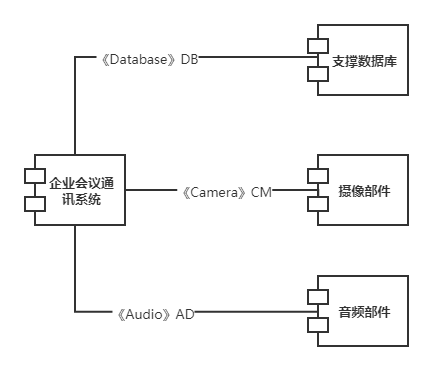


图【20】分享桌面时序图

4.5 接口设计

4.5.1 接口标识和接口图

外部接口示意图如下图所示：



图【30】接口图

对应的接口表如下表所示：

表【7】外部接口表

| 序号 | 接口标识 | 接口名称 | 接口类型 | 接口描述 |
| --- | --- | --- | --- | --- |
|  | 《Database》 DB | 与数据库的接口 | 数据库接口 | 连接数据库，提供用户信息验证 |
|  | 《Camera》CM | 与摄像部件的接口 | API接口 | 连接摄像部件，得到面部信息 |
|  | 《Audio》AD | 音频部件接口 | API接口 | 连接摄像部件，得到声音信息 |

4.5.2 信息接口

信息接口表如下表所示：

表【8】信息接口表

| 接口标识 | 接口名称 | 信源 | 信宿 | 接口内容 | 信息使用说明 |
| --- | --- | --- | --- | --- | --- |
| I Vehicle WB | 机动车系统接口 | 机动车系统 | 驾驶员违规行为检测系统 | 车辆行驶中的信息参数及道路限速等 | 用于针对超速等违反道路限制的违规行为检测 |

4.5.3 API接口：

API接口表如下表所示：

表【9】API接口表

| 序号 | 接口标识 | 接口名称 | 实现方 | 依赖方 | 功能 |
| --- | --- | --- | --- | --- | --- |
|  | I E-A WB2 | 视频传递API接口 | WebRTC等开源软件 | 本系统 | 对实时采集的浏览器摄像头视频流进行p2p传递 |

5 用户界面设计

5.1应当遵循的界面设计规范

1. **易用性**

按钮名称应该易懂，用词准确，摒弃模棱两可的字眼，要与同一界面上的其他按钮易于区分，能望文知意最好。理想的情况是用户不用查阅帮助就能知道该界面的功能并进行相关的正确操作。易用性细则：

1. 完成相同或相近功能的按钮，常用按钮要支持快捷方式。
2. 完成同一功能或任务的元素放在集中位置，减少鼠标移动的距离。
3. 按功能将界面划分局域块。并要有功能说明或标题。
4. Tab键的顺序与控件排列顺序要一致，目前流行总体从上到下，同时行间从左到右的方式。
5. 复选框和选项框按选择几率的高底而先后排列。
6. 界面空间较小时使用下拉框而不用选项框。
7. 选项数较少时使用选项框，相反使用下拉列表框。
8. 专业性强的软件要使用相关的专业术语，通用性界面则提倡使用通用性词眼。
9. **规范性**

规范性细则：

1. 状态条要能显示用户切实需要的信息，常用的有： 目前的操作、系统状态、用户位置、用户信息、提示信息、错误信息等，如果某一操作需要的时间较长，还应该显示进度条和进程提示。
2. 状态条的高度以放置五号字为宜，滚动条的宽度比状态条的略窄。  
    **3. 合理性**

屏幕对角线相交的位置是用户直视的地方，正上方四分之一处为易吸引用户注意力的位置，在放置窗体时要注意利用这两个位置。

合理性细则:

1. 多个子窗体弹出时应该依次向右下方偏移，以显示窗体出标题为宜。
2. 重要的命令按钮与使用较频繁的按钮要放在界面上注目的位置。
3. 错误使用容易引起界面退出或关闭的按钮不应该放在易点位置。横排开头或最后与竖排最后为易点位置。
4. 对可能造成数据无法恢复的操作必须提供确认信息,给用户放弃选择的机会。
5. 非法的输入或操作应有足够的提示说明。
6. 对运行过程中出现问题而引起错误的地方要有提示，让用户明白错误出处，避免形成无限期的等待。
7. 提示、警告、或错误说明应该清楚、明了、恰当。

**4.美观与协调性**

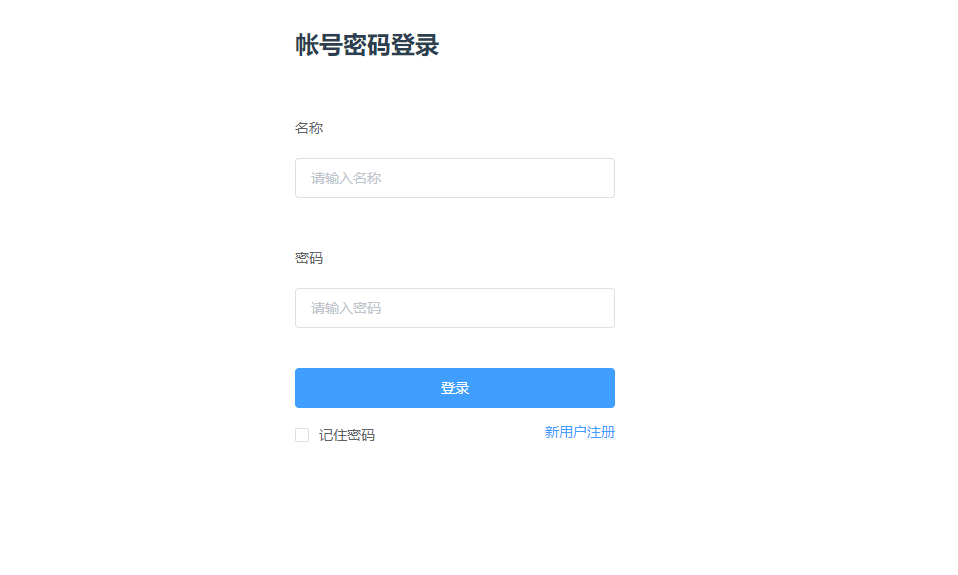
界面应该大小适合美学观点，感觉协调舒适，能在有效的范围内吸引用户的注意力。

美观与协调性细则：

1. 布局要合理,不宜过于密集，也不能过于空旷，合理的利用空间。
2. 按钮大小基本相近，忌用太长的名称，免得占用过多的界面位置。
3. 按钮的大小要与界面的大小和空间要协调。
4. 避免空旷的界面上放置很大的按钮。
5. 放置完控件后界面不应有很大的空缺位置。
6. 字体的大小要与界面的大小比例协调, 通常使用的字体中宋体9-12较为美观，很少使用超过12号的字体。
7. 大型系统常用的主色有"#E1E1E1"、"#EFEFEF"、"#C0C0C0"等。
8. 界面风格要保持一致，字的大小、颜色、字体要相同，除非是需要艺术处理或有特殊要求的地方。
9. 如果窗体支持最小化和最大化或放大时，窗体上的控件也要随着窗体而缩放；切忌只放大窗体而忽略控件的缩放。
10. 对于含有按钮的界面一般不应该支持缩放，即右上角只有关闭功能。
11. 通常父窗体支持缩放时，子窗体没有必要缩放。

5.2界面信息汇总

**1.登录**

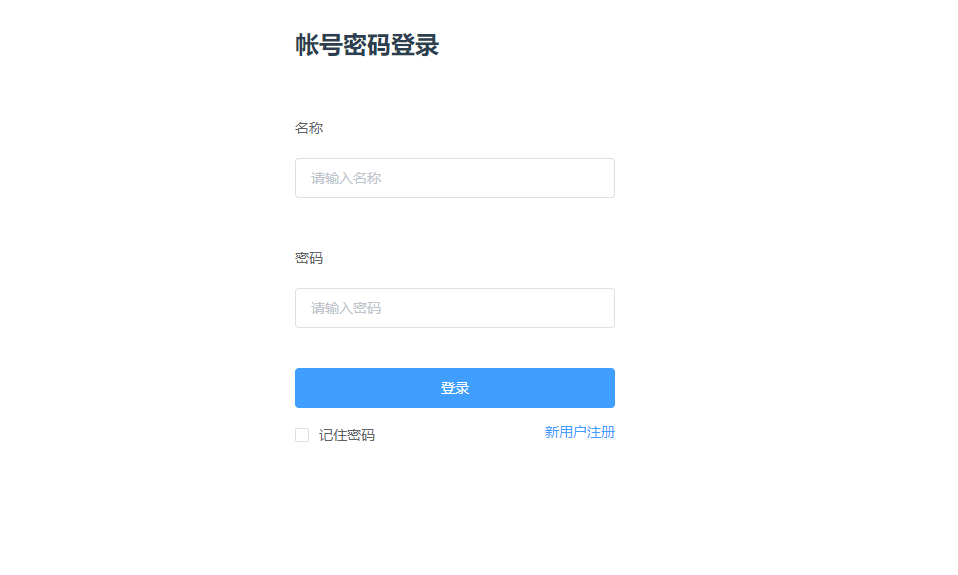


图【31】登录界面图

表【11】驾驶员违规驾驶行为检测软件登录界面汇总表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 标识号（登录主界面） | | | | |
| 序号 | 标识 | 名称 | 说明 | 备注 |
| 1 | 输入栏 | Username | 输入用户名 |  |
| 2 | 输入栏 | Password | 输入密码 |  |
| 3 | 跳转 | Register | 注册 | 跳转注册界面 |

**2.注册**

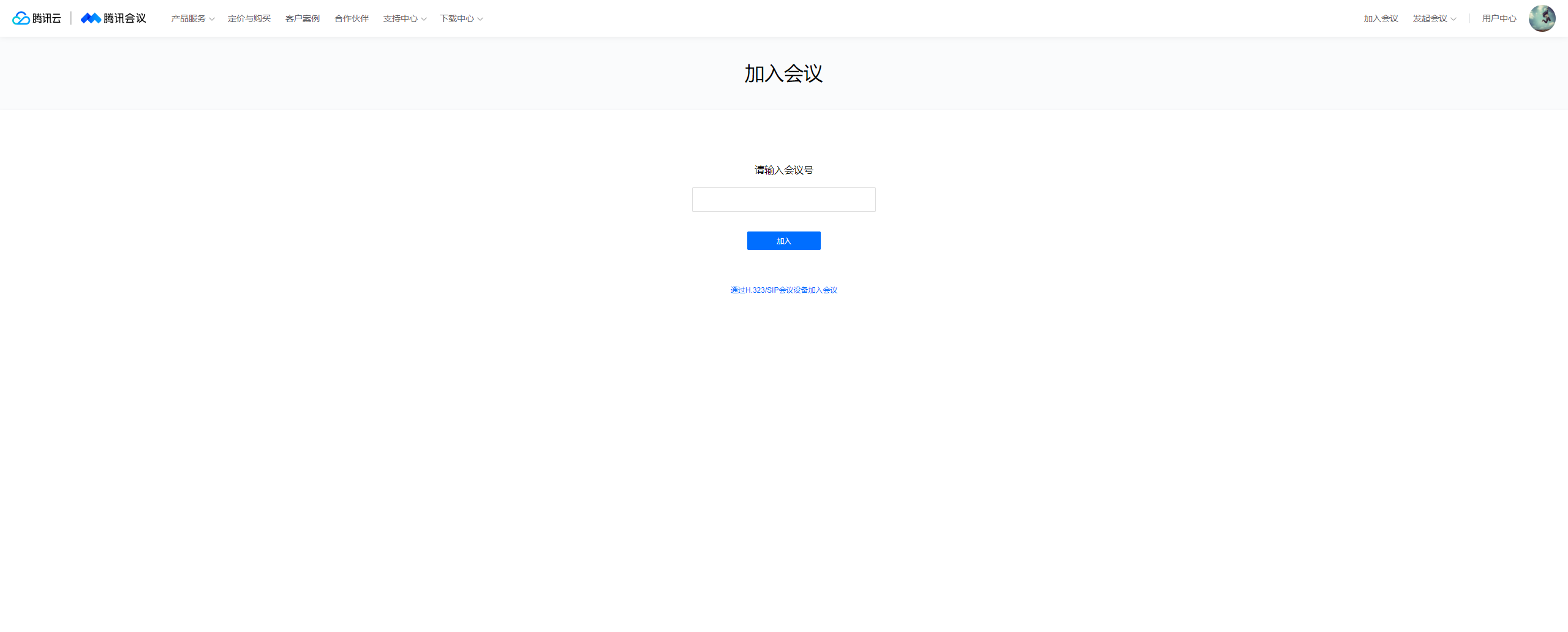


图【32】注册界面图

表【12】驾驶员违规驾驶行为检测软件注册界面汇总表

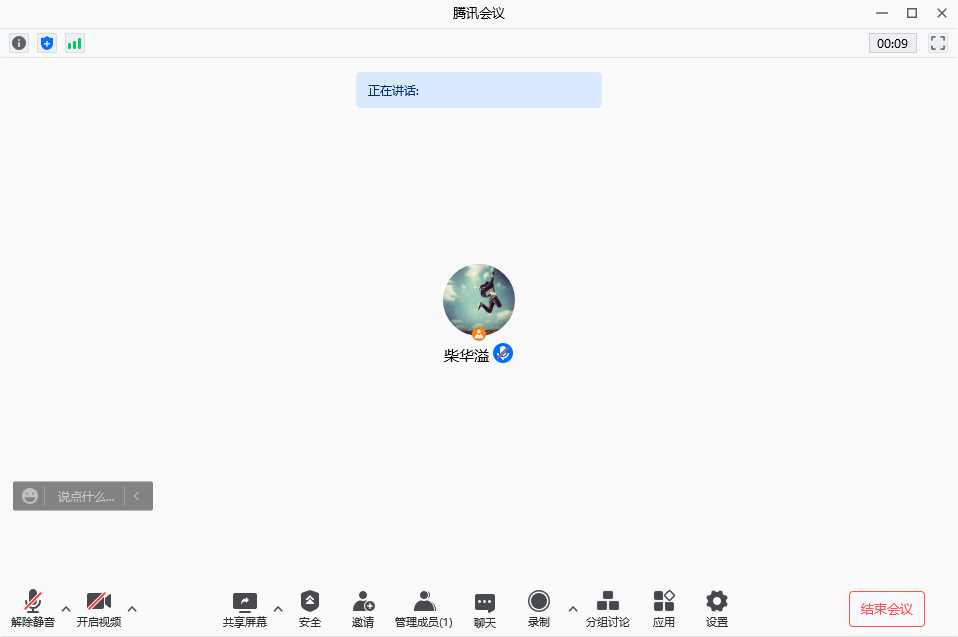
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 标识号（登录主界面） | | | | |
| 序号 | 标识 | 名称 | 说明 | 备注 |
| 1 | 输入栏 | Username | 输入用户名 |  |
| 2 | 输入栏 | Password | 输入密码 |  |
| 3 | 输入栏 | PasswordAgain | 再次输入密码 |  |
| 4 | 输入栏 | Command | 输入口令 | 可选 |
| 5 | 选择按钮 | property | 选择用户类型 |  |

**3.加入会议界面**



图【33】加入会议界面图

**5.会议界面**



图【33】会议界面图

6配置文件设计

数据库配置文件见文档：django连接mysql

7性能设计

性能设计非常重要，对于服务器端实时性来说系统性能的重要性不言而喻，对客户端软件来说性能好的软件也会获得良好的用户体验，从而给用户留下高质量软件的良好印象。因此在进行架构设计中性能设计非常重要。

单就性能设计来说，在架构设计初期就一定要把系统性能考虑在内，否则等开发完成以后测试发现性能不好就比较难办，通常要花费较长的时间来诊断性能瓶颈，找到提升的办法，甚至要改变架构，伤筋动骨，往往造成项目延期。所以性能设计首先要有明确的性能目标，根据用户和软件本身的性能要求来设计，合适的就是最好的。其次，要有适当的度量标准和量化的性能指标。最后，要有相应的设计策略，具体的测试方法。

7.1缓存性能：

在数据层和应用层之间增加数据缓存层，提供全局数据服务。可以大大减少数据库往返次数，可以充分利用大内存，而共享内存更能实现数据并发访问。

7.2多线程性能：

多线程对单CPU系统还只是简单的顺序利用CPU时间和改善用户体验，多CPU系统才是真正的并行。要注意的是多线程不要争抢访问同一资源而导致部分串行操作，要做到真正的并行操作多线程并不容易。另外，在多线程间同步一个庞大的资源，过多创建线程又没有实现线程池也会导致系统性能下降。

7.3数据库优化：

* 标识潜在的索引，但不要创建过多的索引。
* 如果使用 SQL Server，则使用 SQL Server 的事件探查器和索引优化向导。
* 监视处理器的使用；理想范围是：75-80% 处理器时间。
* 使用查询分析器分析查询计划以优化查询。
* 使用存储过程优化性能。
* 标准化写入的大量数据，即写入较少的数据。
* 取消标准化读取的大量数据，即读取较少的数据。

7.4代码性能：

在编程实现上，代码性能设计也很重要，一些昂贵的操作会占用大量的资源和CPU时间。因此要是急用更加快速、节约的算法与数据结构。

8系统出错处理设计

8.1出错信息

用一览表的方式说明每种可能的出错或故障情况出现时，系统输出信息的形式、含意及处理方法。应充分考虑不同层、软件单元/软部件和数据库中的代码，做到：

1. 出错信息描述简明扼要；
2. 通过出错信息可以定位错误源；

表【20】出错信息表

|  |  |  |
| --- | --- | --- |
| **错误类型** | **子项** | **错误原因** |
| 数据库错误 | 连接 | 连接超时 |
| 连接断开 |
| 数据库本身 | 数据库代码错误 |
| 数据库溢出 |
| 系统部分自定义错误 | 权限错误 | 管理员权限设置故障 |
| 输入错误 | 用户ID错误/为空 |
| 用户密码错误/为空 |
| 验证码错误/为空 |
| 链接错误 | 查找错误 | 未找到符合要求的记录 |
| 内部链接错误 | 网页链接错误 |
| 外部连接错误 | 文件/图片链接错误 |
| 页面休整而无法打开 |

8.2补救措施

故障出现后可能采取的变通措施，包括：

1. 周期性地把磁盘信息记录到磁带等存储设备；
2. 降效技术，也就是使用另一个效率稍低的系统或方法来求得所需结果的某些部分。一个自动系统的降效技术可以是手工操作和数据的人工记录；
3. 通过最后接收到的信息用以确认故障点，并根据装有备份的存储设备来恢复及再启动技术，使软件从故障点恢复执行或使软件从头开始重新运行。

表【21】补救措施表

|  |  |
| --- | --- |
| **错误** | **解决方法** |
| 连接超时 | 重新连接或检查网络等硬性情况 |
| 连接断开 | 重新连接 |
| 数据库代码错误 | 改正数据库代码 |
| 数据库溢出 | 扩展数据库 |
| 管理员权限设置故障 | 从后端重新设置权限 |
| 用户ID错误/为空 | 提示输入不正确，检查信息是否正确，并返回 |
| 用户密码错误/为空 | 提示输入不正确，检查信息是否正确，并返回 |
| 验证码错误/为空 | 提示输入不正确，检查信息是否正确，并返回 |
| 未找到符合要求的记录 | 检查是否记录被删除，是，则撤销删除；否，则无符合记录 |
| 网页链接错误 | 改为正确的链接 |
| 文件/图片链接错误 | 改为正确的链接 |
| 页面休整而无法打开 | 提示该网页暂时无法打开，并尽早完成休整 |

8.3系统统维护设计

没什么系统是一劳永逸的，在运行过程中，还有大量运行管理、系统维护、、系统设置等工作要做。系统维护包括程序的维护、数据文件的维护、代码的维护等。而系统的修改，往往会造成系统程序、文件、代码的修改，因此，系统的修改必须经过严密的审核。

系统维护设计主要是了系统运行与维护的方便而在程序内部设计中作出的安排，包括在程序中专门安排用于系统的检查与维护的检测点和专用模块。

8.4错误处理设计

系统在系统检查与维护方面，采用日志模块记录系统运行过程状态以及出现的问题。

系统外部接口调用前后都要进行日志的详细记录，方便接口调试；系统中重要的状态信息的变化都要通过日志记录下来，方便查问题时还原现场，推断程序运行过程；系统入口与出口记录输入与输出，方便定位问题；任何业务异常都进行日志详细记录；日志记录要求清晰准确，能够通过日志准确定位系统问题。

系统输出调试日志一定程度上会影响到系统运行效率，出于系统性能考虑，默认情况下，日志级别设为“ERROR”，调试信息不会输出。当需要输出调试信息时，开发人员手工修改系统配置文件中的日志级别为“DEBUG”，从而获取相关调试信息。

具体出错信息与补救措施详见11.2补救措施。

9需求可追踪性

需求跟踪是指跟踪一个需求使用期限的全过程，需求跟踪包括编制每个需求同系统元素之间的联系文档，这些元素包括其他类型的需求，体系结构，其他设计部件，源代码模块，测试，帮助文件等。需求跟踪为我们提供了由需求到产品实现整个过程范围的明确查阅的能力。

需求跟踪的目的是建立与维护“需求－设计－编程－测试”之间的一致性，确保所有的工作成果符合用户需求。

需求跟踪有两种方式：

（1）正向跟踪。检查《需求分析文档》中的每个需求是否都能在后继工作成果中找到对应点。

（2）逆向跟踪。检查设计文档、代码、测试用例等工作成果是否都能在《需求分析文档》中找到出处。

表【22】需求追踪表（正向）

| 软件概要设计说明 | | 软件需求规格说明 | |
| --- | --- | --- | --- |
| 包标识/名称 | 章条号 | 需求标识/名称 | 章条号 |
| 用户管理包 | 3.1.1 | 用户登录 | 3.4.1 |

表【23】需求追踪表（逆向）

| 软件需求规格说明 | | 软件概要设计说明 | |
| --- | --- | --- | --- |
| 需求标识/名称 | 章条号 | 包标识/名称 | 章条号 |
| 管理员身份验证 | 3.4.20 | 违规驾驶记录管理 | 3.1.4 |

10注释

10.1软件标识号

Alpha（α）：内部测试版，α是希腊字母的第一个，表示最早的版本，内部交流或者专业测试人员测试用，Bug较多，普通用户不要安装。

Beta（β）：公开测试版，β是希腊字母的第二个，这个版本比alpha版发布晚一些，专业爱好者或者忠诚用户大规模测试用，存在一些缺陷，该版本也不适合一般用户安装。

RC：Release Candidate（候选版本），也叫Gamma（γ），比较成熟的测试版，主要着重于除BUG，不会对软件做任何大的更改，功能和最终发布版一样，类似于发行版前的预览版。

Release：这个标识正式发行版本，一般是由RC版本而来。表示经过测试，正式发布的版本，属于可靠稳定的版本。

GA：General Availability，也就是一般可用版本，意思其实和Release的意思是一样的。都属于正式发布版本。也就意味着经过了比较严格的测试，可以放心使用。

Stable：稳定版。意思其实和GA意思一样，只是不同的标识而已。在开源软件中，都有stable版，这个就是开源软件的最终发行版，用户可以放心使用。

10.2 CSCI

CSCI是计算机软件配置项（Computer Software Configuration Item）简称，在软件设计文档中经常用到。

配置与配置项 在配置管理中，“配置”和“配置项”是重要的概念，“配置”是在技术文档中明确说明并最终组成软件产品的功能或物理属性。

因此“配置”包括了即将受控的所 有产品特性，其内容及相关文档，软件版本，变更文档，软件运行的支持数据，以及其他一切保证软件一致性的组成要素，相对与硬件类配置，软件产品的“配置” 包括更多的内容并具有易变性。

10.3 ODBC

开放数据库连接（Open Database Connectivity，ODBC）是为解决异构数据库间的数据共享而产生的，现已成为WOSA(The Windows Open System Architecture(Windows开放系统体系结构))的主要部分和基于Windows环境的一种数据库访问接口标准。ODBC 为异构数据库访问提供统一接口。用ODBC 可以访问各类计算机上的DB文件，甚至访问如Excel 表和ASCI I数据文件这类非数据库对象。

10.4 ISO

ISO标准是指由国际标准化组织(International Organization for Standardization)， ISO)制订的标准。 国际标准化组织是一个由国家标准化机构组成的世界范围的联合会，现有140个成员国。根据该组织章程，每一个国家只能有一个最有代表性的标准化团体作为其成员，原国家质量技术监督局以CSBTS名义国参加ISO活动。

10.5 OS

操作系统（operating system，简称OS）是管理计算机硬件与软件资源的计算机程序。操作系统需要处理如管理与配置内存、决定系统资源供需的优先次序、控制输入设备与输出设备、操作网络与管理文件系统等基本事务。操作系统也提供一个让用户与系统交互的操作界面。

10.6 DBSM与DB

数据库管理系统（Database Management System ），比如Oracle、SQL Server、DB2、MySQL等。

10.7 DBDD

数据库设计说明书(Database Design Description)。