**项目详细设计文档**

**团队名称：Anti-Gay-Fri 第26组**

**团队成员：PM:姜殊 成员：陈元杰 耿昱翔 金炜 莫其凡**

|  |  |  |
| --- | --- | --- |
| 变更时间 | 变更说明 | 变更人 |
| 2012/6/4 | **数据库方面相关文档编写** | **姜殊** |
|  |  |  |
|  |  |  |

1. 全局数据结构说明（全局数据结构说明，包括常量、变量和数据结构）

服务器端：

|  |  |
| --- | --- |
| 数据 | 数据结构 |
| Data.Name 连接信息 | public int x; 用于设定倒计时  public String n; 名字  public MeetingReceiverInterface rmi; |
| Data. OrderInfo 笔序信息 | public OrderRecieverInterface reciever;  public String name; 名字  public int remaintime; 用于设定倒计时 |
| Data.MeetingInfo 会议信息 | public int MeetingState=0;  0：会议未开始 1：会议正在进行  public int MeetingID=0;  public ArrayList<Integer> boardID; |
| Data.Meeting 会议信息（数据库） | private Date startTime=null;  private Date endTime=null;  private ArrayList<String> users =null;  private int id =-1; |
| SQLConnecter ：与数据库连接管理类 | **private** **final** String driverName ="com.mysql.jdbc.Driver";  **private** **final** String database ="seproject";  **private** String userName ;  **private** String password ;  **private** String host = "localhost";  **private** Connection connection |
|  |  |

客户端

|  |  |
| --- | --- |
| 数据 | 数据结构 |
| Data.Name | public int x;  public String n; |
| Data.OrderInfo | public OrderRecieverInterface reciever;  public String name;  public int remaintime; |
| Data.MeetingInfo | public int MeetingState=0;  public int MeetingID=0;  public ArrayList<Integer> boardID; |
| Logic.AudioInfo 语音信息 | public static AudioFormat getAudioFormat(){  float sampleRate=44100.0F;  int sampleSize=8;  int channels=1;  boolean bigEndian=false;  return new AudioFormat(  AudioFormat.Encoding.PCM\_SIGNED,  sampleRate,  sampleSize,  channels,  (sampleSize/8)\*channels,  sampleRate,bigEndian);  }    public static int cTIMES=5;  public static int cD=30000;  public static int cData=3000; |

1. 模块设计（包括用例图、功能设计说明）

1.登录

Client.view.Logging

rmi.server.interface.LoggingService{

public int checkPassword(String id,String password)

}

Server.Server.LoggingServer

客户端Logging通过LoggingService远程调用服务器上的LoggingServer中的方法，实现检验账号密码的作用。下文不再赘述。

2.会议管理

Client.view.MainFrame

Client.view.MeetingReceiver

rmi.server.interface.MeetingControlService{

public static final int PORT=6667;

public static final String NAME="MeetingControlS";

public int getMeetingState();

public int getMeetingNo();

public void startMeeting(String name);

public int stopMeeting(String name);

public int addABoard(String name,int boardID);

public ArrayList<Integer> getBoardList();

public void addBoardID(int boardID);

public void deleteBoardID(int boardID);

public int getMeetingID() ;

}

rmi.server.interface.MemberControlInterface{

public static final int PORT=6668;

public static final String NAME="MemberControlS";

public void refreshMember(String name, MeetingReceiverInterface mri);

public ArrayList<String> getIDs();

}

rmi.client.interface.MeetingReciever{

public int ifAllow(String name) throws RemoteException;

}

Server.Server.meetingControl

Server.Server.memberControl

3.笔序模式

Client.view.MainFrame

rmi.client.interfaces.OrderRecieverInterface{

public int ifAllow(String name);

}

rmi.server.interfaces. OrderControlInterface{

public static final int PORT=6669;

public static final String NAME="OrderControlS";

public void join(OrderRecieverInterface ori,String name);

public void refresh(String name);

public boolean cutIn(OrderRecieverInterface ori,String name);

public void exit(String name);

public ArrayList<String> getOrder();

}

Client.logic.OrderClient

Client.logic.OrderChoose

Server.Server.OrderControl

5.语音交流

Client.logic.Record

Client.logic.PlayBack

Client.logic.VoiceClient

Data.AudioInfo

Server.voiceServer.voiceServer

Server.vioceSercer.Mixer2

6.白板

Board.paintBoard{

MeetingControlService meetingcontrol; //客户端可调用的远程服务器对象

PaintSender paintSender;//客户端可调用的远程服务器对象

PaintReceiver paintReceiver;//服务器可调用的远程客户端对象

Public void setAccess();调整笔序

Public void onStartMeeting();会议开始白板初始化

Public void onStopMeeting();会议结束白板调整

sendAllImgOnStopMeeting();会议结束保存所有白板图片

private void setLineWidth();调整画笔粗细

}

通过会议管理对象meetingcontrol进行会议管理与白板对象之间的交互，paintSender和paintReceiver分别是远程服务器和远程客户端对象，用于白板客户端之间的通信

1. 接口设计

数据层接口

|  |  |  |
| --- | --- | --- |
| 接口ID | 接口名（接口层） | 接口方法及描述 |
| I1 | **Sql.interface**  BoardElementDataInterface  白板管理接口、增删白板、增删白板上数据元、获取白板列表、获取白板上数据元列表 | /\*\*  \* **@param** meetingID  \* **@param** boardID 数据库中记录白板上的某个图元，  \* **@param** username  \* **@param** o  \* **@return** 返回数据库生成的图元ID  \*/  **abstract** **public** **int** writeNewImgeData(**int** meetingID,**int** boardID,String username,Object o);  /\*\*  \* **@param** meetingID  \* **@param** username  \* **@param** boardID 插入在数据库中插入新的白板，白板ID由程序生成  \* **@return**  \*/  **abstract** **public** **int** newBoard(**int** meetingID, String username,**int** boardID);  /\*\*  \* **@param** elementsID  \* **@return** 根据ID读取指定数据元  \*/  **abstract** **public** Object readImageData(**int** elementsID);  /\*\*  \* **@param** boardID  \* **@return** 根据白板ID获得白板上最新的数据元ID  \*/  **abstract** **public** **int** getNewestID(**int** boardID);  /\*\*  \* **@param** boardID  \* **@param** elementsID  \* **@return** 根据白板ID，数据元ID，获得指定白板上需要更新的数据元ID列表  \*/  **abstract** **public** ArrayList<Integer> getUpdateList(**int** boardID,**int** elementsID);  /\*\*  \* **@param** boardID  \* **@param** elementsID  \* **@return** 根据白板ID数据元ID删除指定白板上的一个数据元  \*/  **abstract** **public** **boolean** deleteImageData(**int** boardID,**int** elementsID);  /\*\*  \* **@param** meetingID  \* **@return** 返回某次会议的所有白板列表  \*/  **abstract** **public** ArrayList<Integer> getBoardIDList(**int** meetingID);  /\*\*  \* **@param** boardID  \* **@return** 删除白板存在时返回true,不存在返回false  \*/  **abstract** **public** **boolean** deleteBoard(**int** boardID); |
| I2 | **Sql.interface**  FileNetInterface  文件下载接口 | /\*\*  \* **@param** fileID  \* **@return** 获取指定文件的下载流  \*/  **public** **abstract** InputStream downloadFile(**int** fileID);  /\*\*  \* 关闭流  \*/  **public** **abstract** **void** closeStream ();  /\*\*  \* **@param** fileID  \* **@return** 获得指定文件的长度  \*/  **public** **abstract** **int** getFileLength(**int** fileID);    /\*\*  \* **@return** 返回数据库所有文件的信息列表  \*/  **public** ArrayList<String> getFilesList(); |
| I3 | **Sql.interface**  FileUploadInterface  文件上传接口 | /\*\*  \* **@param** filename  \* **@param** Username  \* **@param** meetingID  \* **@return** 从服务器端上传指定路径的文件至数据库  \*/  **public** **int** uploadFile(String filename,String Username,**int** meetingID); |
| I4 | **Sql.interface**  MeetingDataInterface  会议记录管理接口  增加会议、结束会议、验证会议是否结束、获取最新会议ID | /\*\*  \* **@param** startTime  \* **@return会议ID**  \*/  **public** **abstract** **int** insertMeeting(String startTime);  /\*\*  \* **@param** meetingID  \* **@param** username参会用户  \* **@return** 未插入成功则返回-1 已存在返回0 否则>0  \*/  **public** **abstract** **int** insertMeetingMember(**int** meetingID,String username);  /\*\*  \* **@param** ID会议ID  \* **@param** endTime  \* **@return** DML更新的行数。未成功返回-1  \*/  **public** **abstract** **int** updateMeeting(**int** ID,String endTime);  /\*\*此接口待确认  \* **@param** id  \* **@return**  \*/  **public** **abstract** String findMeetingByID(**int** id);  /\*\*  \* **@return** 获得最后一次的会议ID  \*/  **public** **abstract** **int** getLastMeetingID();  /\*\*  \* **@return** 检查最后一次会议是否正常结束，是则返回true  \*/  **public** **abstract** **boolean** checkLastMeeting(); |
| I5 | **Sql.interface**  UserIndentifyInterface  用户登录验证接口 | **public** **int** *NOTFOUND* =0;  **public** **int** *ERROR* =-1;  **public** **int** *CORRECT* =1;  **public** **int** checkUser(String username,String password); |
| I6 | **Sql.interface**  UsersDataInterface  用户管理接口、用户增删改 | **public** **int** *NOTFOUND*=0;  **public** **int** *EXISTED*=0;  **public** **int** *ERROR* = -1;  **public** **int** *CORRECT*=1;  **public** **abstract** **int** insertUser(String userName,String passWord);  **public** **abstract** **int** updateUser(String userName, String newPassWord);  **public** **abstract** String getUserPassWord(String userName);  **public** **abstract** **int** deleteUser(String userName); |

|  |  |  |
| --- | --- | --- |
| I7 | **Rmi.server.interfaces**  paintSender  白板发送图元信息 | **public** **static** **final** **int** *PORT* =8888;  **public** **static** **final** String *NAME* ="白板服务";  **public** **boolean** sendPaint(ShapeModel model,**int** boardID,**int** meetingID)**throws** RemoteException;  **public** **boolean** sendUndo(**int** boardID)**throws** RemoteException;  **public** **boolean** sendRedo(**int** boardID)**throws** RemoteException;  **public** **boolean** sendRedoModel(ShapeModel model,**int** boardID,**int** meetingID)**throws** RemoteException;  **public** **boolean** sendAllImg()**throws** RemoteException; |
| I8 | **Rmi.client.interfaces**  paintReceiver  白板接收图元信息 | **public** **void** receiveShapeModel(**int** boardID, ShapeModel model)**throws** RemoteException;  **public** **void** receiveUndo(**int** boardID)**throws** RemoteException;  **public** **void** receiveRedo(**int** boardID)**throws** RemoteException;  **public** **void** receiveNewBoard(**int** boardID)**throws** RemoteException;  **public** **void** receiveDeleteBoard(**int** boardID)**throws** RemoteException; |

文件共享接口

|  |  |
| --- | --- |
| 接口 | 接口作用 |
| FileShareService | **public** **int** *PORT*=8889;    **public** String *NAME*="FileShare";    **public** ArrayList<String>getFileList()**throws** RemoteException;    **public** **int** getFileLength(String fileName)**throws** RemoteException;    **public** **byte**[] download(String fileName, **int** start, **int** length)**throws** RemoteException;    **public** **boolean** isFileExist(String fileName)**throws** RemoteException;    **public** **void** upload(String fileName,**byte** b[])**throws** RemoteException;    **public** **void** completeUpload(String fileName,String userName)**throws** RemoteException; |
| MessageService | **public** **int** *PORT*=8887;    **public** String *NAME*="Chatting";    **public** **boolean** sendMessage(Message message)**throws** RemoteException;    **public** **boolean** addReciever(MessageReciever reciever)**throws** RemoteException; |
| ChatLogic | **Boolean** sendMessage(String message); |
| FileShareLogic | **public** ArrayList<String>getFileList() ;    **public** **boolean** downloadFile(String src,String des,ProcedureController procedure) ;    **public** **boolean** uploadloadFile(String fileName,ProcedureController procedure) ; |

1. 数据库设计

数据库名：seproject

数据库表总览：

+---------------------+

| Tables\_in\_seproject |

+---------------------+

| chatmessages |

| elements |

| files |

| meetings |

| users |

| whiteboards |

+---------------------+

表chatmessages:

+---------------------+

| Tables\_in\_seproject |

+---------------------+

| chatmessages |

| elements |

| files |

| meetingmembers |

| meetings |

| users |

| whiteboards |

+---------------------+

表：elements

+------------+-------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+------------+-------------+------+-----+---------+----------------+

| ElementsID | int(11) | NO | PRI | NULL | auto\_increment |

| BoardID | int(11) | NO | | NULL | |

| Shapemodel | mediumblob | YES | | NULL | |

| Type | varchar(30) | NO | | unknown | |

| UserName | varchar(30) | NO | | NULL | |

+------------+-------------+------+-----+---------+----------------+

表files

create table Files(

FileID int not null auto\_increment,

MeetingID int not null,

FileName varchar(60) not null,

FileData MediumBlob not null,

FileLength int not null,

ModifyTime DateTime not null,

LoaderUserName varchar(30) not null default 'system',

primary key(FileID)

);

表meetings

create table Meetings(

MeetingID int not null auto\_increment,

StartTime DateTime not null,

EndTime DateTime default null,

primary key(MeetingID)

);

表users

create table Users(

UserName varchar(30) not null,

Password varchar(16) not null,

primary key(UserName)

);

表whiteboards

create table WhiteBoards(

BoardID int not null ,

MeetingID int not null,

primary key(BoardID,MeetingID)

);

1. 系统性能设计

用户上线在1s内反应

用户下线或断线在5s内反应

进入笔序在1s内反应

离开笔序在1s内反应

突然断线，笔序在5s内反应

结束会议，申请插队，无响应保留时间为20s

语音根据网络流畅程度，延迟在1s至2s，网络不通畅可能导致卡音