第一次会议记录：

首先，我们确定整个桌面监控系统分为主控端和被监控端，即服务器和客户端。而两者之间的通讯，采用Java 语言来实现，在主控端机器上采集事件(一般是键盘和鼠标的事件)，然后封装成消息类传输到被监控端。在被监控端上重演主控端的动作事件。并在被监控端运行client端，接收消息，如果主控端有请求操作的消息，用Robot截下当前屏幕，传给主控端，主控端显示被监控端的屏幕，是一个位图；然后接收在这个位图上的鼠标事件和键盘事件，并把鼠标位置(位图上的坐标换算成对应的屏幕上的坐标)和键值送到被监控端上，在被监控端上重演同样的事件。

计划中的整个系统运行概述如下：

1．启动被监控端，打开指定的UDP端口号。用于读取命令。

2．被监控端读取命令（命令格式为ordername:port）ordername为命令名字，port为主控端打开的TCP端口。

3．接到主控端连接后，被监控端就对当前用户的桌面采用屏幕截取，然后发送给主控端。依被监控端设计的不同，可以设定屏幕截取的时间间隔，时间间隔短一点就可以获得连续屏幕变化了。

4．主控端在画布上对鼠标、键盘事件进行监听，被监控端重演主控端上的事件

5．主控端和被监控端读取和发送数据，分别来实现文件上传和下载。

1. 在被监控端实现DOS命令的执行。

第二次会议记录：

这次我们讨论了系统运行的几个主要流程。

1.主控端启动流程

在主控端安装服务器程序，主控端首先启动其控制窗口类，初始化图形在主控端安装服务器程序，主控端首先启动其控制窗口类，初始化图形显示，

然后用户输入被监控端的IP地址。接着主控端开启任意TCP端口，并向该IP配置的UDP端口发送初始化命令（命令格式为ordername:port），ordername为命令名 字，port为主控端打开的TCP端口号。主控端对开启的TCP端口进行监听，如果超时，连接失败。否则，主控端读取被监控端的状态对象，接着启动控制模块和屏幕监视模块。

2.被监控段启动流程

在被监控端安装客户端程序，程序启动的时候，尝试开启UDP端口，如果防火墙或者杀毒软件阻止，尝试开启配置的其它UDP端口，如果都尝试完毕，启动失败。否则，将开启的UDP套接字传入，然后开启命令接收线程，目的是进行命令获取，并对获取的命令进行处理。然后将自己复制到自启动项，并改名为Explorer.jar.。

3.命令接收和处理流程

被监控端UDP端口启动成功后，就开始对该UDP端口进行监听，如果监听到有命令，就读取它，并判断命令是否有效，命令以ordername:port为格式，如果无效，返回继续读取命令，否则，调用相应的命令处理模块对命令进行处理，命令处理完后将处理结果返回给主控端。

4.图形监视线程启动流程

主控端开启任意TCP端口，向被监控端的UDP端口发送“要求被监控端连接”的UDP命令，命令形式为screen：TCP Port。如果命令超时，命令执行失败，否则，如果被监视端来连接，生成Soket对象，从Soket中读取被监视端发送过来的图形对象，并将图象显示在画布上，每隔一定时间（若干秒），重新读取图象，更 新画布显示，从而实现连续获得对方屏幕变化。

5.图形监视线程启动流程

主控端开启任意TCP端口，向被监控端的UDP端口发送“要求被监控端连接”的UDP命令，命令形式为screen：TCP Port。如果命令超时，命令执行失败，否则，如果被监视端来连接，生成Soket对象，从Soket中读取被监视端发送过来的图形对象，并将图象显示在画布上，每隔一定时间（若干秒），重新读取图象，更 新画布显示，从而实现连续获得对方屏幕变化。

最后将任务分发给小组各成员。

需求分析；刘小强

需求功能图以及优先级；陈树聪

会议记录者；范畅

登录页面代码设计；钟嘉扬