

## Executive Summary:

The client, who is a healthcare systems provider, wanted to build an application which would enable patients to set up video conferencing appointments with physicians/system users. The need was to allow a user to send an invite for a video conference to others and all those who accepted the invite to be able to meet up in a group video conferencing setup. The idea was to eliminate locational constraints to the extent possible in communication. We enabled the video-logs feature to allow storing of data for future references if needed. The client wanted to optimize the application performance and reduce the network bandwidth usage. The development team at Mindfire implemented peer connectivity and performed several load time optimizations to achieve it.

## About our Client:

**Client name:** Confidential | **Location:** US | **Industry:** Healthcare

## Technologies:

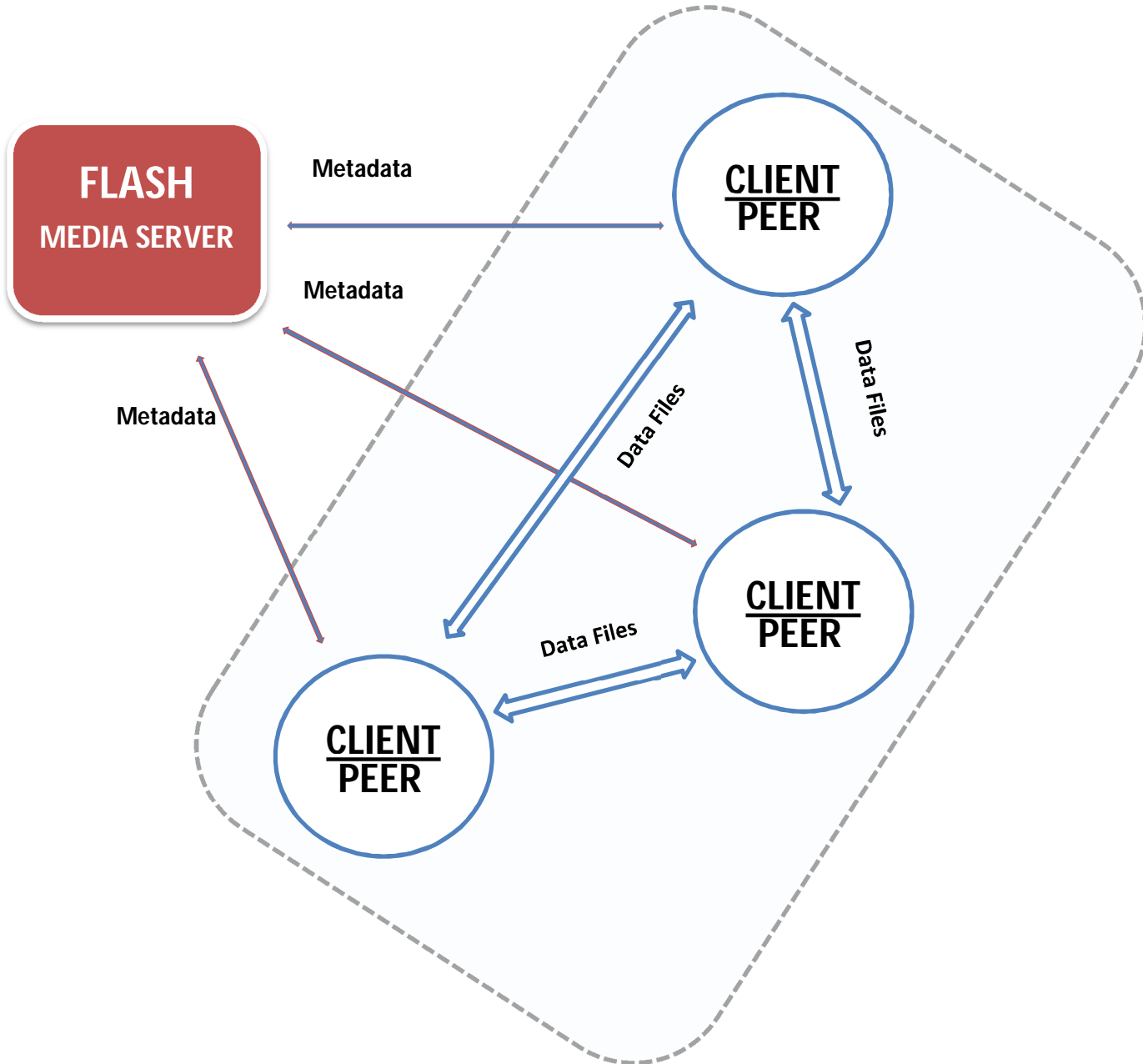
Flex 4.5, Action script 3.0, FMS 4, PHP, JavaScript, XML, MySQL, SVN

## Solution:

We designed and implemented a web based video conferencing widget to be used as an extension to a health care application. The web application implements a link that allows users to schedule a meeting in advance and send an email with the invitation link to the patient/group of attendees. The attendees can then click on the link and join the conference at the scheduled time. The widget allows for recording of the conference sessions and presents users with the ability to get into group chats; besides allowing them to see each other and talk on an audio video interface. At the end of a conference session, the users can download their copy of the text chat and recorded video. Among the features we implemented were standard controls for video/audio including controls to mute/pause video stream and control the microphone and speaker volume. We also implemented the option to ping users by getting the widget on the machines to ring in order to get their attention. By way of consideration of bandwidth availability, we implemented peer-2-peer connections on demand in order to deal

with instances when recording of conference is not needed. Flash Media Server was used as the back end server and implemented meeting request synchronization and authentication for user widgets.

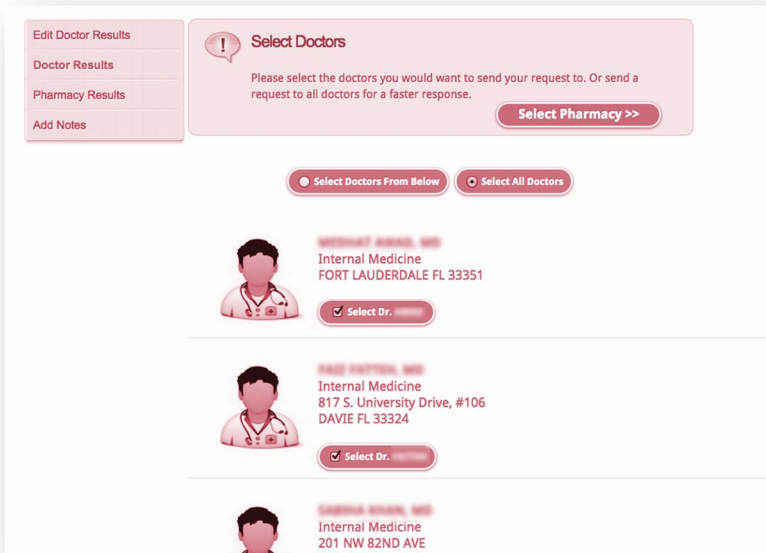
**Flow Diagram:**



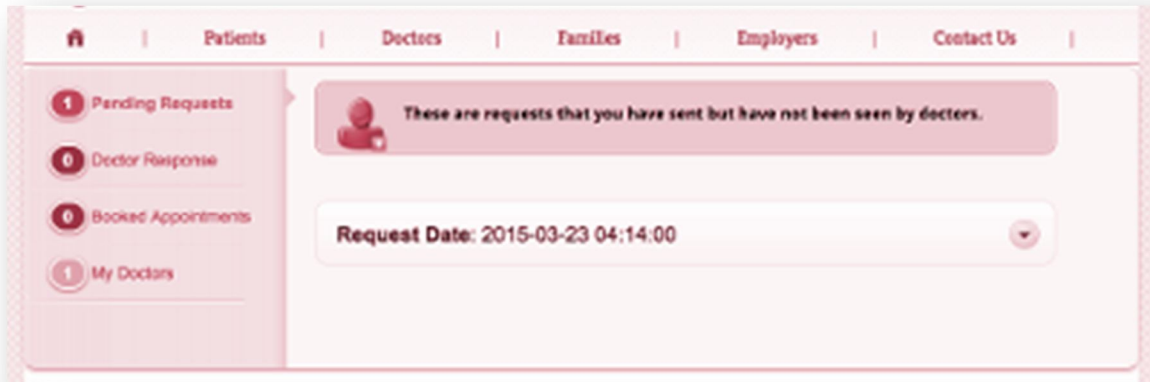
Screenshots:



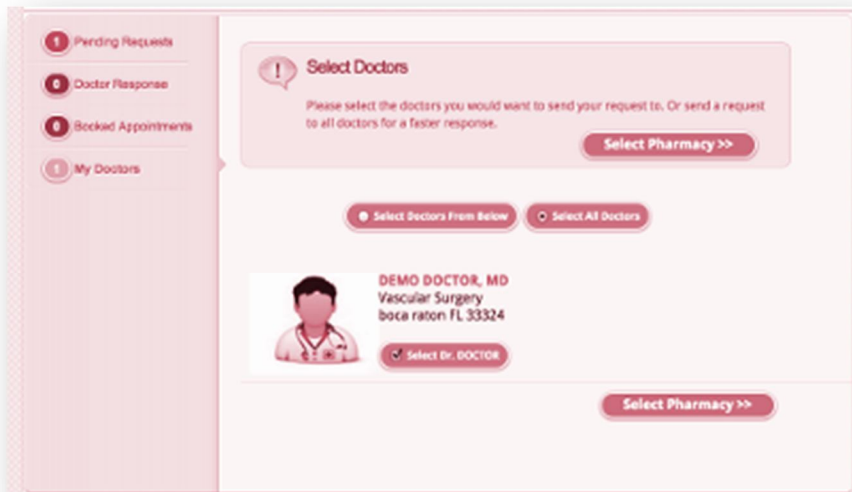
**Screen 1:** After logging in, the patient searches for a physician, based on proximity, by specialty and zip code. The system performs the search in the database by specialty and zip code. Physicians within a 10-mile radius get identified.



**Screen 2:** List of searched physicians. Patients can pick one and more physicians and send requests for a telemedicine session.



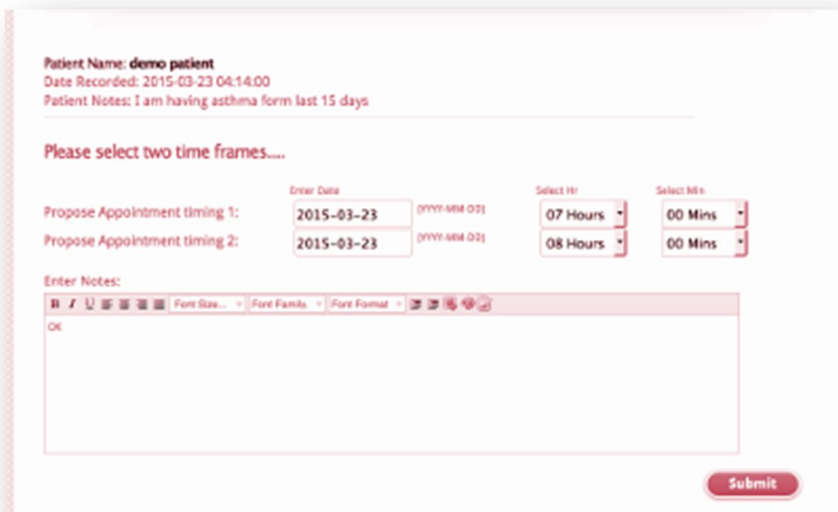
**Screen 3:** Lists all pending requests from patients which have been sent to physicians and have not been seen.



**Screen 4:** List of physicians the patient has sent requests for telemedicine sessions.



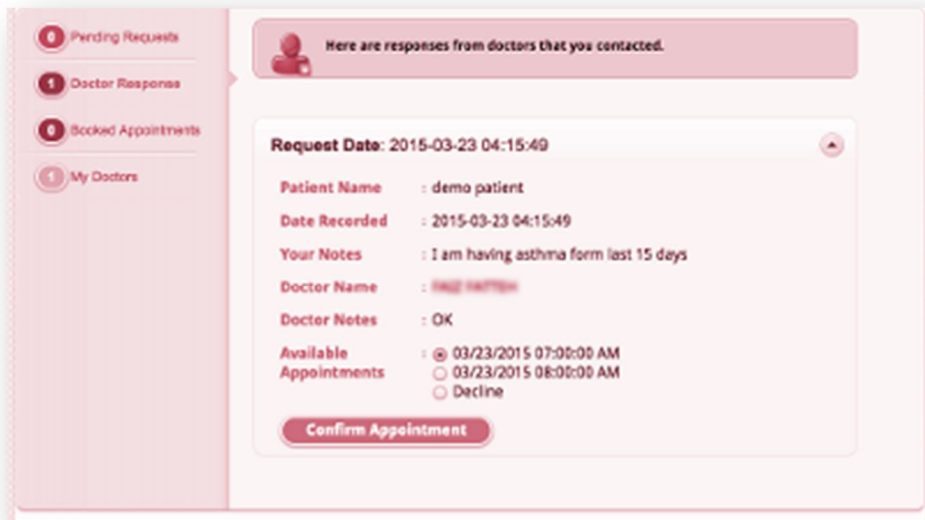
**Screen 5:** Physician account, from where a physician can see all telemedicine requests received from various patients.



**Screen 6:** Physician sends two availability time slots to the corresponding patient for scheduling the session. Patient has the option to choose the one which is more convenient.



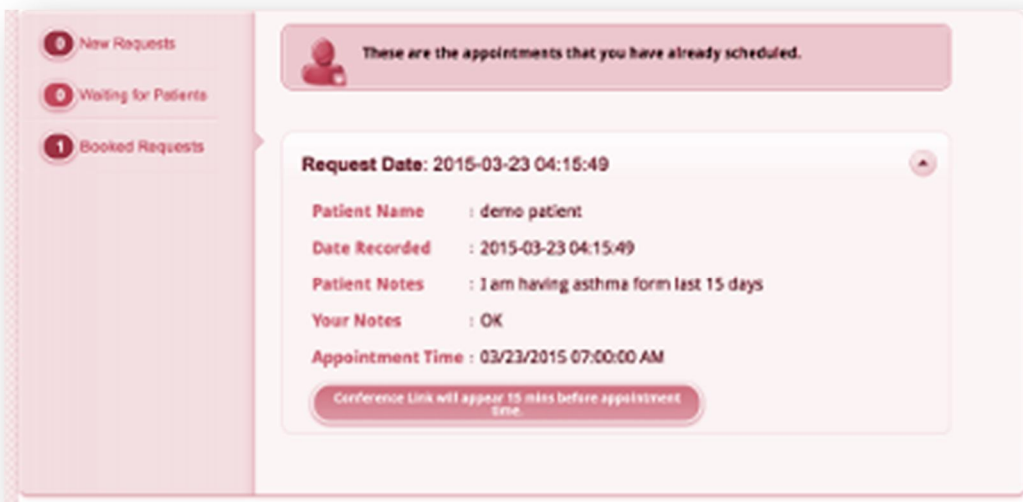
**Screen 7:** Physician can see all requests responded to and not yet confirmed by patients.



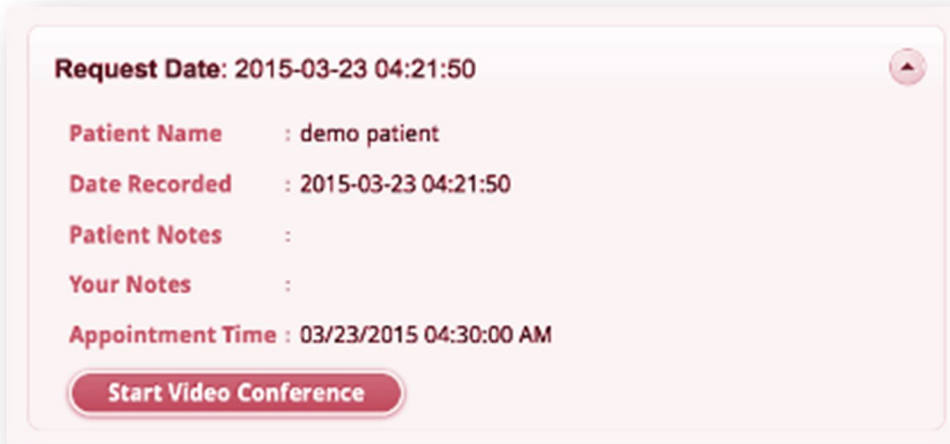
**Screen 8:** Patient seeing the response received from the physician for the requested telemedicine session. Based on availability, the patient can choose the session timing and confirm the same.



**Screen 9:** Patient can see all booked/confirmed sessions and can join a session from there only 15 minutes before the session start time.



**Screen 10:** Physician can see all the confirmed sessions from patients.



**Screen 11:** Session joining link activated 15 minutes before the scheduled start time



**Screen 12:** Physician and patient in an audio/video session can do text chat as well.