Med Connect

Project plan & 1st Inc by Group 17

Praveen Kumar Surapaneni 58

Sarath Chand Lingamgunta 43

Suramya Chukka 11

1. **Introduction:**

For the past few years we have been seeing a lot of companies adopting video conferencing as their primary medium of communication with their customers. The technological advancements done in the field of Video conferencing are more than enough to make the customers think that VC is better medium of communication, as it provides lot of benefits like zero time for travel, reduced cost for infrastructure. There are lot of Video Conferencing Applications like Skype, Hangouts which provide best possible way to video chat with others, but these doesn’t offer scheduling of the meetings, record the meeting, searching through the meetings and chatting with the customer. We want to create an Android/IOS app for medical field which can streamline the interactions between the doctor and patient by letting them video chat with each other.

1. **Project Goal and Objectives:**

**Overall Goal**

The purpose of this App is to let the doctor video chat with the patient, for doing so we have to implement a WebRTC which transfers the real-time video data across the systems. Two new REST services has to be built which will take care of uploading the data about the meeting in to user’s personal calendar using Google Calendar API. Another service has to be created to upload all the meeting videos in to Cloud databases (Amazon S3) which must be saved for further references. A new message Queue must be built to handle the chat between the patient and doctor. Implement push notifications to send remainders to the users about their appointments.

**Specific objectives**

Visiting a doctor can be huge task for a patient if he is unable to move, if the patient wants to know what doctor said during his previous visit, if the patient want to receive notifications about the Appointment, if the patient want a log of everything that happened between him and his personal doctor. To provide all these needs to the user the App must have the below functionalities.

* Provide the time slots of all the doctors that are open to the patients
* Send a notification to the doctor if he is booked for a particular time slot
* Create a meeting in the doctor and patient’s calendar
* Suggest the best doctors for a particular health condition
* Provide rating system for the patients to rate the doctors
* Record the meeting and save in the patient’s history to watch it later
* Provide a search engine to search for doctors and their videos
* Maintain the chat history between the doctor and his patient
* Provide users an easy access to their previous and upcoming appointments

We want to bundle all of these tasks in a single app which will help our app stand out and deliver best features to the users.

**Significance**:

While there are lot of high end applications available in the market that will let provide best video conferencing tools to the users, these aren’t on par with the business standards when comes to the scheduling and logging part. Right now if we want to schedule a meeting we have to use outlook calendar to send out the invites for the meeting and then open another video conferencing app for the actual meeting. There is some redundancy in doing this work and there is no certain log as there are two different applications. So there is lot of improvements that we can offer through or App, when it comes to scheduling and searching through the previous records.

1. **Project Background and Related Work**

As stated above there are lot of solutions available which will cost a lot to the customers, of them [Citrix GoToMeeting](https://www.citrix.com/products/gotomeeting/overview.html) is one of the best and most used video conferencing Application. Another popular video conferencing App is [Skype for Business](http://www.skype.com/en/business/%20) by Microsoft which has been giving a tuff competition to Citrix GoToMeeting, but Skype doesn’t support calendar integration natively and it has to depend on the third party app for doing so.

Major problem with these two is these are not custom made for medial field, and cannot be an effective solution in for patients consulting the doctors. These doesn’t offer an option to the doctor where he can prescribe the drugs for a particular user, and patients won’t be able to search for the doctors that are available in their area. We are planning to record the sessions with the doctor so that patient can review them whenever he has some queries related to their previous meeting. Skype doesn’t offer a solution for recording the meetings.

Calendar Integration is one of the major feature we are trying to implement in our project which will intelligent sync the users calendar with the recent appointments and remind them about their future appointments using push notifications. Skype relies on outlook for calendar integration and most normal users doesn’t have outlook, so we are planning to implement Google calendar which will be used by most of the average users.

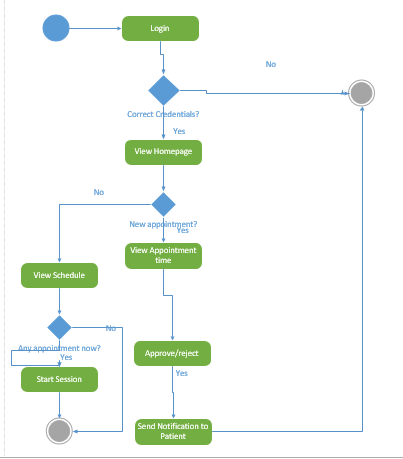
Cost effectiveness is primary thing, Citrix and skype both cost a lot for users. Skype is charging around $2 per month and Citrix is also charging the same for its users. We are planning to let the users use our Application for free which will help our App to reach wider audience.

1. **Proposed System:**

**Requirement Specifications:**

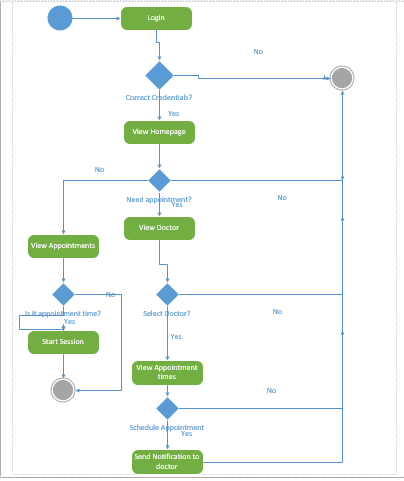
* All Patients shall create their own account in order to access this system.
* All doctors shall create their own account in order to access this system.
* A registered patient shall view all the doctors and their specializations in the field.
* Patient shall select a doctor.
* After selecting the doctor, he shall view the doctor’s schedule.
* A registered doctor shall prepare his own available time slots.
* Those time slots shall be viewed to the patient after selecting the doctor.
* Patient shall pick up the available time slot.
* Then a notification shall go to the Doctor.
* Doctor shall either accept or reject that notification.
* After the doctor’s action, an acknowledgement should go to the patient.
* Patient/Doctor shall view their dash board that shows their schedule.
* Send an email notification to both doctor and patient 1 hour before the schedule time.
* At the schedule time they should start a video session.
* Should provide a chat interface also.
* System should capture the conversation history.
* A registered user shall browse the captured conversation history

**Work Flow Analysis - UML Activity Diagram:**



**Doctor’s Activity Diagram**

We can describe the workflow from both patient and doctor’s perspective. In the system all the users (patients/doctors) can register their accounts. A patient can view his scheduled timing after logging into his account. If a there is a schedule on this time then he can start session with the doctor. A patient can also pick new appointment. By selecting view doctors button, he can view all the doctors and their available time slots who are in the system. After selecting a particular doctor he can view the doctor’s available time slots. Patient can pick a time slot and then a notification will be sent to doctor. Doctor can approve/reject the appointment. Then an acknowledgement will reach out to the patient



**Patient’s Activity Diagram**

**Technical Requirements:**

Our system will support android operating system. Here we are using Java as programming language. We are using WEBRTC as framework in order to make video calls.

For the database we are using Amazon Cloud Services to make our system more scalable. Amazon S3 for data storage and for the relational calculations we are using Amazon RDS. We will use XML as a medium for the request and response from client to web services.

We are planning to create our own service to store metadata information of the data (conversation) that is stored in Amazon S3 into Amazon RDS. By doing this we don’t any external trigger that will push the information of the data (conversation) into amazon RDS. We are planning to use calendar API to schedule appointments.

**Architectural requirements:**

We are following 3-tier architecture. In presentation tier we are using android framework. Most of the logic tier will be in Java and coming to the data tier we are using amazon cloud services like S3 and RDS. All the services we are using are REST, and as such we will confirm the Rest architectural style.

**Framework Specification:**

We are using WebRTC framework for audio and video conferencing. WebRTC is a free, open project that provides browsers and mobile applications with Real-Time Communications (RTC) capabilities via simple APIs. The WebRTC components have been optimized to best serve this purpose. WebRTC offers web application developers the ability to write rich, real-time multimedia applications (think video chat) on the web, without requiring plugins, downloads or installs. Its purpose is to help build a strong RTC platform that works across multiple web browsers, across multiple platforms.

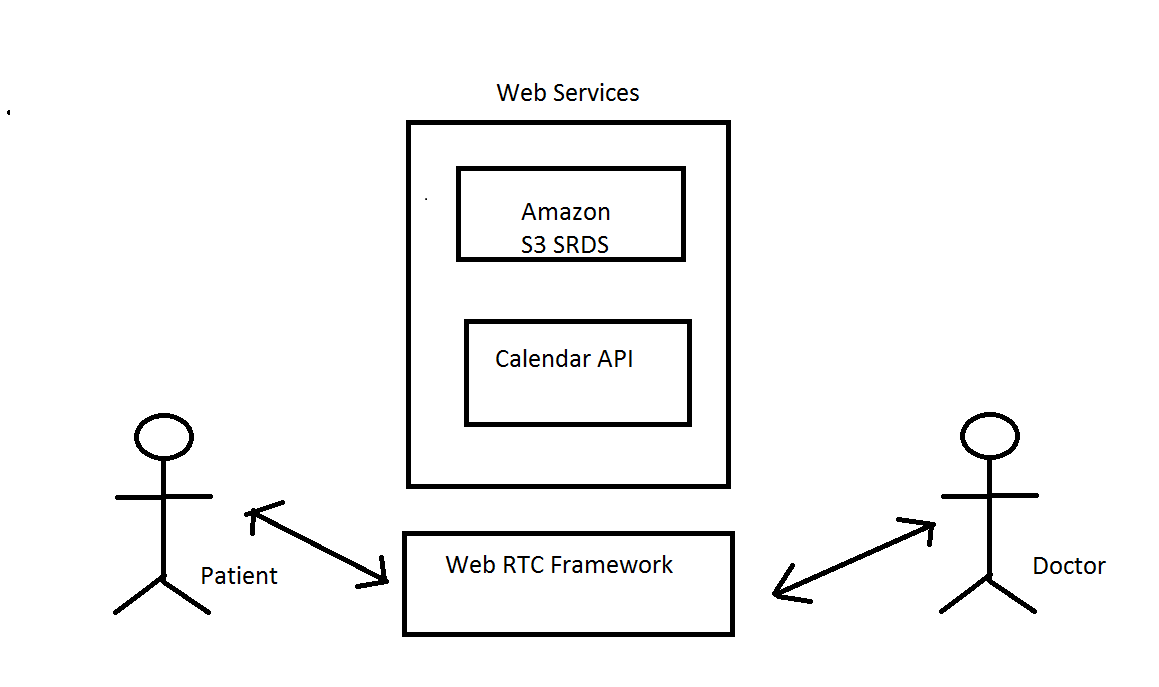
[](http://www.webrtc.org/_/rsrc/1317202919504/reference/WebRTCpublicdiagramforwebsite%20(2).png)

**Architecture of WebRTC**

Several JavaScript APIs are being standardized to enable third party developers to develop web applications using WebRTC technology. The one we are interested in is specified in the below link. <http://w3c.github.io/mediacapture-main> - Media Capture and Streams

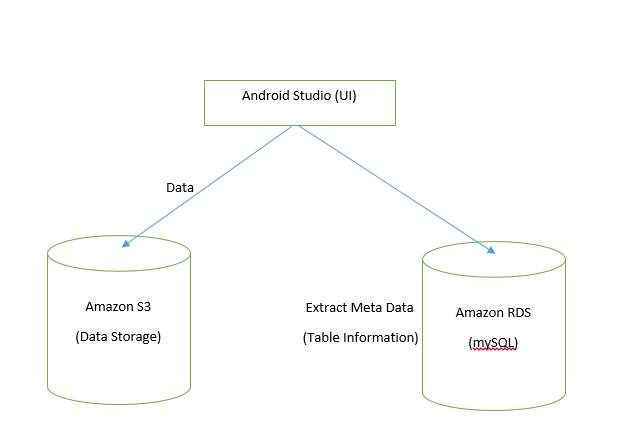
**System Architecture Diagram:**

The below architecture diagram gives you a brief idea of what the system is? And how the system will build? Main part of the architecture is WebRTC framework through which doctor and patient will be connected. On top of the framework we are using web services to send/receive data. We are trying to use calendar API and Amazon cloud Services as a Database.



**System Specifications:**

Existing Services: Amazon S3, Amazon RDS, Google Calendar API, Google OAUTH, Google SpeechToText API, Google TextToSpeech API.



We upload data (conversation) from android application to Amazon S3 and the data about data (Meta data) i.e. the information about the file, uploader and locking information is stores in Amazon RDS MySQL DB instance. S3 is online cloud storage service provided by amazon and we are using this free storage service to save all the data (conversation), and the last important dependency of our app is Amazon RDS. RDS is a relational data base service offered by Amazon in which we create an instance of a database of our needs. In our case we used MYSQL database as this is an open source data base and we will be using this database.

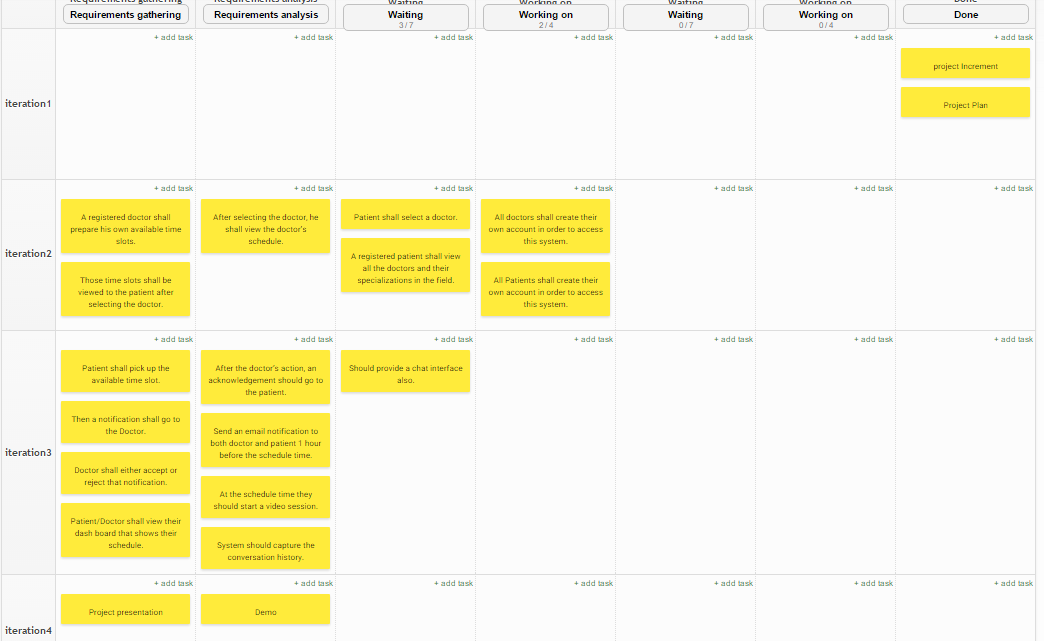
**New Services:**

We wish to create four new services, one to handle all the calendar data, one to push data into Amazon S3, one to push RDS data in to Amazon RDS and last one to implement Google SpeechToText API.

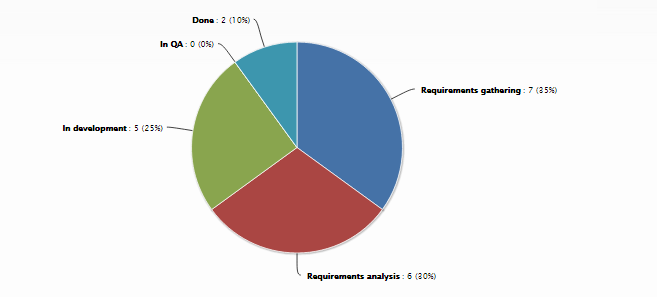
1. **Project Plan:**

**Kanban Tool:**

Here all the stories/ use cases are mentioned. By observing this diagram we can analyze the current status of each story.

****

**Snap of stories in Kanban Tool**

****

**Project Timeline:**

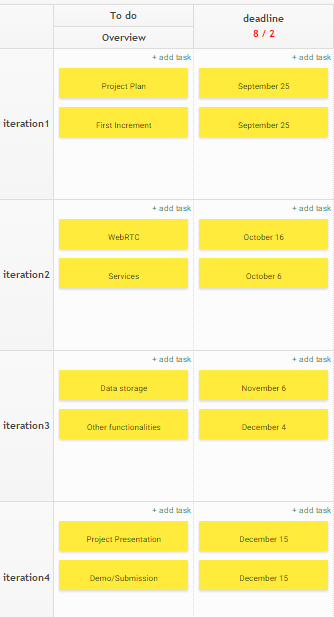
Plan & First Increment: September 25 (F)

Second Increment: October 16 (F)

Third Increment: November 6 (F)

Fourth Increment: December 4 (F)

Project Presentation/Demo/Submission: December 15 (T)



**Project Members & Role:**

**Suramya Chukka: Design and back end coding for texting Android UI**

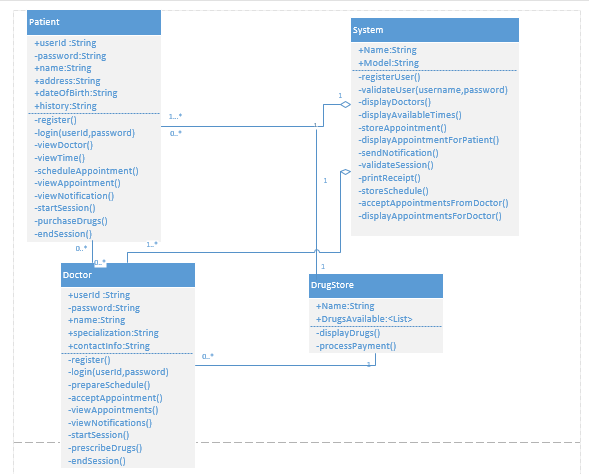
**Praveen Surapaneni: Design and coding of Video Chat**

**Sarath Chand Lingamgunta: Design and coding of Scheduling, Google Calendar API**

1. **First Increment Report**

**Class Diagram:**

Our Application typically has two types of Users – Doctor and Patient. Hence, a class for each of them is designed, which specifies their individual attributes and operations. System, here depicts the overall functionality about how each and every request from users is processed or forwarded. Drug Store is the additional feature which we are planning to implement so that the Patient makes use of it to purchase drugs, whenever Doctor prescribes any.



**Class Diagram**

**Patient Class:**

Being the primary user of our service, a patient must first be registered in order to use this application. Hence, his personal attributes are essential. Apart from that, he is assigned a username and password, which are the only credentials for him to access all the services.

A Patient can choose Doctors based on their specialization. Once a doctor is chosen, he can view Doctor’s availability and schedule appointment accordingly. Notifications are sent when required and a Patient can start the session when it is the time. Purchasing Drugs is optional, if and only if he desires to do it online when the doctor has prescribed any.

**Doctor Class:**

Doctor is the primary resource of our application. Similar to the Patient, he has to register and hence his personal attributes. Username and password are his credentials to use the application. In order for the Patient to make appointment, a doctor must first schedule an appointment based on his free time availability. He can also choose to accept or decline any appointment. Viewing appointment and Starting session is same as the Patient, notifications sent to him would come in handy in this scenario. A session can be of two forms: Video or chat, depending on Patient Specification.

**Drug Store:**

This is the optional service which contains details about drugs. Also, when the user wishes to purchase, checking and validating the purchase is done here.

**System:**

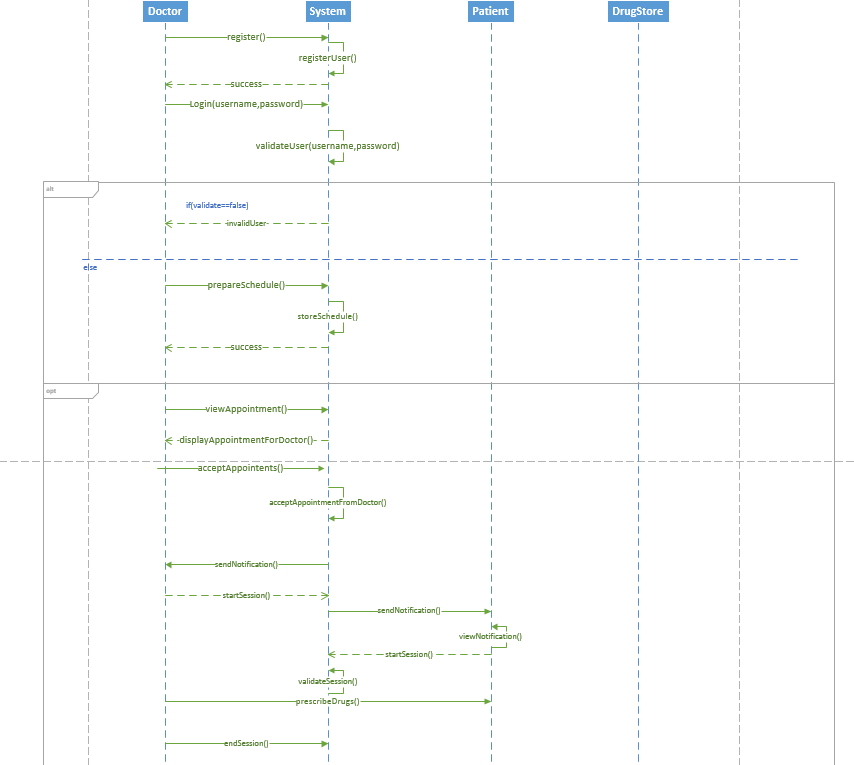
Core part of our application, this is responsible for all the processing and internal functionality. Registering and validating user based on their credentials is the main responsibility of the system. Later on, this system can take the advanced roles such as storing schedule and appointments in its storage system. Whenever a user request is made, information is retrieved from the stored data. Sessions can be proceeded if and only if the System validates it based on stored data. Even for a session to be ended, it must be approved by7 the system (that is how it works, generally). We did not elaborately include all the other default functionalities like displaying screens etc. as things get developed more as we move further in the development of our project.

However, the class diagram we produced here is a mere depiction of how our overall system works and what are the major classes, based on our initial understanding and research.

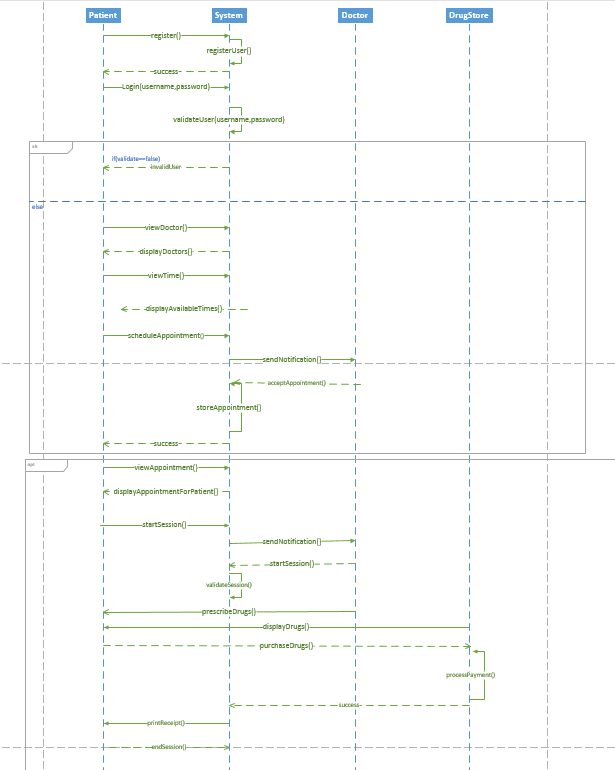
**Sequence diagram:**

Sequence diagram an Interaction diagram. It will Show how process operate among different objects in what order. Name of objects can be class name in class diagram or instance names.

Here doctor can register/login into the system, he can prepare his schedule and view the scheduled time slots. If any patient asks for an appointment he will get a notification and he can approve/reject that appointment. He can start a session with the patient and prescribe drugs. In the same way we also have patient’s sequence diagram.



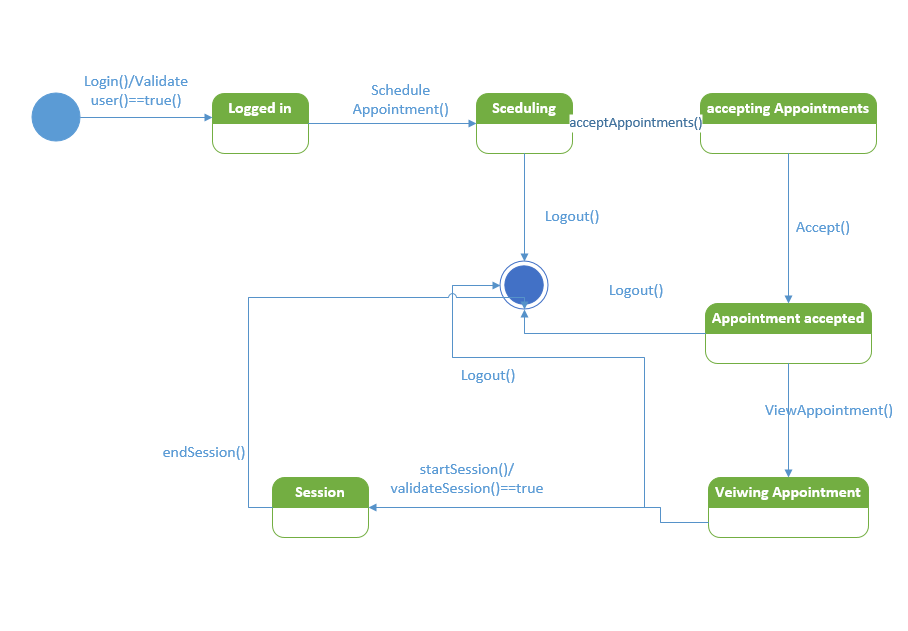
**Doctor’s sequence diagram**

****

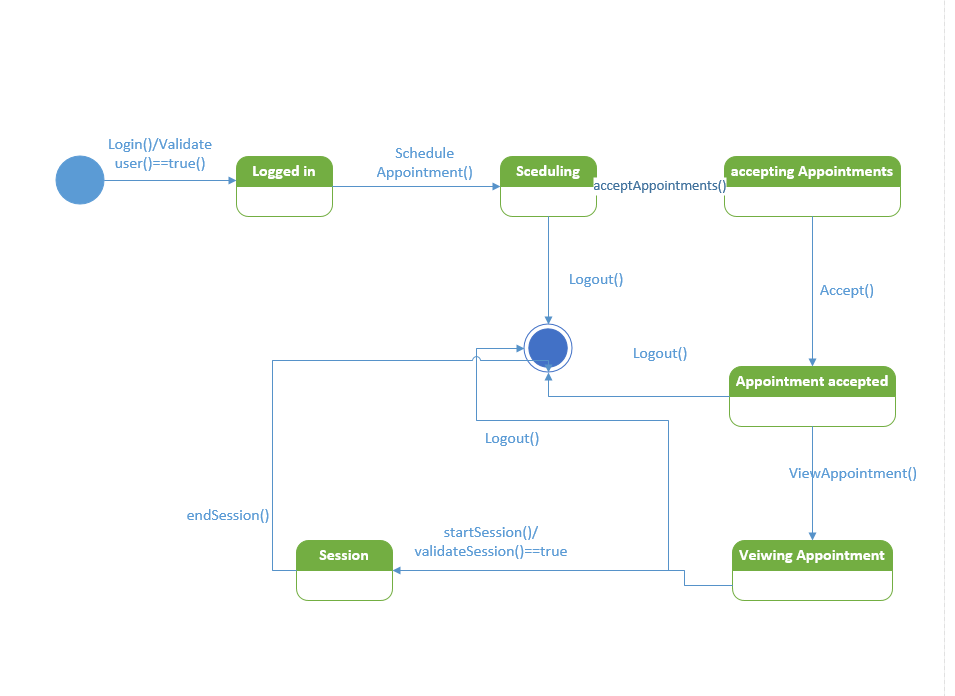
**Patient’s sequence diagram**

**State diagram:**

State diagram describe behavior of system. Here we can establish different states of the system. Primarily we considered Logging in, Scheduling, Accepting appointment, View Appointment and Start session as main states. At any point of the system user can logged out of the system.

****

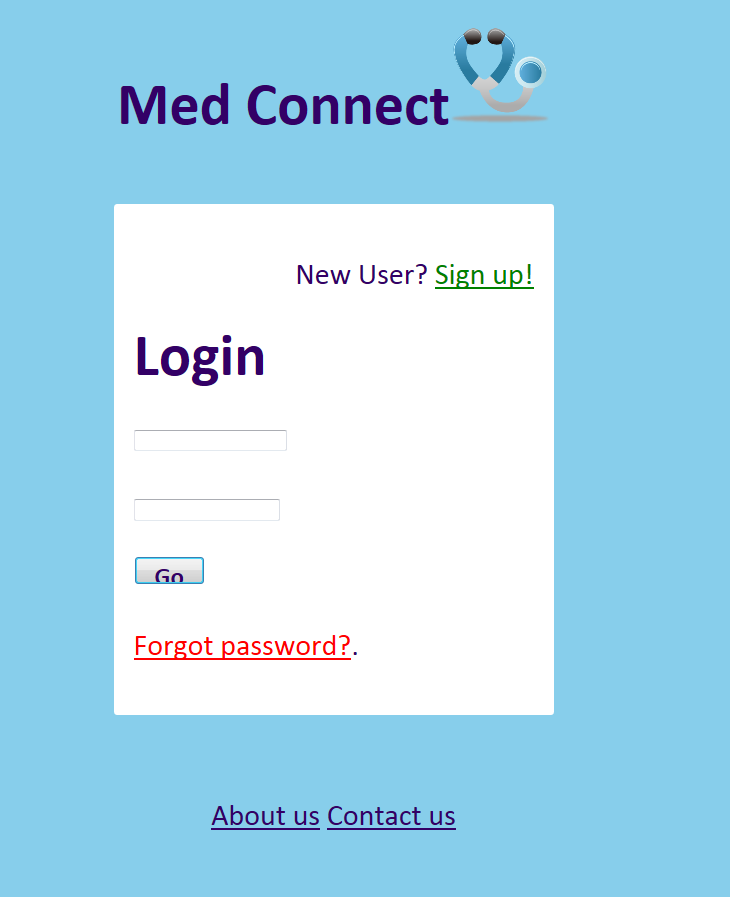
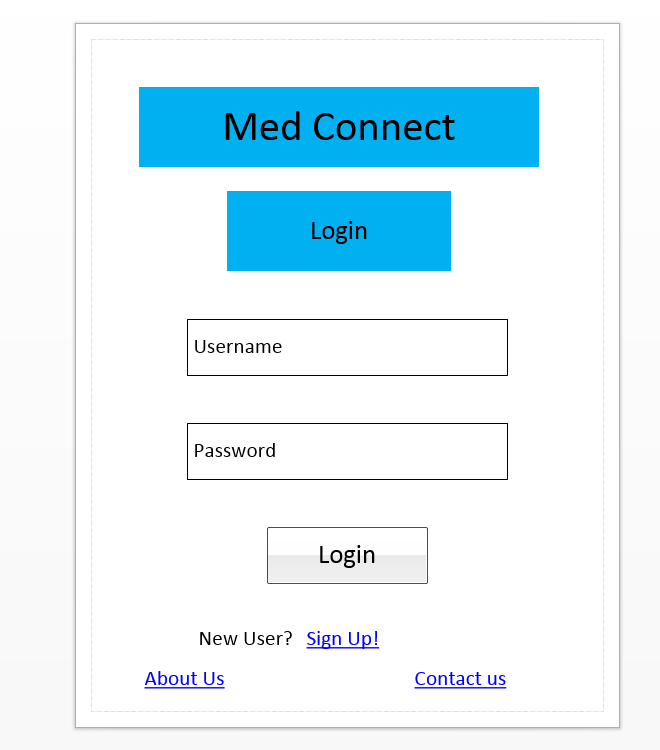
**State Diagram for Doctor**

****

**State Diagram for Patient**

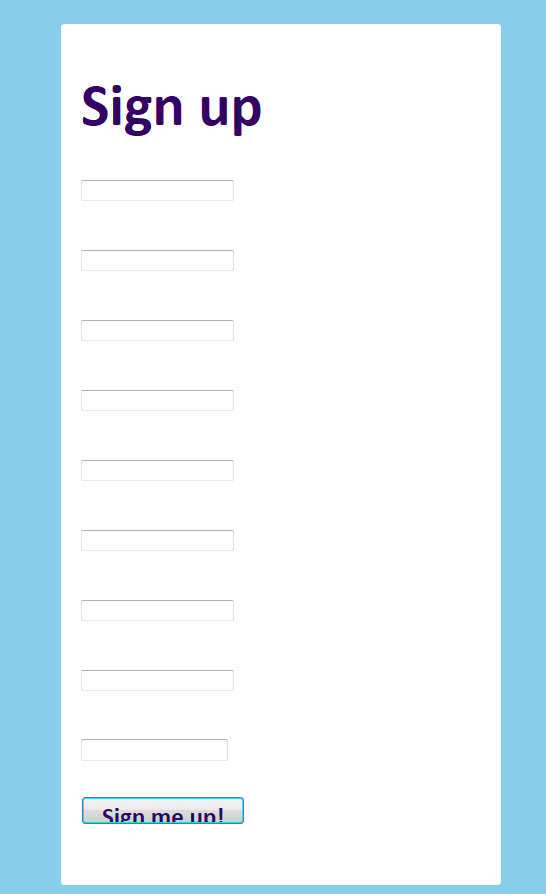
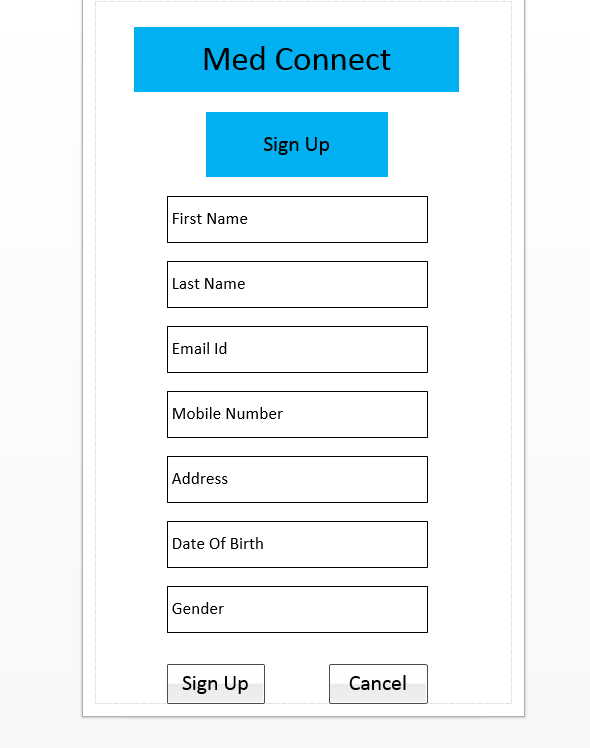
**Mock Ups:**

Below is the mockup of login screen where users can login with their username and password, if the user ever forgets his password we are providing a link to change his password. New users can sign up by clicking on the Signup link.

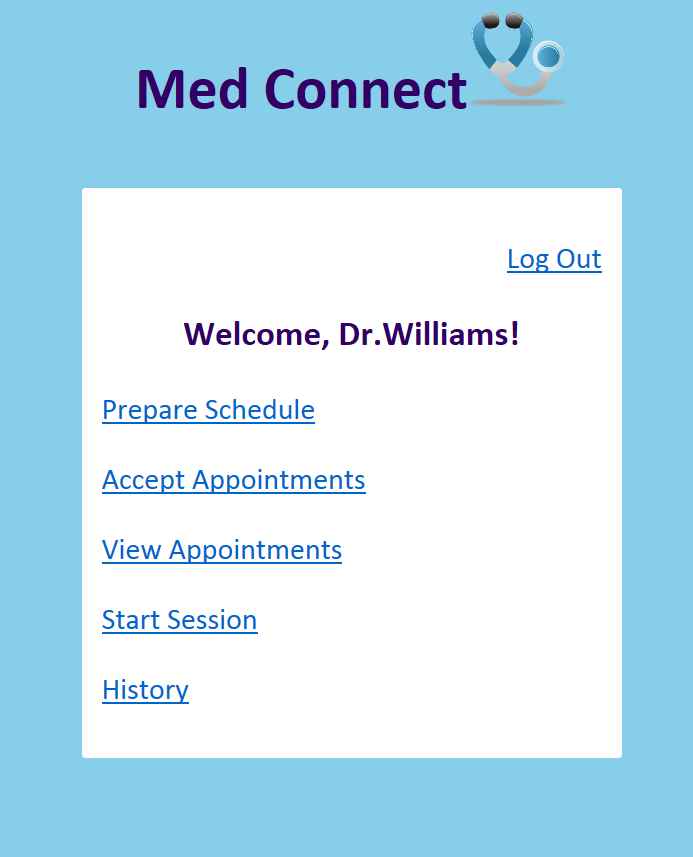
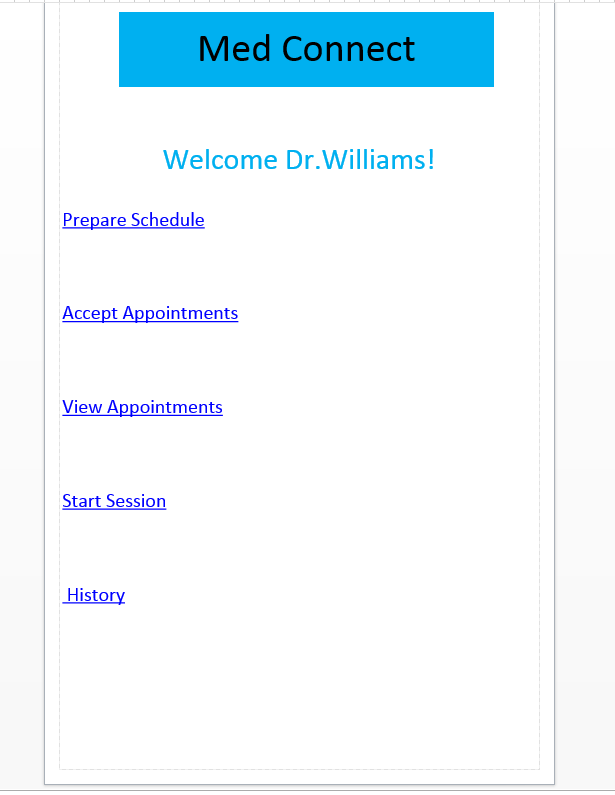
**Login\_MockUp Login\_WireFrame**

Below is the mockup of sign up Screen where user has to give his personal details and our app will create a unique handle for each user to identify them.

 ****

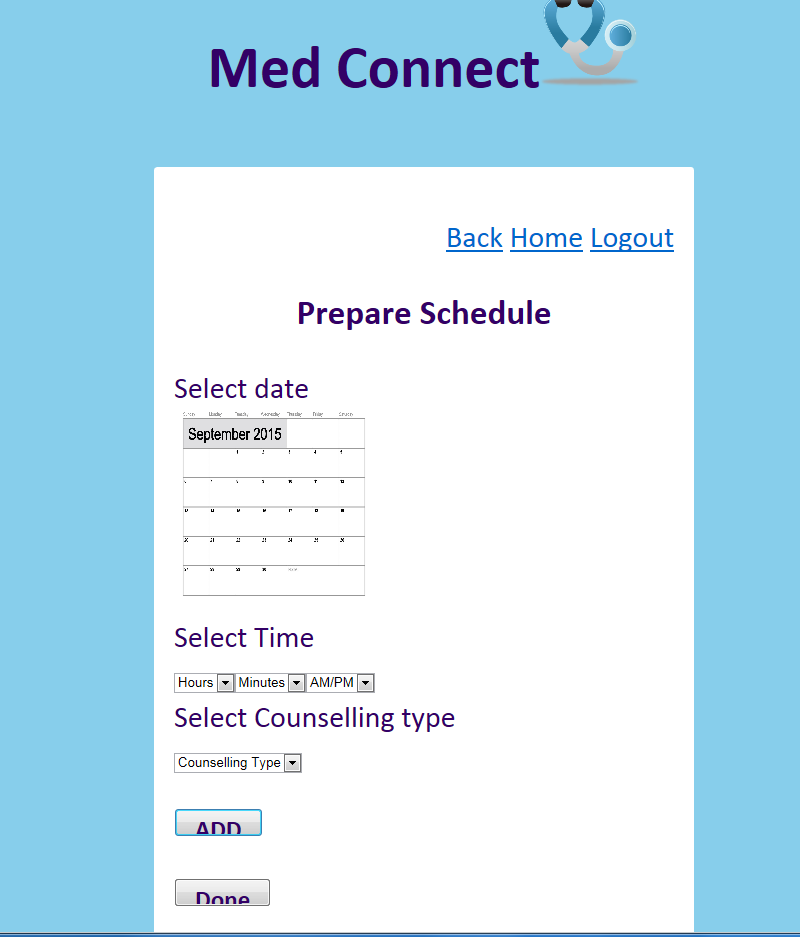
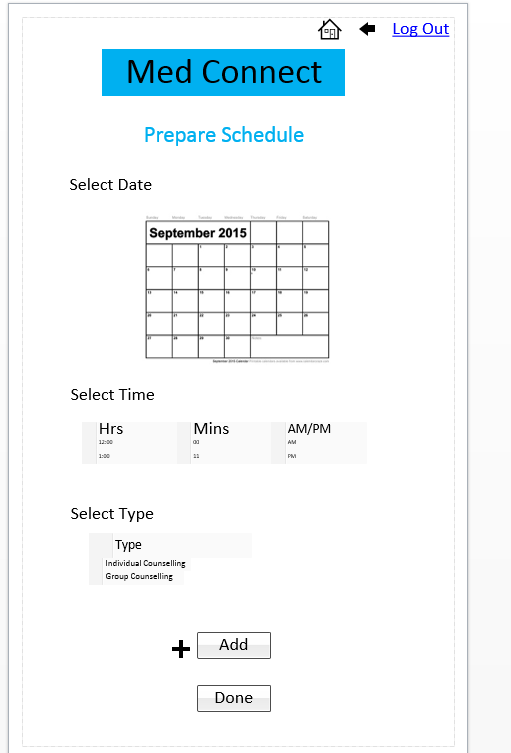
**Signup MashUp Signup WireFrame**

Below is the home page of a doctor, this is the screen what a doctor will see once he logs in to our Application

 ****

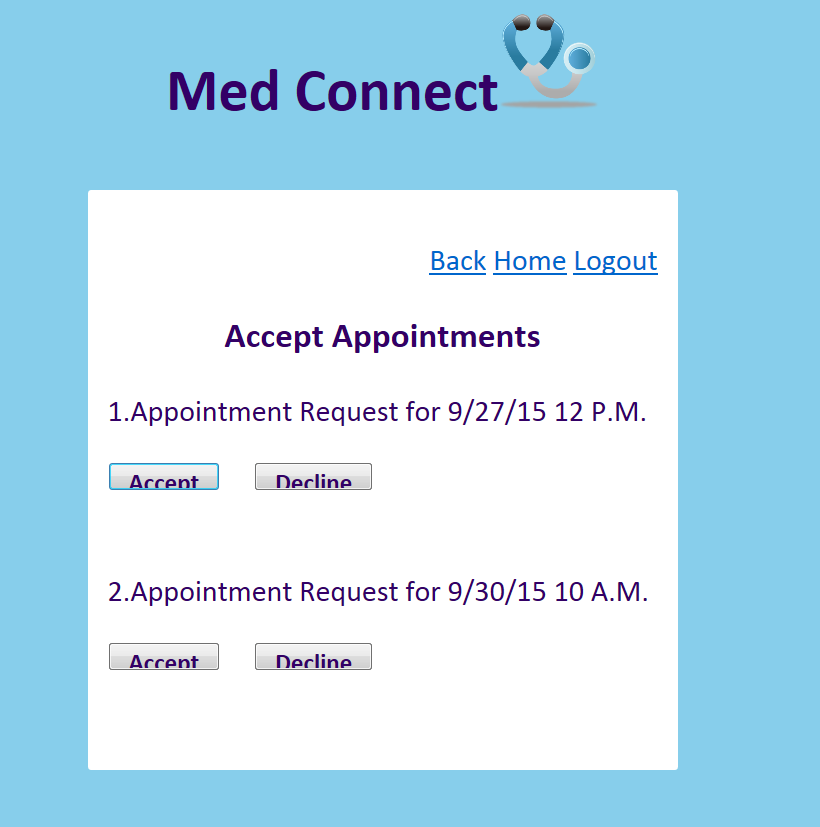
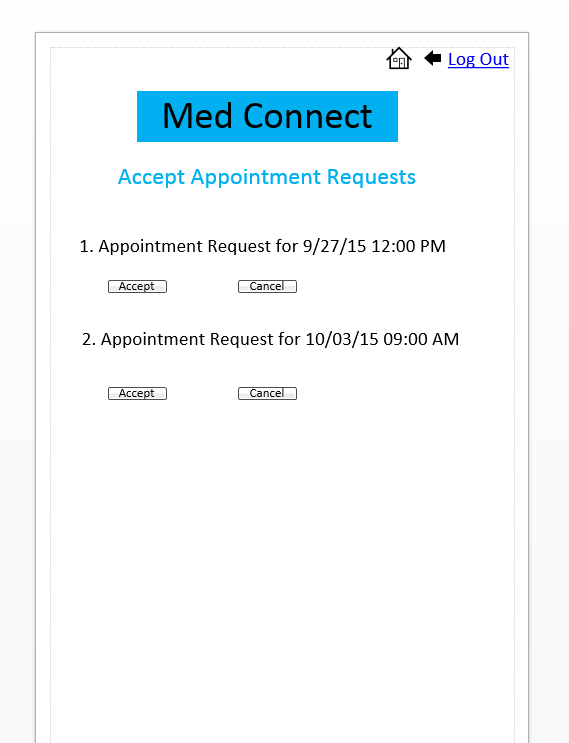
**Doctor\_Home Mahup Doctor\_Home WireFrame**

Here doctor can upload all of his schedule in to a clod based calendar

  ****

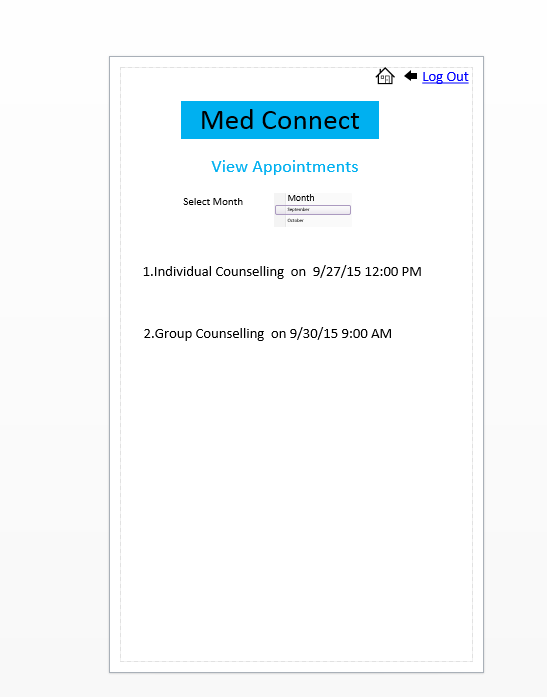
**Doctor\_ Prepare Schedule Mashup WireFrame**

Below screen where Doctor can accept or reject the patients Appoinment. Once he accepts it a notification will be sent out to the patient saying that his appointment has been accepted.

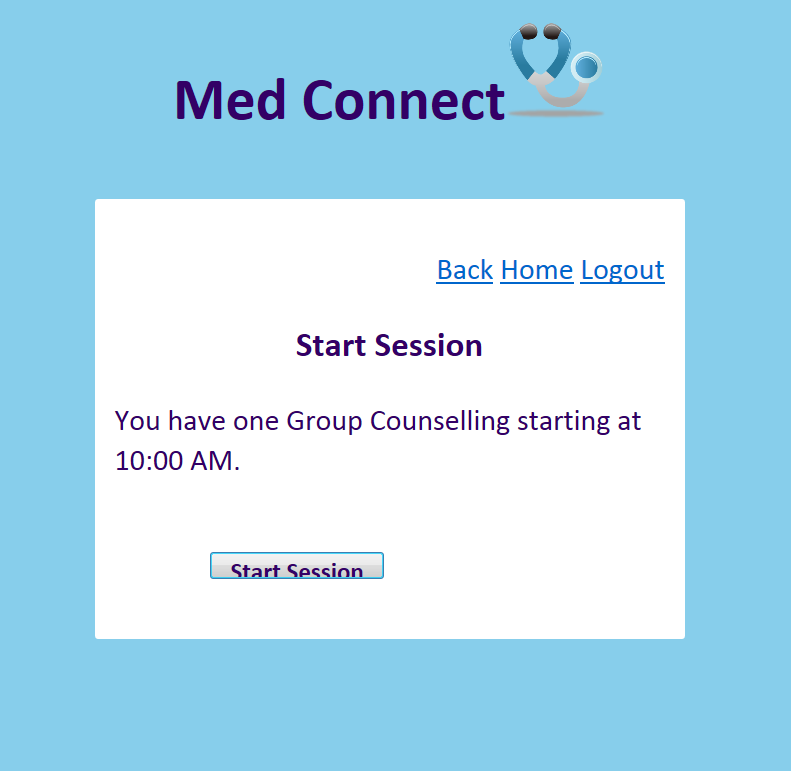
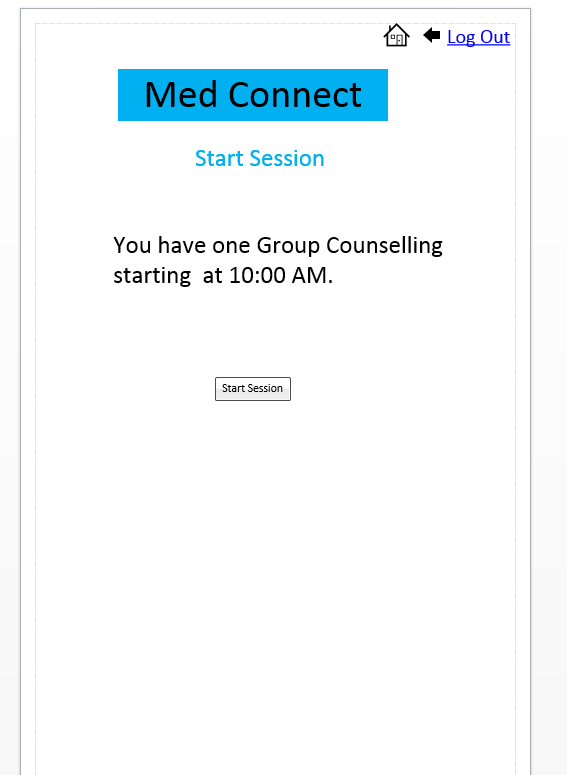
 ****

**Doctor\_Accept Appoinment MashUp WireFrame**

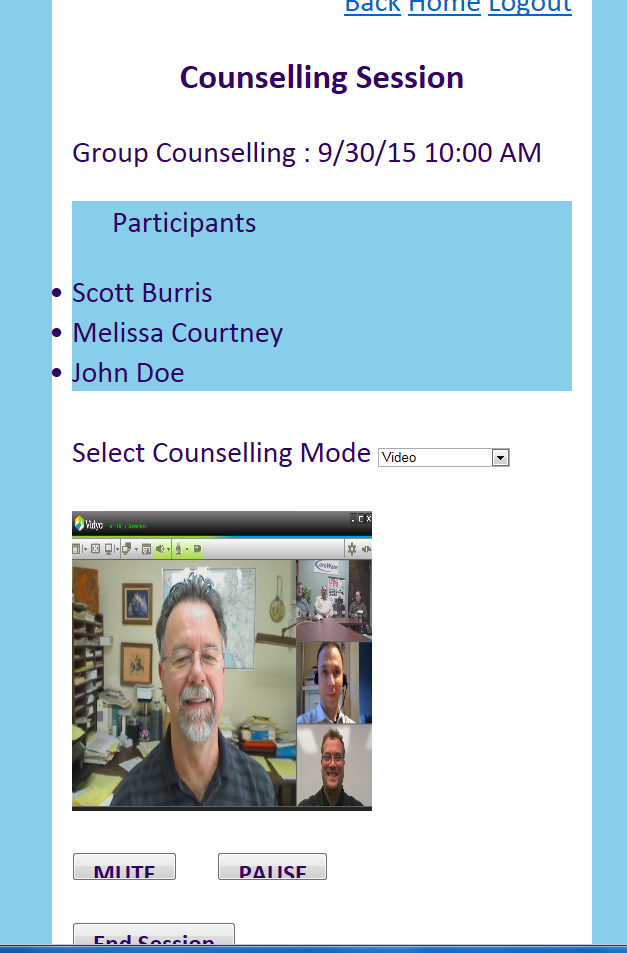
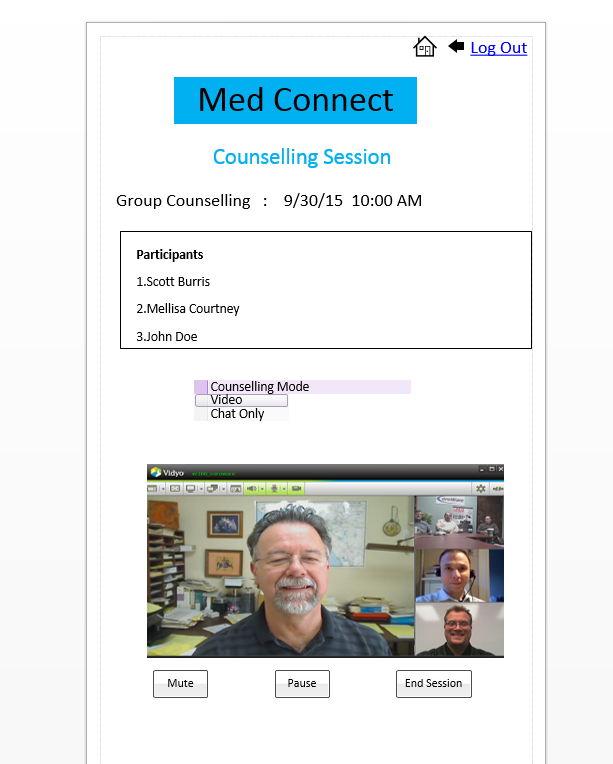
**View Appointments for Doctor:**

** **

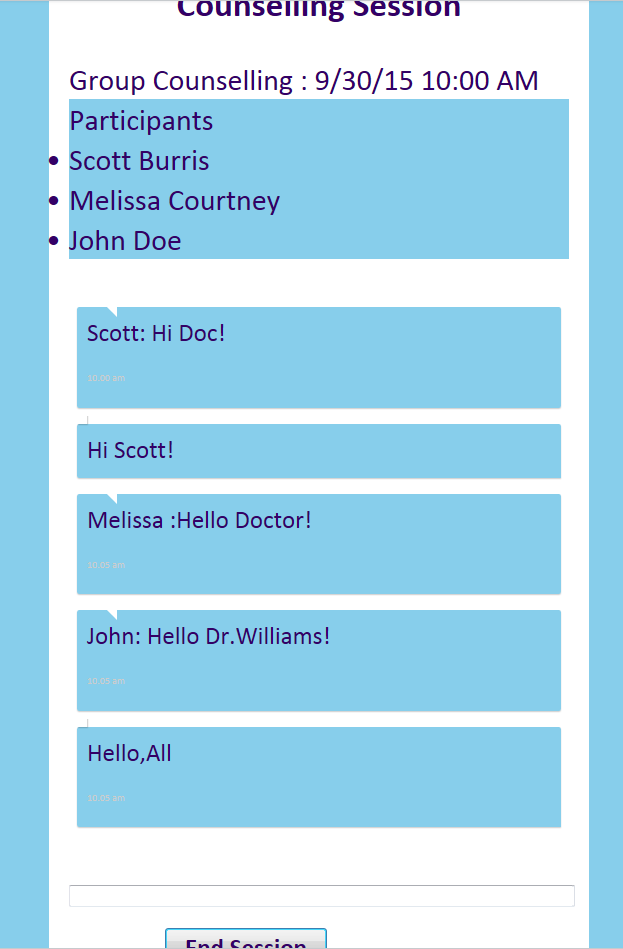
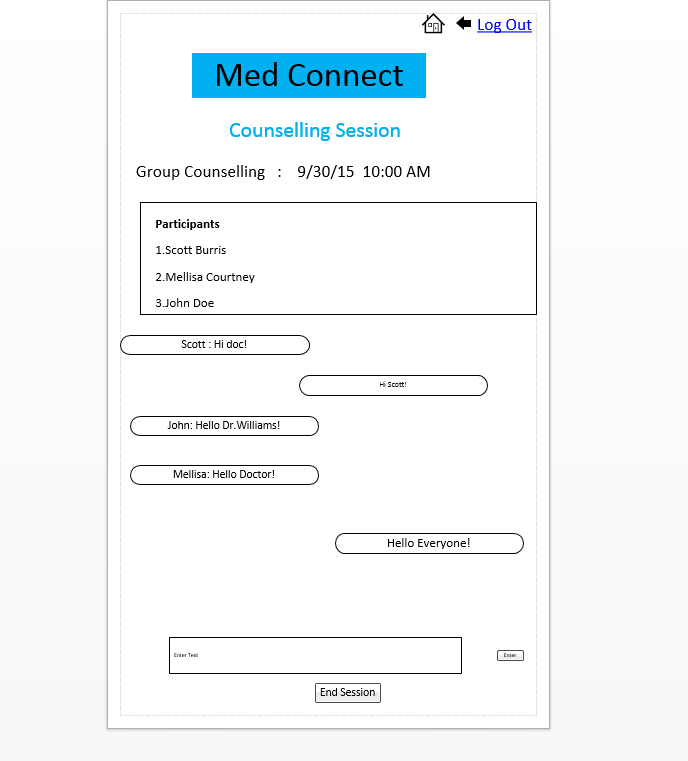
**Start Session Page for Doctor:**

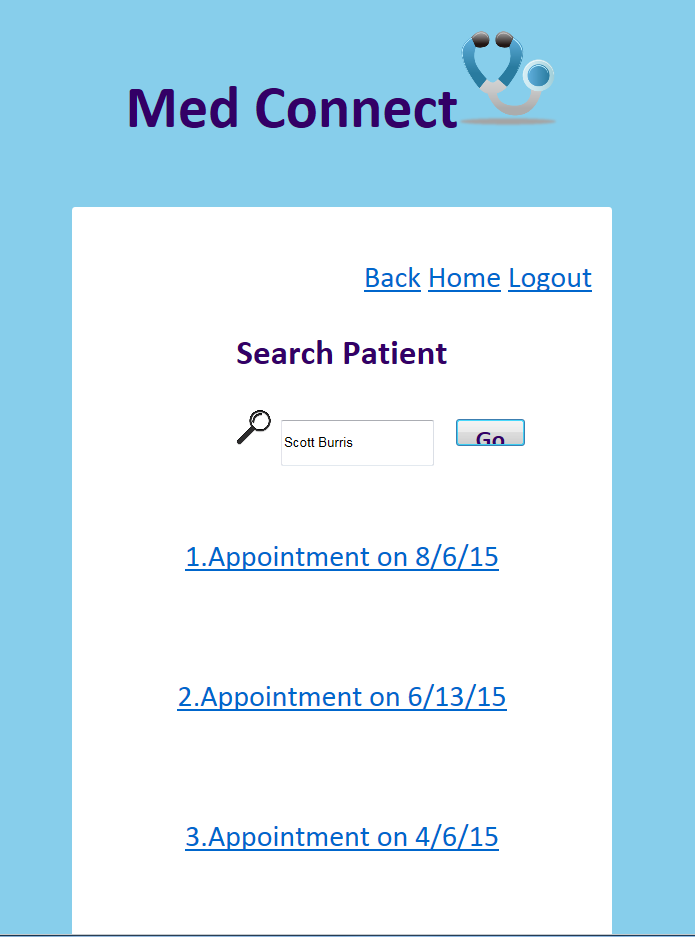
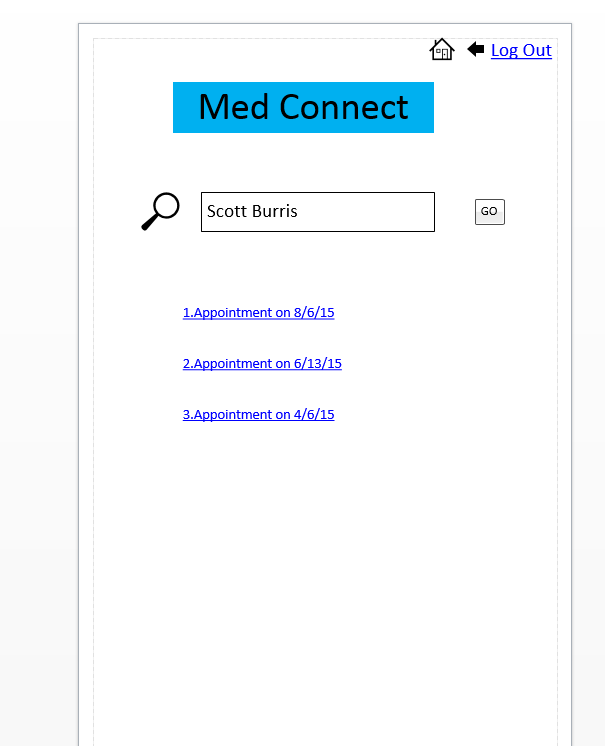
**Group Session page for Doctor:**

****  ****

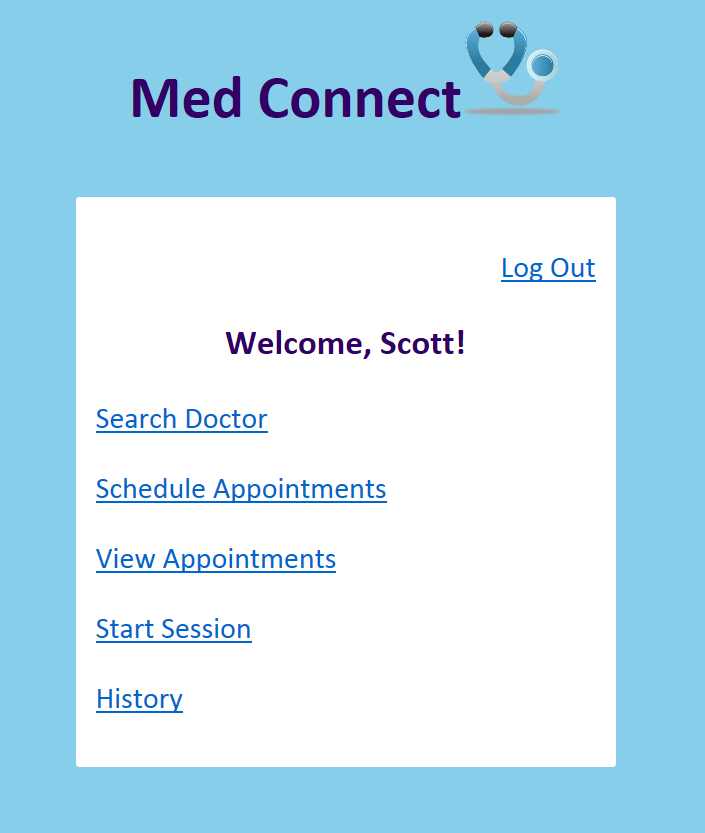
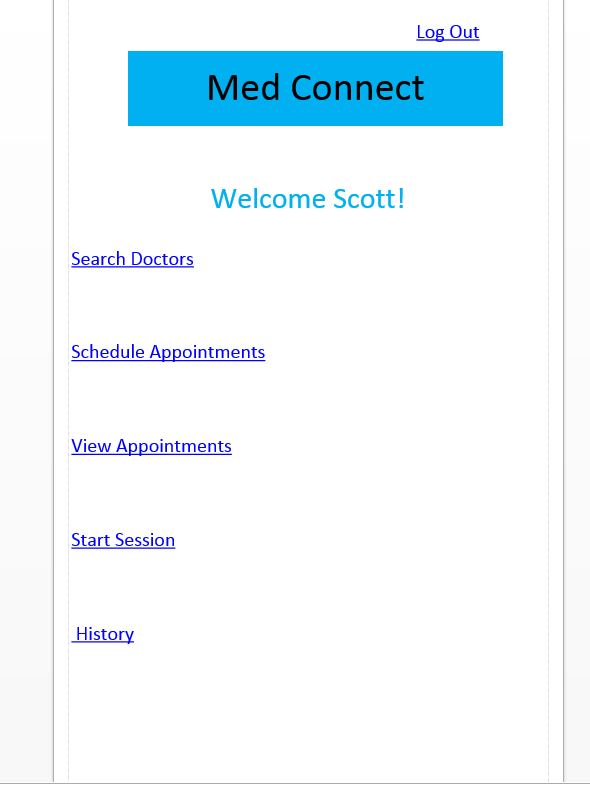
**Chat Only Session For Doctor:**

****  ****

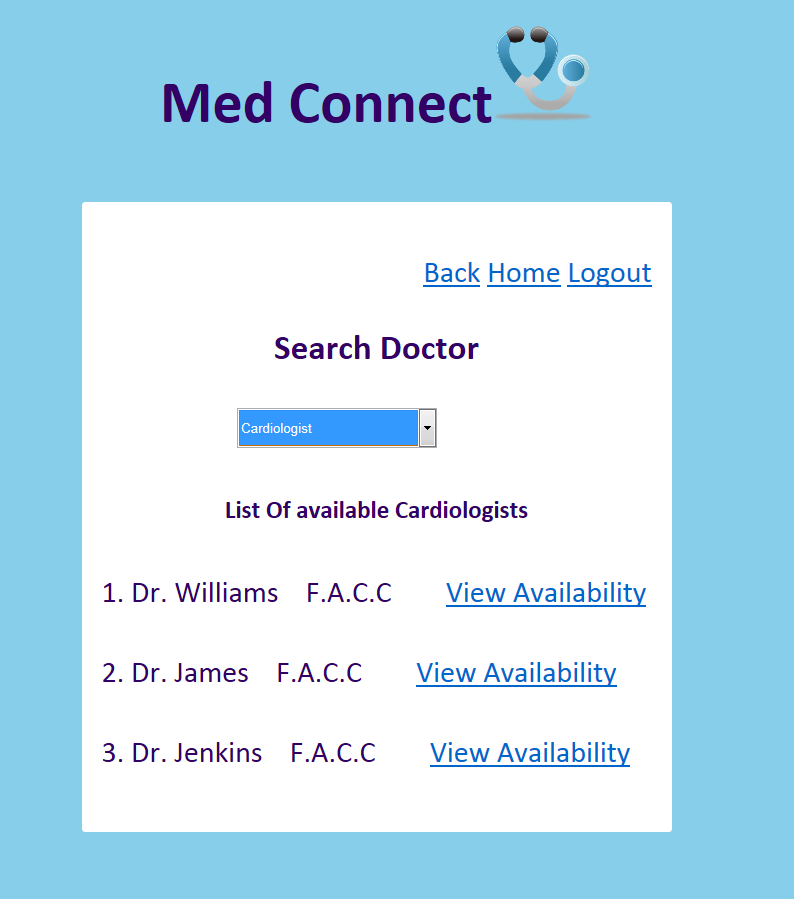
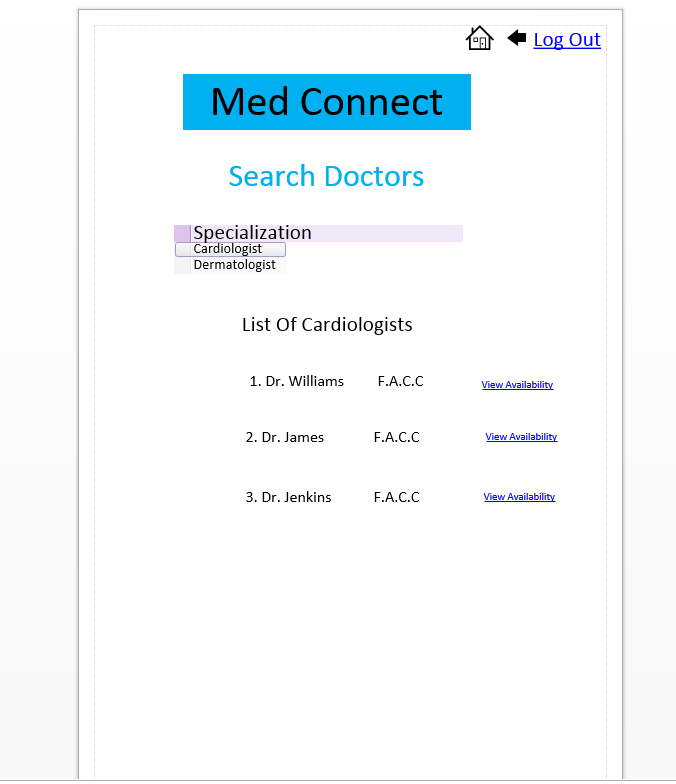
**History page for Doctor:**

****  ****

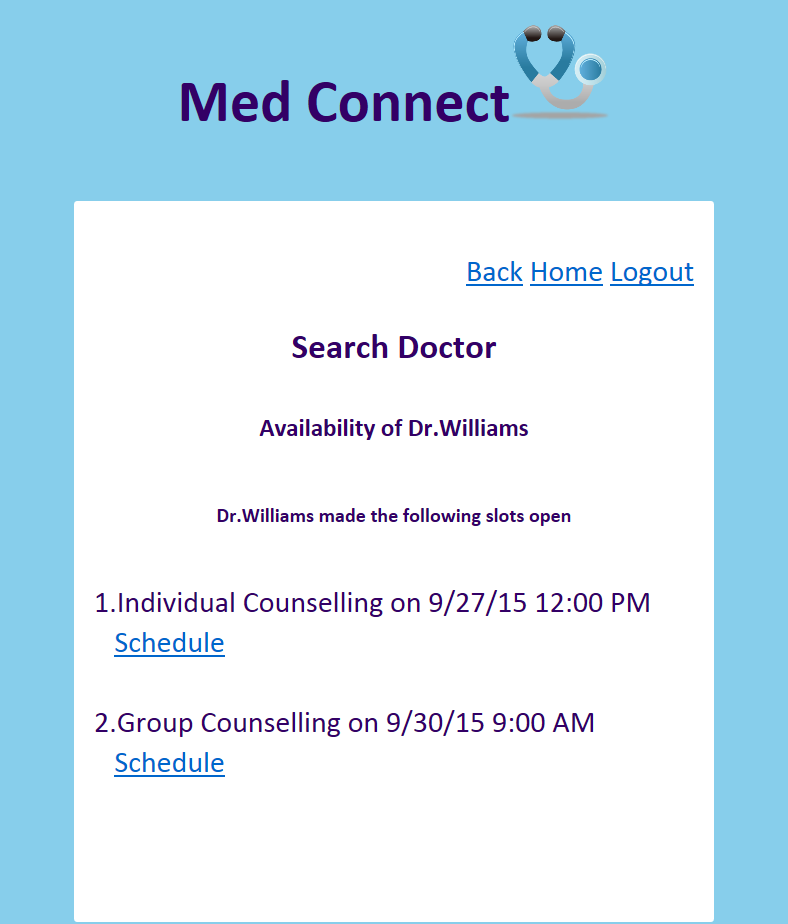
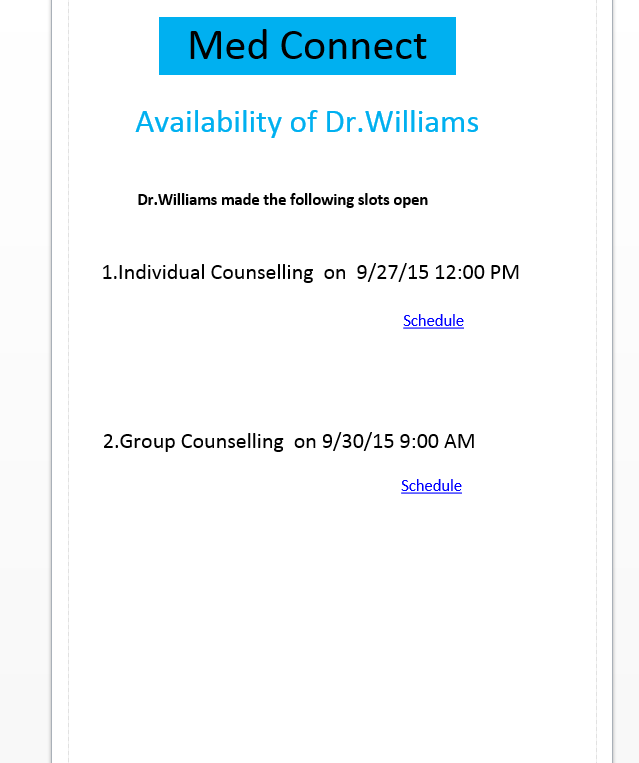
**Home Page for Patients :**

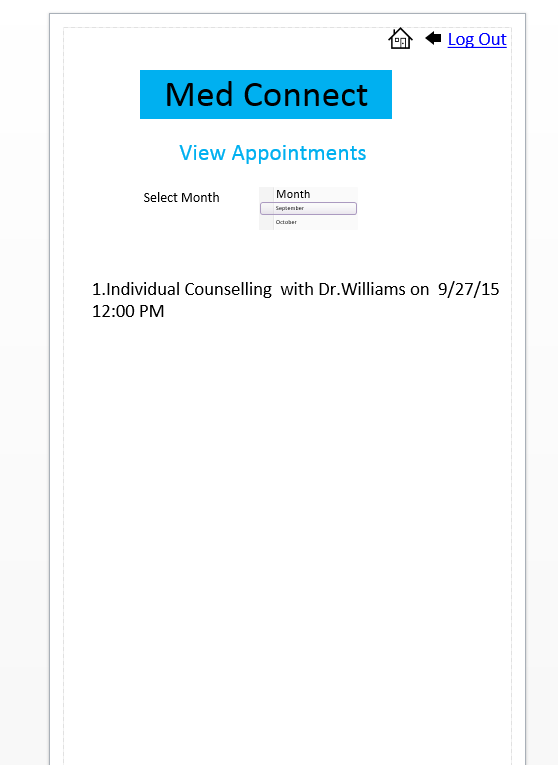
**Search Doctors:**

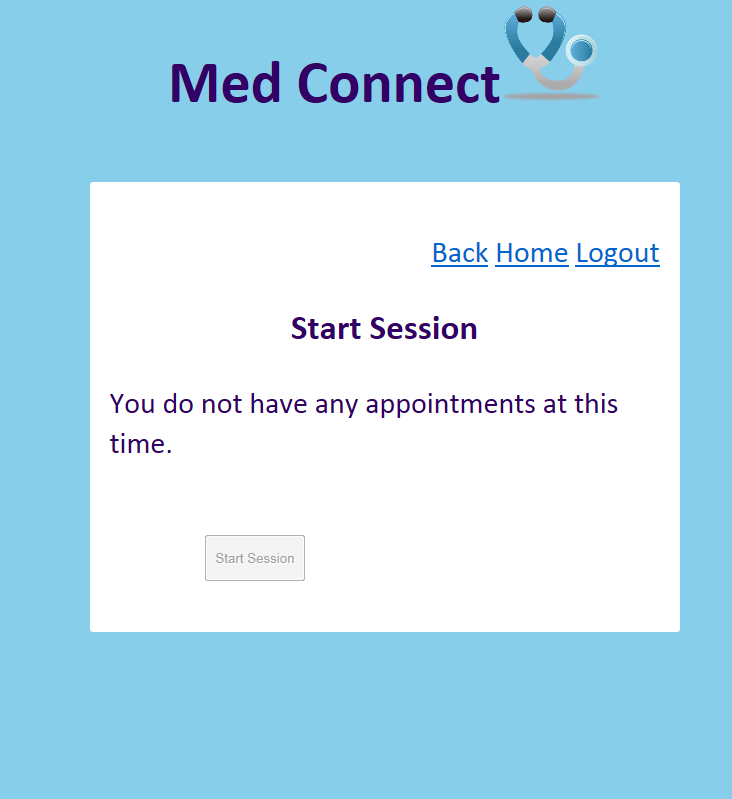
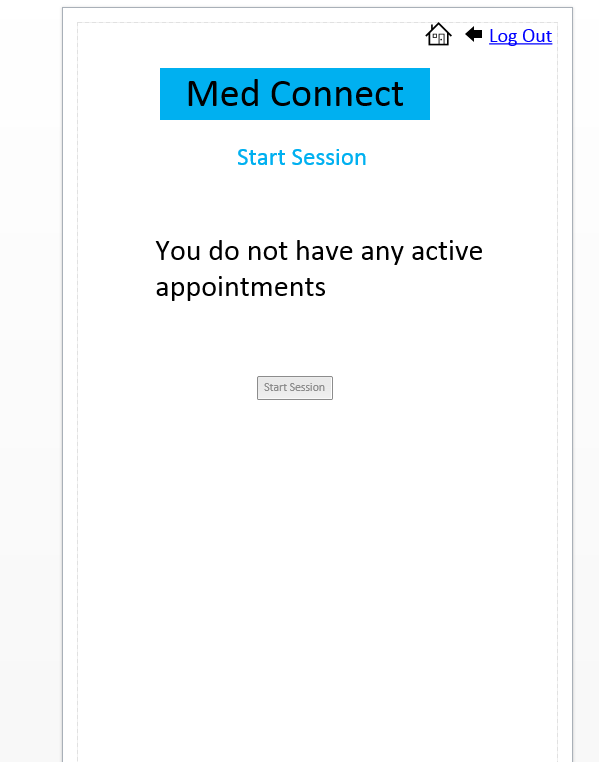
**View Availability Of Doctors :**

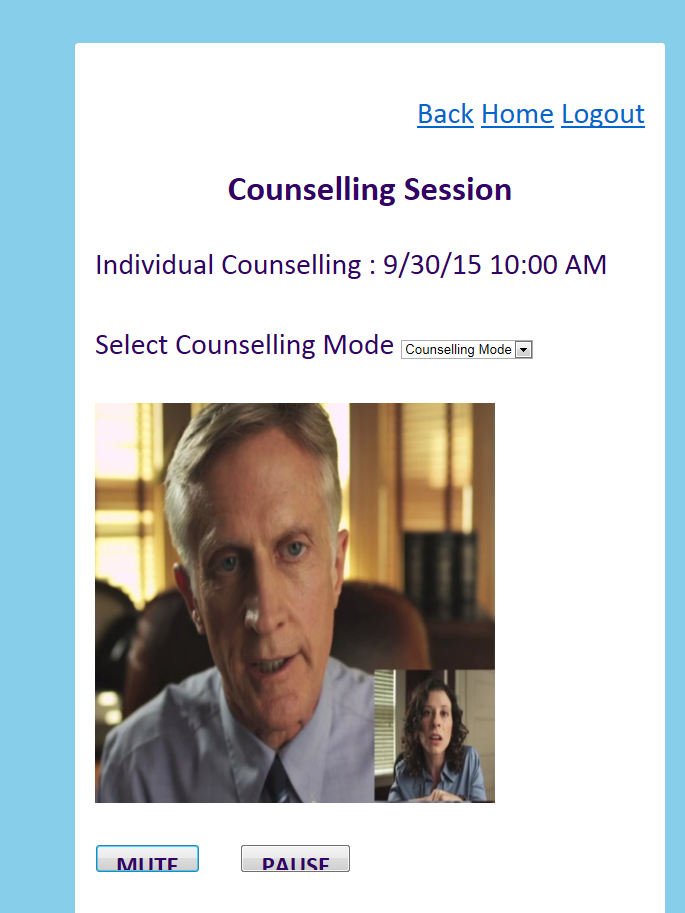
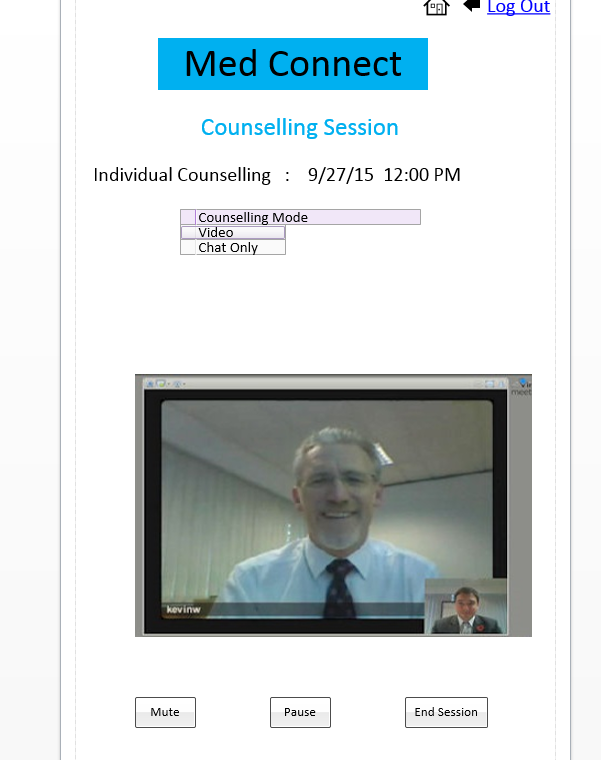
**Patient View Appointments Page:**

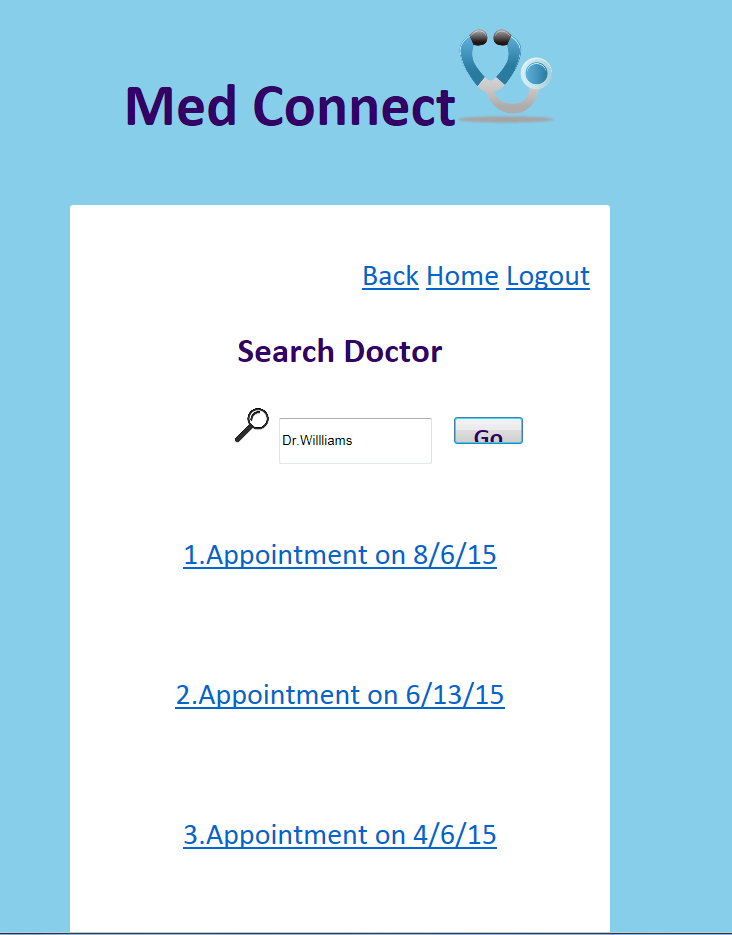
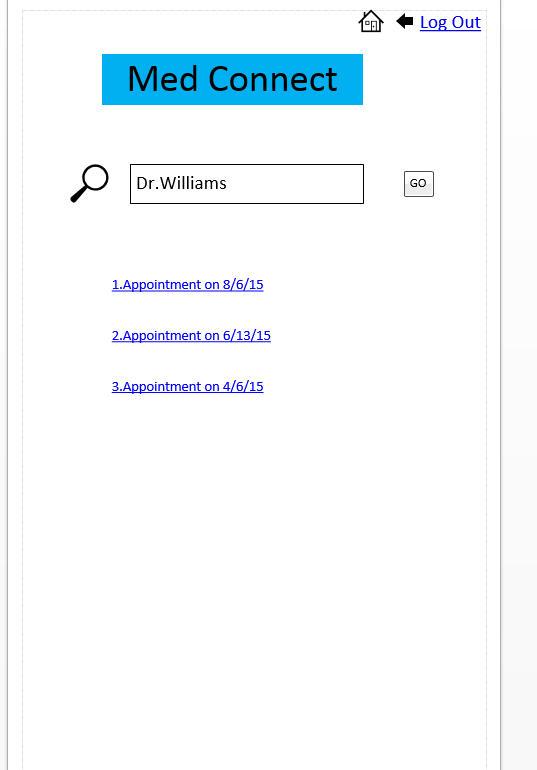
**Patient Start Session Page:**

**Patient Session Page:**

**Patient History:**

1. **Bibliography:**

<http://www.webrtc.org/native-code/android>

<https://en.wikipedia.org/wiki/WebRTC>

<https://frnds.kanbantool.com/dashboard>

<https://www.gliffy.com/uses/uml-software/>

<https://aws.amazon.com/s3/>

<https://aws.amazon.com/rds/>

<http://developer.android.com/guide/topics/providers/calendar-provider.html>

<http://developer.android.com/index.html>

<http://www.frengly.com/>

<http://w3c.github.io/mediacapture-main/archives/20150925/getusermedia.html>