[[1]](#footnote-32264)Fewa Telemedicine Technical document

Version 1.0

Contents

[Introduction and Domain Explanation 2](#_Toc52470664)

[Vocabularies 2](#_Toc52470665)

[Technologies used 2](#_Toc52470666)

[Overall Static Architecture 2](#_Toc52470667)

[Important Activity flows 3](#_Toc52470668)

[Provider Login 3](#_Toc52470669)

[Sending a chat message 3](#_Toc52470670)

[Waiting.. 3](#_Toc52470671)

[Code repository 3](#_Toc52470672)

[Building and running the project 3](#_Toc52470673)

[Prerequisites 3](#_Toc52470674)

[Running the application 3](#_Toc52470675)

[Clone or download repository 3](#_Toc52470676)

[Database Creation 4](#_Toc52470677)

[FewaTelemedicine Basic Changes, Build and Run Project 4](#_Toc52470678)

[DB Design understanding. 5](#_Toc52470679)

[Practice table 5](#_Toc52470680)

[Provider Table 6](#_Toc52470681)

[Patient Table 6](#_Toc52470682)

[Understanding folder structure 7](#_Toc52470683)

[Angular client-side folder explanation 7](#_Toc52470684)

[Server side folder structure 7](#_Toc52470685)

[Video embed 7](#_Toc52470686)

## Introduction and Domain Explanation

This a technical document which explain the project and architecture of Fewa telemedicine.

This is a simple telemedicine project which helps doctor and patients to connect using video conferencing and chat. You can see the full demo of the same at <https://www.youtube.com/watch?v=GslXbdrWbgk> . The video is in Nepalese language but you should be able to follow it up.

Below is the basic flow of the project.

* Doctor logs in , Patient logs in by filling his issues.
* Both go in to video conference call and discuss issues.
* Doctor fills advice.
* Patient can print advice after the call.
* Doctor and patients can also send chat messages to each other
* Patients can also share documents with doctor.

## Vocabularies

Telemedicine is a very simple project but it follows the health care domain vocabulary very religiously. As a common person you would use vocabularies like hospital , patient and doctor.

But when it comes to health care industry these vocabularies are more generalized with words like practice , and provider.

Practice can be Hospital,or Clinic consisting of Solo or group of Providers.

Provider can be Doctor or Nurse that work under a practice.

## Technologies used

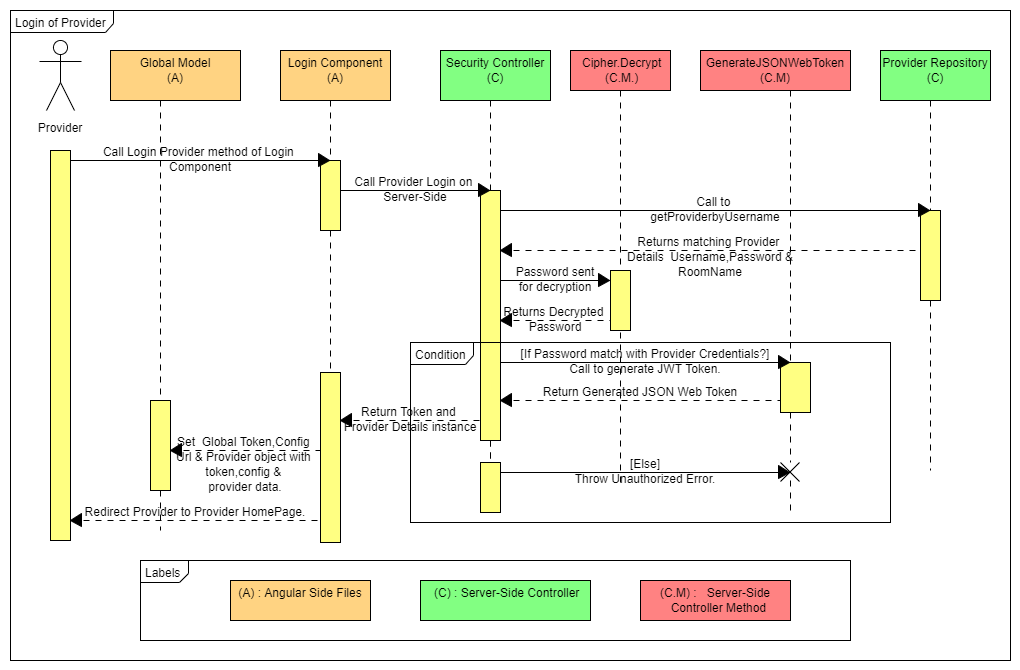
This project is created using Angular as frontend, MVC Core as back end , EF core as ORM and database is postgre.

## Overall Static Architecture

By Shiv

## Important Activity flows

### Provider Login:

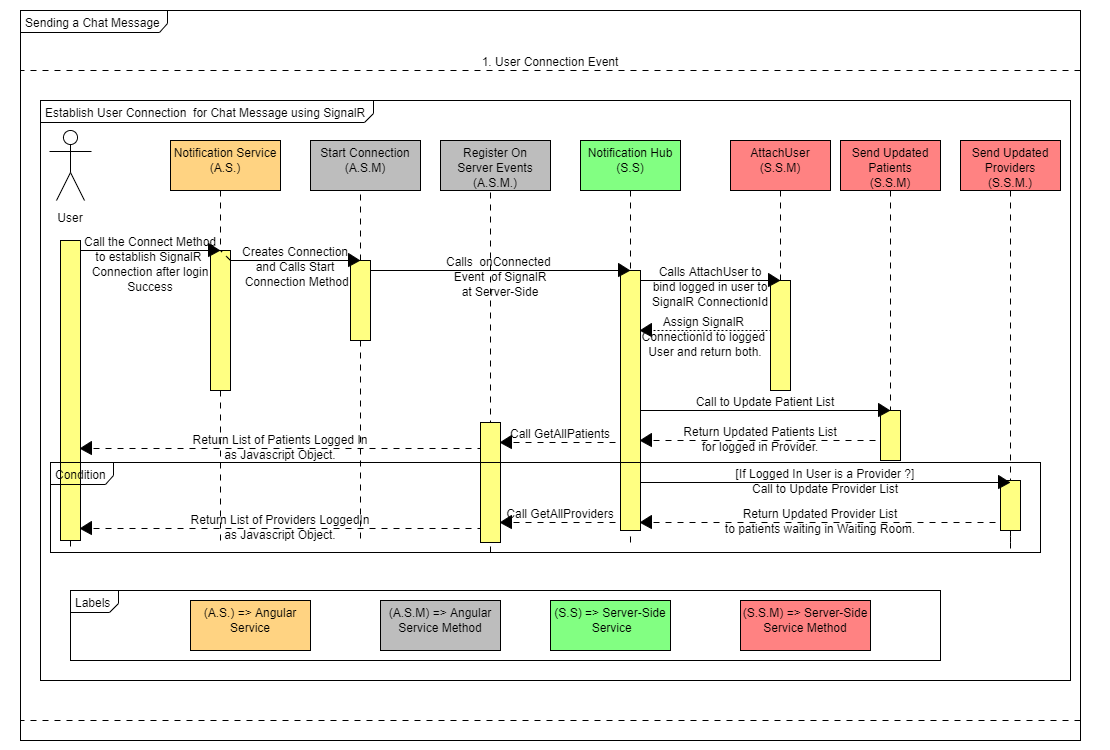


Explanation:

1. Provider Enters Login Credentials and clicks on Login Button.
2. This request then goes to Login Method in login component and a http call is made to MVC Core Security Controller Login Method which then calls the provider repository using getProviderbyUsername method.
3. The Provider Repository checks the username with existing username in database and returns matching Provider object from database to Login method at Server-Side.
4. Server-Side Login Method then verifies username and password with entered credentials and if they match then Password is Decrypted using Cipher.Decrypt Method and a JWT Token is generated using GenerateJSONWebToken Method.This token and Provider Object consisting of provider details is returned to Angular Side Component.
5. The value of token obtained from server-side is passed to Global Model and is used to set value of global token object. Similarly, value of provider object obtained from server-side is used to set value of Global Provider Object and Global config URL is updated with doctor username.

### Sending a chat message:

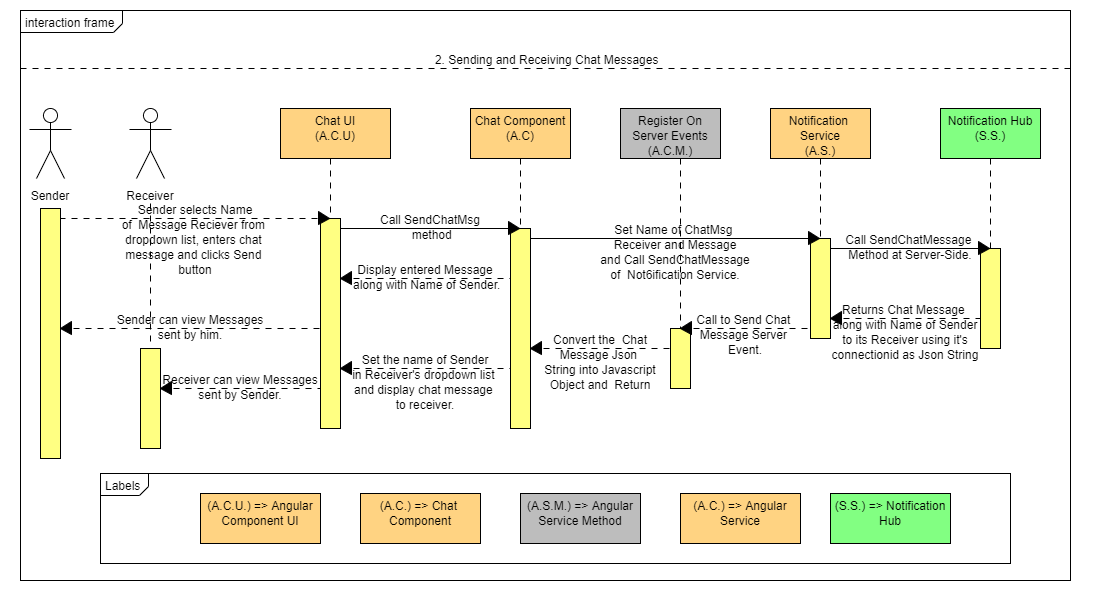
### Step 1 : OnConnected Event :



#### Explanation:

1. After Successful Login of User (Provider or Patient).A Connect method of Notification Service gets called by the constructor.
2. The Connect method internally creates connection and calls start connection method of  
    notification service.
3. This further calls onConnectedAsync Event of SignalR at Server-Side in NotificationHub Service.
4. In SignalR OnConnected Event first goes to map loggedIn username with SignalRConnectionId.
5. Second Call goes to Send Updated List of Patients and is send to all Providers logged In.
6. And Thirdly, If logged in user is provider then provider list is also updated and sent to   
   all patients who are logged in.

Step 2 : Sending and Receiving Chat Messages.



#### Explanation:

#### After successfully connection to SignalR , Sender selects receiver from dropdown list ,Enters message to be sent and clicks on send button.

#### Then Call is made to SendChatMsg method present in component where chat UI is present.

#### The SendChatMsg method saves the name of receiver and chat message in a chat message object and makes call to SendChatMessage method of notification service and display the sent message on Chat UI of sender.

#### The SendChatMessage of notification service calls the server side SendMessage in Notification Hub.

#### This further depending on User who had sent message sets the Name Value of Chat Message with that of Sender and ChatMessage value with Message Text and Sends it to receiver of Message using SignalR .

#### Later, Register On Server Events method of Angular catches this Server-Side Call and Stores the data returned from Server as Javascript Object.

#### This then Calls the Chat Component of receiver and sets the name value with sender name and message text and Displays it to the Receiver at Chat UI.

### 3)Waiting ...

## Code repository

Code is open source and you get the latest version from <https://github.com/opensource-emr/Telemedicine/>

## Building and running the project

As discussed the overall architecture session of project has been given software requires as follow.

1. Visual studio 2019 community edition.
2. Vs code.
3. Postgre sql.

## Prerequisites

You need VS Code(for Client), visual studio(for API), Postgre sql server(for Database), Angular 7

Download VS Code from [here](https://code.visualstudio.com/download)

Download Visual Studio from [here](https://visualstudio.microsoft.com/downloads/)

Download Postgre SQL server from [here](https://www.postgresql.org/download/)

Download Angular 7 from [here](https://cli.angular.io/)

## Running the application

### Clone or download repository

git clone https://github.com/opensource-emr/Telemedicine

NPM Installation for Angular Project

To Install and Run angular project go through below steps:

Step 1: Go to **Telemedicine-master\FewaTelemedicine\ClientApp** path and copy.

Step 2: Open Your Node.js Command Prompt paste the copied path and execute

**npm install** command.

Step 3: Once the **npm install** done successfully than execute **ng build --watch** command. So, some of you wondering that what is **ng build and --watch**. Here it is **ng build** ( It build you angular code) and **--watch** ( It runs in background so that whenever you change the code and save it. It gets build automatically.)

Note :

you can also open Angular Project in visual studio code from there you will do **npm install** and **ng build --watch**.

### Database Creation

**Create Database with below steps**

1. Go to Start and serch pgAdmin and open
2. Create one Data Base **Telemedicine.**

### FewaTelemedicine Basic Changes, Build and Run Project

Step 1: Go to **Telemedicine-master\ FewaTelemedicine.sln** path and double click on solution file to open project in Visual Studio 2019.

Step 2: Open Solution Explorer and find **appsetting.json** file into FewaTelmedicine Web Application and change the **connectionstring** properties as per database and server name.

Step 3:Copy the script from ClientApp->index.html and past it from Views->Home->Index.cshtml

Step 4: open Tools->NugetPackageManger->Package Manger Console -> After opening the package manger console run this command **add\_migration m** The build is successful complete after run thisCommand **update-database .**

Step 3: Once changes done then save the file.

Step 4: Now build **Telemedicine** web Application and run it.

## DB Design understanding.

### Practice table

|  |  |
| --- | --- |
| PracticeId | This is a unique identifier in DB for values. |
| Name | The name of the practice. This can be the hospital name , doctor name , clinic name or any entity/organization which provides service. |
| Address | The address where the practice resides. |
| Contact Number | The contact number of the practice. |
| Email | The Email address of practice .Can be doctor email or clinic email. |
| Description | The Description of practice.Can be short information about the hospital or clinic. |
| Calling Platform | The Calling Platform is the type of video conferencing that is used for patient and doctor call . We are currently using Jitsi and Tokbox where Jitsi is Free Video conferencing platform. |
| Url | The Url of provider that is appended after Hospital url and is visible in address bar to sort out patients depending on provider. |
| Logo Path | The file path of the Logo of respective Practice that is stored within project solution folder. |
| Email API Key | The Email API Key holds the value of Twilio API Key that is used to send Email Invitation to Patients. |
| Email API Name | The Email API Name represents name of the Email that is sent as Invitation. |
| Email Plain Body |  |
| Email Subject | The Email Subject represents the Subject of the Email Content send as Invitation. |
| Email Html Body | The Email HTML Body represents the body of the Email in HTML Format. |
| Email Additional Content | The Email Additional Content represents the additional content that should be displayed in Email Content apart from the HTML Body. |
| Email Message |  |
| SMS API Account SID | The SMS API Account SID represents the username which is available in twilio console and is used to send SMS Invitation using Twilio API. |
| SMS API Auth Token | The SMS API Auth Token represents the password which is available in twilio console and is used to send SMS Invitation using Twilio API. |
| SMS Phone Number | The SMS Phone Number represents the contact number from which SMS will be sent. |
| Server Name | The Server Name represents name of server on which Fewa Application is running. |

### Provider Table

|  |  |
| --- | --- |
| ProviderId | This is unique identifier in DB that stores a specific provider details. |
| Username | The Username is the username of a specific provider where provider can be doctor Name |
| Password | The password is the Password of a specific provider. |
| NameTitle | The NameTitle represents the first title of the provider Name can be Dr,Mr,Ms or Nurse. |
| Name | The Name represents Name of Provider. |
| Email | The Email Address represents Email of Provider. |
| Designation | The Designation represents Designation of the Provider. |
| MedicalDegree | The MedicalDegree is a string represented by MedicalDegree of Provider. |
| Mobile Number | The Mobile Number represents mobile number of the provider. |
| Image | The Image field stores Image ie Profile Picture of the Provider in byte format . |
| Room Name | The Room Name represents name of the Jitsi Room of Provider. |
| Room Key | The Room Key represents Key of the Room of Jitsi Video Platform. |
| Url | The Url represents Url Parameter (ie Default provider Name) that is to be appended to Application Url . |
| PracticeId | The PracticeId is a foreign key relationship that maps every provider with a specific Practice. |

### Patient Table

|  |  |
| --- | --- |
| PatientId | This is a unique identifier in DB for a specific patient. |
| Appointment Date | The Appointment Date is a Date field that stores the Date of Appointment. |
| Start Time | The Start Time stores the time when video conferencing of patient and doctor starts. |
| End Time | The End Time stores the time when video conferencing of doctor and patient ends. |
| Url | The Url consists of Url parameter (Name of Attending Doctor) that is appended to application Url. |
| ProviderId | This is foreign Key relationship that maps patients to a particular provider with the help of providerId. |

## Understanding folder structure

Currently the project has two projects one is Angular and The other server side using MVC Core.

### Angular client-side folder explanation

|  |  |
| --- | --- |
| Patient Folder |  |
| Provider |  |
| Security |  |
| Common |  |
| Models |  |
| Vendors |  |

### Server side folder structure

## Video embed

Currently the project supports two types of video embeds :-

* Jitsi
* TokBox

1. [↑](#footnote-ref-32264)