

Medical Consultation system for community
clinics in remote area



Inspiring Excellence

Md. Fuad Hassan Talukder ID:13301056

Farzana Yeasmin ID:14301111

Supervisor: Dr. Md Ashraful Alam

Department Of Computer Science and Engineering

BRAC University

Abstract

In our country many people in the rural areas suffer from various diseases but they do not get proper treatment because of the scarce of good doctors. Most of the people in our country come to the capital city Dhaka as most of the good doctors and facilities are in Dhaka but to do so they have to come from far, from various corners of the country. Not only this costs a lot but many of them become more sick in the process of doing so. Our aim is to build a system that can reduce peoples suffering as well as reduce the death rate of our country by giving them consultation. For this we need a video calling process. Through which the patient and doctor can communicate with each other. The Doctor can check the patient's initial report and also the history (if available) and will give the necessary treatment. The main role of our project is played by a video call and for this we used WebRTC, which enables real time communication in browser. For the platform we used Node.js, express and MongoDB. There are more than 14000 community clinics in total in rural areas of Bangladesh. The patient can come the nearest community clinic and tell the representative person about his/her problem and on the basis of that through our internal process he/she will get necessary consultation through video call. We tried to make the system less expensive and also reliable for the rural people. We hope this sort of system will play a vital role in our medical care system.

Declaration

We hereby declare that this thesis is completely based on the results obtained from our own work. Proper acknowledgement has been made in this paper while using all the material. This thesis, neither in whole or in part, has been previously submitted to any other institute for the award of any degree or diploma.

Signature of supervisor

Dr. Md Ashraful Alam

Assistant Professor

Department of Computer Science and Engineering

BRAC University

Signature of Author

Md. Fuad Hasan Talukder

Farzana Yeasmin

Acknowledgement

First of all, we would like to thank Allah for enabling us to work on this project which has been a great learning experience for us. By the grace of Allah, we could put our best efforts and try our best to successfully complete it on time.

Secondly, we would like to convey our gratefulness to our supervisor Dr. Md. Ashraful Alam for his contribution and guidance throughout the whole phase of our work and also to write this report. From the very beginning to the end of the work he has provided us with all kinds of help and inspired us to move forward to our goal.

We are also very thankful to our parents, friends, and well-wisher who have helped us directly or indirectly while working on our research and continuing our work. We would also like to acknowledge the work of our fellow researchers.

At last, we would like to thank BRAC University for giving us the opportunity to conclude the research and for giving us the chance to complete our Bachelors Degree.

Contents

0.1	Chapter 1	6
0.1.1	Introduction	6
0.1.2	Motivation	7
0.2	Chapter 02	8
0.2.1	Background Study	8
0.2.2	Related Work	9
0.3	Chapter 03	10
0.3.1	Architecture of the proposed model	10
0.3.2	Work Procedure	17
0.3.3	Implementation and Results	25
0.4	Chapter 4	30
0.4.1	Future Plan	30
0.4.2	Conclusion	31
0.5	Reference	33

List of Figures

1	Basic Concept of Proposed Consultation System	10
2	System Architecture of proposed consultation system	11
3	Doctors End	13
4	Patients End	14
5	Work-flow Diagram of proposed consultation system	16
6	Database	20
7	Data saved in cloud database in mLab	21
8	Restful routing table	22
9	Relationship diagram of database.	23
10	Deployed in Heroku	25
11	IDE	26
12	File tree of Medico	27
13	Landing Page of our platform	28
14	Registered Doctors Pool	29
15	WebRTC implemented	29

0.1 Chapter 1

0.1.1 Introduction

Computer technology has gone through very drastic changes and the changes have also influenced many fields and revolutionized function. One of the major fields which has been revolutionized by computer technology is medicine and health-care field. It has brought many changes to this field ranging from the organization of patients data to the management of operating theatres. Medical expert system has also created with many application systems to complement the duties of a doctor, online consultation between doctor and patient and so one.

Medical expert systems have been applied in a wide range of industrial and commercial problems including diagnosis, planning, scheduling, decision support and process monitoring and control. These systems capture and deliver knowledge that is not easily represented using traditional computing approaches. Many countries now use the Web Based Online Medical Diagnosis System (WOMEDS). WOMEDS will be explained in terms of three modules such as registration and administration, diagnosis and treatment and health monitor and tips. Web based medical treatment is becoming very popular and also helpful now a days. Sometimes people are not able to visit doctors because of time issue and also for serious health problem, so they can use the software and communicate with respective doctor.

Our country is still lagging behind from this technology but we need this type of software for the sake of our rural people specially, because they suffer a lot. Due to distance sometimes they cannot able to visit clinics, on the other hand, Doctors are not that much efficient, so they suffer a lot and at the end lots of people die without treatment.

0.1.2 Motivation

All lives matter. When it comes to medical treatment haste and availability of doctors or equipment is of utmost important. There are many people in the world out there who are dying just because they can not receive the right treatment in the right time. If they were to receive treatment in the right time, their lives could be saved which is the most precious thing in a persons life, their lives.

In our country many people in the rural areas suffer from various diseases but they can not get proper treatment because of the scarce of resources. By that we mean the good doctors. Most of the people in our country come to the capital city Dhaka as most of the good doctors and facilities are in Dhaka but to do so they have to come from far, from various corners of the country. Not only this costs a lot but many of them becomes more sick in the process of doing so. So we thought about what can we do for the people so that their suffering will be lessen and by doing so we came up with an idea which can prove to be quite effective.

Our aim is to build a system that could reduce peoples suffering, as well as reduce the death rate of our country by giving them consultation. For this we can use a video calling process. Through which the patient and doctor can communicate with each other and talk frequently. The Doctor can check the patient initial report and also the history (if available) and will give the necessary treatment. The main role of our project is played by a video call and for this we used WebRTC, which enables real time communication in browser. Beside this, there will be community clinics which plays a important role in our system. The patient can come the nearest community clinic and tell the representative person about his/her problem and on the basis of that through our internal process he/ she will get necessary treatment. We try to make the system less expensive and also reliable for the rural people. We hope this sort of system will play a vital role in our medical care system.

0.2 Chapter 02

0.2.1 Background Study

Different databases for example- MEDLINE and Inspec were searched for related publications mainly within the past five years. Multiple search terms were used, combining online consultation with health or medical, using the MeSH term remote consultation, or using e-visit, e-consultation, and video consultation. After multiple searches we found a mixed review on online consultation or video Consultation.

In 1998 the Swedish noncommercial public health service Infomedica opened an Ask the Doctor Service on its Internet portal. At no charge, anyone with Internet access can use this service to ask questions about personal health-related and disease-related matters. A study found in Australia showed that only 29 percent of young people were willing to participate in a video consultation regarding their sexual health issues, while 63 percent gave higher preference to telephone consultation (Garrett et al., 2011). Another review showed a slow uptake of telepsychiatry in Australia despite its reported successes internationally. A recent media release by the royal Australian college of general practitioners (RACGP) stated that the service delivery model of certain OMC(Online Medical Consultation) providers adds more difficulty for doctors, who may have to diagnose the patient without fully understanding the medical and social context , or being able to do a physical examination. On the other hand, lots of people are appreciating these online based consulting systems because they can save their time and the cost is lesser than visiting a doctor face to face. Research found that People especially the women appreciate this type of technology, because they sometimes feel shy to discuss their problem in front of other rather than discuss it only with Doctor. Also the working women have a time issue to visit to a doctor. According to CBHC(Community based healthcare) (June, 2016) there are one clinic for 6,000 population and more than 13136 community clinics in all over the country and which can come to use in this situation. Around 38 people visit these community clinics per day.

0.2.2 Related Work

The literature provides so many examples of Web-based infrastructures which is aiming to improve the management of patients diseases. Few of them are based on old communication technologies. We are trying to present a summary of the main solutions that uses WebRTC.

Here we try to focus on recent solutions some of which use WebRTC. We provide a correlative analysis of some representative works. A recent work uses in addition to the audio/video communication part of WebRTC, the Data Channels to exchange medical data between different medical entities. Moreover the patients have access to medical sensors. In [15], the authors propose an e-health platform, providing specific services to the diabetic persons (Type 1 diabetes mellitus (T1DM)). The proposed solution uses medical sensors and a humanoid robot. The robot interacts/dialogs with the patient in order to get better information on his state. All of the data (sensors and dialog) are collected by the robot, and then sent to the health carers. The work [14] presents an extension to WebRTC to enable peer-to-peer exchange of sensor data, and a proposal to enable Web applications to access sensor data and to bring nearby sensor streams to Web applications and multimedia communication over the Web. They mainly propose to extend the MediaStream API of WebRTC in order to manipulate also the sensor streams and to provide them to the customers. The approach is only theoretical without any implementation. In [13], 4 Authors Suppressed Due to Excessive Length the authors present a video conferencing system allowing online meetings between remotely located care coordinators and patients at their homes. They use a special device (TeleMedCare) to monitor the vital sign of the patients. In this work, there is no interaction with medical sensors apart from the TeleMedCare.

0.3 Chapter 03

0.3.1 Architecture of the proposed model

The name of our proposed model is Medico. In this model there are several interfaces.

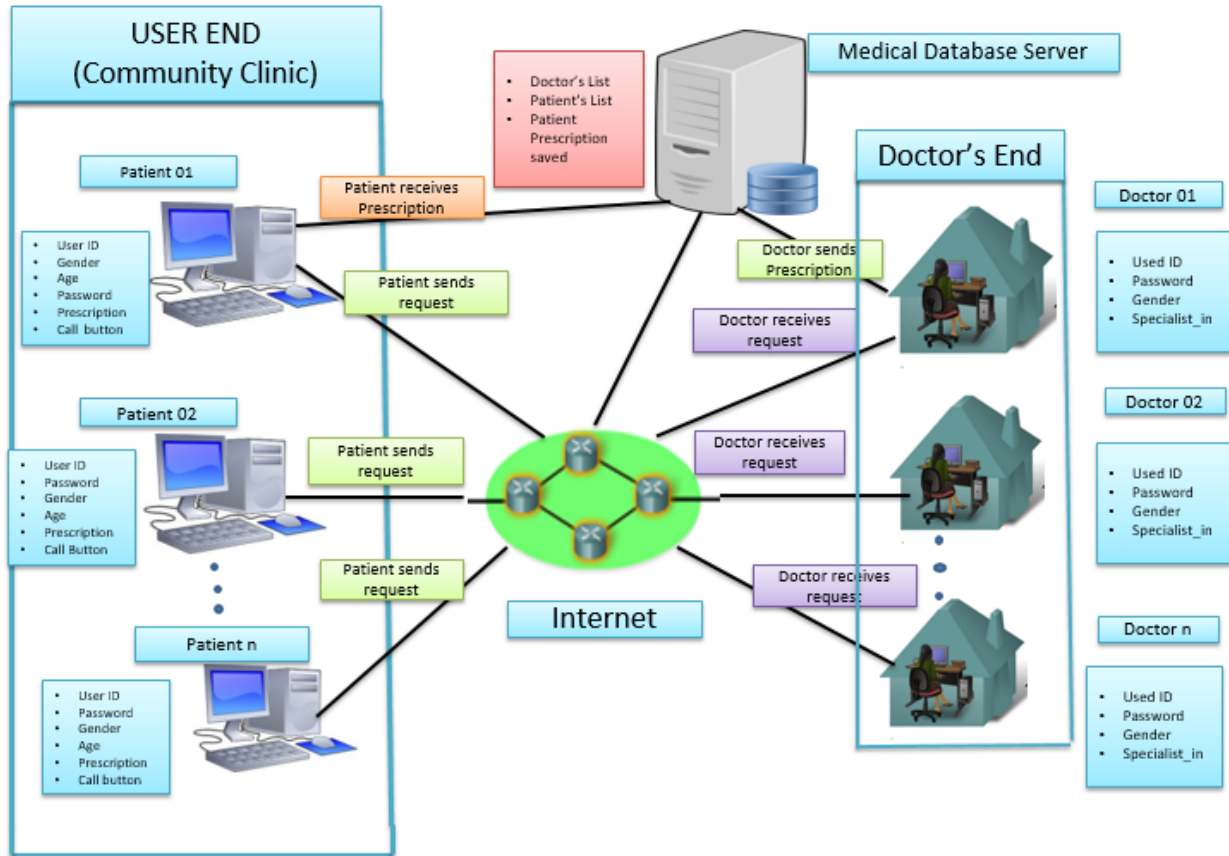


Figure 1: Basic Concept of Proposed Consultation System

Medico

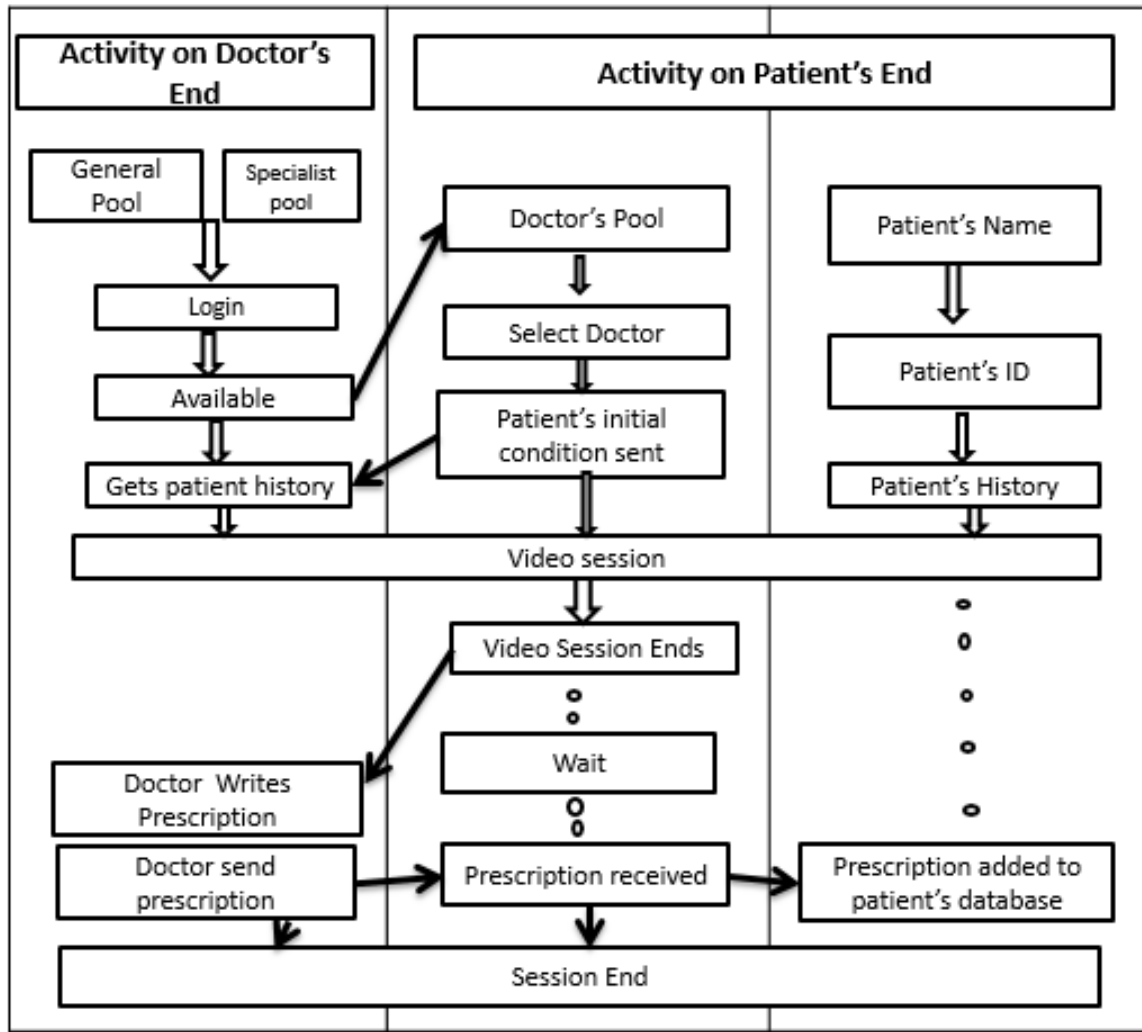


Figure 2: System Architecture of proposed consultation system

Doctors Side:

There are 2 categories of doctor, one is consisted of General Doctors and another one is consisted of Specialized Doctors. General doctors are the doctors who fall in the general category meaning that patients whose conditions are not very critical can consult the doctors from the generalized doctors pool.

On the other hand we have specialized doctors who specializes in any sector. For the patients who need extra attention and whose sickness is more complex than others can consult a specialized doctor. Like people who are cancer patients can consult a specialized

doctor.

- First of all the doctor will login into their account and when she or he logs in, the doctor will be added in the available and active Doctors pool and he will be available for consultation.
- Before the first patient call him he will get the initial condition of the patient in a message like format where he will assess the primary situation.
- Then when he is ready the the video call session will begin.
- After the doctor has completed the video call he will write another prescription and send it to the community clinics patient page.
- The session will end.

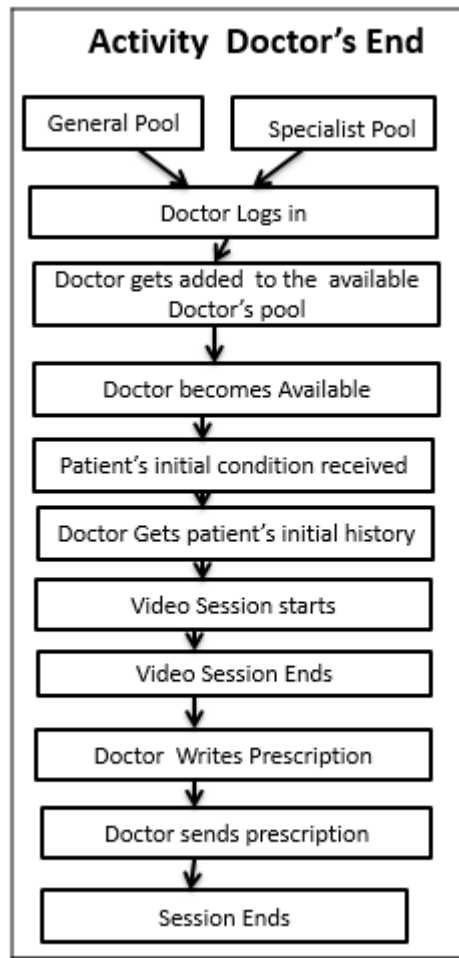


Figure 3: Doctors End

Patient's Side:

- Patient will log in.
- Search for available doctors.
- Finds doctors and connects to doctor.
- Before starting video call Patients sends in the primary condition report.
- After signal the video call session with doctor starts.
- After the video call is finished patient waits while the doctor writes prescription.
- Patient gets the prescription.

- Session ends.
- Patient history is saved in database so that it can be viewed later and any doctor can see it later.

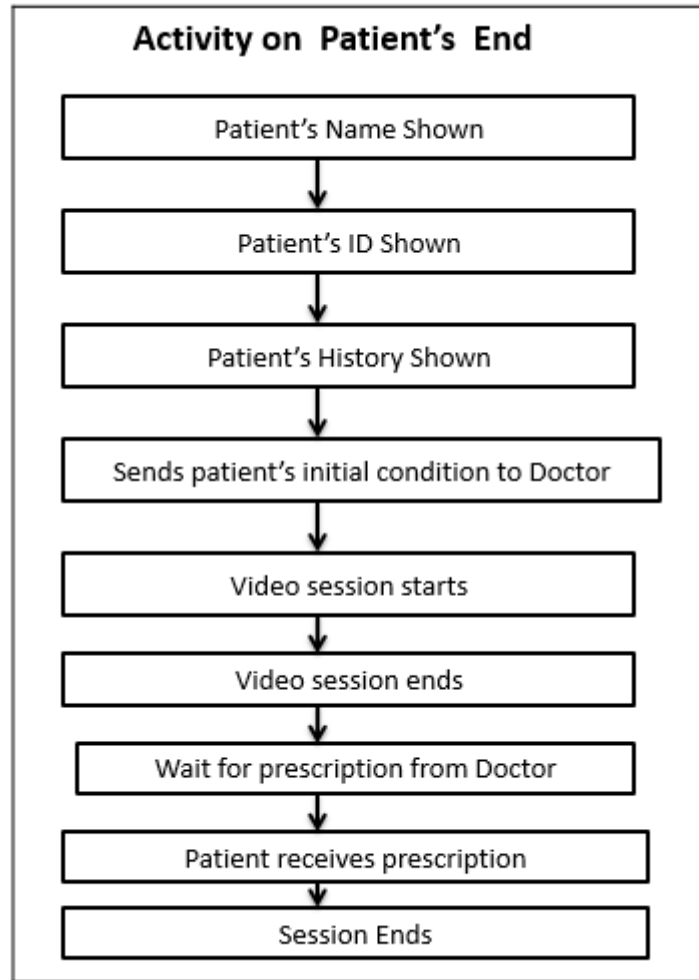


Figure 4: Patients End

Our proposed model will work like, the user needs to log-in to our page. Our previous description showed what will be the situation if a doctor is available but in case if a doctor is not available then the following steps will happen.

- After login, patient will press the button the of Available Doctors.
- If in case no Doctor is available at that moment, then the system will show $DP=0$, that means Doctor pool is 0 and the patient then need to wait for 1 min.

- After 1 min the patient can again start the process by pressing the Doctor Available button. At this moment, if the Doctor is available then the system will show a option for the patient to choose the General Doctor (GD) or Specialist Doctor(SpD).
- Next, the user/patient can choose his/her Doctor, according to his/her need. After choosing General or specialist Doctor, the system show the patient the pool of the Doctors, which is basically the list of specialist or General Doctors list.
- Then the patient can request for his or her Doctor and if the doctor is free at that moment then the patient can communicate with the doctor through video calling process.
- On the other hand, if the requested Doctor is already in an appointment with other patient then the patient need to wait or he/she can choose other doctor from the pool.

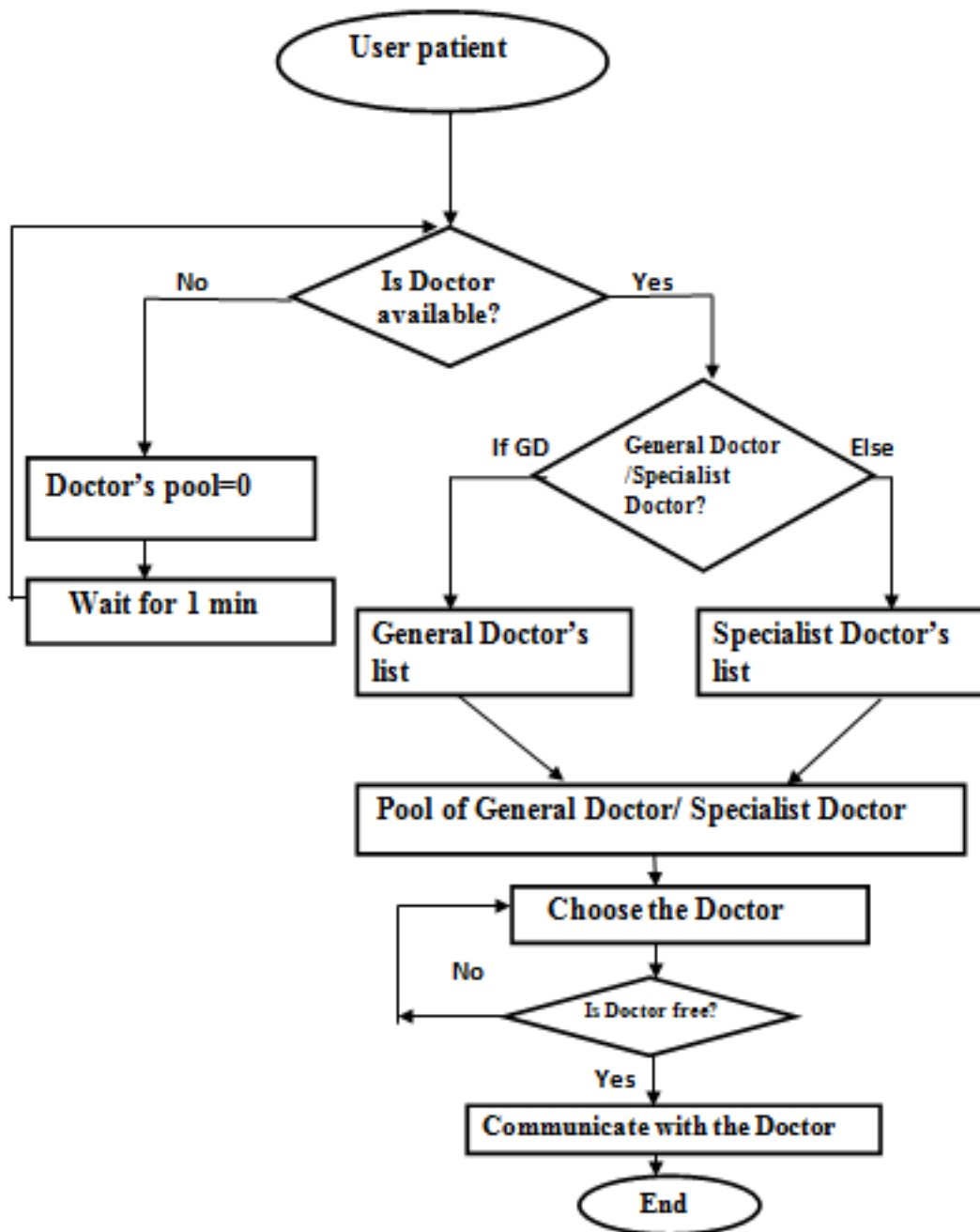


Figure 5: Work-flow Diagram of proposed consultation system

0.3.2 Work Procedure

1. In the backend part we send our queries to our ISP. Inside the ISP, the DNS takes the domain name and turns it into an IP address. A request is sent to the required IP address via HTTP. The requested server figures out what exactly what we are asking for. The server responds with a combination of HTML, CSS and JavaScript. We use HTML(Hypertext markup language) and also CSS(Cascade Styling Sheets). HTML creates content page. For example- Written text and CSS for designing the layout, visual effect. HTML itself is called authoring language, used to create documents in the web-page and CSS is a language used in web pages for describing the presentation create by HTML. We used Bootstrap, which is a framework and that helped us to design our web page easily. It is combined with HTML, CSS, JavaScript. Bootstrap is a framework that helps a developer to save his coding effort by giving less CSS functionality.
2. The most important part of our work is JavaScript. We used JavaScript as our client side scripting language. We wrote JavaScript code in the HTML page so that when our patient or user sends request, the script then sends it to the browser and browser starts working on the basis of user request. JavaScript has the ability to produce the same result on all modern browsers. In client-side execution, no matter where we host JavaScript we execute always on client environment to save bandwidth and make execution process fast. Without reloading the entire page, JavaScript can load content into the document if necessary. We can write JavaScript code between `<script>` tags. Loading and running scripts is blocking and immediate. That means when the browser comes across the `<script>` , the browser stops reading the HTML, instead loads and run the script. The browser starts reading and rendering the HTML code after running the script.
3. After JavaScript, the another important programming interface that we use is DOM. DOM stands for Document Object Model. It is a programming interface for HTML.

It represents the web page as nodes and objects so that programs can change the document structure, style, and content. That way, programming languages can connect to the page. DOM is a object oriented programming language of web page, which can modify by scripting language, like- Javascript. Without it, the JavaScript language would not have any model or notion of web pages, HTML documents, and their component parts. If we run getElementByDocName method the code will return us the list of Doctors name and HTMLTableElement is the interface of accessing the HTML table.

4. As a framework of JavaScript, we used jQuery. It is not a programming language but it makes easier to use JavaScript on our web page. It is less weighted JavaScript library. The jQuery library contains - CSS manipulation, HTML/DOM manipulation, effect and animation and more. It can not exist on its own. It works with JavaScript to make JavaScript much easier and it reduces the development time. Where JavaScript needs 3 to 4 lines of code to show a result, jQuery can do it within one single line of code.
5. To interact with the computer we use command line. Command line is the ultimate seat of power on our computer. Command line can perform amazing power of enhancement and speed, dominate our computer and helps us to do precisely what we want to do. Commands itself have three parts- the utility, the flags, and the arguments. The utility always comes first. ls is a utility. Utilities indicate the general idea of what we want so they are also known as commands all on their own. Most of the time, we can simply run a utility all by itself, without any flags or arguments. Most commands only have one utility. We also use the -l flag to indicate to the utility that we want more information than it usually provides and so it shows us the directory contents in a long format.
6. After that, we use Node.js to run our JavaScript code on the server side. Node.js uses non-blocking I/O models that helps us to make our work efficient. Along with that

we use npm. Npm has a command line tool which helps us to use different packages in our code. Node.js has some extraordinary features when it comes to developing a real-time application. Node.js can be used in front-end and back-end. It helps us to speed up our framework.

7. We used express as a framework of our work. Express itself is a web developing framework. It is designed for building web applications and APIs. Express is the back-end part of our MEAN stack. Here we use `--save` flag which works for installing our packages, we used `package.json` to hold relevant meta-data and it also can handle the projects dependencies, along with that we used `npm init` which works for creating new packages.
8. We used Embedded JavaScript (EJS) to return to our code to get a maintainable HTML structure. EJS is a JavaScript template library. It is a client side JavaScript code, which is embedded within HTML documents between a pair of `<script>` and `</script>` tags. In JavaScript we use `npm install ejs` to use the dependency.
9. To connect our application, we use Application Programming Interface(API). APIs are interfaces that interact with our code. It actually works as our messenger. We use it to send our request to receiver and also get the response from the receiver.
10. To connect our data and also to interact with all the information we need a database that can give us more flexibility. Where if we change the field of our document, database structure will change over time so we used MongoDB Database. MongoDB is an object-oriented, simple, dynamic database. It helped us to store data objects as separate documents inside a collection. We use different types of Mongo command like- `find`, `show`, `update`, `insert`. In the backend we are using mLab which is cloud hosted MongoDB to store the data.

The screenshot shows a dark-themed web interface for a database. At the top, there is a navigation bar with five tabs: 'Collections' (highlighted in blue), 'Users', 'Stats', 'Backups', and 'Tools'. Below the navigation bar, the 'Collections' section is displayed. It includes two buttons: 'Delete all collections' (with a trash icon) and '+ Add collection'. Below these buttons is a table with the following data:

NAME	DOCUMENTS	CAPPED?	SIZE
comments	2	false	8.20 KB
doctors	3	false	11.44 KB
users	3	false	13.94 KB

Each row in the table has a small trash icon on the right side, indicating that individual collections can be deleted.

Figure 6: Database

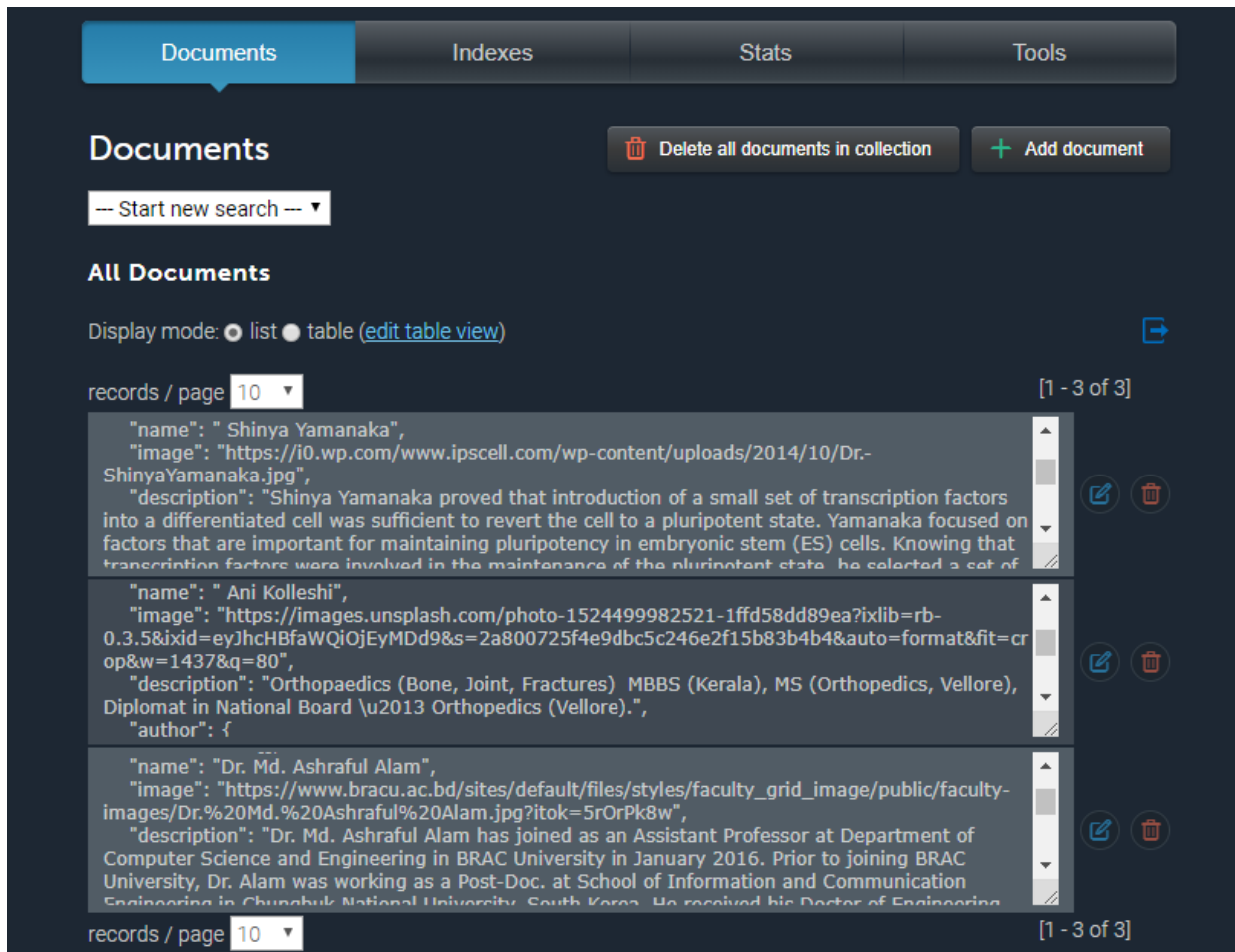


Figure 7: Data saved in cloud database in mLab

- In the routing part, to map the HTTP and CRUD(CREATE,READ, UPDATE, DELETE) functionality, we used RESTful routing. REST stands for representational states transfer. It is a tool to find out routes. There are 7 different RESTful routes that we use. At first, to list our Doctors' name and info, we use index routes (/doctors) to list all the Doctors and it is a GET request from HTTP. Then (/doctors/new) to show all the new Doctors form. We use create routes (/doctors), which is a POST request from HTTP and it create a new doctor form which can be redirect. To show the specific information of a doctor if we write (/doctors/:id) it will show us the specific information about a doctor. To edit the information we use (/doctors/:id/edit), it is a GET request from HTTP and it will edit one doctor's information. To update information, (/doctors/:id) is used and it is a PUT request from HTTP which helps to update specific doctor's information and it can redirect. Finally, to delete or remove a

doctors data from the list, we use (/doctors/:id/) as a delete request from the HTTP which will delete the specific doctor's information from the list.

Name	Path	HTTP Verb	Purpose
Index	/doctors	GET	List all doctors
New	/doctors/new	GET	Show new Doctors form
Create	/doctors	POST	Create a new doctor, then redirect somewhere
Show	/doctors/:id	GET	Show info about one specific doctor
Edit	/doctors/:id/edit	GET	Show edit form for one doctor
Update	/doctors/:id	PUT	Update a particular doctor, then redirect somewhere
Destroy	/doctors/:id	DELETE	Delete a particular doctor, then redirect somewhere

Figure 8: Restful routing table

- To make the relationship between our data, we use association. Association creates relationship between two entities based on common attributes. This relationship could be one-to-one, one-to-many, many-to-many. The association also allows entity objects to access the data of other entity objects through a persistent reference. The common example of association is Facebook, like as- (user, post, photos, share, comment, like) these all entities are connected with each other.

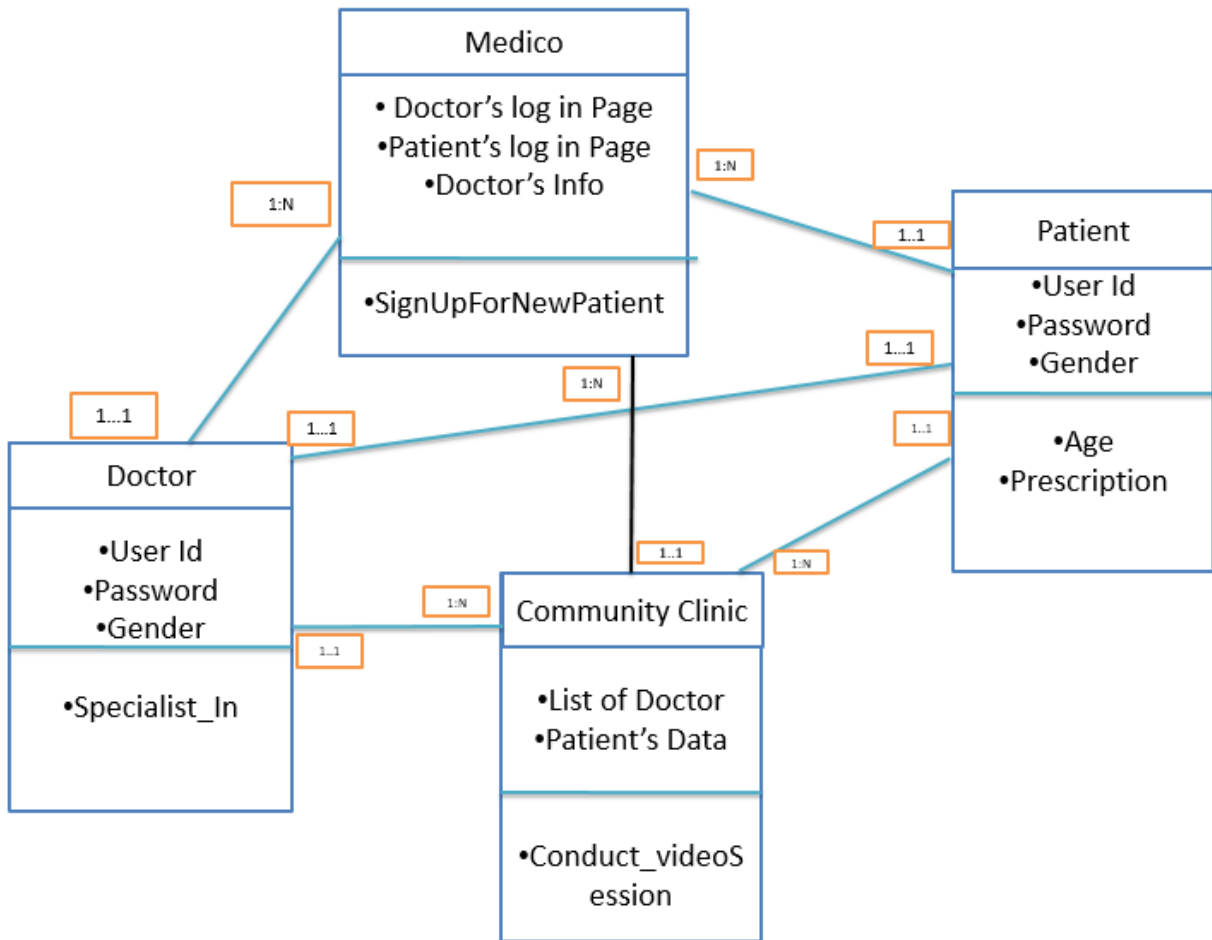


Figure 9: Relationship diagram of database.

Here, we can see the relationships between a doctor and a patient. A doctor can consult many patients and an patient can consult with various doctors but at a moment only 1 doctor and 1 patent will be connected. A patient will have all the prescriptions saved in his profile so that if that patient decides to consult a doctor at a later stage the doctor will be able to see the patients history and understand more about the patient.

13. In Medico we create a doctor file and inside it we insert a models directory and a module.exports file. Which is basically a Javascript file. Inside the models we execute comment.js, doctor.js and user.js file.

In Doctors schema, we add name, image, description, id and the username of the Doctor. It contains all the information of the Doctor.

14. We use Website authentication for our user, so that they can access their account in a process which is called user authentication. For this human to computer interaction process user need to create an account using his/her username, password and they need to give this information everytime they want to login the account. We need authentication to make our system more reliable to the user. Only the registered user can access our system with his/her valid credentials. The purpose of logging in to the application is to let it know who you are, so it can grant you the necessary privileges.
15. We choose WebRTC. WebRTC stands for Web Real-Time Communication. It is an open source project which can enable real-time communication of audio, video and data in Web. WebRTC uses RTCPeerConnection to communicate streaming data between two browsers. The main objective of WebRTC is, it can create peer-to-peer connection based on peers external network IP address and network bandwidth. WebRTC helps to add standard APIs (Application Programming Interfaces) and also built-in real-time audio and video capabilities and codecs to browsers without a plug-in. By writing a few lines of JavaScript, we can add high quality peer-to-peer voice, video, and data channel communications to their collaboration. It also helps to make our system secure by Javascript APIs. WebRTC allows a desktop and also the mobile browser to access the devices microphone and video camera. The browser Then notify the user that an application is requesting access to their computers camera and microphone. Once the user/ patient allows to access the devices, WebRTC then create individual streams of transferable audio and video data which is generated by that input devices. This data is then transmitted via network data channels established. As we already mentioned, WebRTC has several Javascript APIs. like- if we write getUserMedia it will capture audio and video. RTCPeerConnection will steam audio and video between two users etc.
16. Next, we used Heroku, which is basically a server side tool. It is a cloud based

development platform. It uses Git as the primary means for deploying a system. It will only be applied when the root directory has a package.json file. The package.json file defines the dependencies which must be installed with our system. In case a dependency is missing from our package.json file, then the system will show us the error that a specific module is absent. The main reason for using Heroku is because it is already pre-planned. Heroku helps us to run our system on the server. To do this, we need to create a Simple Node.js Express App and run it locally. Then, we need to log into Heroku from our terminal and deploy our Node App, finally opening the application in live.

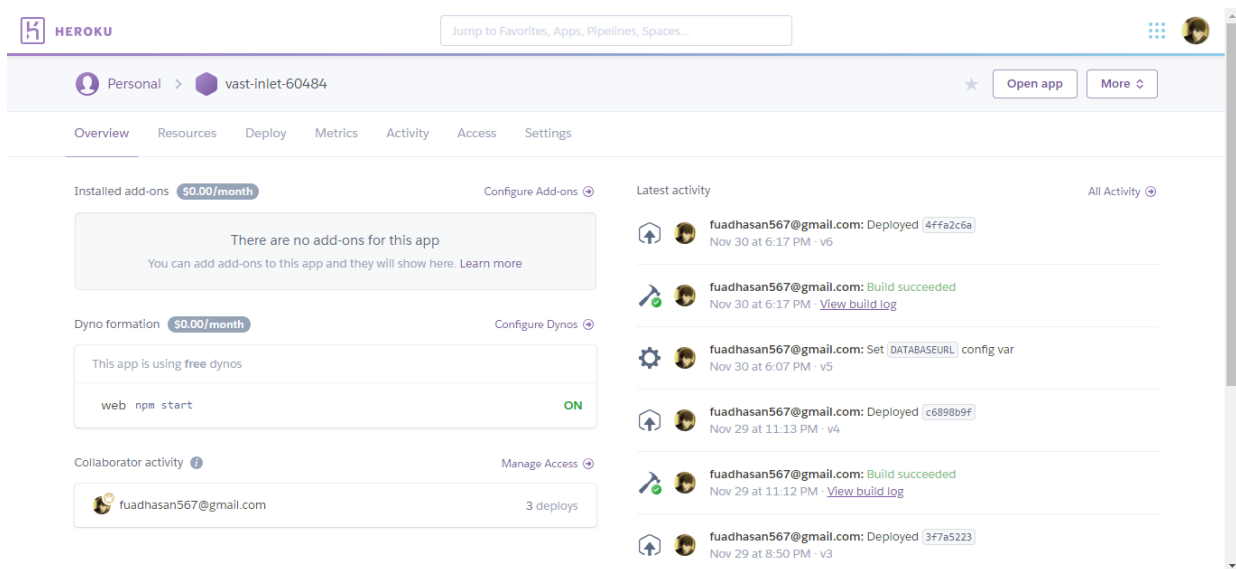


Figure 10: Deployed in Heroku

0.3.3 Implementation and Results

Programming and concepts languages used:

1. HTML
2. CSS
3. JavaScript

4. Node.js
5. Express
6. MongoDB
7. jQuery
8. Bootstrap
9. WebRTC

We did most of our coding part in Cloud9. Cloud9 is a cloud-based integrated development environment (IDE) that let's write, run, and debug code with just a browser. Cloud9 makes it easy to write, run, and debug serverless applications. Cloud9 also provides an environment for locally testing and debugging using command prompt. This allows to iterate on your code directly, saving time and improving the quality of code. We developed a part and tested and debugged it on Cloud9.

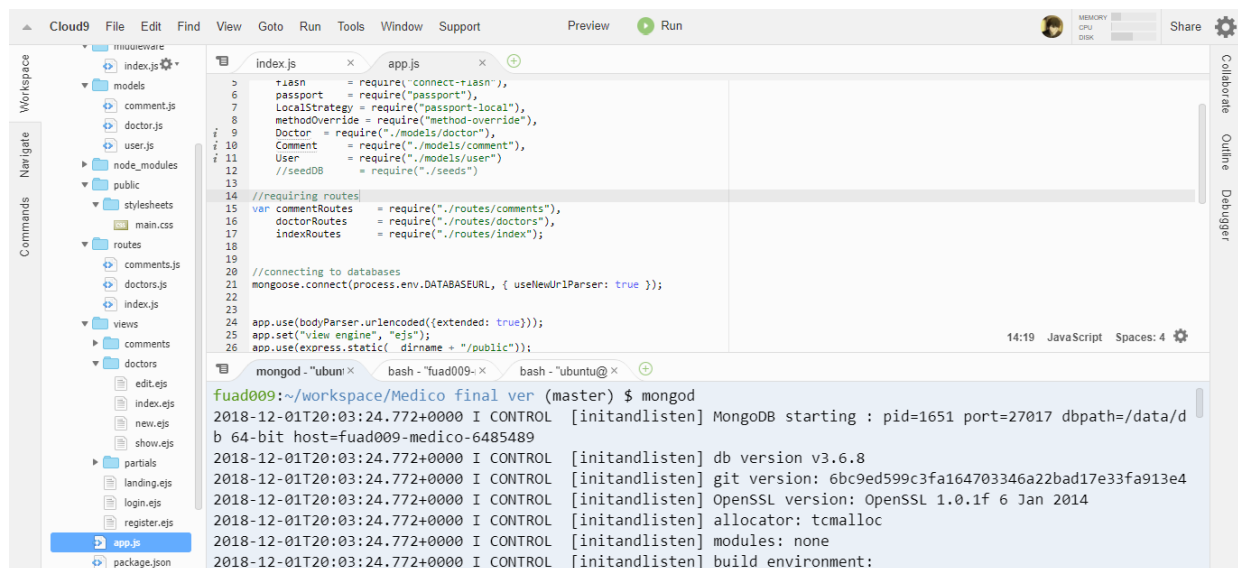


Figure 11: IDE

Our file structure developed in this IDE is given below.

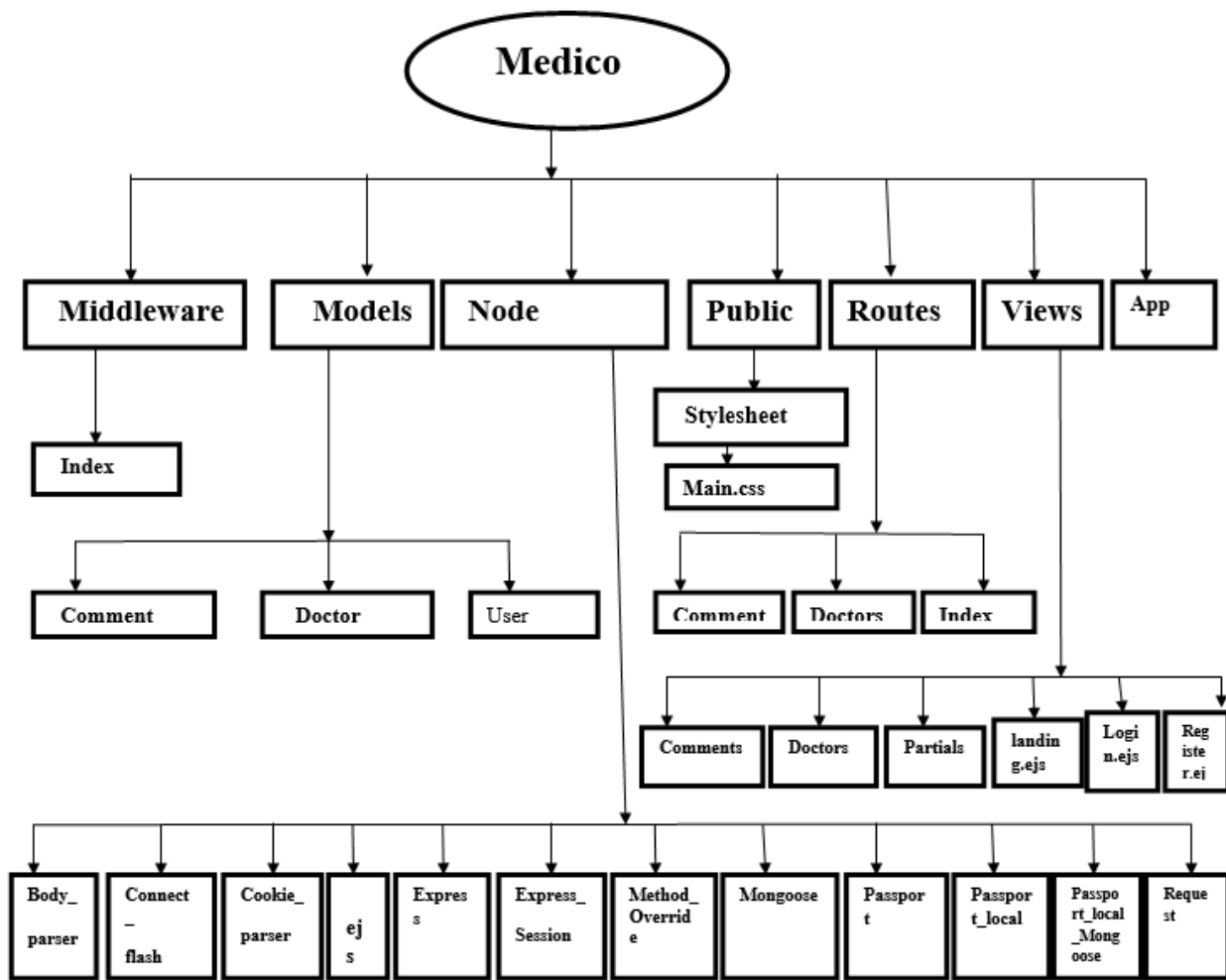


Figure 12: File tree of Medico

This the site we were able to make. Basically this is the interface of the platform that both doctors and patients will use for interacting and consultation.

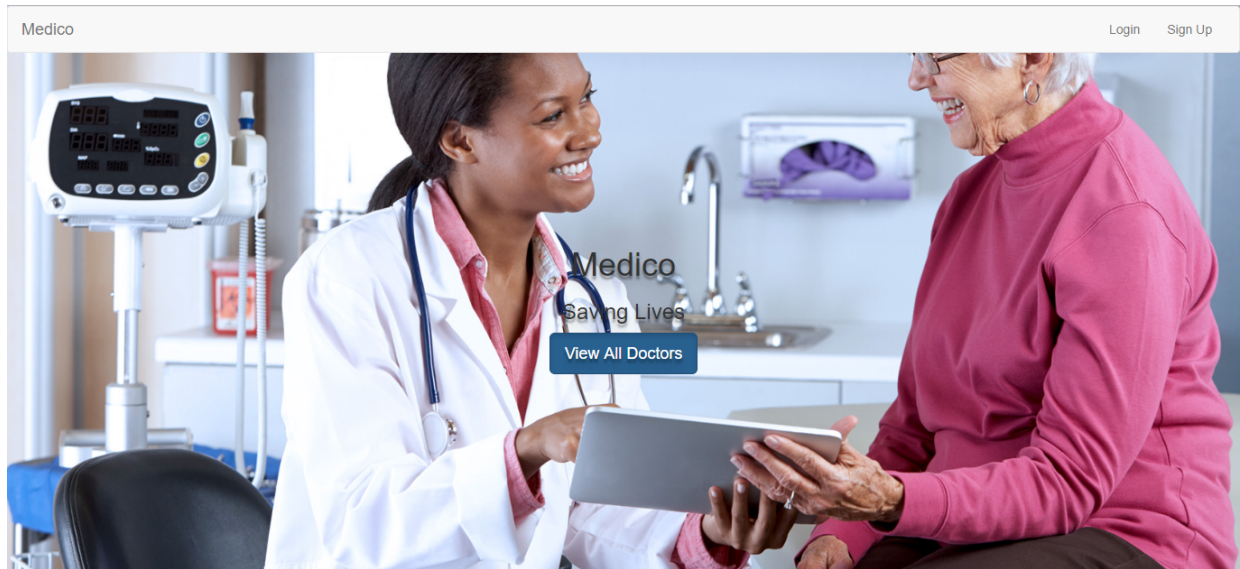


Figure 13: Landing Page of our platform

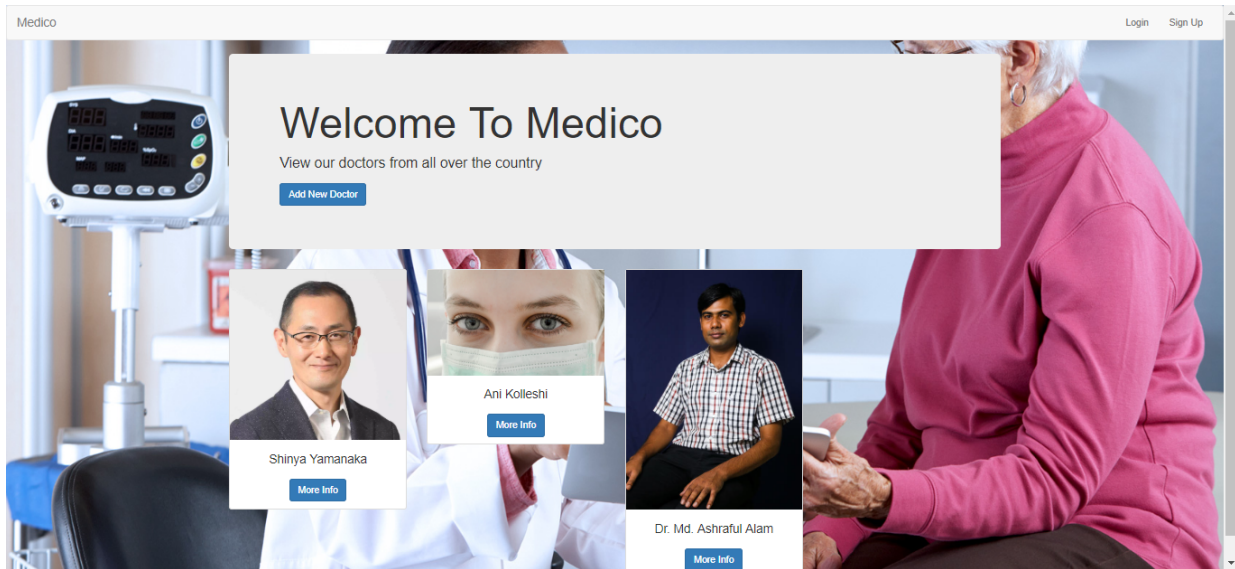


Figure 14: Registered Doctors Pool

This page shows the list of doctors enlisted in the doctors database.

After implementing WebRTC:

This is the video we get after implementing WebRTC. One window will be of doctors and another window will be of patients.



Figure 15: WebRTC implemented

0.4 Chapter 4

0.4.1 Future Plan

As our future work, at first, we will try to improve the efficiency of our model so that it can play a significant role in healthcare in Bangladesh. Currently, we have a lot of scope to improve the system.

If we can apply machine learning we will be able to predict what disease the patient has using various machine learning algorithms. We can then send the predicted data to a doctor first so that the doctor can have ease in determining the disease correctly. We can also get a clear picture of which disease is affecting which area by analysing this data. We can categorize diseases according to area then. This will help us in a lot of ways to eradicate various diseases by prevention.

We will take as much as instances of data to increase the accuracy level. The more we take data for training our system, the accuracy will be increased more. Along with that, we will work on adding more features to our system. For example- adding to do different health tests of a patient. The more we provide information to our system and train it with the information, the more it can improve the accuracy level. If we could get more data from the patient, the system could provide better results. So, gathering practical data can be an important task to be carried out to achieve a better accuracy level. We can use processed data which will bring a better outcome and we must work on it in the future.

One of our aims with our project is to make the system efficient and also reliable for the user, to do so, we will take help from different experts in the medical field to create final attributes, which we can apply in most cases. Understanding a disease through consulting in a video call is tough for a doctor, so, we will try to make the system more realistic by adding more features. Though it is a video calling consultation system, so, there could be some prob-

lem regarding appointment with the Doctor and the availability of the Doctor on specific field. To solve this problem we will try to appoint more Doctor in every specific disease. We have a dream to serve the patient 24 hours by our system, which we could not do at present.

In future, we will try to add a feature of rating Doctor. Through this feature the patient who already consult to a Doctor can rate the Doctor. It will be helpful for the new patient to make quick decision in terms of choosing a Doctor. Last but not the least, we have aim to make our system mobile application based also, so that people can use it easily.

0.4.2 Conclusion

In this work, we have proposed a system which can play a important role in terms of health-care. As a core feature of our system is video calling, we tried to give immediate treatment to the rural people and try to reduce their suffering. Through the video calling communication, the Doctor communicate with the patient, asking all the problems he/she faces, give necessary medicine and also test (if necessary) or if necessary, doctor will recommend him/her to visit physically.

Using WebRTC, we connect the video calling process and also use several technologies have been used to run it in the web platform. The system will also be able to help the patient to choose his/her favourite Doctor. By our system, the user will be able to get emergency support which is found missing in many cases in Bangladesh, as a result every year lot of people die. The aim of our project is to develop a system which can act as a tool in health-care especially give immediate support to save rural people life. They are the main sufferer and most of the death rate in our country is because of the death occurs in rural area, they deserve a better and immediate treatment. We try our best to make our system reliable to them and still work on it to make it more effective. Last but not the least, we believe that our proposed model will be helpful enough to improve the overall

health-care system in Bangladesh and create a positive impact in health-care field .

0.5 Reference

1. F. Rhinow, P. P. Veloso, C. Puyelo, S. Barrett and E. O. Nuallain, "P2P live video streaming in WebRTC," 2014 World Congress on Computer Applications and Information Systems (WCCAIS), Hammamet, 2014, pp. 1-6.doi: 10.1109/WCCAIS.2014.6916588
2. Z. Hao, "Adaptable P2P panoramic video streaming system based on WebRTC and WebGL," 2016 International Conference on Audio, Language and Image Processing (ICALIP), Shanghai, 2016, pp. 166-169.doi: 10.1109/ICALIP.2016.7846552,
3. S. Vashishth, Y. Sinha and K. H. Babu, "Addressing Challenges in Browser Based P2P Content Sharing Framework Using WebRTC," 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), Crans-Montana, 2016, pp. 850-857.doi: 10.1109/AINA.2016.143
4. M. Varma, H. K. Yarnagula and V. Tamarapalli, "WebRTC-based peer assisted framework for HTTP live streaming," 2017 9th International Conference on Communication Systems and Networks (COMSNETS), Bangalore, 2017, pp. 415-416. doi: 10.1109/COMSNETS.2017.7945420
5. IOT BASED HEALTHCARE MONITORING SYSTEM WITH ALERT AND NOTIFICATION SYSTEM. (2016). International Journal of Advance Engineering and Research Development, 3(04).
6. V. Lanza, "Connecting Hospitals through a Distance Learning Center and Permanent Remote Support Network - A Sicilian Project to Improve Patient Care," Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06), Singapore, 2006, pp. 38-38. doi: 10.1109/CCGRID.2006.1630931
7. Caiata-Zufferey, M., Abraham, A., Sommerhalder, K., Schulz, P. J. (2010). Online Health Information Seeking in the Context of the Medical Consultation in Switzerland. Qualitative Health Research, 20(8), 10501061. <https://doi.org/10.1177/1049732310368404>

8. Fiordelli, M., Diviani, N., Schulz, P. J. (2013). Mapping mHealth research: a decade of evolution. *Journal of medical Internet research*, 15(5), e95. doi:10.2196/jmir.2430
9. Lunt, N., Hardey, M., Mannion, R. (2010). Nip, tuck and click: medical tourism and the emergence of web-based health information. *The open medical informatics journal*, 4, 1-11. doi:10.2174/1874431101004010001
10. A Linard, I Merckaert, Y Libert, N Delvaux, S Marchal, J Boniver, A-M Etienne, J Klastersky, C Reynaert, P Scalliet, J-L Slachmuylder, D Razavi; Factors that influence cancer patients' anxiety following a medical consultation: impact of a communication skills training programme for physicians, *Annals of Oncology*, Volume 17, Issue 9, 1 September 2006, Pages 1450-1458, <https://doi.org/10.1093/annonc/mdl142>
11. Lienard, A. , Merckaert, I. , Libert, Y. , Delvaux, N. , Marchal, S. , Boniver, J. , Etienne, A. , Klastersky, J. , Reynaert, C. , Scalliet, P. , Slachmuylder, J. and Razavi, D. (2008), Factors that influence cancer patients' and relatives' anxiety following a threeperson medical consultation: impact of a communication skills training program for physicians. *PsychoOncology*, 17: 488-496. doi:10.1002/pon.1262
12. Heritage, John Stivers, Tanya. (2000). Online Commentary in *Acute Medical Visits: A Method of Shaping Patient Expectations*. *Social science medicine* (1982). 49. 1501-17. 10.1016/S0277-9536(99)00219-1.
13. Kim, J., Van, L. M., Kim, J. (2016). Load Balancing Methodology for Efficient Real-Time Streaming Service Based on WebRTC. *Lecture Notes in Electrical Engineering*, 1331-1337. doi:10.1007/978-981-10-0557-2-125
14. Azevedo, J., Pereira, R. L., Chainho, P. (2015). An API proposal for integrating sensor data into web apps and WebRTC. *Proceedings of the 1st Workshop on All-Web Real-Time Systems - AWeS '15*. doi:10.1145/2749215.2749221
15. *Telemedicine Platforms: The Potential for WebRTC*. (n.d.).

16. Telehealth WebRTC webinar with Wellbeing Consult - WebRTC.Ventures. (2015, October 29).
17. Jian, Cui, and Zhuying Lin. "Research and Implementation of WebRTC Signaling via WebSocket-based for Real-time Multimedia Communications." (2016).
18. ("Community Clinic in Bangladesh: Taking Health Services to Rural People — daily sun", 2018)
19. Singh, K., Yoakum, J. (2015). Vclick: Endpoint Driven Enterprise WebRTC. 2015 IEEE International Symposium on Multimedia (ISM). doi:10.1109/ism.2015.92
20. A Privacy Safeguard Framework for a WebRTC/WoT-Based Healthcare Architecture Saad El Jaouhari, Ahmed Bouabdallah 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC) 2018
21. ("Community clinics in Bangladesh: Bringing health care to the doorstep of rural people", 2018)
22. Browser-to-Browser Security Assurances for WebRTC Richard L. Barnes, Martin Thomson IEEE Internet Computing 2014
23. Telemedicine for emergency care management using WebRTC Vidul A P, Shibin Hari, Pranave K P, Vysakh K J, Archana K R 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI) 2015
24. WebRTC: delivering telehealth in the browser. (2016, April 13). Retrieved from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5344122/>
25. El Jaouhari, Saad, Bouabdallah, Ahmed. (2018). A Privacy Safeguard Framework for a WebRTC/WoT-Based Healthcare Architecture. 10.1109/COMPSAC.2018.10278.
26. Jacqueline Lee, J. (2014, January 20). Telemedicine Platforms: The Potential for WebRTC.

27. TelehealthWebRTC webinar with Wellbeing Consult.(2015, October 29).Retrived from <https://webrtc.ventures/2015/10/telehealth-webrtc-webinar/>
28. Home. (2018). Retrieved from <http://www.communityclinic.gov.bd/>
29. Difference between SQL and NoSQL - GeeksforGeeks. (2018).
30. Chhetri, N. (2016). A Comparative Analysis of Node. js (Server-Side JavaScript).
31. Skaalsveen, A. (2015). Development of smart devices in Node. JS. An exploratory approach (Master's thesis).
32. Ruppen, A., Pasquier, J., Hurlimann, T. (2011, December). A RESTful architecture for integrating decomposable delayed services within the web of things. In 2011 IEEE 17th International Conference on Parallel and Distributed Systems (pp. 860-865). IEEE.
33. Wandschneider, M. (2016). Learning Node. js: A Hands-on Guide to Building Web Applications in JavaScript. Addison-Wesley Professional.
34. Ortiz, A. (2014, March). Server-side web development with JavaScript and Node. js. In Proceedings of the 45th ACM technical symposium on Computer science education (pp. 747-747). ACM.
35. Yoona, H. K. R. H. J., Stankovicb, S. H. S. J. A., Koc, J. HealthNode: Software Framework for Efficiently Designing and Developing Cloud-Based Healthcare Applications.
36. Zemzans, O. (2016). EXPLORING NOSQL DATABASES: Comparison of Databases.
37. Banker, K. (2011). MongoDB in action. Manning Publications Co..
38. Chodorow, K. (2013). MongoDB: The Definitive Guide: Powerful and Scalable Data Storage. " O'Reilly Media, Inc."

39. Boicea, A., Radulescu, F., Agapin, L. I. (2012, September). MongoDB vs Oracle–database comparison. In 2012 third international conference on emerging intelligent data and web technologies (pp. 330-335). IEEE.
40. Membrey, P., Plugge, E., Hawkins, D. (2011). The definitive guide to MongoDB: the noSQL database for cloud and desktop computing. Apress.