System for Remote Medical Scheduling and Consultation

Haritha Poornachandran, Amritha Barade and Akshita Arunkumar

***Abstract*—Hospitals often face crowding and see their resources and personnel stretched thin due to overcrowding. This project aims to design a digital system that would allow smoother and more organized scheduling of appointments for both consultation as well as secondary medical necessities such as scans and tests. It also provides a platform for people with mild or sporadic symptoms to consult doctors online without needing to go to a hospital. The purpose of this is to minimize the number of people with benign conditions from crowding hospitals as well as save time and provide more instantaneous results. This program is written in Java and uses My-SQL for back-end support.**

***Index Terms*—MySQL, NetBeans, Prepared Statements, Swing Components.**

I. Introduction

Hospitals often serve a large number of patients everyday. They deal with major surgeries and trauma cases to small afflictions such as the common cold. They have the need to store all the data regarding a patients complete medical record to doctor availability, medication, scans, test results and more.

It is a common occurrence to walk into a hospital and wait for quite some time to see a physician. This is commonly seen in the cases of non-urgent problems. While a case such as an accident or rapid blood loss may garner quick response, most other cases have a waiting period as well as paper work to fill. Even in institutions with digitized records one still needs to travel to the hospital before he or she can see a doctor.

This along with the cost of in person consultations has made many people reluctant to visit hospitals for what they may think is a minor problem. However, this can lead to patients ignoring symptoms that may point to a more serious underlying problem. On the other hand this could also lead to patients self medicating after simply looking up online for their symptoms with some over the counter medication that is unsuitable.

This project aims to solve all these problems by providing a working model where patients can maintain accounts with the hospital with all relevant information including any current treatments available to the doctor. Moreover it also has a simple feature to allow booking of appointments for a particular time slot and date with a particular doctor practising a particular specialization. Most importantly this project also gives the patients the ability to fill in a form with their base symptoms and have it sent to a doctor for online consultation. The doctor based on his or her discretion can either prescribe a cure, book an in person follow up with the patient or schedule a scan or test to gain more information.

II. Background

This project is built using Net-beans Swing components for the front end, with Java based prepared statements to interact with the Database. The Database is defined using My-SQL.

A. Net-Beans

Net-Beans IDE is a free, open source, integrated development environment (IDE) that enables you to develop desktop, mobile and web applications. The IDE supports application development in various languages, including Java, HTML5, PHP and C++. The IDE provides integrated support for the complete development cycle, from project creation through debugging, profiling and deployment.[1] The IDE fully supports Java EE using the latest standards for Java, XML, Web services, and SQL and fully supports the GlassFish Server, the reference implementation of Java EE.[2]

B. MySQL

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL)[3]. MySQL is based on a client-server model. The core of MySQL is MySQL server, which handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded (or linked) into seperate applications. MySQL is written in C and C++ and accessible and available across over 20 platforms[4]

III. User Interaction

The primary actors on this system are the patients and the doctors. The doctors have their information already stored in the hospitals database and hence do not need to register separately on this portal. The patients however, need to create an account on their first entry. Each can then login and perform any of the available actions available in their homepage. Depending on whether their account is a patient or doctor account they are provided the choice to either schedule appointment, Fill online consultation form or cancel appointment or for doctors, View patient records, View upcoming appointments,

This paper is written as part of an assignment for the Comprehension and Technical Report Seminar (CS7713) and is based on a project for the Case Tools Lab carried out by Haritha Poornachandran and Keerthana Sundaresan, under the guidance of Ms. R.Bhuvaneshwari.

Haritha Poornachandran, 2017103057, current student of Computer Science and Engineering at CEG, Anna University. (e-mail: 2017103057@annauniv.edu.in)

Amritha Barade, 2017103055, current student of Computer Science and Engineering at CEG, Anna University. (e-mail: 2017103055@annauniv.edu.in)

Akshita Arunkumar, 2017103508, current student of Computer Science and Engineering at CEG, Anna University. (e-mail: 2017103508@annauniv.edu.in)

respond to online queries or give a prescription and then discharge. For some cases the doctors can also schedule a scan for a particular patient.See Fig.1 for Use case diagram.

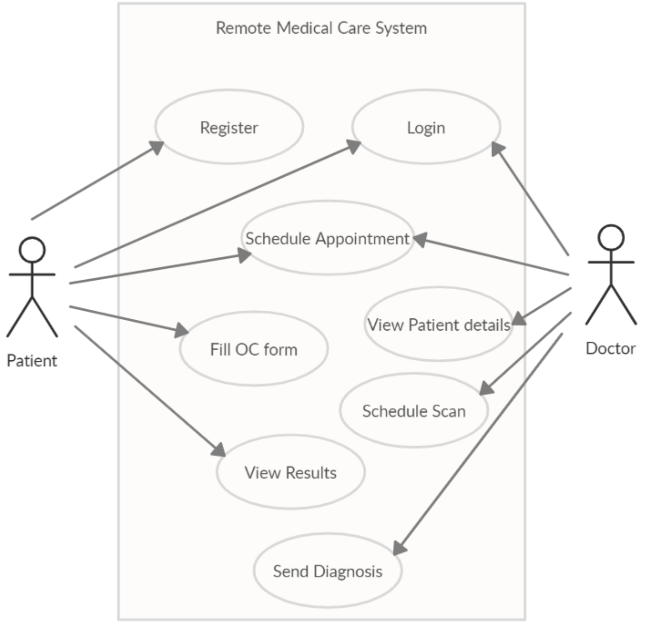


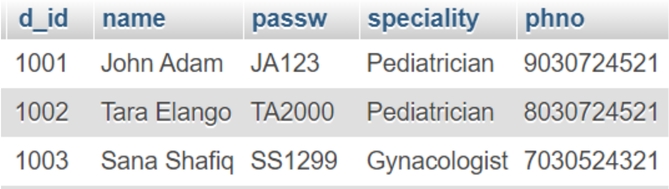
Fig. 1. Use case diagram to demonstrate user interaction with the system.

IV. Description of Tables

We have declared a few tables for the project to store information about the patients, doctors, appointments etc.

A. Doctor

The table doctor has five fields. It is to store the details of the doctors. The ”d-id” field is the doctor ID, unique to every doctor so it is the primary key. The ‘name’ field has the name of the doctor. The ”passw” field has a unique password for each doctor. The specialty of the doctor is given in ‘speciality’ field. The ”phno” field has the contact number of the doctor.

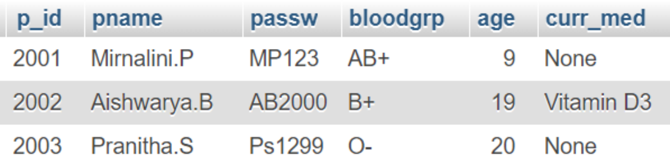


d_id	name	passw	speciality	phno
1001	John Adam	JA123	Pediatrician	9030724521
1002	Tara Elango	TA2000	Pediatrician	8030724521
1003	Sana Shafiq	SS1299	Gynacologist	7030524321

Fig. 2. Sample Doctor table.

B. Patient

The table patient has seven fields. It is to store the details of the patients. The ‘p-id’ is the patient ID, unique to every patient so it is the primary key. The ‘pname’ field has the name of the patient. The ‘passw’ field has a unique password for each patient. The blood group of the patient is given in ‘bloodgrp’ field. The ‘age’ field has the age of the patient. The ‘curr-med’ field has the information regarding any current medicine the patient may be currently prescribed.



p_id	pname	passw	bloodgrp	age	curr_med
2001	Mirnalini.P	MP123	AB+	9	None
2002	Aishwarya.B	AB2000	B+	19	Vitamin D3
2003	Pranitha.S	Ps1299	O-	20	None

Fig. 3. Sample Patient table.

C. Appointment

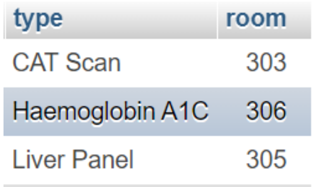
The appointment table has all the appointments being made. It has five fields. The ‘p-id’ has the patient Id.The ‘d-id’ has the doctor ID. The speciality of the doctor chosen is in ‘speciality’ field. The date and slot of the appointment made is in ‘date-of-app’ and ‘slot’ field respectively. Since the ‘d-id’, ‘date-of-app’ and ‘slot’ is unique to each appointment, it is made the primary key.Since the ‘d-id’ and ‘p-id’ are from the ‘doctor’ and ‘patient’ table ,the are the foreign keys.

D. OnlineCon

The Onlinecon table has all the information regarding the online consultations. When a patient fills in the form, information such as ’p-id’, ’d-id’, nausea, sleep as well as other symptoms are stored. The doctor’s name which corresponds with the ’d-id’ are also stored as well as the patients blood group and gender.

E. Scan

The scan table is used to store details of the scans taken. It has two fields. The ‘type’ field stores the type of the scan taken and the ‘room’ field stores the room in which that type of scan is taken.



type	room
CAT Scan	303
Haemoglobin A1C	306
Liver Panel	305

Fig. 4. Sample Scan table.

F. Scan-Date

The Scandate table stores the details of the date in which the scans are taken. It has six fields. The patient id and doctor id are stored in ‘p-id’ and ‘d-id’ ,which is from the ‘doctor’ and ‘patient’ tables respectively.Hence are foreign keys. The type of the scan is stored in ‘type’ field ,which is from the ‘scan’ table.Hence is a foreign key. The date and slot of the appointment made is in ‘date-of-app’ and ‘slot’ field respectively. Since the ‘d-id’, ‘date-of-app’ and ‘slot’ is unique to each appointment, it is made the primary key.The speciality of the scan is in ‘spec’ field.

V. Explanation of code and output

This project uses Java Netbeans to create the frontend of the application. The frontend consists of GUI with which the patient or the doctor can interact with. It uses MySQL database for the backend services. It stores relevant tables consisting of information on the various classes. The project allows either patient or doctor to login. Once logged in they can view results schedule appointments, consult online etc.

The code with documentation can be found at this link: XXX

Below included is the explanation of the main modules and the code associated with each.

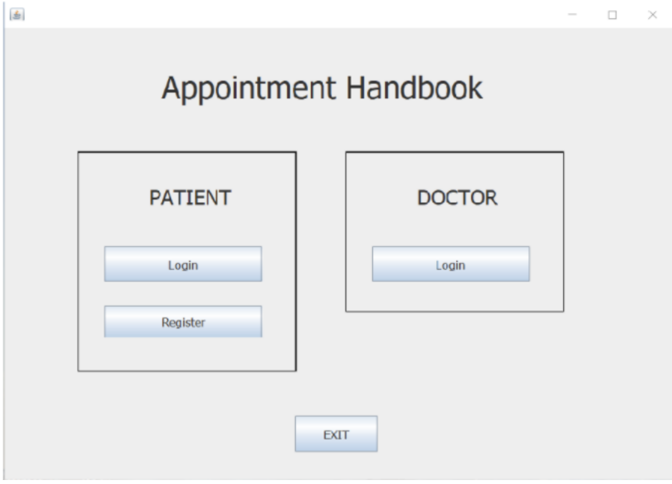
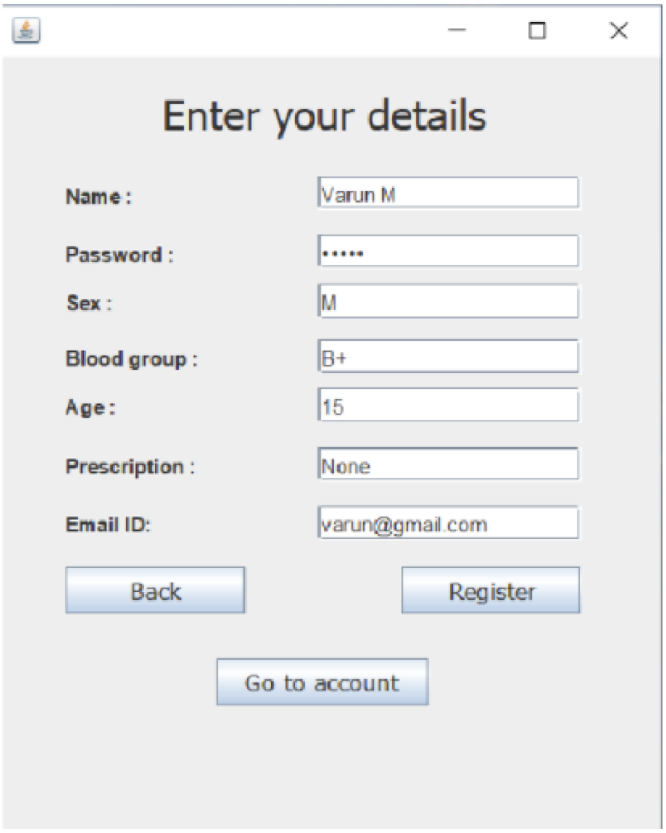


Fig. 5. Opening screen of application.

A. Registration

This module is available for patients to register themselves with the hospitals database. Only first time patients need to do so. The doctors do not need to use this as their details would practically already be available to the hospital. *RegistrationForm.java* handles this part be taking in patient details and storing all this information in the *Patient* table.



Enter your details

Name : Varun M

Password : •••••

Sex : M

Blood group : B+

Age : 15

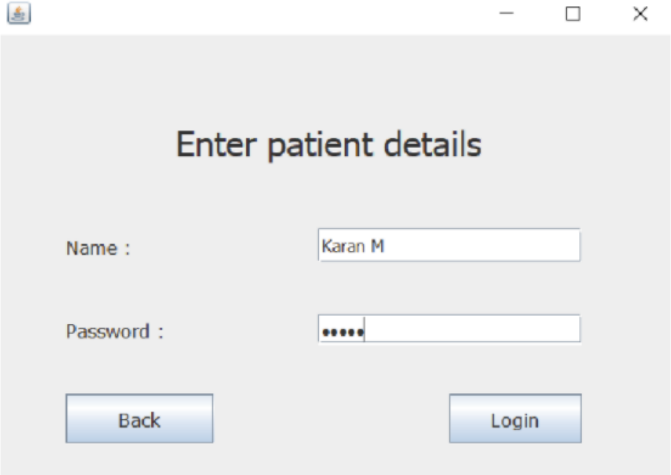
Prescription : None

Email ID: varun@gmail.com

Fig. 6. Registration form for patients.

B. Login

This is open to both patients and doctors. The Login page allows them to choose category an then login. For patients the login is handled by *PatientLogin.java* which fetches information from the *Patient* table. For doctors the same action is handled by the *DoctorLogin.java* which fetches login details from the *Doctor* table.



Enter patient details

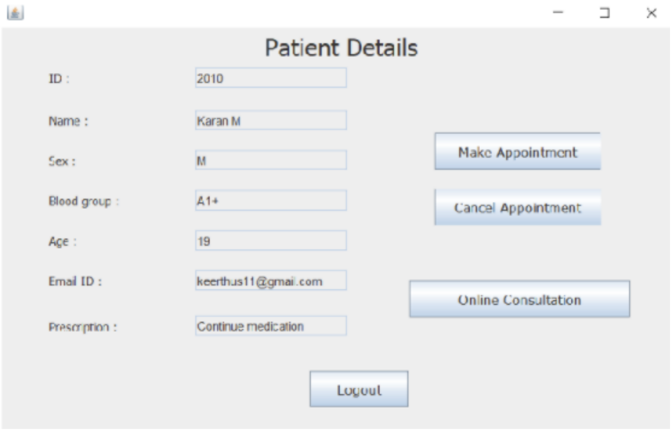
Name : Karan M

Password : •••••

Fig. 7. Login form.

C. Homescreen

This is the page that greets the user after login. It acts as gateway to all other action that can be performed by the user with respect to whether they are a doctor or a patient. This is handled by the *DoctorView.java* file for doctor, which allows them to choose if they want to see their patient details, schedule a scan or view the results. The patient side is handled by the *PatientView.java* code which allows them to schedule appointments, cancel, view their current details, fill in an online consultation form with their symptoms etc.


Patient Details

ID : 2010

Name : Karan M

Sex : M

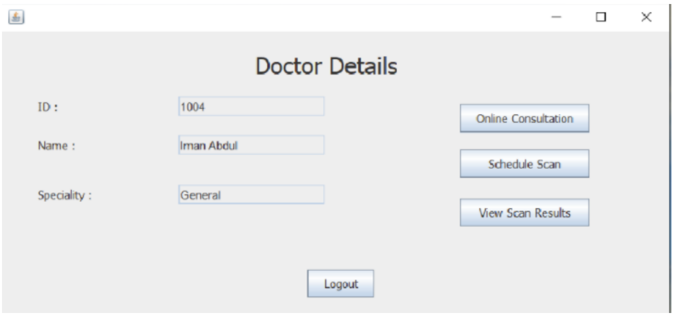
Blood group : A1+

Age : 19

Email ID : keerthus11@gmail.com

Prescription : Continue medication

Fig. 8. Homepage for Patient.


Doctor Details

ID : 1004

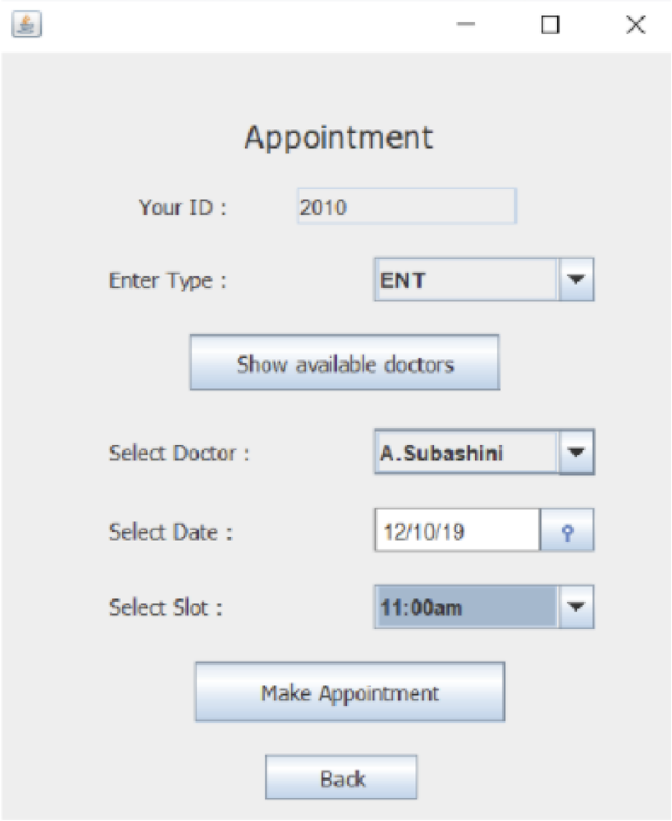
Name : Iman Abdul

Speciality : General

Fig. 9. Homepage for doctor.

D. Appointments

The user have the ability to make appointments by choosing a specialist, selecting date and a slot from the given list of available physicians. This is handled by the *Appointment.java* program. The patient also recieves a confirmation email in their given email address. They can also cancel any appointments they do not wish to continue. This is done by the *CancelApp.java* file. Both these actions interact with the *Appointment* table.


Appointment

Your ID : 2010

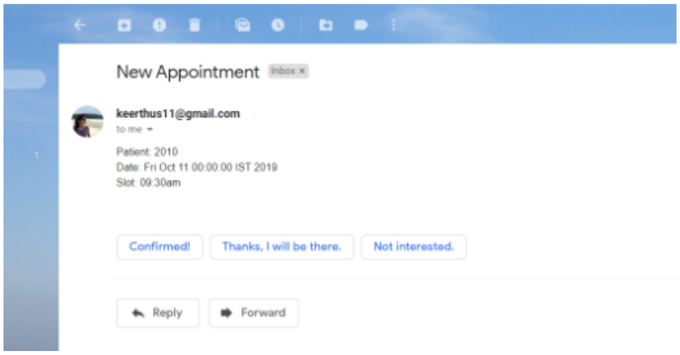
Enter Type : ENT

Select Doctor : A.Subashini

Select Date : 12/10/19

Select Slot : 11:00am

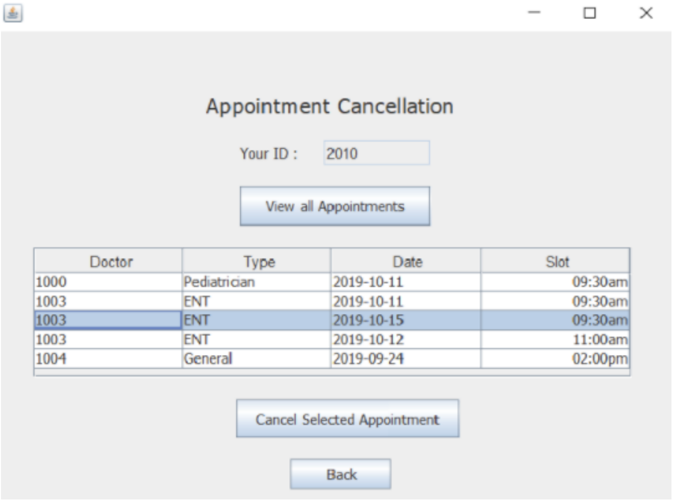
Fig. 10. Appointment form.


New Appointment

keerthus11@gmail.com
to me

Patient: 2010
Date: Fri Oct 11 00:00:00 IST 2019
Slot: 09:30am

Fig. 11. Appointment confirmation.



Appointment Cancellation

Your ID : 2010

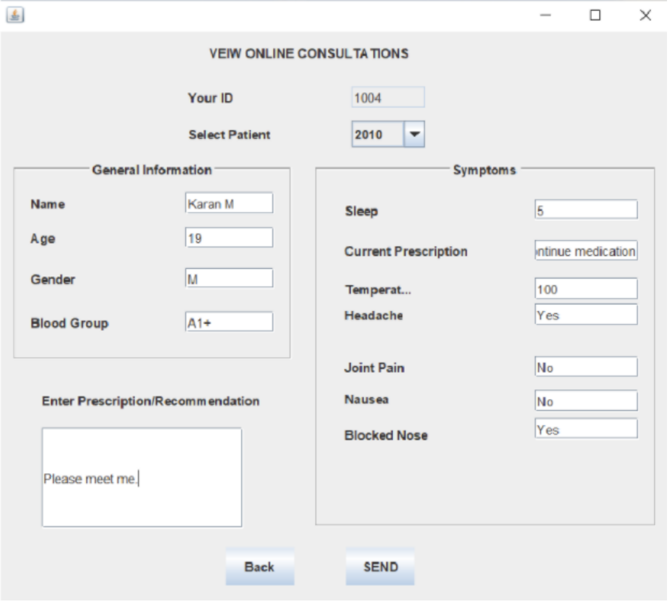
Doctor	Type	Date	Slot
1000	Pediatrician	2019-10-11	09:30am
1003	ENT	2019-10-11	09:30am
1003	ENT	2019-10-15	09:30am
1003	ENT	2019-10-12	11:00am
1004	General	2019-09-24	02:00pm

Fig. 12. Appointment cancellation page.

E. Online Consultation

First the patient initiates the process by filling out the online consultation form. In it details such as sleep patterns, joint pain, and other symptoms. All this is stored in the *OnlineCon* table. They also select a particular doctor to send this information to. this entire action on the patients side is handled by the *OnlineCon.java* file.

On the Doctors side all they can select patient through a drop down menu option that will let them see the patients details and symptoms. They can then write a prescription or schedule a scan or a test if they feel it is necessary. They can also ask them to meet them in person by booking an appointment. This prescription is also directly sent to the patients email ID. This is handled by the *DocOC.java* file.



VEIW ONLINE CONSULTA TIONS

Your ID: 1004

Select Patient: 2010

General Information

Name: Karan M

Age: 19

Gender: M

Blood Group: A1+

Enter Prescription/Recomm endation

Please meet me.

Symptoms

Sleep: 5

Current Prescription: ntinue medication

Temperat...: 100

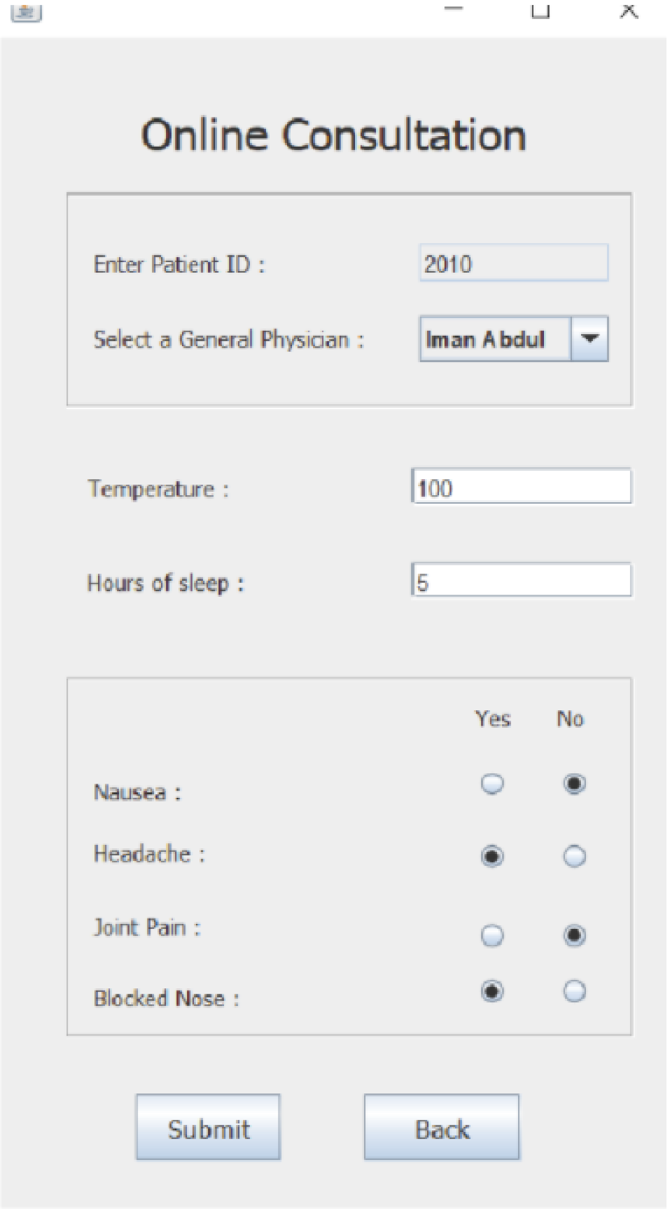
Headache: Yes

Joint Pain: No

Nausea: No

Blocked Nose: Yes

Fig. 13. Online consultation - doctor's side.



Online Consultation

Enter Patient ID : 2010

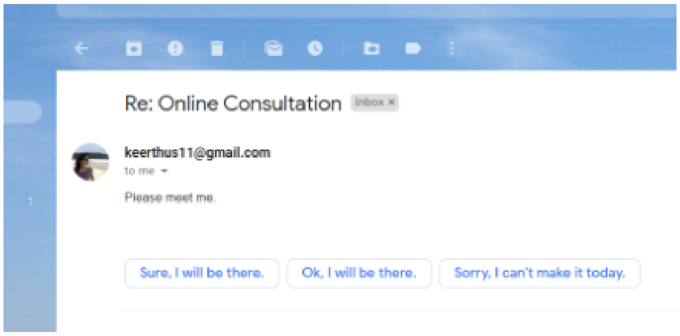
Select a General Physician : Iman Abdul

Temperature : 100

Hours of sleep : 5

	Yes	No
Nausea :	○	●
Headache :	●	○
Joint Pain :	○	●
Blocked Nose :	●	○

Fig. 14. Online consultation form - patient side.



Re: Online Consultation

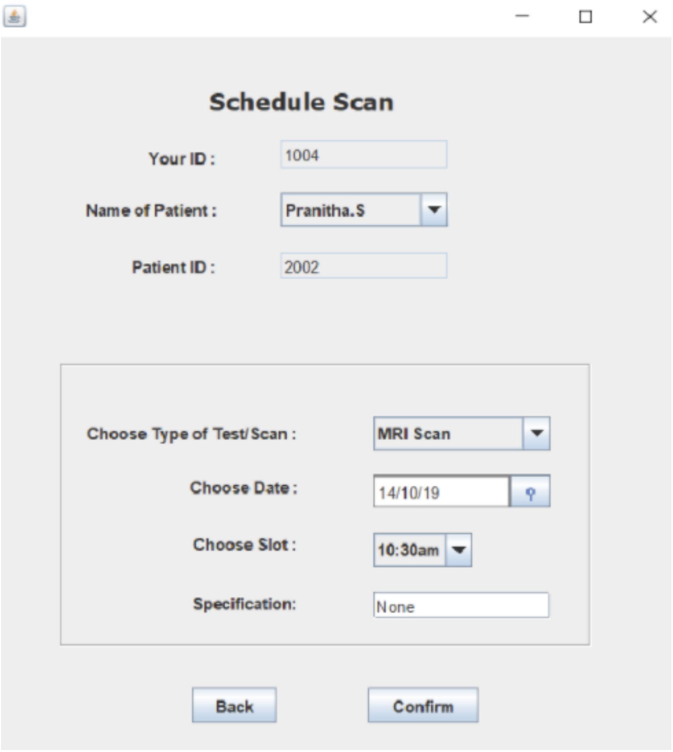
keerthus11@gmail.com
to me

Please meet me.

Fig. 15. Response from doctor to patient.

F. Scan

the doctors have the option to schedule a scan or a test from a given list for any of the patients currently consulting with them. They can choose the patient, fill in the required details and this information is stored in the *scandate* table. The patient will get a reminder of this. This action is handled by the *ScanSched.java* file.



Schedule Scan

Your ID : 1004

Name of Patient : Pranitha.S

Patient ID : 2002

Choose Type of Test/Scan : MRI Scan

Choose Date : 14/10/19

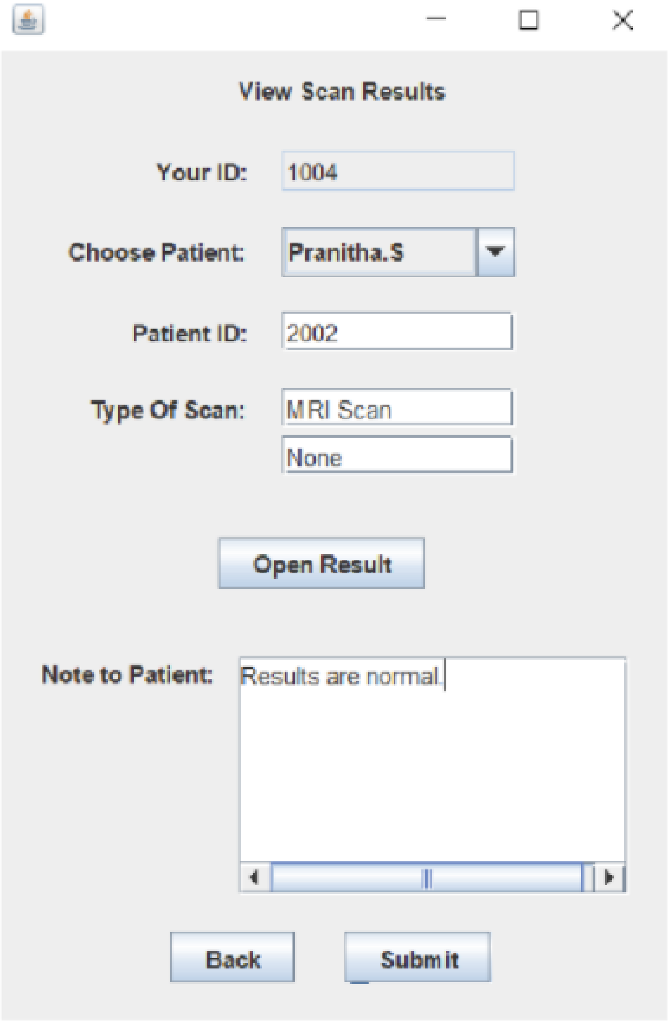
Choose Slot : 10:30am

Specification: None

Fig. 16. Schecduling scan by the doctor for patient.

G. View Results

Finally on the doctors side, they can view the results for their patients once it comes in. They can open an external file containing the results for a particular patient and can then send the patient their diagnosis and remarks. This reaches the patient through their email.



View Scan Results

Your ID: 1004

Choose Patient: Pranitha.S

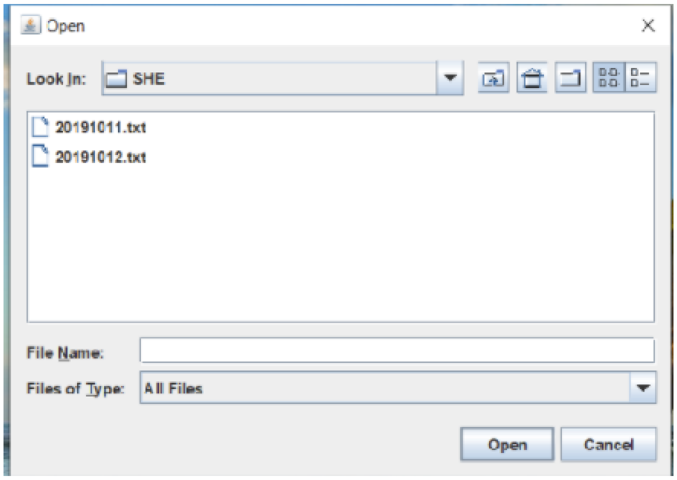
Patient ID: 2002

Type Of Scan: MRI Scan

None

Note to Patient: Results are normal.

Fig. 17. Scan results.



Open

Look In: SHE

20191011.txt

20191012.txt

File Name:

Files of Type: All Files

Fig. 18. File chooser.

VI. Possible future additions

The 'Smart Appointment and Scheduling Systems for Hospitals' application which aims at scheduling appointments for online consultation and diagnosis, can be further expanded with additional functionalities due to the increasing advancements in Machine Learning and other technologies in the medical industry. With Machine Learning, we can use prediction modules (predictive algorithms like Decision Tree, Random Forest etc...) to predict/diagnose the disease that the patient might be having with the inputs he/she feeds in the interface. Also, we can use recommendation models to provide a prescription corresponding to the predicted disease. In this way, you can avoid rushing to the hospitals even for the slightest discomfort and also the avoid hysteria by Googling the symptoms and assuming you have the worst. The implementation of remote healthcare practices and technology has been showing an increased adoption among healthcare providers, institutions and users.

VII. Conclusion

In this manner we built this project which would streamline the process of getting medical aid. By conducting initial consultations online as well as allowing the scan appointments to be fixed remotely, face to face communication would only be required in later stages. This process would be quicker and more time efficient in the long run.

Acknowledgment

We would like to thank Ms. R. Bhuvaneshwari for her guidance throughout this project. We would also like to thank Keerthana Sundaresan for collaborating on this project and allowing us to use it as a base for this paper.

VIII. References

- X. Chen, L. Wang, J. Ding and N. Thomas, "Patient Flow Scheduling and Capacity Planning in a Smart Hospital Environment," in IEEE Access, vol. 4, pp. 135-148, 2016, doi: 10.1109/ACCESS.2015.2509013.
- N. Kohli and N. K. Verma, "Performance issues of hospital system using MySQL," 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, 2010, pp. 497-501, doi: 10.1109/ICCSIT.2010.5564100.
- A. Rizal, Y. Lin and Y. Lin, "Contactless vital signs measurement for self-service healthcare kiosk in intelligent building," 2018 3rd International Conference on Intelligent Green Building and Smart Grid (IGBSG), Yi-Lan, 2018, pp. 1-4, doi: 10.1109/IGBSG.2018.8393548.