An Electronic Prescription System powered by Speech Recognition, Natural Language Processing and Blockchain Technology

Jitendra Mahatpure, Dr Mahesh Motwani, Dr Piyush Kumar Shukla

Abstract— This paper presents a new healthcare system that would change the way of storing and processing health records. It will digitize the complete healthcare process. There won't be any need to carry paper prescriptions on revisiting doctors. The system will generate an electronic prescription using speech recognition and natural language processing. A QR code on a patient's smartphone is used to retrieve the digital prescription record stored on a blockchain network. A patient will be able to share historic prescription records to a new doctor. The System enables the patient to manage the privacy of their personal health record. Patient Health Record can only be accessed by using the QR code from the patient's smartphone as an identity. The system proposed in this paper is targeted to those doctors and clinics that are still using paper-based handwritten prescriptions and cannot afford the existing Electronic Health Record systems available. This system can be fully operated even through a single smartphone. It is a combination of a group of 5 modules working together. Those modules include Hyperledger Composer Blockchain Network, Node.js server REST APIs for communicating with the blockchain network, Python Django REST API server for Natural Language Processing or text processing, a React JS based admin panel and a React-Native based mobile application for Doctors and patients.

Index Terms— Blockchain, Electronic Prescription, Hyperledger Composer, Natural Language Processing, QR code, Speech to text, Text Processing.

1 Introduction

A major issue in India is that generally prescriptions are still written by hand and the readability of such handwritten prescription is very limited. Several cases have come to light, where an error in understanding the prescription by chemist has lead to the wrong medication, which caused severe health issues to the patient[1]. But usage of traditional Electronic Health Record System(EHR) in generating an electronic prescription is tedious and a bit costly for a doctor in India. A proper infrastructure would be needed for setting up such a system. Also, the operating cost and time for the system would also add up. Either the doctor would operate the system on his own consuming time or an operator if employed would cost money. Furthermore, patient record addition and retrieval consumes 49 per cent of doctor's time in using an EHR system[2].

Smartphones have fairly penetrated the Indian market, so a smartphone-based electronic prescription system using Speech to Text and patient identification using QR code will be accessible and cost-effective for every doctor. In this paper, Authors are submitting an effective system that will help the doctor in minimizing their interaction with the EHR system and increase the patient-doctor facetime.

- *Jitendra Mahatpure is currently pursuing dual degree Integrated Post Graduate programme in Computer Science & Engineering in University Institute of Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal(M.P.), India, PH-+917354918469. E-mail: jmahatpure01@gmail.com*
- *Dr. Mahesh Motwani is currently a Professor in Computer Science & Engineering Department in University Institute of Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal(M.P.), India*
- *Dr. Piyush Kumar Shukla is currently an Assistant Professor in Computer Science & Engineering Department in University Institute of Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal(M.P.), India*

In this system, Authors are using the doctor's voice as input that would allow reducing the time required to write a prescription. When storing personal health record (PHR), Patient data privacy and security is always a challenge for Authority. For transparency and trust, blockchain technology can be utilized for the storage of generated prescriptions and other patient data. The patient will have all the access control over their personal data, what and to whom they want to share medical records. Another challenge is giving access to patient data to the doctors in the hospital without any manual input to reduce Doctor-EHR time. To achieve this, a QR code is generated on the mobile app for Patients, linked with the Unique Identity of Patient can be utilized for automatic patient record retrieval on the visit.

2 Literature Review

The idea of electronic prescription is nothing new, there are a few systems in the Indian market that offers a way to digitize the generation and storage of prescriptions. Some of them are described below. IRX Clinics[3] is a complete clinic management suite developed by Trata E Systems. It includes a smartpen, a smart prescription pad, and a tablet. The system allows doctors to continue writing prescriptions by hand and automatically capture the data in the background. The smartpen and prescription pad work in collaboration to save the data digitally. The tablet is preloaded with the clinic management suite which can be used to manage prescriptions, appointments, diagnoses. Prescrip[4] is an app that maintains patient records and lets doctors print the prescription in seconds. The app allows doctors to create prescription with ease, they just have to input the diagnosis. The app will automatically generate the prescription, and it gets better with use. It predicts the medicines on the basis of diagnosis by the doctor. E-Prescription by i-Grandee Software Technologies Pvt.

Ltd.[5] is a client-server system with capabilities to share E-Prescription with pharmacies. Doctors create a prescription and transfer it to the pharmacy. A fully paperless prescription system which allows access to the electronic prescription by all the authorized healthcare providers.

Prescription pad[6] is one of the widely used prescriptions writing medical software. It provides Safe, first-rate, foolproof prescription writing facility, comparative analysis of patient's medical history.

SlashDr[7] is a virtual physician assistant. It generates the Medical Council of India Compliant prescription, doctors have to select the symptoms diagnosed and the system will automatically generate the prescription.

Gboard[8] is the google keyboard mobile application which has voice input method that can be utilized to fill any field in a form using speech to text.

Most of the above prescription systems take manual input from the doctor and then generate a prescription. Some of them predict medicines on the basis of symptoms.

3 PROPOSED SOLUTION

In the current health system, when patients arrive at the hospital, the first step is to identify the patient before proceeding ahead. Patient identification is performed by mobile number or medical card or by identity/name. Medical staff verifies the provided information and book consultation for the patient. After that, the patient may be seen by the doctor. In each step, Patient identification and record retrieval are repeated which is a time-consuming process. As a paper-based prescription is used at most of the places, The review of record and updating of EHR is again time-consuming and doesn't benefit doctors in the current scenario. The hospitals where EHR is managed and used for patient records are updated by medical staff. In rush hours, Each medical staff has to update several patient records using the patient identifier, which may lead to the wrong updation in the patient record. That can cause serious consequences.

A solution to these issues is described in this section. The proposed solution eliminates patient misidentification and rationalizes the use of time in patient care. The computer with a webcam and mobile devices used by medical staff and doctors can be used to identify the patient using QR Code from the patient's smartphone and voice-based e-prescription to update EHR in real-time.

The system can be used to deploy self-registration kiosk in hospitals to reduce the load on the medical staff. The patient medical history can be retrieved by any doctor with the right permissions to access the patient identification service. The patient can be identified with the simple gesture of scanning QR Code. Thanks to the use of smartphones, the system can be used without additional hardware to identify patients, retrieve medical history and update health records in real-time.

The authors are proposing a Healthcare System based on Voice recognition, Natural Language Processing and Blockchain technology. There are 5 modules in this system. Fig. 1, shows the overall system architecture.

1. Hyperledger Composer Blockchain Network
2. Node.js REST APIs
3. Python Django REST APIs
4. React-Native Mobile App
5. React JS Web Application

Out of the 5 modules or components, authors have deployed the blockchain network peer, Node.js REST API server and Python Django REST API server on AWS EC2 t2.micro Virtual Machine.

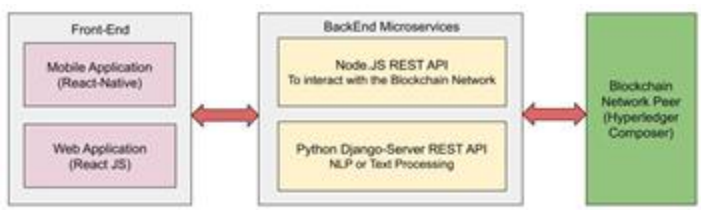

Figure1. System Architecture

3.1 Hyperledger Composer Blockchain Network

The Authors have used Hyperledger Composer framework[9]. It is an open-source set of tools for developing a blockchain network that makes it simple for developers to create blockchain applications. The framework allows defining the network model in the form of objects in a .cto file. And the permissions associated with each participant of the network is defined in a .acl file.

TABLE 1. COMPARISON TABLE OF ELECTRONIC PRESCRIPTION SYSTEMS

Attributes	IRX Clinics[1]	Prescrip[2]	E-Prescription[3]	Prescription Pad[4]	SlashDr[5]
E-Prescription mechanism	Smartpen	Ipad App	Web interface	Desktop and tablet interface	Ipad App
Company	Trata E Systems	Prescrip	i-Grandeee	Prescription Pad	SlashDr
Data Storage	Encrypted on tablet	On-device	Server	Both option on server or device	On a device with backup service
Appointments	Supported	Not supported	Not supported	Supported	Supported
Doctor Referrals	Supported	Supported	Not supported	Supported	Supported
Pill Reminders	Supported	Supported	Not supported	Not supported	Not supported
E-Prescription access to patient	Yes	No	Yes	No	No

3.1.1 Model

Fig. 2, gives an overview of the blockchain network model. In a blockchain network model, there are 3 main components i.e. Participant, Asset, Transaction.

There are 5 types of participants in the network.

1. Patient
2. Doctor
3. Lab
4. Hospital
5. Pharmacy

And one asset i.e. Prescription with following attributes

1. Symptoms
2. Medicines(an array of Medicine name and dosage pair)
3. owner(Patient)
4. currentDoctor(the doctor creating the prescription)
5. doctors(an array of doctors with access privileges to this prescription)
6. labs(an array of labs with access privileges to this prescription)(optional)
7. pharmacies(an array of pharmacies with access privileges to this prescription)(optional)
8. Datetime
9. reportsFileUrl(a reports file can be uploaded by Laboratories)(optional)
10. labTests(a string of lab tests)(optional)
11. notes(a string)(optional)
12. public(a boolean, if true then the prescription is accessible to other doctors)(default true)

There are 3 transactions i.e. methods for granting access of a prescription to various participants.

1. grantAccessToDoctor: invoked by owner patient of the prescription
2. grantAccessToLab: invoked by the current Doctor of the prescription
3. grantAccessToPharmacy: invoked by the current Doctor of the prescription

These 3 transactions have 2 attributes each one is the Prescription (a reference to the prescription object using Id) and the new participant to be granted access i.e. Doctor, Lab, Pharmacy.
The blockchain network deployed works on the principle of proof of identity, that is only the participant with the required access permissions can access and operate on the network. An identity digital certificate card is issued to each participant during registration, which is used for connecting to the blockchain network. Furthermore, delete operation is disabled on the network for all types of objects.
All the participants have an associated asset responsible for separating email and password from the actual object and act as a mapping between the email address and participant object.

Doctor participant object has the following attributes:

1. Id
2. Name
3. RegistrationNumber
4. Mobile
5. Hospital(a reference to the hospital object)
6. Patients (an array of reference to patient objects)

EmailToDoctor asset has email, password and a reference to the Doctor object as attributes.

Patient participant object has the following attributes:

1. Id
2. Name
3. Age
4. Mobile
5. Gender(optional)
6. PicUrl(optional)

EmailToPatient asset has email, password and a reference to the Patient object as attributes.

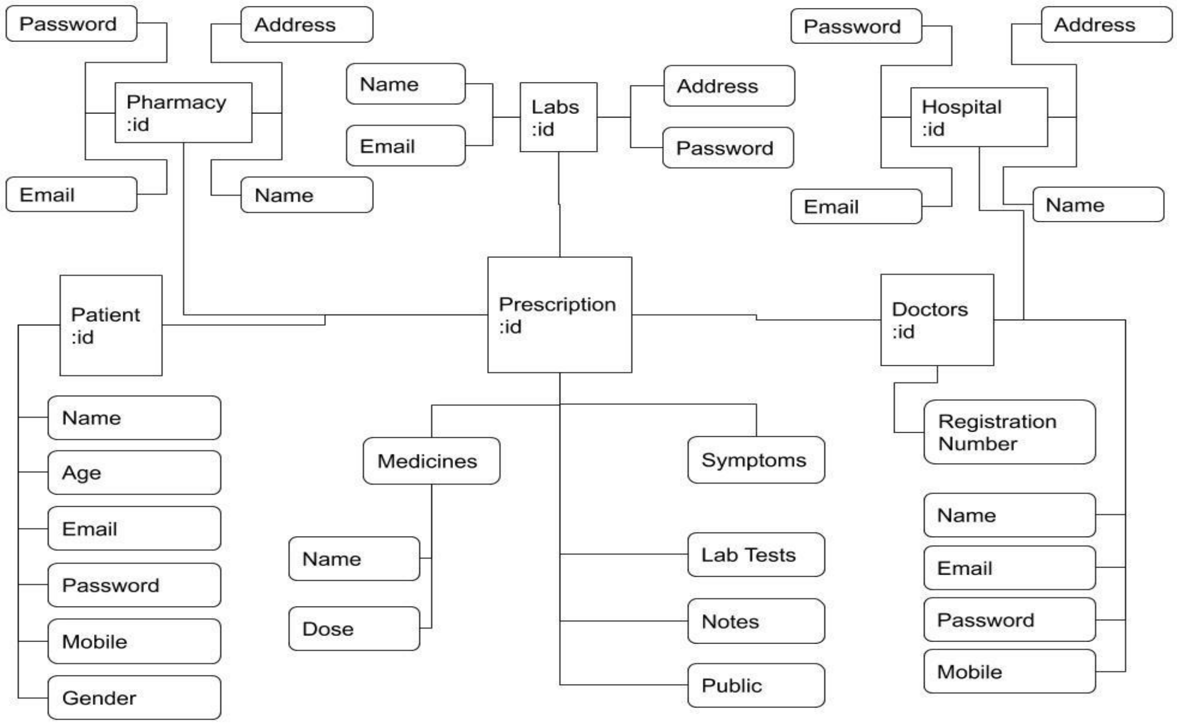

Figure 2. Blockchain Model Overview

Hospital participant object has the following attributes:
1. Id
2. Name
3. Address
4. Mobile

EmailToHospital asset has email, password and a reference to the Hospital object as attributes.

Lab participant object has the following attributes:
1. Id
2. Name
3. Address
4. Mobile

EmailToLab asset has email, password and a reference to the Lab object as attributes.

Pharmacy participant object has the following attributes:
1. Id
2. Name
3. Address
4. Mobile

EmailToPharmacy asset has email, password and a reference to the Pharmacy object as attributes.

3.1.2 Permissions

The blockchain network is accessed through an identity card i.e. a cryptographic identity issued to each participant of the blockchain network on registration. Each identity card has associated permissions.

1. Doctors can create prescriptions.
2. Doctors can read prescriptions if their reference is present in the doctors array of prescription object.
3. A doctor can read all public prescriptions of a patient by scanning the patient's QR code.
4. Doctors can update prescriptions if they are referenced as currentDoctor in the prescription object.
5. Doctors can create Patient and Doctors, can read patient details if the patient reference is present in patients array of doctor object. Patient reference is added to the doctor object when a prescription is created by the doctor with the patient as owner.
6. A patient can create, read, update prescriptions. Read and update operations are allowed to the owner patients of the prescription.
7. Doctors can read all Labs and Pharmacies. Also, Doctors can read Hospital objects which are referenced in the doctor object.
8. Labs and Pharmacies can read prescription object if they are referenced in the labs and pharmacies array in the prescription object.

3.2 Node.js REST API Server

Hyperledger Composer[9] has a client-side node.js library for communicating with the blockchain network Peer deployed on the same server as the node.js application. The authors created GET, POST APIs for creating and listing all the participants, asset and transactions using express[10] library for node.js. Each of these APIs takes a mandatory field i.e. identity card name of the participant requesting the data or submitting it.

3.3 Python Django REST API Server

The authors are proposing an approach to using Speech Recognition to speed up the process of prescription generation. Speaking a sentence will consume less amount of time than writing it. And Natural Language Processing(NLP) to extract prescription information from the transcript. For speech to text, Google's speech recognition API is the best available API as it supports Indian English, also it is available in Android smartphones as an in-built free to use service.

Fig. 3, describes the text processing done on the python Django REST server. The HTTP POST request from the mobile app is received and the transcript array is decoded from it. The first transcript string from the array is considered for further processing. The Text extraction process involves the removal of unimportant words or stopwords in NLP terms from the transcript string like: is, are, and, etc. Then the keyword-based text extraction is done. There are 4 basic components of any prescription i.e. Patient Name, Patient Age, Symptoms, and Medicines with dosage. The patient details will come by scanning QR code from patient's mobile app so, the patient name and age can be skipped here. The QR code provides the patient ID so, patient name and age can be retrieved through the patient ID and displayed for cross-checking. The patient ID is added to the prescription object. Hence, the 2 main components that are needed to be extracted from the transcripts are symptoms and medicine names with dosage. Also, there are 2 other optional components Lab tests and notes, that can be utilized by the doctor to prescribe a lab test and for some extra remarks respectively.

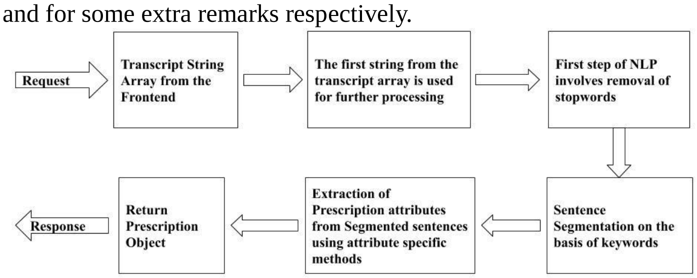

Figure 3. Voice Prescription Backend Process

A list of 1111 symptoms is created by scraping wikipedia[11], healthline.com[12] and rxlist.com[13]. And is used for identifying individual symptoms from the fragmented text and for creating a comma-separated list of symptoms from the transcript text. Similarly, medicine details from healthos[14] API are cached on the server for faster search. The medicine name extracted from the transcript is searched first in the cached list of medicines and then through an API call to healthos endpoint.

The keywords are searched one by one in the order written. For example, if *lab test(s), test(s) and note(s)* are written, so *lab tests*, *lab test*, *tests*, *test*, *notes* and *note* is the order. If the search of the first keyword is successful then the rest of the keywords are not searched and boolean is set accordingly. This order of keyword searching is followed wherever the search is done in the below algorithm.

Algorithm for keyword-based text extraction

Step 1: Preparing the transcript for extraction. First, the removal of unnecessary words or stopwords like is, am, are, and, the, in, etc. A list of such words for the English language is taken from NLTK data stopwords corpus[15]. Second, changing the case of the transcript to lowercase.

1. Tokenize the transcript using NLTK tokenizer[16].
2. Iterating over all the tokens from the above sub-step, a sentence is formed by concatenating tokens which are not present in the stopwords list.
3. The sentence from above sub-step is converted to lowercase.

Step 2: In the sentence from the above step, *symptom(s)* keyword is searched, if found the keyword is removed from the sentence and a boolean symptomsFound is set to true otherwise the boolean is set to false.

Step 3: Next, Medicines are searched by keywords *medication(s), medicine(s)* in the sentence from the previous step if one of the keywords is found then the sentence is split into two fragments at the place of the keyword, removing the keyword and medicineFound boolean is set to true. If the above keywords are not found, one more condition is checked to test the presence of medicines i.e. a regex condition that checks for the presence of name keyword (regex expression: "([\w]+)\sname"). If this test condition is true than the sentence is split in two fragments at the word just before the name keyword and medicationFound boolean is set to true.

Step 4: Symptoms Extraction

1. If symptomsFound and medicationFound both booleans are set to true then out of the two fragments formed in the previous step, the first fragment of the sentence is processed using the list of 1111 symptoms prepared before-hand and a comma-separated list of symptoms is set to symptoms attribute of the prescription object.
2. If symptomsFound boolean is set to true but medicationFound is not set to true then we search for the presence of lab tests or notes in the sentence from step 2. For that, keywords *lab test(s)*, *test(s)* and *note(s)* are searched. If any one of the keywords is found then the sentence is split into two fragments at the keyword, removing the keyword and the respective booleans labTestFound or notesFound are set to true. Out of the two fragments formed, the first one is processed and a comma-separated list of symptoms is assigned to the symptoms attribute of the prescription object.

Step 5: If medicationFound boolean is set to true.

1. The presence of lab tests and notes is checked in the second fragment out of the fragments created in step 3 using the keywords *lab test(s)*, *test(s)* and *note(s)*. If one of the keywords is found then the fragment is further split into two fragments at the keyword, removing the keyword and the respective boolean labTestFound or notesFound is set to true.
2. Out of the two fragments formed in the previous sub-step, the first fragment is assigned to a variable medicines_text which is going to be used for extracting each medicine name and dose.
3. medicines_text is searched for *name* keyword, if found then medicines_text is split into an array of strings each containing medicine name and dose, the *name* keyword is removed from the strings of the array.

Step 6: Medicines Extraction.
Now, the array of strings formed in the previous step is iterated and the following sub-steps are performed for each string in the array:

1. The string is searched for *dose*, *dosage* keyword, if found then the string is split into two fragments at the place of keyword, removing the keyword and out of the two fragments, the second one is assigned to a variable with name dose. Also, the first fragment contains the name of medicine.
2. The first fragment from the above sub-step is first searched in medicines list cached on the server if not found then is sent to healthos[14] medicine search API. The medicine with the best match is returned from the API.
3. The dose variable and the medicine name returned from the healthos API is appended to the medicines array in the prescription object.

Step 7: If neither the symptomsFound boolean is set to true nor the medicationFound boolean is set to true then the initial sentence is searched for the presence of keywords *lab test(s), test(s)* and *note(s)*. If one of the keywords is found then the sentence is split into two fragments at the location of that keyword, removing the keyword and the respective boolean labTestFound or notesFound is set to true.

Step 8: If labTestFound boolean is set to true then out of the fragments formed in step 4 or 5 or 7 the second fragment is searched for the presence of *note(s)* keyword, if found then the fragment is further split into two fragments at the location of the *note(s)* keyword, removing the keyword and the first fragment is assigned to labTests attribute of prescription object and the second one is assigned to the notes attribute of prescription object.

Step 9: If notesFound boolean is set to true then out of the fragments formed in step 7 the second fragment is assigned to notes attribute of the prescription object.

Example Transcript received:
"Symptoms are cough runny nose and fever medicine name cheston cold dose twice daily after meal for 3 days name aciloc 150 dose twice daily after meal for 3 days lab test test for malaria notes this is hard demo prescription"

Step 1: "symptoms cough runny nose fever medicine name cheston cold dose twice daily after meal for 3 days name aciloc 150 dose twice daily after meal for 3 days lab test test for malaria notes hard demo prescription"

Step 2: "cough runny nose fever medicine name cheston cold dose twice daily after meal for 3 days name aciloc 150 dose twice daily after meal for 3 days lab test test for malaria notes hard demo prescription"
symptomsFound: true

Step 3: "cough runny nose fever", " name cheston cold dose twice daily after meal for 3 days name aciloc 150 dose twice daily after meal for 3 days lab test test for malaria notes hard demo prescription"
symptomsFound: true, medicationFound: true

Step 4: Symptoms: "runny nose, fever, cough", "name cheston cold dose twice daily after meal for 3 days name aciloc 150 dose twice daily after meal for 3 days lab test test for malaria notes hard demo prescription"
symptomsFound: true, medicationFound: true

Step 5: Symptoms: "runny nose, fever, cough", medicines_text: " name cheston cold dose twice daily after meal for 3 days name aciloc 150 dose twice daily after meal for 3 days", medicines_text_array: [" cheston cold dose twice daily after meal for 3 days ", " aciloc 150 dose twice daily after meal for 3 days "], " lab test test for malaria notes hard demo prescription"
symptomsFound: true, medicationFound: true, labTestFound: true

Step 6: Symptoms: "runny nose, fever, cough", medicines:[{"name": "CHESTON COLD SYRUP", dose: "twice daily after meal for 3 days"}, {"name": "ACILOC 150 MG TABLET", dose: "twice daily after meal for 3 days"}] , " lab test test for malaria notes hard demo prescription"
symptomsFound: true, medicationFound: true, labTestFound: true

Step 7 is skipped because of the boolean values.

Step 8: Symptoms: "runny nose, fever, cough", medicines:[{"name": "CHESTON COLD SYRUP", dose: "twice daily after meal for 3 days"}, {"name": "ACILOC 150 MG TABLET", dose: "twice daily after meal for 3 days"}] , labTests:"test for malaria", notes: "hard demo prescription"
symptomsFound: true, medicationFound: true, labTestFound: true

Step 9 is skipped because of the boolean notesFound value.

The final prescription object is formed:

{"medicines": [{"name": "CHESTON COLD SYRUP", "dose": "twice daily after meal for 3 days"}, {"name": "ACILOC 150 MG TABLET", "dose": "twice daily after meal for 3 days"}], "name": "", "age": "", "symptoms": " runny nose, fever, cough", "labTests": "test for malaria", "notes": "hard demo prescription"}

3.4 React-Native Mobile Application

The Mobile Application is developed to help doctors utilize the system. A doctor can register patients, identify the patient and can generate e-prescription using the mobile app. This app can also be used by the patients to view their prescriptions.

The Mobile application in Doctors mode provides an interface for speech recognition and text processing to fill the details in a prescription form.

Fig. 4, shows the process on the mobile application side. In the mobile app, voice is recorded with the microphone present in the device and this recorded voice is sent to the Google speech recognition engine, which returns an array of transcript strings. This array is arranged on the basis of the best match first i.e. the first string from the array will be the best match of the voice input provided. This array is sent to the server through an HTTP POST request for further processing.

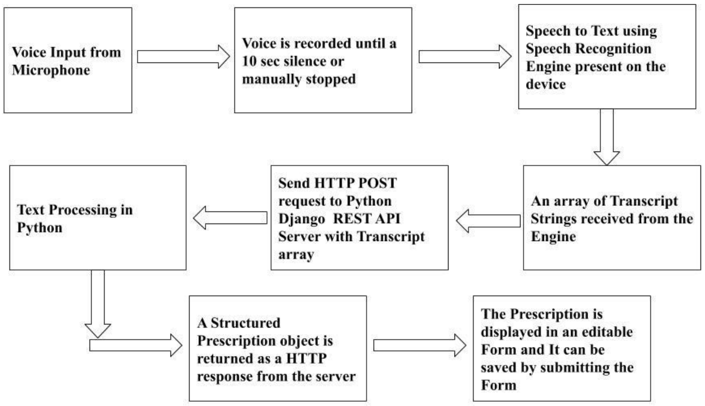

Figure 4. Voice Prescription Frontend process

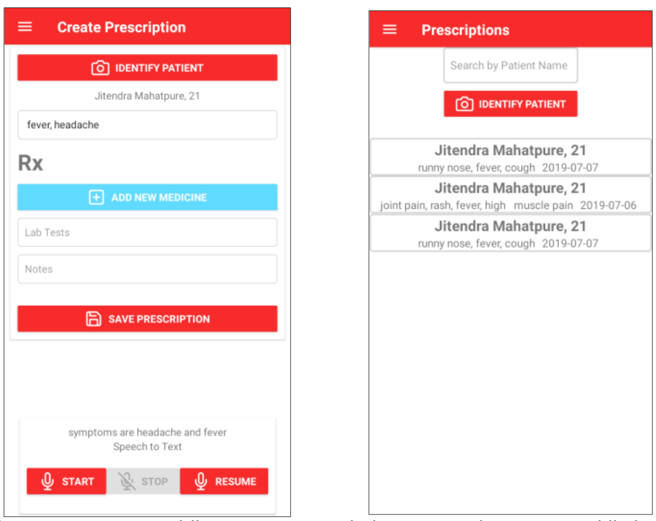

Figure 5. Doctor Mobile App: E-Prescription generation page and listing of Prescriptions

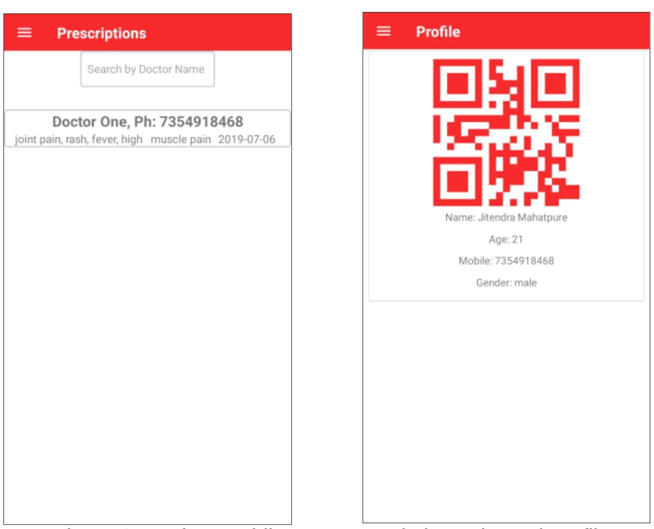

Figure 6. Patient Mobile App: Prescriptions List and Profile

3.5 React JS Web Application

The React JS Web Application provides the listing of participants and prescriptions on the basis of permissions. It also provides an interface for the registration of new participants. Fig. 7, shows a list of prescriptions for the logged-in doctor. Furthermore, electronic prescriptions can be generated from this web application using speech recognition shown in Fig. 8 and 9.

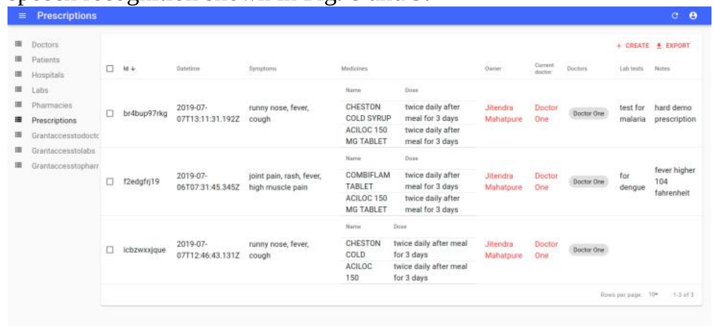

Figure 7. Web Application: Prescription List

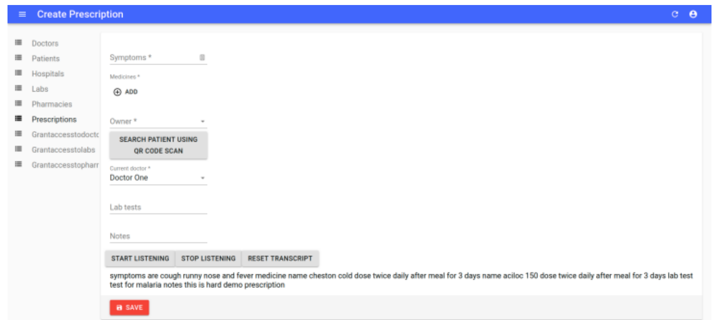

Figure 8.Web Application: Create Prescription - Speech Recognition

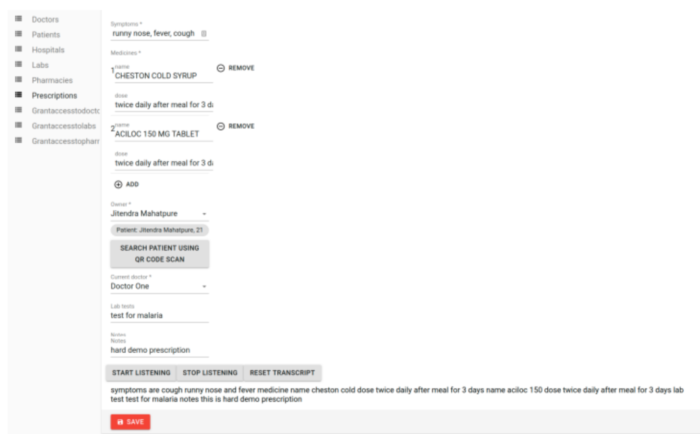

Figure 9. Web Application: Create Prescription - Auto Populated Fields using NLP

4 Result Analysis

The proposed and implemented system has two advantages over any existing electronic prescription generating system:

1. Cost
2. Time

In the proposed system only open source technology is utilized and the interface is available as a web application and mobile application both. So, the complete system's effective price is less and the usability is more. There is no requirement of proper infrastructure, a single smartphone is sufficient for operating the whole system. It is targeted at those small clinics and doctors that are still using handwritten prescriptions.

The proposed system is compared to the existing EHR systems that use manual input from a person to create patient records. This system utilizes speech input to fill all the fields of a prescription. So, the amount of time consumed for creating a prescription is less in the proposed system than by using manual input or field by field voice input on a mobile application using voice input keyboards like Gboard.

An Experiment to decide on the time consumption factor is described below:

4 different prescription transcripts with different lengths listed below are used to depict that with the increase in content the time required in manual input rises more rapidly than the voice input method of this system.

1. Symptom is headache and medicine name is Crocin dose once after meal whenever you feel pain.
2. Symptoms are runny nose and fever medicine name is Cheston cold dose is twice daily after meal for 3 days and name aciloc 150 dose twice daily after meal for 3 days.
3. Symptoms are fever body pain and chills medicine name is d'cold total dose twice daily after meal for 3 days and name aciloc 150 dose twice daily after meal for 3 days lab test test for malaria.
4. Symptoms are headache, muscle pain, joint pain, high fever, rashes medicine name combiflam dose twice daily after meal for 5 days name Aciloc 150 dose twice daily after meal for 5 days lab test test for dengue notes fever was above 104 Fahrenheit.

Fig. 10, is a graph showing the time consumed in filling prescription fields by three methods

1. Manual Input
2. Proposed Systems method
3. Gboard Voice input method

For the Experiment, the time taken is an average of multiple attempts. And it clearly shows that the proposed system can cut down the time consumption significantly for bigger prescriptions. A difference of around 20-30 secs is visible from manual input and 10-20 secs from the Gboard Voice input method in an Android smartphone.

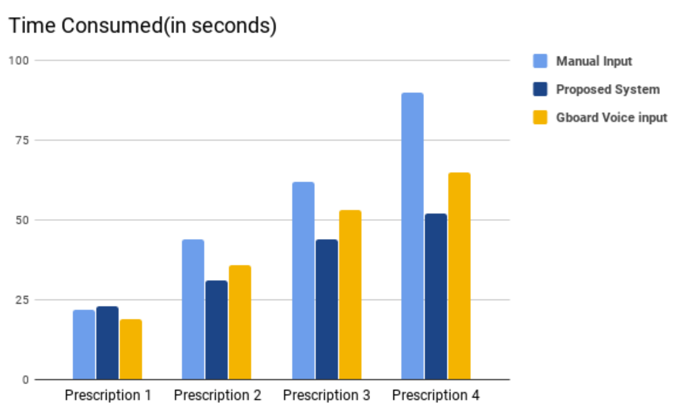

Figure 10. Time consumption

5 Conclusion & Future Work

The proposed and implemented system aims to reduce the amount of time consumed in creating and accessing patient records. It also provides the features of an EHR in a Mobile Application. Thus, making this system available to all the doctors of India through their smartphone.

Authors implemented an innovative solution to solve the problem of illegible handwritten prescriptions. Voice-based e-prescription needs a minimal change in the workflow of doctor but in the long run, it will create a huge impact in developing a digital ecosystem for patients. E-prescription system helps in managing EHR in real-time while maintaining the patient's privacy.

The implemented system will reduce the patient record access time and maintain high security and privacy of patient data. The patient is given the control to share their health records with other doctors.

In the near future, Authors are planning to integrate and use the system in the real hospital ecosystem to test and validate the implementation and to analyze the impact it will create in the healthcare domain.

References

[1] D. K. Sokol and S. Hettige, "Poor handwriting remains a significant problem in medicine," J. R. Soc. Med., vol. 99, no. 12, pp. 645–646, Dec. 2006.

[2] C. Sinsky et al., "Allocation of Physician Time in Ambulatory Practice: A Time and Motion Study in 4 Specialties," Ann. Intern. Med., vol. 165, no. 11, p. 753, Dec. 2016.

[3] iRxClinics, "Digital Prescription Writing & Smart Pen | iRx Clinics – Digitally Yours | Digitized patient record." [Online]. Available: https://www.irxclinics.com/. [Accessed: 13-May-2019].

[4] "Prescription app, e-prescribing, doctors prescription software, prescription app for doctors." [Online]. Available: http://www.prescrip.in/Index.aspx. [Accessed: 13-May-2019].

[5] "E-PRESCRIPTION | e prescription software | e prescription India | e prescription benefits." [Online]. Available: http://www.igrandee.com/jsp/domain/health/product/eprescription.jsp. [Accessed: 13-May-2019].

[6] "Prescription Pad | Prescription Writing Medical Software for Doctors." [Online]. Available: http://www.prescriptionpad.in/index.aspx. [Accessed: 13-May-2019].

[7] "SlashDr - Your own virtual physician assistant." [Online]. Available: https://slashdr.com/. [Accessed: 13-May-2019].

[8] "Gboard - the Google Keyboard – Apps on Google Play." [Online]. Available: https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin&hl=en_IN. [Accessed: 09-Jul-2019].

[9] "Hyperledger Composer - Create business networks and blockchain applications quickly for Hyperledger | Hyperledger Composer." [Online]. Available: https://hyperledger.github.io/composer/latest/. [Accessed: 09-Jun-2019].

[10] "Express - Node.js web application framework." [Online]. Available: https://expressjs.com/. [Accessed: 09-Jul-2019].

[11] "List of medical symptoms," Wikipedia. 30-Dec-2018.

[12] "Directory Symptoms." [Online]. Available: https://www.healthline.com/directory/symptoms. [Accessed: 04-Jul-2019].

[13] "Symptoms & Signs A-Z List - A on RxList," RxList. [Online].

Available: https://www.rxlist.com/symptoms_and_signs/alpha_a.htm. [Accessed: 05-Jul-2019].

[14] "2Tab | HealthOS - API for medicine data, food and nutrition data." [Online]. Available: http://www.healthos.co/#/home/search. [Accessed: 09-Jun-2019].

[15] "NLTK Data." [Online]. Available: http://www.nltk.org/nltk_data/. [Accessed: 02-Jul-2019].

[16] "nltk.tokenize package — NLTK 3.4.3 documentation." [Online]. Available: https://www.nltk.org/api/nltk.tokenize.html. [Accessed: 02-Jul-2019].