

Visvesvaraya Technological University

Jnana Sangama, Belagavi - 590018



A Project Report on

“Voice Based E-Prescription”

*Project Report submitted in partial fulfilment of the requirement for the
award of the degree of*

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

ABHISHEK GOWDA M V	1KS17CS002
ASHISH K AMAR	1KS17CS013
KRITHIKA JAGANNATH	1KS17CS036
MEGHANA G	1KS17CS042

Under the guidance of
Dr. Rekha B. Venkatapur
Professor and Head
Department of Computer Science & Engineering
K.S.I.T, Bengaluru-560109



KSIT
K.S. INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

K. S. Institute of Technology

#14, Raghuvanahalli, Kanakapura Road, Bengaluru - 560109

2020 - 2021

K. S. Institute of Technology

#14, Raghuvanahalli, Kanakapura Road, Bengaluru - 560109

Department of Computer Science & Engineering



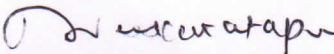
KSIT
K.S. INSTITUTE OF TECHNOLOGY

CERTIFICATE

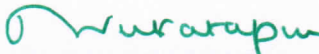
Certified that the project work entitled “Voice Based E-Prescription” is a bonafide work carried out by:

ABHISHEK GOWDA M V	1KS17CS002
ASHISH K AMAR	1KS17CS013
KRITHIKA JAGANNATH	1KS17CS036
MEGHANA G	1KS17CS042

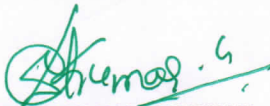
in partial fulfilment for VIII semester B.E., Project Work in the branch of Computer Science and Engineering prescribed by Visvesvaraya Technological University, Belagavi during the period of April 2021 to July 2021. It is certified that all the corrections and suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The Project Report has been approved as it satisfies the academic requirements in report of project work prescribed for the Bachelor of Engineering degree.


.....
Signature of the Guide

[Dr. Rekha B. Venkatapur]
[Professor and Head]


.....
Signature of the HOD

[Dr. Rekha B. Venkatapur]


.....
**Signature of the Principal/
Director**

[Dr. DILIP KUMAR K]

External Viva

Name of the Examiners

Signature with date

1.

2.

DECLARATION

We, the undersigned students of 8th semester, Computer Science & Engineering, KSIT, declare that our project work entitled “**Voice Based E-Prescription**”, is a bonafide work of ours. Our project is neither a copy nor by means a modification of any other engineering project.

We also declare that this project was not entitled for submission to any other university in the past and shall remain the only submission made and will not be submitted by us to any other university in the future.

Place: *Bangalore*
Date: *23/7/2021*

Name and USN

ABHISHEK GOWDA M V (1KS17CS002)

ASHISH K AMAR (1KS17CS013)

KRITHIKA JAGANNATH (1KS17CS036)

MEGHANA G (1KS17CS042)

Signature

.....*Abhishek*.....

.....*Ashish*.....

.....*Krithika*.....

.....*Megha*.....

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task will be incomplete without the mention of the individuals, we are greatly indebted to, who through guidance and providing facilities have served as a beacon of light and crowned our efforts with success.

First and foremost, our sincere prayer goes to almighty, whose grace made us realize our objective and conceive this project. We take pleasure in expressing our profound sense of gratitude to our parents for helping us complete our project work successfully.

We take this opportunity to express our sincere gratitude to our college **K.S. Institute of Technology**, Bengaluru for providing the environment to work on our project.

We would like to express our gratitude to our **MANAGEMENT**, K.S. Institute of Technology, Bengaluru, for providing a very good infrastructure and all the kindness forwarded to us in carrying out this project work in college.

We would like to express our gratitude to **Dr. K.V.A Balaji**, CEO, K.S. Group of Institutions, Bengaluru, for his valuable guidance.

We would like to express our gratitude to **Dr. Dilip Kumar K, Principal/Director**, K.S. Institute of Technology, Bengaluru, for his continuous support.

We like to extend our gratitude to **Dr. Rekha.B.Venkatapur, Professor and Head**, Department of Computer Science & Engineering, for providing a very good facilities and all the support forwarded to us in carrying out this project work successfully.

We also like to thank our Project Coordinators, **Mr. K Venkata Rao, Associate Professor, Dr. Vaneeta M, Associate Professor, Mr. Raghavendrachar S, Asst. Professor, Dr. Aditya Pai H, Asst. Professor, and Mrs. Sneha K, Asst. Professor, Department of Computer Science & Engineering** for their help and support provided to carry out the project and complete it successfully.

Also, we are thankful to **Dr. Rekha.B.Venkatapur, Professor and Head**, for being our Project Guide, under whose able guidance this project work has been carried out and completed successfully.

We are also thankful to the teaching and non-teaching staff of Computer Science & Engineering, KSIT for helping us in completing this project work.

**ABHISHEK GOWDA M V
ASHISH K AMAR
KRITHIKA JAGANNATH
MEGHANA G**

ABSTRACT

COVID-19 Pandemic has changed the lifestyle of most of the people and one of the changes is meeting anyone in person. People in remote areas have difficulty traveling to a clinic. But as of now, most of the consultations are done online / over the phone, which results in miscommunication of the medicines prescribed.

Also, if the patient forgets the medicine name, dosage, he /she has to keep contacting the doctor again and again. Our idea mainly focuses on eliminating these errors and helping the doctors generate prescriptions by voice commands and send the same to the patient as SMS/PDF.

This application helps the doctors to generate patient prescriptions using voice commands and send the same as an SMS for people who do not have the facility of a smartphone or as a pdf to the patients over social media. Furthermore, the application has features wherein the doctors can access records of patients and send messages. Patients can book appointments based on doctor's availability on our app or using the IVR system.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
1.	INTRODUCTION	01
1.1	Over View	01
1.2	Purpose of the project	02
1.3	Scope of the project	02
1.4	Definitions	02
2.	LITERATURE SURVEY	05
2.1	Problem Statement	8
3.	SYSTEM REQUIREMENT SPECIFICATION	9
	12	
3.1	Hardware Requirements	9
3.2	Software Requirements	9
	3.2.1 Adobe XD	10
	3.2.2 Node JS	10
	3.2.3 My SQL	10
	3.2.4 Postman	11
	3.2.5 React JS	11
	3.2.6 AWS	11
	3.2.7 Google Speech to text	11
	3.2.8 Twilio	12
	3.2.9 HTML	12

	3.2.10 CSS	12
3.3	Non-Functional Requirements	13
4.	DESIGN	14
4.1	System Architecture	14
4.2	Data Flow Diagram	16
	4.2.1 Constructing Data Flow Diagram	17
4.3	Sequence Diagram	17
4.4	Class Diagram	19
4.5	Activity Diagram	20
4.6	Use Case Diagram	21
5.	IMPLEMENTATION	24
5.1	Programming Language Selection	24
5.2	Factors to Consider	24
5.3	Coding	27
	5.3.1 Speech-to-text conversion using web kit speech recognition API.	27
	5.3.2 Code for Interactive Voice Response(IVR) system.	28
6.	TESTING AND RESULTS	30
6.1	Types of Tests	30
	6.1.1 Unit Testing	30
	6.1.2 Integration Testing	30
	6.1.3 Functional Testing	31

	6.1.4 System Testing	31
	6.1.5 White Box Testing	31
	6.1.6 Black Box Testing	32
	6.1.7 Acceptance Testing	32
6.2	Test Cases	32
6.3	Performance Evaluation	34
	6.3.1 Evaluation of booking system	34
	6.3.2 Evaluation of speech recognition module	34
	6.3.3 Evaluation of prescription formatting module	34
	6.3.4 Evaluation of IVR system	34
7.	SNAPSHOTS	35
7.1	Home Page	35
7.2	Patient Login Page	36
7.3	Patient Register Page	36
7.4	Doctor Login Page	37
7.5	Doctor Register Page	37
7.6	Patient Dashboard Page	38
7.7	Patient Appointment Page	38
7.8	Slot booking based on doctor	39
7.9	Slot time selection page	39
7.10	Slot booking based on category	40
7.11	Doctor Dashboard Page	40

7.12	Doctor Appointment Page	41
7.13	Add Patient Page	41
7.14	Prescription Generation Page	42
7.15	Generated Prescription Page	43
8.	CONCLUSION	44
9.	FUTURE ENHANCEMENTS	45
	REFERENCES	46
	CERTIFICATES	47

LIST OF FIGURES

Fig. No.	Figure Name	Page No.
4.1	Architecture of proposed system	15
4.2	General notation of DFD	16
4.3	Sequence diagram of proposed system	18
4.4	Class diagram of proposed work	19
4.5	Activity diagram of proposed work	20
4.6	Use Case diagram for doctor	22
4.7	Use Case diagram for patient	23
7.1	Home Page	35
7.2	Patient Login Page	36
7.3	Patient Register Page	36
7.4	Doctor Login Page	37
7.5	Doctor Register Page	37
7.6	Patient Dashboard Page	38
7.7	Patient Appointment Page	38
7.8	Slot booking based on doctor	39
7.9	Slot time selection page	39
7.10	Slot booking based on category	40
7.11	Doctor Dashboard Page	40
7.12	Doctor Appointment Page	41
7.13	Add Patient Page	41
7.14	Prescription Generation Page	42
7.15	Generated Prescription Page	43

LIST OF TABLES

Table. No.	Table Name	Page No.
2.1	Comparison between TATA health and E Sanjeevini	6
2.2	Comparison between Tele Arogya and board of doctors	6
6.2	Test Cases	32

Chapter 1

INTRODUCTION

1.1 Overview

The global spread of Novel Coronavirus illness poses a serious threat to millions of individuals around the world. Annual conferences of various sizes, as well as small group meetings, have migrated to the internet. The new format (virtual mode) presents various significant technical and organizational obstacles, but it also presents opportunity. Using our app, one may virtually reach out to patients and consult with them.

Furthermore, bad drug effects are a leading cause of death worldwide, with tens of thousands of people dying each year as a result of medication or prescription errors. Due to indecipherable handwritings, drug interactions, and ambiguous drug names, many of these errors have an impact on caregivers' administration of the incorrect substance or dosage to patients. Voice-based e-prescription, that allow prescription information to be recorded as a voice and then heard by voice response rather than in the doctor's handwriting, may avoid some of these flaws. The fundamental goal of this project is to create and build a voice-based E-Prescription with the option to book appointments utilizing an IVR system.

This application helps the doctors to generate patient prescriptions using voice commands and send the same as an SMS for people who do not have the facility of a smartphone or as a pdf to the patients over social media. Furthermore, the application has features wherein the doctors can access records of patients and send messages. Patients can book appointments based on doctors availability on our app or using the IVR system.

1.2 Purpose of the project

The whole idea came into picture due to COVID and how it has made it difficult to meet anyone in person. People in remote areas have difficulty traveling to a clinic. Now that most of the consultations are done online / over the phone, which might result in miscommunication of the medicines names.

Also if the patient loses the prescription, they have to keep contacting the doctor again which is troublesome.

Our idea mainly focuses on eliminating these errors. As a matter of fact, few articles show that thousands of people die in a year due to shabby handwriting of doctors. And Modi government wants to define e-prescription format to regulate online sales of medicines. This is the reason the whole idea of voice based E-Prescription came into picture, to have a contactless way for the doctors to treat patients, and this idea also saves time and energy for the doctors as they need not keep writing prescriptions by hand.

1.3 Scope of the Project

The scope of this project is as follows:

1. Achieve contactless consultation which is a major advantage during the pandemic.
2. Provide a system to reach out to all the people irrespective of the type of device they own.
3. Propose a system for easy access of prescriptions on both the patient and the doctor side.
4. Provide a system for the doctor to make the process of generating prescriptions easily.
5. Help promote the moto of 'Digital India' by saving trees as this project is generating digital prescriptions.

1.4 Definitions

1.4.1 IVR

Interactive Voice Response (IVR) is an acronym for Interactive Voice Response system. It is a method of communicating with your users over the phone. Voice and DTMF tones produced by phones while pressing buttons on the keypad are used to control IVR. Considering how one deals with Airtel's customer service.

1.4.2 Database

A database is a data management tool that enables for quick data storing and retrieval. There are other sorts of databases, but the most common is a relational database, which stores data in tables with the same type of information in each row. These refer to the way data is kept and the relationships between tables.

1.4.3 Adobe XD

Adobe Experience Design is another name for this. It's a user experience design tool for online and mobile apps that's built on vectors. This is used to provide a more user-friendly interface.

1.4.4 React JS

It's mostly used to manage the view layer for mobile and web apps, and it's an open-source JavaScript framework that's primarily used to create single-page applications.

1.4.5 Node JS

This is typically used outside of a browser to run JavaScript code. It's an open source back-end, cross-platform Javascript runtime environment. Another benefit is that it enables developers to use JavaScript to create command-line tools and server-side scripting, which aids in the creation of dynamic web pages prior to transmission to the user's browser.

1.4.6 MySQL

Mysql is a relational database system that can be used for e-commerce, data warehousing, and logging, among other things. This is a database management system that uses a structured query language .

1.4.7 Postman

This is a client for an API. This enables sharing, testing, creating, and documenting APIs simple for developers like us. This is accomplished by sending and receiving simple and complicated HTTP/s queries, as well as reading their answers. This would result in less time-consuming effort and a more efficient approach.

1.4.8 Google speech to text API

The Google Speech API, also known as Cloud Speech-to-Text, is a complex tool that converts voice to text using Google's machine learning technology. Speech to text is performed in this application while producing prescriptions, which is one of the key features.

1.4.9 Webkit speech recognition API

The Speech Recognition Event is sent by the recognition service module, and here is how it is handled. The Web Speech API's Speech Recognition interface is simply a controller for the available recognition service.

1.4.10 Twilio

This is a global messaging API that can send and receive MMS, OTT, and SMS messages. It simply makes use of intelligent sending characteristics to ensure that messages reach end users consistently, no matter where they are, with minimal procedure and faults.

1.4.10 Fast 2 SMS

Fast2SMS is a well-known bulk SMS service in India. You may anticipate high-quality SMS services from us, as known for its performance-driven messaging services. In addition to the two significant features, In this project Fast2SMS is used for sending SMS to the non - smartphone users.

Chapter 2

LITERATURE SURVEY

For the Literature Survey, Tata health, E-sanjeevini, Tele Arogya and Board of Doctors are taken into consideration. All of the compared applications are up and running. In these applications, The prescription generation is manual unlike the proposed system.

All the applications have an option to book appointments online but the proposed system not only gives the option to book consultations online but also gives option for the non-smartphone users to book consultations via the IVR service. This IVR service is unique to our proposed system. All of these applications generate E-prescription but the proposed system not only generates E-prescription but also generated audio prescriptions and these are stored in the application itself, which can be accessible to the patients with smartphone or without smartphone.

The non-smartphone users can fetch their audio prescriptions via the IVR service. Most of the mentioned applications have video consulting which is not the provided option in the proposed system.

Finally, after comparing with the related work it can be told that the existing systems offer consultation facilities but it limits its facilities to only high end devices like smartphone/PC users. In the proposed system, all the users will have to go through a one time registration process, which helps the system maintain patients' profiles as well as classify smartphone and basic cell phone users.

Table 2.1 Comparison between Tata health and E-sanjeevini

Attributes	Tata health	E-sanjeevini	Proposed System (Voice Based E-Prescription)
E-Prescription generation (by Voice/manual)	Manual	Manual	Voice
Appointments	Yes	Yes	Yes
IVR Service	No	No	Yes
Prescription sending format	In App	Link	SMS / Social Media App/ In App
Access to E-prescriptions and audio prescriptions	E-prescription	E-prescription	E-prescription and audio prescription
Video Call	Yes	Yes	No
Link to the App	https://www.tatahealth.com/	https://esanjeevini.tatahealth.com/	In Progress

Table 2.2 Comparison between Tele Arogya and Board of Doctors

Attributes	Tele Arogya	Board Of Doctors	Proposed System (Voice Based E-Prescription)
E-Prescription generation (by Voice/manual)	Manual	Manual	Voice
Appointments	Yes	Yes	Yes
IVR Service	No	No	Yes
Prescription sending format	In App	In App	SMS / Social Media App/ In App
Access to E-prescriptions and audio prescriptions	E-prescription	E-prescription	E-prescription and audio prescription
Video Call	Yes	Yes	No
Link to the App	https://telearogya.com/	https://boardofdoctors.com/	In Progress

Comparing each of these applications in detail:

Tata health:

It provides Safe and private chats - Instant consultation ,Online Doctor Consultation, Video Consultation ,Access to doctors for any medical issues, Footsteps tracker built-in - Lab and medicine integration - Medicine reminders. In this application there is manual generation of prescriptions, IVR service is not available unlike the proposed system.

E-Sanjeevini:

It provides free service with sending e-prescriptions and they also have appointment booking page on the website for easy consultation with token generation. They send E-prescription via SMS/Email. But, when compared to the proposed system this does not have Voice based generation of prescription and also this is only available to people with smartphone, which also means no IVR service.

Tele Arogya:

Connecting with doctors is made easy and consultations are made faster, secure and also easier, but this app also does not provide generation of voice prescription and also this is not available for all the users(smartphone and non-smartphone users)

Board of Doctors:

This web application provides a wide range of options to their users like viewing appointments, Uploading their receipts, updating their medical history, getting to choose between specialists and also viewing their prescription but this application also does not offer voice based e prescription generation and this is also limited only to the smartphone users.

2.1 Problem Statement

Due to the current covid-19 situation most of the consultations are done online/over a phone call due to which there are chances of having miscommunication in the medicines prescribed. Our idea mainly focuses on eliminating these errors and helping the doctors generate prescription by voice commands send the same to the patient as SMS/PDF.

Chapter 3

SYSTEM REQUIREMENT SPECIFICATION

A software requirements specification (SRS) is a detailed outline of the software's intended function and environment. The SRS explains in detail what the programme will do and how it is anticipated to perform. Software requirements definition allows for a thorough evaluation of needs prior to the start of design, reducing the need for subsequent rework. It must also offer a genuine service. It should also serve as a solid foundation for predicting product prices, risks, and timelines.

To extract the requirements, one must have a clear and thorough understanding of the goods that will be developed or that are currently being developed. The software requirements specification document enumerates all of the necessary requirements for the project's development. This is accomplished and fine-tuned by extensive and ongoing contact with the project team and the customer until the software is finished.

3.1 Hardware Requirements

The following are the hardware requirements:

- Processor: Intel i3 2nd generation and above
- RAM:4GB
- Hard Disk: 500GB

3.2 Software Requirements

The following are the software requirements:

- Adobe XD
- Node JS
- My SQL
- Postman
- React JS
- AWS
- Google speech to text
- Twilio
- HTML
- CSS

3.2.1 Adobe XD

Adobe Experience Design is another name for this. It's a user experience design tool for online and mobile apps that's built on vectors. This is used to provide a more user-friendly interface.

It is a tool with vector-based interface. It may be used to generate and collaborate on everything from prototypes to mockups to complete designs.

3.2.2 Node JS

This is mostly for running JavaScript code outside of a browser. Another benefit is that it enables developers to use JavaScript to create command-line tools and server-side scripting, which aids in the creation of dynamic web pages before they are sent to the user's browser. It's a free, open-source back-end Javascript runtime environment.

As a consequence, Node.js symbolises a "JavaScript everywhere" paradigm, bringing together web application development under a single programming language rather than two distinct languages for server-side and client-side scripts.

Regardless of the fact that .js is the normal JavaScript filename suffix, "Node.js" refers to the overall entity rather than a single file.

3.2.3 My SQL

Mysql is the most popular system that can be used for e-commerce, data warehousing, and logging, among other things. This is a relational database management system that uses a structured query language (SQL).

MySQL is offered under a variety of proprietary licences in addition to the GNU General Public License, which makes it free and open-source software. MySQL AB, a Swedish company that was later acquired by Sun Microsystems, designed and maintains the database (now Oracle Corporation). After Oracle bought Sun in 2010, Widenius forked the open-source MySQL project to create MariaDB.

3.2.4 Postman

This is a client for an API. This enables sharing, testing, creating, and documenting APIs simple for developers like us. This is accomplished by sending and receiving simple and complicated HTTP/s queries, as well as reading their answers. This would result in less time-consuming effort and a more efficient approach.

3.2.5 React JS

It's mostly used to manage the view layer for mobile and web apps, and it's an open-source JavaScript framework that's primarily used to create single-page applications.

React, often known as React.js or ReactJS, is a free and open-source JavaScript front-end toolkit for developing user interfaces and UI components. React may be used to create single-page and mobile apps as a foundation. React, on the other hand, is just interested in state management and processing information. Because React programmes typically require the use of additional frameworks for routing and client-side functionality, building React apps frequently demands the use of additional frameworks.

3.2.6 AWS

AWS CodeDeploy is a completely managed software delivery solution for Amazon EC2, AWS Fargate, AWS Lambda, and on-premises servers. AWS CodeDeploy can be used to streamline software deployments, eliminating the need for time-consuming and error-prone manual operations.

3.2.7 Google Speech to text

The Google Speech API, also known as Cloud Speech-to-Text, is a complex tool that converts voice to text using Google's machine learning technology. Speech to text is performed in this application while producing prescriptions, which is one of the key features.

3.2.8 Twilio

This is a global messaging API that can send and receive MMS, OTT, and SMS messages. It simply makes use of intelligent sending characteristics to ensure that messages reach end users consistently, no matter where they are, with minimal procedure and faults.

3.2.9 HTML

HTML is the industry standard markup language for web-based text, commonly known as HyperText Markup Language. Web browsers convert HTML texts obtained from a web server or locally saved files into multimedia web pages using Cascading Style Sheets (CSS) and computer languages such as JavaScript. HTML supplied visual indications for the document's presentation and logically reflected the structure of a web page in its early versions..

HTML elements are the components that make up an HTML page. Images and other objects, such as interactive forms, can be included into the created page using HTML techniques. HTML helps you to produce well-organized documents by specifying structural semantics for text components such as headings, paragraphs, lists, links, quotations, and other elements. HTML allows programming language like JavaScript to add programmes to web pages that change their ways and content. CSS governs how a piece of information appears and is organised.

3.2.10 CSS

Cascading Style Sheets, abbreviated CSS, is a style sheet language for specifying how a document produced in a markup language like HTML should appear. CSS is an integral aspect of the World Wide Web, just like HTML and JavaScript. CSS is a style sheet that allows you to customise layout, colors, and fonts by separating presentation from text. This separation of presentation and content, give more freedom and control in the specification of appearance features, and allow several web pages to share formatting by expressing the appropriate CSS in a distinct file..

CSS sheet, resulting in less repetition and intricacy. allowing the.css file to be cached in order to speed up page loading Because styling and content are separated, the same HTML page can be displayed in a variety of styles for multiple rendering modalities, such as on-screen, in print, by voice (via a speech-based browser or screen reader), and on Braille-based interactive applications. CSS is a collection of rules for various formatting that can be accessible on a smartphone if the data is accurate.

3.2 Non-Functional Requirements

A non-functional specification is a requirement in systems engineering and system requirements engineering that defines criteria that can be used to judge the performance of a system rather than specific behaviours.. The system meets the non-functional requirements listed below:

- 3.2.1 Efficiency
- 3.2.2 Extensibility
- 3.2.3 Maintainability
- 3.2.4 Platform compatibility
- 3.2.5 Portability
- 3.2.6 Quality
- 3.2.7 Reliability
- 3.2.8 Robustness
- 3.2.9 Supportability
- 3.2.10 Testability
- 3.2.11 Usability

Chapter 4

DESIGN

The process of establishing the architecture, components, modules, interfaces, and data for a system in order to meet specific criteria is known as systems design. The application of systems theory to product development is known as systems design..

The significance of this phase can be appreciated because it entails identifying data sources, as well as the nature and type of data that is available. In order to construct a salary system, for example, inputs such as attendance, leave details, additions or deductions are required. This makes it easier to figure out what kind of data is accessible and who supplies it to the system, allowing the system to be developed with all important elements in mind. Furthermore, system design ensures that the system is developed in such a way that it meets the needs of the users and puts them at rest, as it is user-oriented. One of the key goals of this phase, in terms of flexibility, is to create a system that can be dynamic in nature and sensitive to changes as needed. Another essential goal of the system design process is to create a system that can perform efficiently, giving the desired output while also being sensitive to time within a set time limit.

4.1 System Architecture

A system's architecture is a conceptual model that describes the system's structure, behaviour, and other aspects. A systematic study and representation of a system arranged in a way that facilitates reasoning about the system's structures and behaviours is known as an architecture description. It could be made up of system components that will successfully integrate the overall system in place. Figure 4.1 represents the main architecture of our web application.

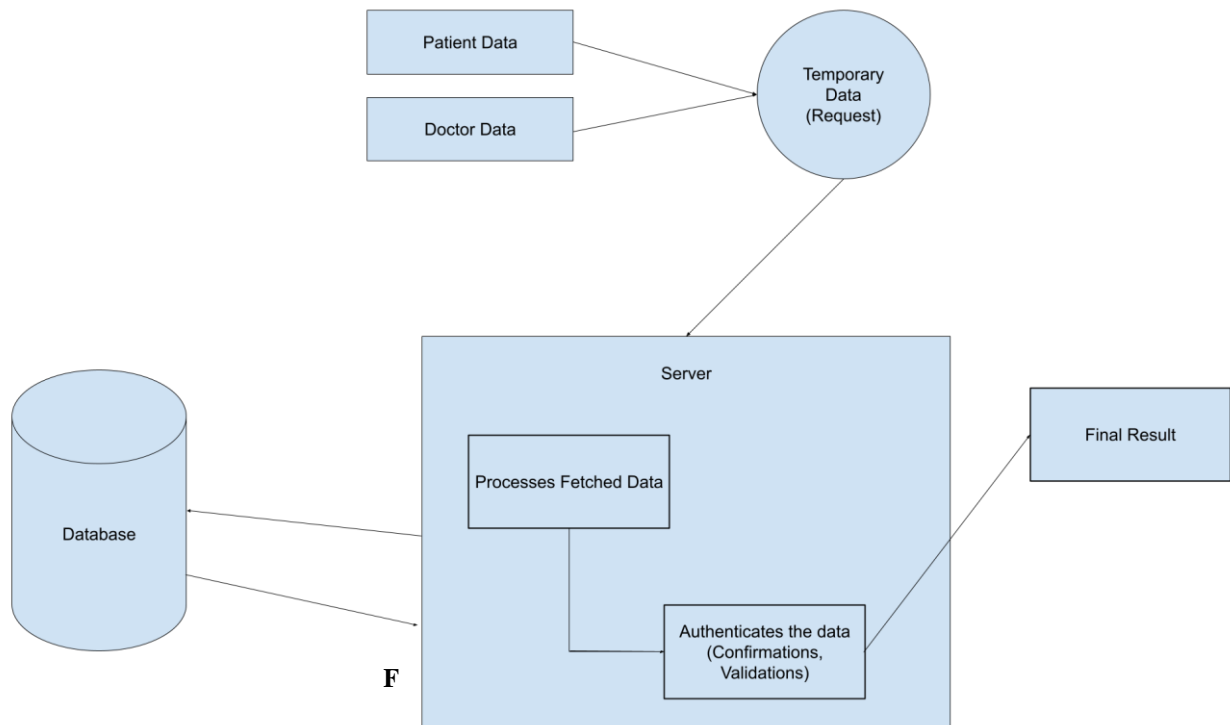


Figure 4.1: Architecture of Proposed system

The above shown architecture shows us how the flow happens. Let's say the patient or doctor send requests (logs into our app or makes an operation say fetches the appointment details). That request will be sent to the server for processing as a request object. You can see more of it in the class diagram below.

Now, that request object is processed by the server and it makes appropriate database requests (for logging in or making the queries for list of doctors or list patients available). Once that data is received from the database, if it is there or not, it will be processed or cleaned and sent back to the user as a final result. That is, it updates the content in the application for that particular action performed by the respective user.

4.2 Data Flow Diagrams

Data flow diagrams are the fundamental building blocks that define the flow of data in a system to a certain destination, as well as the variation in flow when a transition occurs. By splitting the procedure into sub-processes, it makes the entire procedure look like a decent document and makes it simpler and easier to grasp for both programmers and non-programmers. Dataflow diagrams are simplistic blocks that demonstrate the interaction between various functional units and provide a high-level overview, as well as the limits and details of the system's constituents.

The dataflow diagrams begin at the origin and terminate at the destination level, decomposing from high to low levels. The following are the most significant points to remember regarding dataflow diagrams: It shows the dataflow in one direction but not in loop structures, but it doesn't really show the time factors. This report explains the dataflow analysis, which includes information on the data that was utilised, the classification of dataflow diagrams based on their purposes, and the many levels that were employed in the project. The general notations for constructing a block diagram are shown in the below Figure 4.6:



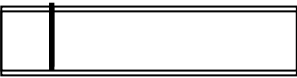
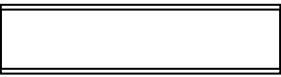
	Data flow process
	Process
	Data store
	Source

Figure 4.2: General notation of DFD

Data flow process: This determines the direction of data flow from one entity to another.

Process: The source from which the output is generated for the specified input is defined by the process. It describes the operations that are carried out on data in order to change, store, or distribute it.

Data store: The space or actual location where the data is held once it has been extracted from the data source is known as the data store.

Source: It is the data's starting point or destination point, the point at which an external entity acts as a catalyst for the data to flow to its intended destination.

4.2.1 Constructing Data-Flow Diagram

Dataflow diagrams can be created by splitting the process into several levels, such as DFD 0, DFD 1, DFD 2, and so on. These easy procedures must be performed in order to complete this process:

- The dataflow diagram can only be created if the process has only one dataflow in and one dataflow out.
- The incoming and departing data should be modified by the procedure.
- The data storage should not be isolated; it should be linked to at least one other process.
- Only one dataflow should be used by the process's external entities.

The dataflow in data processing should be from left to right and from top to bottom. The data stores and their destinations should be called with capital letters in the dataflow diagram, and the data flow and process should be named with a tiny capitalising initial letter.

4.3 Sequence Diagram

A sequence diagram is a graphical representation that depicts how processes interact with one another and in what order they occur. It's a Message Sequence chart construct. A sequence diagram depicts item interactions in chronological order. It illustrates the scenario's objects and classes, as well as the sequence of messages sent between the objects required to carry out the types of scenarios functionality. In the Logical View of the system under development, sequence diagrams are often related with use case realisations. Sequence diagrams are sometimes known as event diagrams or scenario diagrams.

A sequence diagram depicts multiple processes or things that exist simultaneously as parallel vertical lines (lifelines), and the messages passed between them as horizontal arrows, in the order in which they occur. This enables for the graphical specification of simple runtime scenarios.

The Figure 4.3 shows the sequence diagram of the predictor tool:

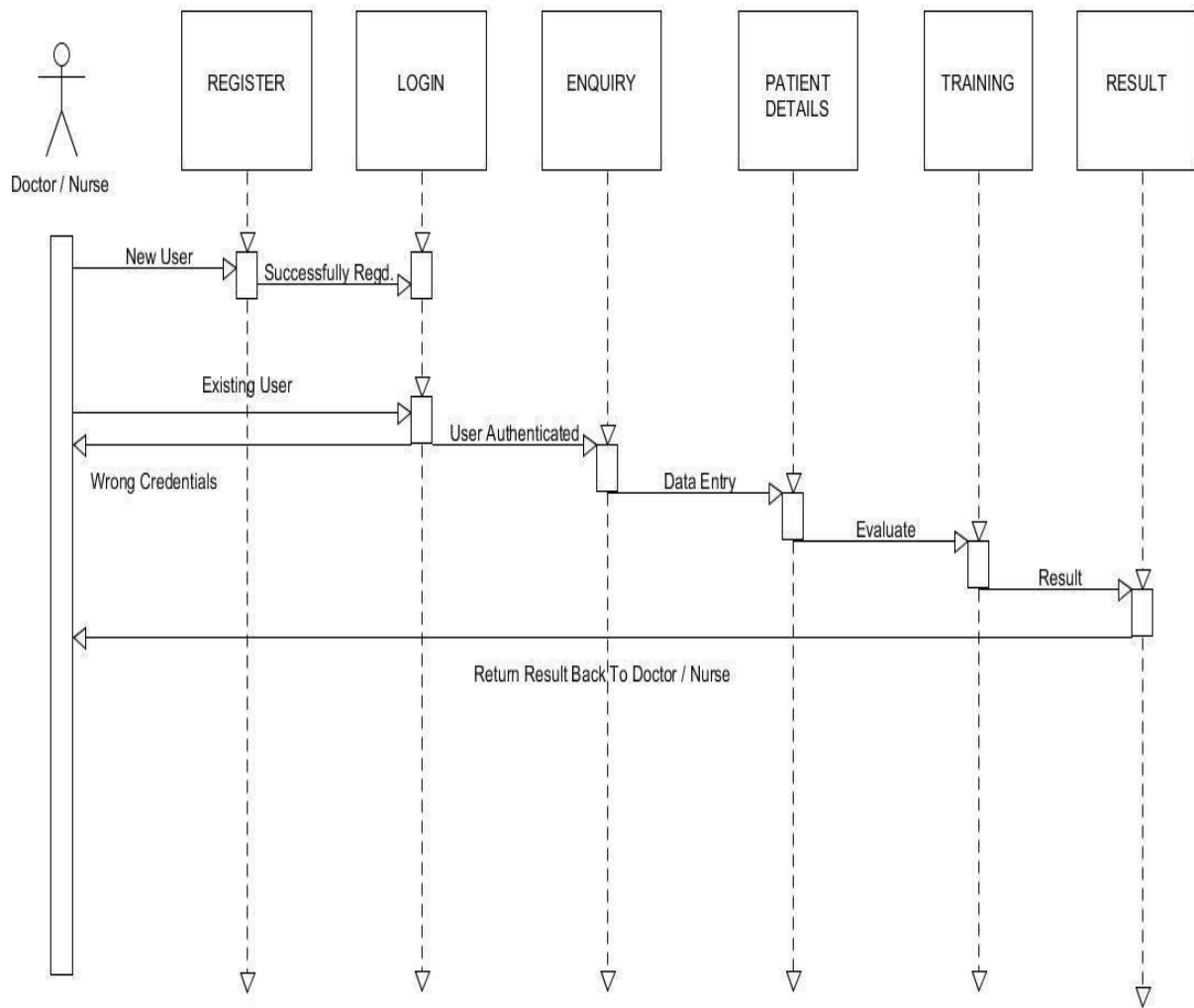


Figure 4.3 Sequence diagram of proposed work

4.4 Class Diagram

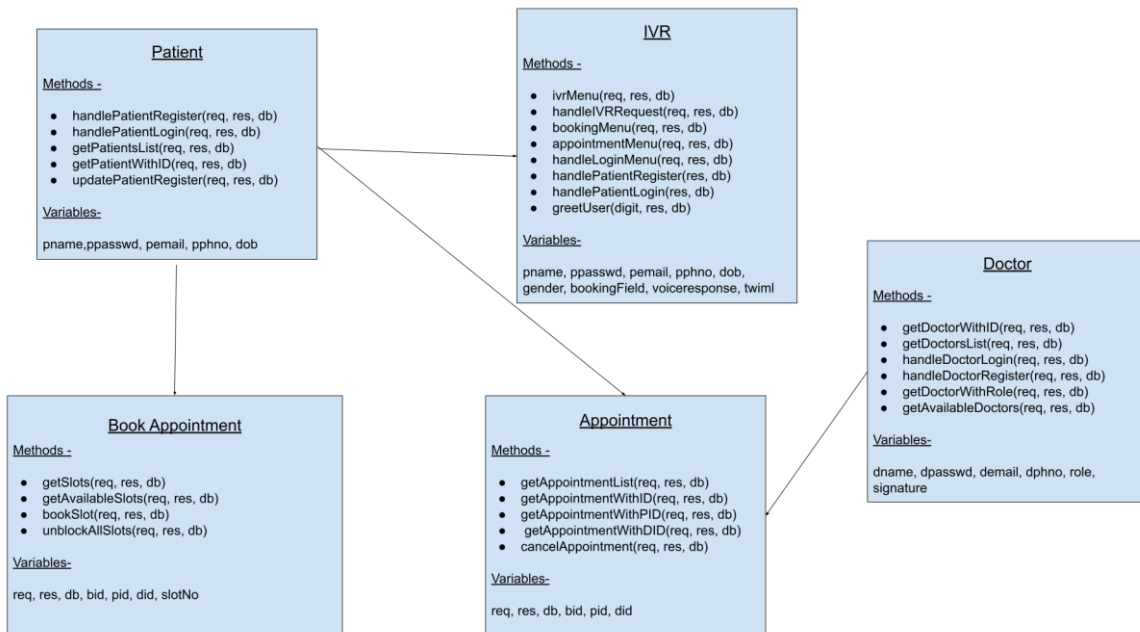


Figure 4.4 Class Diagram of the proposed work

A class diagram in the Unified Modeling Language (UML) is a form of static structural diagram in software engineering that depicts the structure of a system by displaying the system's classes, properties, operations (or methods), and relationships among objects. In the class diagram that is shown in the figure 4.4, dependency relationship is used. The changes made at the head of the arrow reflect in the one at tail.

In figure 4.4, it shows a class diagram of the overall system. It can be seen that the methods of patient can be used to trigger the methods in book appointment. In the same way, patient can also trigger events in the IVR module and the appointment module. The doctor can also trigger actions in appointments like canceling patient appointments or viewing them by making changes on his / her side and they will be reflected in the appointment module.

4.5 Activity Diagram

Activity diagrams are visual portrayal of workflows that include choice, iteration, and concurrency. Activity diagrams are used to represent both computational and organisational processes in the Unified Modelling Language (i.e. workflows). Activity diagrams show the overall flow of control.

Activity diagrams are made up of a small number of shapes that are connected by arrows. The most important form categories:

- 4.5.1 Actions are represented by rounded rectangles;
- 4.5.2 Diamonds are symbolic of choices;
- 4.5.3 The start (split) and end (join) of concurrent activities are represented by bars;
- 4.5.4 The start (starting state) of the workflow is represented by a black circle;
- 4.5.5 The end is symbolised by a black circle that is encircled (final state).
- 4.5.6 The chronology in which actions occur is represented by arrows that run from the beginning to the conclusion.

Flowcharts can be thought of as activity diagrams. Concurrency cannot be expressed using traditional flowchart techniques. However, the join and split symbols in activity diagrams only fix this for simple scenarios; when they are arbitrarily mixed with decisions or loops, the model's meaning is unclear. The activity diagram of IVR system is shown in Figure 4.5 .It represents the entire workflow of the project.

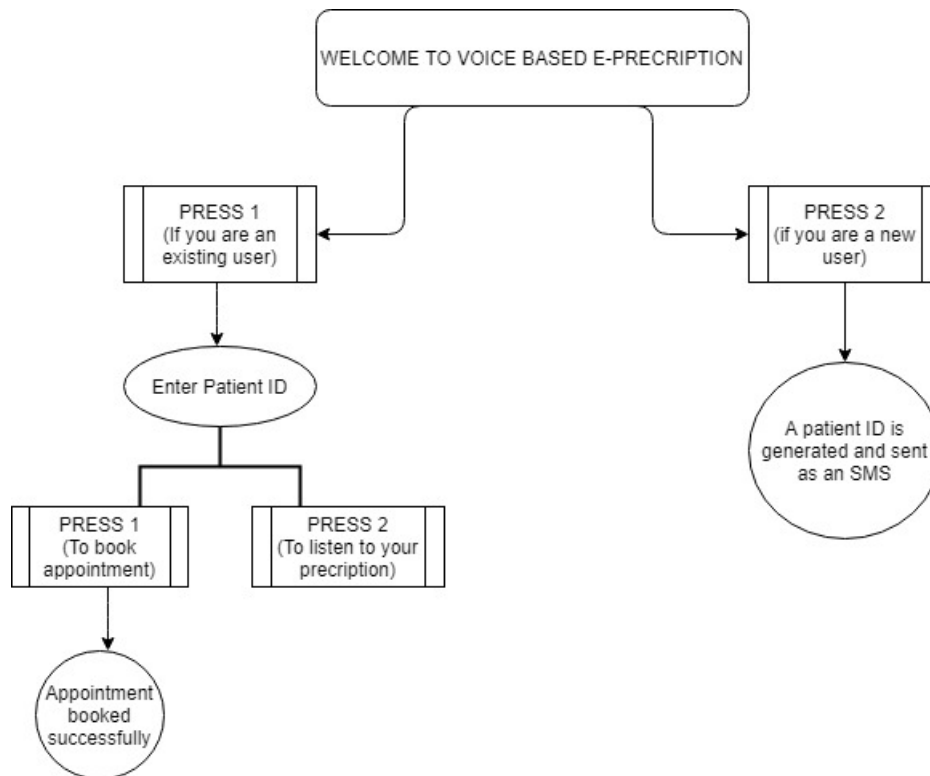


Figure 4.5: Activity diagram of proposed work

The first event is triggered by an incoming call to the IVR system. Based on user input the IVR flow goes along with it. If it's a new user to the system, he / she will press 2 and a record will be created.

If the patient is an existing user then he / she will choose option 1 and go with the flow to book an appointment or listen to the prescription.

4.6 Use Case Diagram

At its most basic level, a use case diagram depicts a user's interaction with the system by illustrating the relationship between the user and the many use cases in which the user is involved. A use case diagram can be used to identify the many sorts of users of a system as well as the various use cases, and it is frequently supplemented by other types of diagrams.

A doctor can register/login to the application, can enquire about a patient, register

a new patient by entering the patient's medical details, can retrieve an existing patients detail, form prescription and also view the result. The below Figure 4.10 shows the use case diagram of the proposed system:

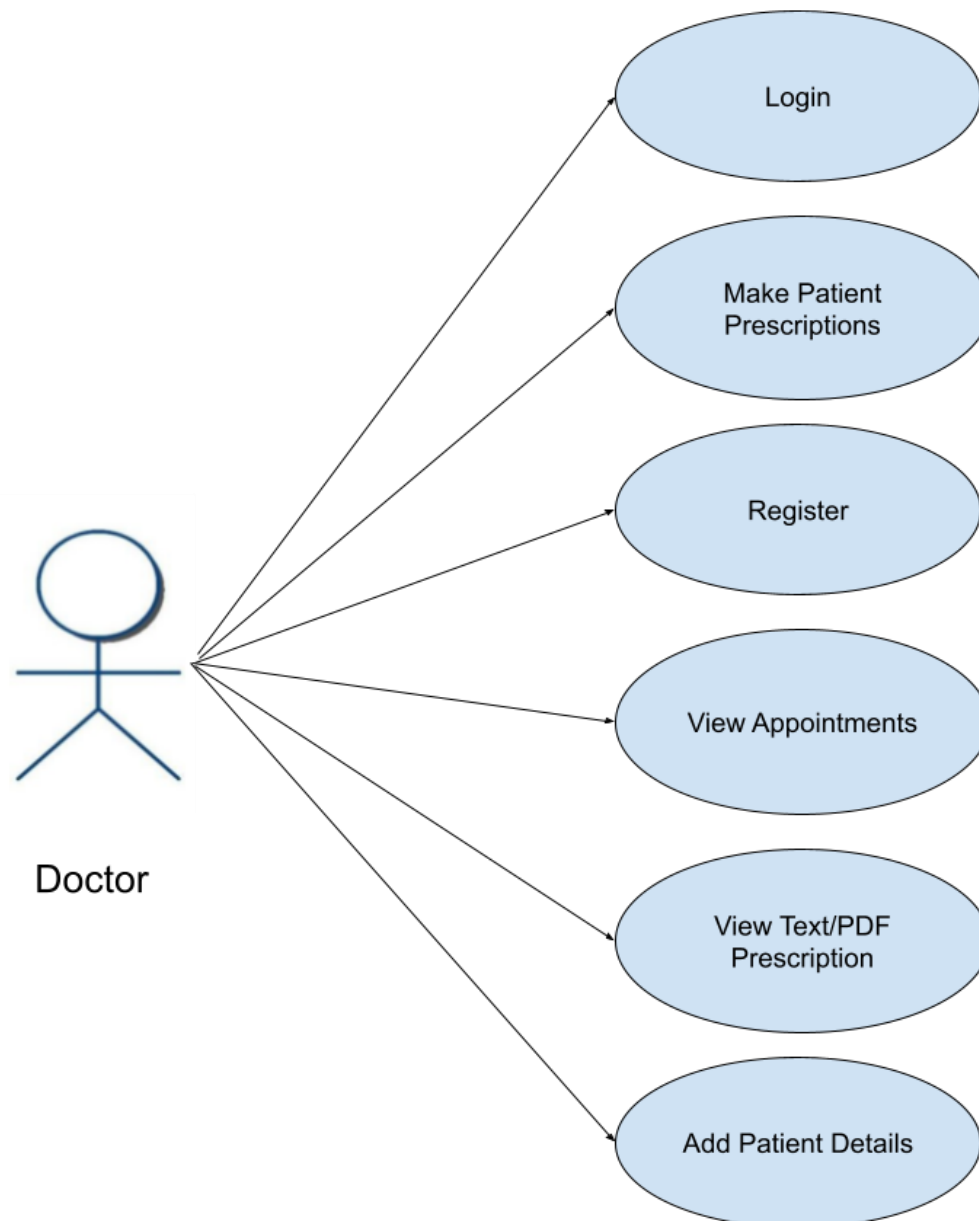


Figure 4.6 Use Case diagram for doctor

Now, let's look at the patient's UML diagram. The patient will be able to login/register with our application. In this project similar features for a user with a smartphone have been provided and another user without a smart phone. Regardless of the type of device the patient has he / she will be able to book appointments, login, register, view the prescription (both audio and the file) and view appointments. The below Figure 4.11 shows the use case diagram of the proposed system:

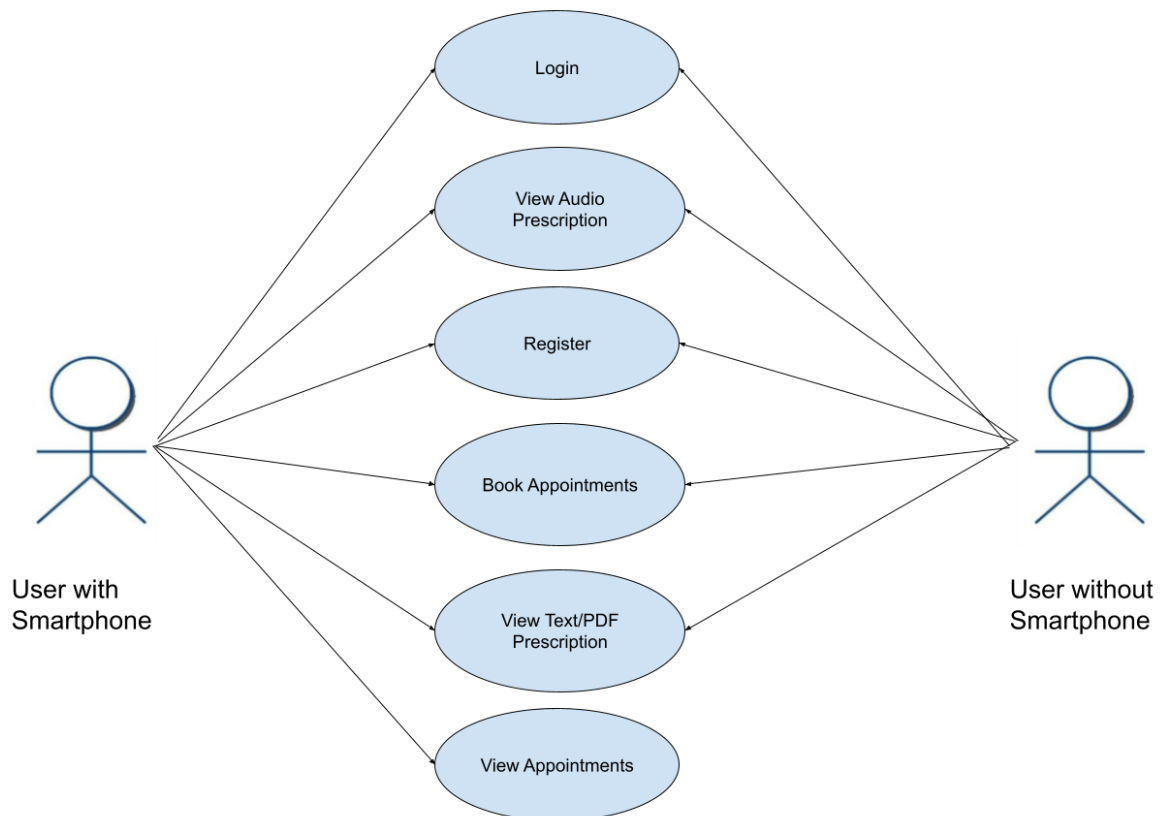


Figure 4.7 Use Case diagram for Patient

Chapter 5

IMPLEMENTATION

5.1 Programming Language Selection

When it comes to developing software, there are a lot of different programming languages to pick from, and it's easy to become lost in the details of each one. Many things can influence your language choice. If it's for a personal project or pastime, you might as well choose a language you're already familiar with. If you base your decision on available resources, you may wind up with quite cryptic solutions. Alternatively, you may spend a lot of effort creating reusable components, which will make documentation a pain. There isn't a duplicate comparison of procedural, object-oriented, and functional languages in this article. It demonstrates how to choose a language for your project that is both efficient and easy to develop using real-world examples and scenarios. It enables you to review a number of things to consider when choosing a programming language, whether for personal usage or for a large project within a company.

5.2 Factors to consider:

When it comes to choosing a programming language, there isn't just one thing to consider. JavaScript, for example, is a simple, easy-to-learn language structure that emphasises coherence and thereby lowers the cost of programme support. There is no single language that is the "best option." Though you may prioritise certain variables in enterprise applications, such as performance and security, other ones, such as fewer lines of code, may be lower on your priority list. There's always a trade-off to be made. Before you start working on a project or assignment, you'll probably need to conduct some preliminary work. By far the most disregarded aspect of this preparation is language selection. You can choose a personal favourite when choosing a language for a personal project. The number of lines of code is critical here; the obvious choice is a language that can complete the task in 10 rather than 20 lines of code. You should focus on getting the solution out first, then on neatness and performance. It's a different storey when it comes to initiatives for huge corporations.

To tackle a certain problem, they interact and integrate with one another. The language chosen may be influenced by considerations like as the ease with which the software can be ported to another platform or the availability of resources. Choosing the correct programming language can result in solutions that are compact, simple to debug, expand, document, and fix.

Factors to considered for selecting programming language/tech stack:

- 1) Building APIs
- 2) A language's adaptability
- 3) Production Time
- 4) Support for mobile applications
- 5) The support and community

Today's greatest popular software ecosystem is the JavaScript world. Its growth drove it from a basic scripting tool to the foundation of today's internet. Every day, browsers become more powerful, the NPM registry increases to an astounding 1 million packages, and Node has long proven itself as a capable back-end language.

5.2.1 Building APIs

Node is the logical heir to the legacy of JS, which was created to be used on the backend. If the code is placed on a Node backend, a simple URL call can be used to run it.

Representational State Transfer (REST) is an acronym for Representational State Transfer. REST, a web-based architecture, use the HTTP protocol. Every component is a resource, and each resource is accessed through a standardised interface that adheres to HTTP best standards.

A REST Service just provides access to resources, but a REST client accesses and modifies those resources using the HTTP protocol. URIs/global IDs are used to identify each resource. REST supports a variety of representations to represent a resource

5.2.2 Library and frameworks

One of the reasons why JavaScript is so popular is its widespread libraries and frameworks. As a developer, one can manage both client and server side programs with knowledge of JavaScript.

A library is a reusable piece of code with a single main use case. Depending on the language, a library can have multiple functions/objects/methods. To gain access to a library's functionality, the application can link to it.

A JS framework, on the other hand, gives more control over your app. It aids in the direction of the architecture and subsequent project. Frameworks are made up of numerous libraries that come with hooks and callbacks so you may build on top of them.

5.2.3 Asynchronous and callbacks

JavaScript, in general, executes code in a non-blocking manner. This means that code that takes a long time is executed in the background while the rest of the code is done in parallel. Asynchronous programming is a word used to explain this behavior. Since JavaScript is processed in a non-blocking manner, extra care has to be taken to deal with it if you need the result before proceeding with other code.

Callbacks are short functions that alert the calling instance when an asynchronous script has completed and the result is accessible.

5.2.4 Support for mobile applications

One of JavaScript's key advantages is its cross-platform compatibility. Nowadays, mobile devices are commonly used to access the internet. JavaScript can also be used to create applications for non-web contexts. The capabilities and uses of JavaScript make it a powerful tool for creating mobile apps. The React Native framework for designing mobile apps is a popular JavaScript framework. React Native allows the developer to construct mobile apps for a range of operating systems. It is not necessary for the developer to write separate code for the iOS and Android platforms. It simply has to be written once and can run on a variety of platforms.

5.2.5 Support and Community

A programming language should have a strong community behind it, just as excellent software needs a community to help it thrive. A language with an active forum is more likely to be popular than a wonderful language that lacks support. Wikis, forums, tutorials, and, most importantly, additional libraries are all created as a result of community assistance. The days of people working in silos are long gone. People do not want to sift through all of the paperwork just to remedy a simple issue. If a language has a large following, there's a significant possibility that someone else has encountered your problem and documented it in a wiki or forum.

5.3 Coding

The program code for speech to text conversion using Web kit speech-recognition and the IVR system and Booking appointment in FCFS basis are as follows:

5.3.1.Speech-to-text conversion using web kit speech recognition API.

Importing the speech-recognition API which uses webkit-speechRecognition under the hood.

```
import SpeechRecognition, { useSpeechRecognition, } from "react-speech-recognition";
const { transcript, resetTranscript } = useSpeechRecognition({ commands });
```

Checking client(Browser) support for the API

```
if (!SpeechRecognition.browserSupportsSpeechRecognition()) {
  return alert("Browser not supported!");
}
```

Invoking the startListening method to start transcribing.

```
SpeechRecognition.startListening({
  continuous: true,
  language: "en-IN",
});
```

Invoking stopListening method to stop transcribing

```
SpeechRecognition.stopListening();
```

Invoking resetTranscript method to reset the transcription

```
resetTranscript();
```

Code to format the prescription to standard 1-1-1 format.

```
const formatFrequency = (frequency) => {
  let frequencyArr = { morning: 0, afternoon: 0, night: 0 };
  frequency.split(" ").map((word) => {
    word = word.toLowerCase();
    word = word.replace(/[\.,/#!$%^&*;:{}=\-_`~()]/g, ""); //removing punctuations
    word = word.replace(/\s{2,}/g, " "); //removing 2 or more spaces if any
    if (word in frequencyArr) {
      frequencyArr[word] = 1;
    }
    return word;
  });
  let formattedData = Object.values(frequencyArr).join("-");
  return formattedData;
};
```

5.3.2 Code for Interactive Voice Response(IVR) system.**List of all APIs used to build IVR.**

```
app.post("/ivr-request", (req, res) => {
  ivrHandler.handleIVRRequest(req, res, db);
});

app.post("/ivr/login", (req, res) => {
  ivrHandler.handleLoginMenu(req, res, db);
});

app.post("/ivr/menu", (req, res) => {
  ivrHandler.ivrMenu(req, res, db);
});

app.post("/ivr/booking-menu/:id", (req, res) => {
  ivrHandler.bookingMenu(req, res, db);
});

app.post("/ivr/appointment/:id", (req, res) => {
  ivrHandler.appointmentMenu(req, res, db);
});
```

Gathering user input and routing to menu.

```
const handleIVRRequest = (req, res, db) => {
  const voiceResponse = new VoiceResponse();
  voiceResponse.say({ loop: 1 }, "Welcome to voice based e-prescription");
  const gather = voiceResponse.gather({
    action: "/ivr/menu",
    numDigits: "1",
    method: "POST",
  });

  gather.say(
    "Please press 1 if you're an existing user. Press 2 if you're a new user",
    { loop: 2 }
  );
  res.type("text/xml");
  res.send(voiceResponse.toString());
};
```

Get entered digit from input stream

```
const ivrMenu = (req, res, db) => {
  let body = "";
  req.on("data", (chunk) => {
    body += chunk;
  });
  req.on("end", function () {
    let enteredDigit = parseInt(querystring.parse(body).Digits);
    greetUser(enteredDigit, res, db);
  });
  res.type("text/xml");
};
```

Routing user to menu based on entered digit.

```
const greetUser = (digit, res, db) => {
  const twiml = new VoiceResponse();
  if (digit === 1) {
    handlePatientLogin(res, db, twiml);
  } else if (digit === 2) {
    handlePatientRegister(res, db, twiml);
  }
};
```


Chapter 6

TESTING AND RESULTS

The goal of testing is to find mistakes. Testing is the practise of attempting to find all possible flaws or weaknesses in a work product. It allows you to test the functionality of individual components, subassemblies, assemblies, and/or a whole product.

It is the process of testing software to ensure that it meets its requirements and meets user expectations, and that it does not fail in an unacceptable way. There are many different types of tests. Each test type is designed to meet a certain testing need.

6.1 Types of Tests

There are several forms of testing, including unit, integration, functional, and system testing, which are all used here.

6.1.1 Unit Testing

Unit testing entails creating test cases to ensure that the program's internal logic is working properly and that programme inputs result in valid outputs. Validation of all decision branches and internal code flow is required. The testing of individual software units of an application is known as unit testing. Prior to integration, unit testing is performed once each individual unit has been completed. This is an intrusive structural test that relies on prior knowledge of the structure. Unit tests are used to verify a specific business process, application, or system configuration at the component level. Unit tests guarantee that each individual path of a business process follows the published specifications and has clearly defined inputs and outputs.

6.1.2 Integration Testing

Integration tests are used to see if two or more software components can work together as a single application. Testing is event-driven, with a focus on the basic consequence of screens or fields. Integration tests prove that, while the components were individually satisfying, the combination of components is right and consistent, as demonstrated by successful unit testing.

Integration testing is a type of testing that focuses on uncovering issues that occur from the combination of components. After unit testing and before system testing, integration testing is the phase of software testing in which separate modules or groups of modules are combined and tested as a unit.

6.1.3 Functional Testing

Functional tests demonstrate that the functions being tested are available in accordance with business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : Valid input must be accepted in identified classes.

Invalid Input : Invalid input must be rejected in the identified classes.

Functions : Functions that have been identified must be used.

Output : It is necessary to exercise identified kinds of application outputs.

Systems/Procedures: It is necessary to use interface systems or procedures. Functional tests are organised and prepared around requirements, important functions, or unique test cases. In addition, systematic covering of business process flows is required; For testing, data fields, established processes, and subsequent processes must all be considered.

Additional tests are identified before functional testing is completed, and the effective value of present tests is determined.

6.1.4 System Testing

System testing guarantees that the complete integrated software system complies with the specifications. It checks a setup to guarantee that the results are known and predictable.

The configuration oriented system integration test is a form of system testing. Process descriptions and flows are used to test systems, with an emphasis on pre-driven process connections and integration points.

6.1.5 White Box Testing

White Box Testing is a type of software testing in which the software tester is familiar with the software's inner workings, structure, and language, or at the very least its purpose. It's used to test regions that aren't accessible using a black box level.

During a system level test, it can test paths within a unit, paths between units during integration, and paths between subsystems.

Techniques for designing white box tests include:

- Control flow testing
- Data flow testing
- Branch testing
- Path testing

6.1.6 Black Box Testing

Testing software without knowing the inner workings, structure, or language of the module being tested is known as black box testing. Like most other types of tests, black box tests must be written from a definitive source document, such as a specification or requirements document.

6.1.7 Acceptance Testing

Acceptance by the users Testing is an important aspect of any project, and it necessitates active engagement from the end user. It also guarantees that the system satisfies the functional specifications.

6.2 Test Cases

Table 6.2 Test cases

MODULE	GIVEN INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	REMARKS
Doctor Login	Correct phone number and password is given as inputs	Depending on the number and password, a user should be able to login	Login successful	Pass
Doctor Registration	Doctor Registration with details along with specialization	Doctor details are validated and stored	Registration successful	Pass

Patient registration	Basic details about patient are taken as input	User details are validated and stored	Registration successful and patient ID is generated	Pass
Patient login	Correct phone number and password is given as inputs	Depending on the number and password, a user should be able to login	Login successful	Pass
IVR login	Patient ID is taken as input	If the patient ID is present in the database, user should be able to login and make booking	Login successful	Pass
Prescription audio over IVR	Patient ID	Audio version of recent consultation prescription will be played if patient ID is found in database	Prescription audio of most recent consultation is played	Pass
Appointment Booking	Booking slot and doctor ID	Patient should be able to book appointment according to doctor and slot availability	Booking successful	Pass

6.3 Performance Evaluation

The system provides both smart and non-smart phone users, the ability to book appointment in hassle freeway. The IVR system is as accurate as the application built for smart phone users. On the other hand, it lets doctor generate a well formatted prescription by just voice commands thereby reducing the workload. There are several other parameters other than this, each entity can be evaluated as follows.

6.3.1 Evaluation of booking system

Booking system employs First come first serve (FCFS) approach, as it implemented by SQL transactions. This way the system is robust enough to manage user traffic effectively for all operations across the application and IVR system.

When a user makes a booking request, the database is updated by blocking the available slot and doctor. This way the system handles the booking system uniformly for all the users.

6.3.2 Evaluation of speech recognition module

The speech recognition module is over 95%, approximately. Most of words including medicine names were transcribed during testing phase. However, in few cases, more sophisticated medicine names and even common words weren't predicted efficiently. Hence, additional code had to be written to handle the edge cases.

6.3.3 Evaluation of prescription formatting module.

The doctor generates the prescription via the commands, but entities like, frequency, has to be formatted to a standard format as the doctor cannot spell these digits which would cause bottle neck issues in conversion of this commands to audio version. This challenge was resolved by employing a method which converts the text format of frequency like Morning, after and night to 1-1-1 format.

6.3.4 Evaluation of IVR System.

The IVR system, as discussed before handles user registration, login and fetches the consultation details effectively. However, in addition to this it also performs actions like sending SMS alerts to user about details of the registration and booking made. This helps the non-smartphone user to quickly lookup on their details like patient ID and booking details like slot, doctor phone number etc.

Chapter 7

SNAPSHOTS

A snapshot is a representation of a system's state at a specific time. It can refer to a physical duplicate of a system's state or a feature that certain systems provide.

The below attached are the snapshots of the project “Voice Based E-Prescription”, each of the modules are shown and explained in detail.

7.1 Home page

The first page upon opening the application is the home page, as shown in Figure 7.1. The Doctor or the patient can Login/register by clicking on either of the cards as shown in the figure 7.1



Figure 7.1 Home Page

7.2 Patient Login Page

If any user clicks on the patient login card then they land on this particular page where he/she should login in order to access the application as shown in the figure 7.2

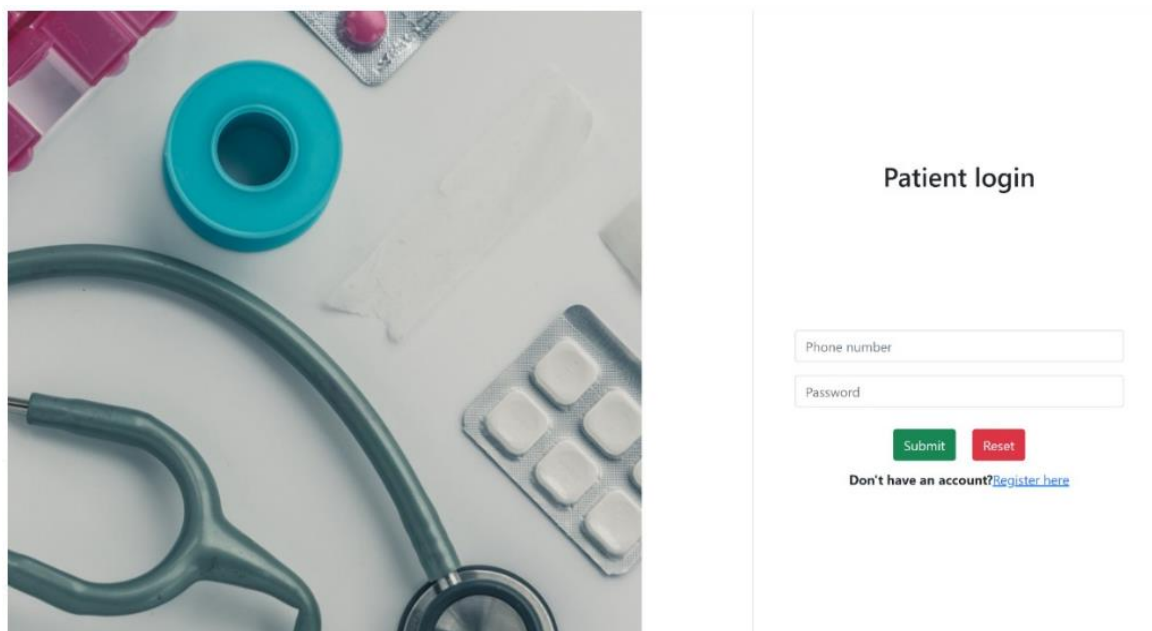
The image shows a patient login page. On the left, there is a photograph of medical supplies: a stethoscope, a blue pill bottle, a blister pack of white pills, and a pink pill container. On the right, the login form is displayed. It has a title 'Patient login' at the top. Below the title are two input fields: 'Phone number' and 'Password'. Underneath these fields are two buttons: a green 'Submit' button and a red 'Reset' button. At the bottom of the form, there is a link that says 'Don't have an account? Register here'.

Figure 7.2 Patient Login Page

7.3 Patient Register Page

If that particular user is a new user then he/she would have to register in order to use the application by filling in the required fields as shown in figure 7.3

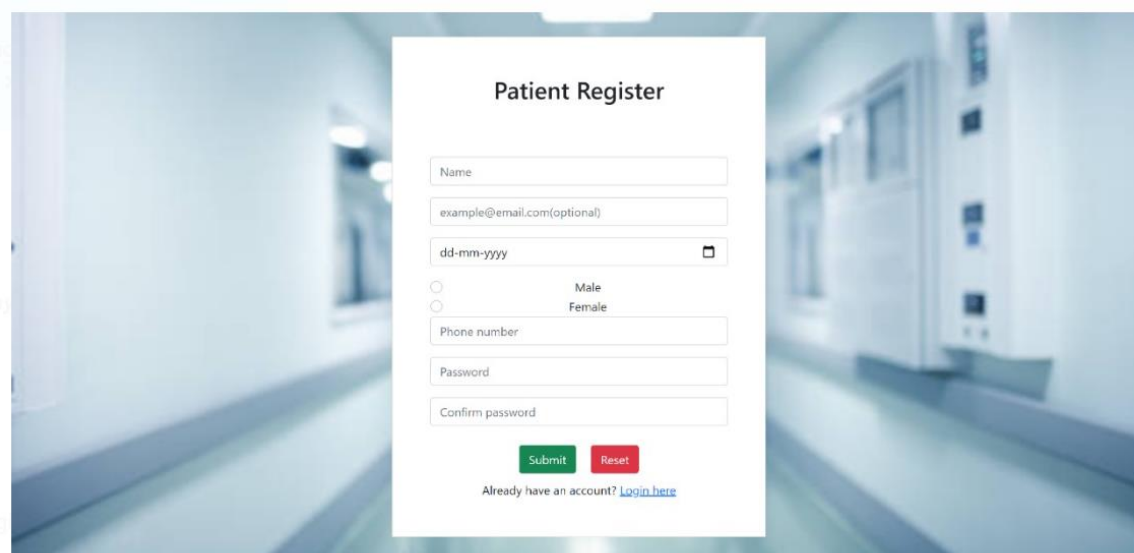
The image shows a patient registration page. The background is a blurred hospital hallway. The registration form is centered and has a title 'Patient Register'. It contains several input fields: 'Name', 'example@email.com(optional)', 'dd-mm-yyyy' with a calendar icon, two radio buttons for 'Male' and 'Female', 'Phone number', 'Password', and 'Confirm password'. At the bottom of the form are two buttons: a green 'Submit' button and a red 'Reset' button. Below the buttons is a link that says 'Already have an account? Login here'.

Figure 7.3 Patient Register page

7.4 Doctor Login Page

If any user clicks on the Doctor login card then they land on this particular page where he/she should login in order to access the application as shown in the figure 7.4

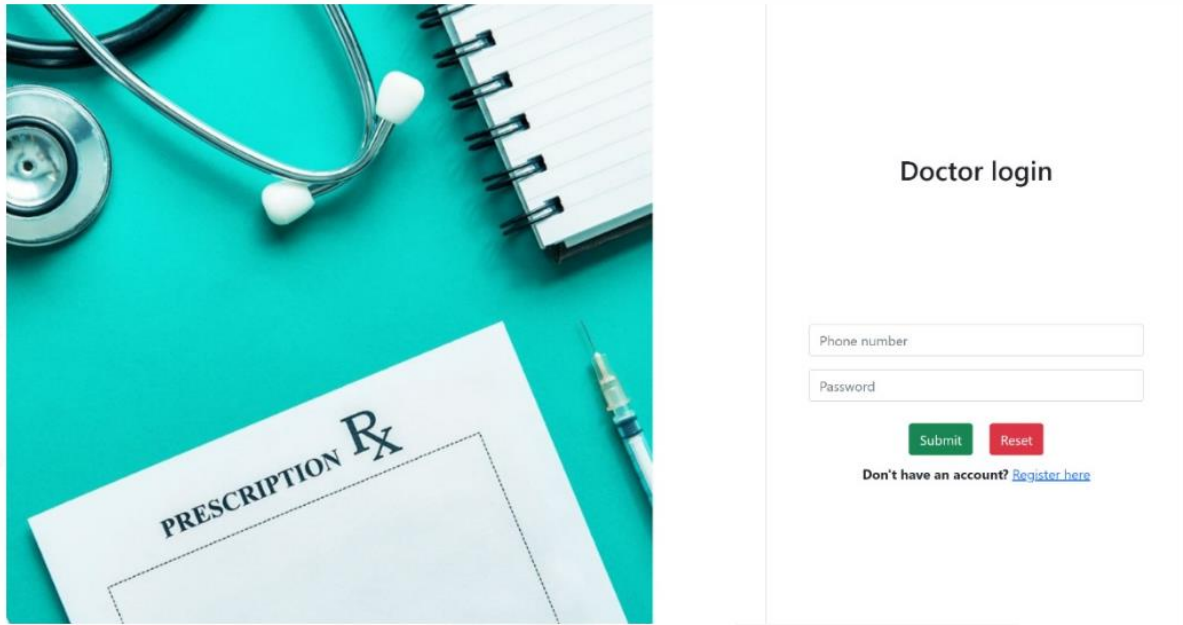


Figure 7.4 Doctor Login page

7.5 Doctor Register Page

If that particular user is a new user then he/she would have to register in order to use the application by filling in the required fields as shown in figure 7.5

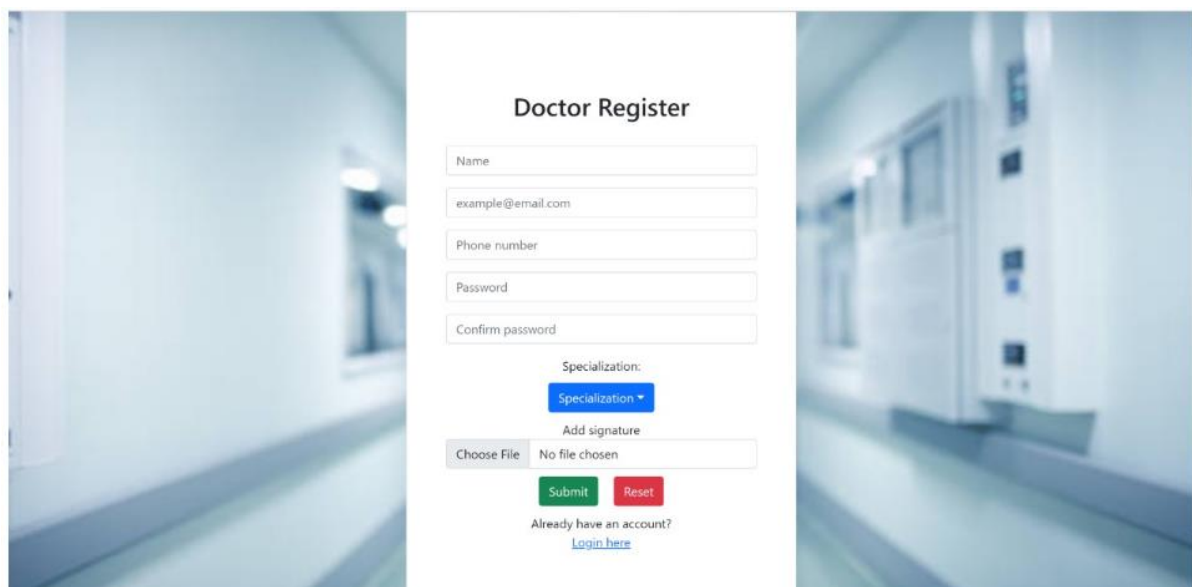


Figure 7.5 Doctor Register page

7.6 Patient Dashboard Page

After the patient logs in to his/her particular account, he can view his dashboard where he can listen to his previous audio prescription and also see the doctor details with whom he consulted previously and also download his E-prescription as a PDF as shown in the figure 7.6. If he is a new user and does not have any previous consultations, the dashboard will be empty

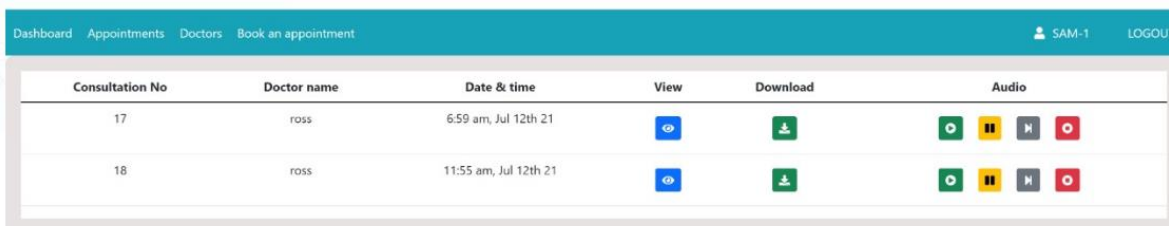


Figure 7.6 Patient Dashboard Page

7.7 Patient Appointment Page

After the patient logs in to his/her particular account, he can book an appointment with a particular doctor or a particular category and once it is booked the booking details will be shown in the appointment page with the details of the doctor, time and date and incase the patient wants to cancel that particular booking he can do so by clicking the cancel button given. The same is shown in figure 7.7



Figure 7.7 Patient Appointment Page

7.8 Slot booking based on doctor

After the patient logs in to his/her particular account, he can book an appointment either by choosing a particular doctor or by choosing a particular category. The below figure 7.8 shows slot booking based on doctor

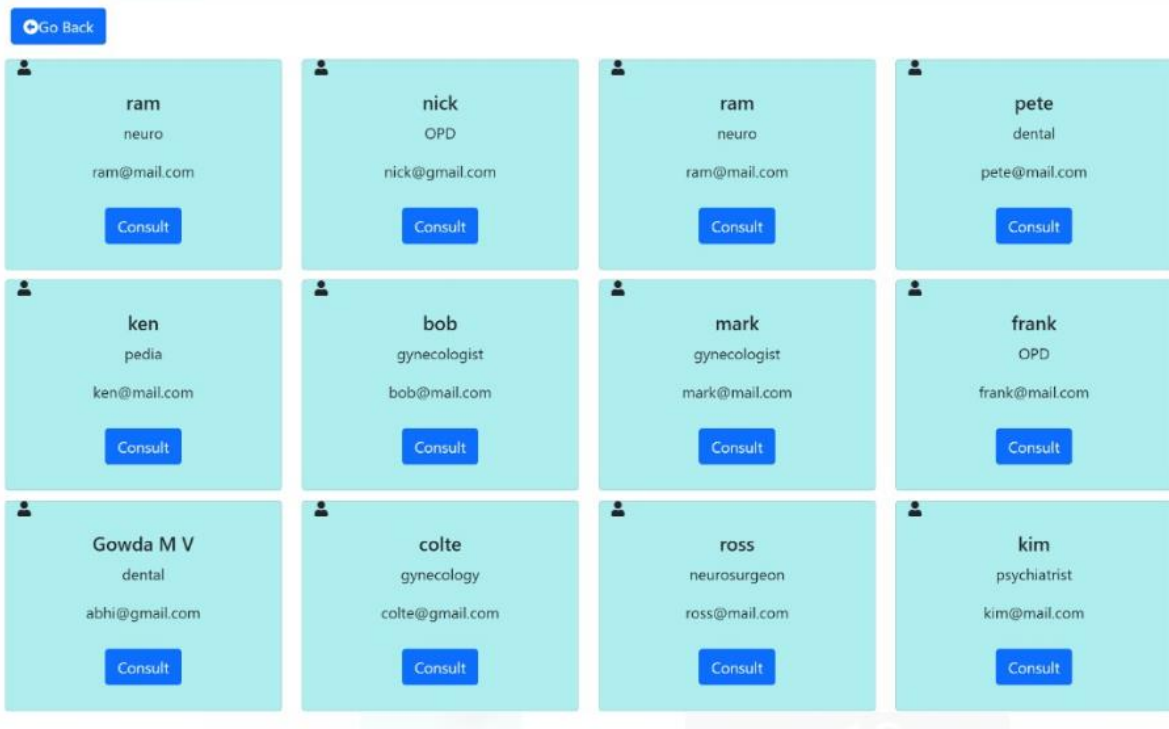


Figure 7.8 Slot booking based on doctor

7.9 Slot time selection Page

After the patient logs in to his/her particular account, he can book an appointment either by choosing a particular doctor like shown below and after he clicks consult the below page is seen and the user can book by choosing his/her convenient time



Figure 7.9 Slot time selection Page

7.10 Slot booking based on Category

After the patient logs in to his/her particular account, he can book an appointment either by choosing a particular doctor or by choosing a particular category. The below figure 7.10 shows slot booking based on category

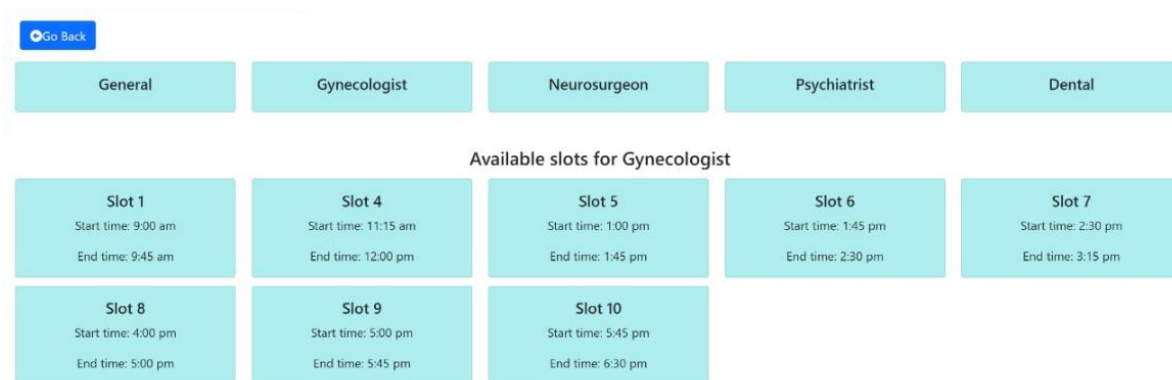


Figure 7.10 Slot Booking based on Category

7.11 Doctor Dashboard Page

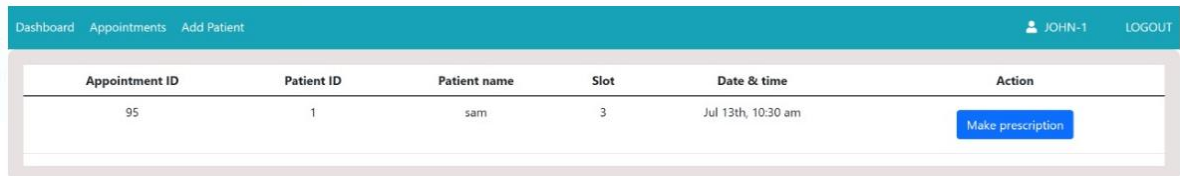
After the Doctor logs in to his/her particular account, he can see his previously consulted patients in the dashboard section as shown in the figure 7.11. He will be able to check all the details of that particular patient like name, time and date of consultation and also the prescriptions like shown.

Consultation No	Patient name	Date & time	View	Download	Audio
2	ram	3:17 am, Jul 5th 21			
3	Abhishek Gowda M V	2:14 am, Jul 7th 21			
4	Abhishek Gowda M V	3:40 am, Jul 7th 21			
12	Abhishek Gowda M V	4:49 am, Jul 7th 21			
13	Abhishek Gowda M V	4:58 am, Jul 7th 21			
16	Abhishek Gowda M V	6:51 am, Jul 12th 21			

Figure 7.11 Doctor Dashboard Page

7.12 Doctor Appointment Page

After the Doctor logs in to his/her particular account, he can view his appointments made by the patients for that particular day and after the conversation between the doctor and the patient is done, the doctor can visit this page and make the prescription on the basis of the consultation done, the same is shown in the figure 7.12

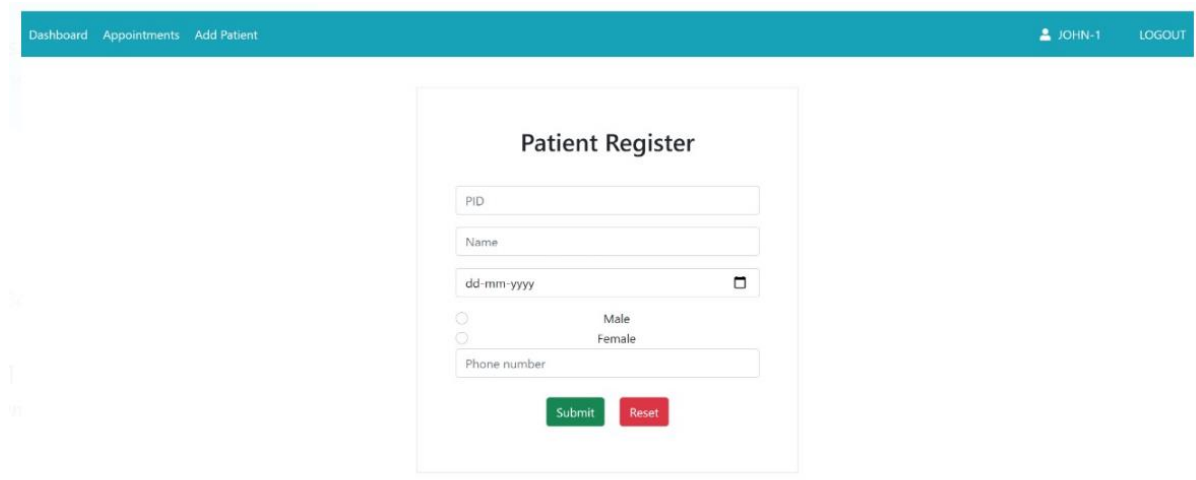


Appointment ID	Patient ID	Patient name	Slot	Date & time	Action
95	1	sam	3	Jul 13th, 10:30 am	Make prescription

Figure 7.12 Doctor Appointment Page

7.13 Add patient Page

If the patient is a non-smartphone user and he registers and books an appointment with a particular doctor using IVR, not all the details will be taken at the time of registration, hence the doctor would have to add certain details using the add patient feature given in his profile, as shown in the figure 7.13



Patient Register

PID

Name

dd-mm-yyyy

Male

Female

Phone number

Figure 7.13 Add Patient Page

7.14 Prescription Generation Page

After the doctor clicks on “make prescription” a page like below is shown where the doctor would give voice commands to the particular textboxes like diagnosis, medicines, frequency, duration of intake and also quantity. This can be done by just hovering over the text boxes and giving voice inputs. Finally, the doctor can give a advice if required and add his signature and save the prescription, the same is shown in the below figure 7.14

The screenshot displays the 'Prescription Generation Page' interface. At the top left, it identifies the doctor as 'Dr. john', an 'Orthopedic' with contact information 'john@email.com' and '9123132122'. The patient's details are listed as 'Name: sam', 'AGE: 3', 'Gender: M', and 'PID: 1'. The date is '13 Jul, 2021'. The MediPro logo is in the top right. A table with columns 'Rx', 'Frequency', 'Duration', and 'Quantity' is present. Below the table are input fields for 'Diagnosis: Ex: Fever', 'Medicine: Ex: paracetamol', 'Frequency: MN-AF-NT', 'Duration: 'X' day(s)', and 'Quantity: dosage'. There are buttons for 'Add advice', 'Add signature', and 'Save'.

Rx	Frequency	Duration	Quantity
----	-----------	----------	----------

Diagnosis:

Medicine:

Frequency:

Duration:

Quantity:

Figure 7.14 Prescription Generation Page

7.15 Generated Prescription Page

After the doctor creates a prescription, he would have to save it and once it is saved the same can be accessed by the doctor and the patient in their dashboards and the final prescription would look like the below figure 7.15

Dr. admin
General
admin@mail.com
123

Name: sam
AGE: 3
Gender: M
PID: 1

Diagnosis: Fever and cough

Date: 13 Jul, 2021

	Rx	Frequency	Duration	Quantity
1	Crocin	1-0-1 (after food)	5 days	2 mg
2	Paracetamol	0-1-0 (after food)	2 days	4 mg
3	Dolo 650	0-1-0 (after food)	2 days	1 mg

ADVICE TO PATIENT:
Take rest for 2 days


Sign: 

Figure 7.15 Generated Prescription Page

Chapter 8

CONCLUSION

Contactless consulting aids in the reduction of human interaction, resulting in a safer encounter during the current covid situation which is one of the main objective of our application .Prescriptions can be sent to persons who have or don't have a smartphone. Our application also assists people with hearing aid and also reduces the amount of paper used. Access to previous reports/prescriptions is also easy.

The application forms prescription with voice commands by the doctor and the doctor can save the generated prescriptions so the patients can view them through the application. Patients can book appointments with a particular doctor by selecting the doctor along with the available slot for that doctor. Patient can also listen to the audio of the prescription on the application. Patients with basic sets are able to register themselves through the IVR service of our application and the Pid will be sent as a message and the audio version of the prescription can be accessed by the non-smartphone users.

Chapter 9

FUTURE ENHANCEMENT

- Improve look and feel of the web application, so that it increases user usability and experience.
- Further regional languages can be added for the IVR service so that people can select the language of the service they are familiar with.
- The proposed web application can be extended to a mobile application for better reachability.
- Improvement on the IVR could be done by routing the call to a virtual help desk. Include chatbot service for booking lab tests and scans that are prescribed by the doctor.
- Include video consultation service in the application itself which makes it more efficient and interactive.
- Include the payment module for the doctor's consultation. Include the module to order the medicines and other medical equipments online through our application.

REFERENCES

- [1] Paper on “Paperless prescription using voice” by Mohammed Abrar Ahmed and Tirumala Shravika. October 2020.
- [2]“Speech to text and text to speech recognition systems”-Areview Ayushi Trivedi,Navya Pant, Pinal Shah,Simran Sonik and Supriya Agrawal Dept.of Computer Science, NMIMS University,Mumbai,India. Corresponding Author:Navya Pant, Published on Mar-apr 2018.
- [3] Suman K. Saksamudre, P.P. Shrishrimal, R.R. Deshmukh, A Review on Different Approaches for Speech Recognition System, International Journal of Computer Applications (0975 8887) Volume 115 No. 22, April 2015.
- [4]”An Electronic Prescription System powered by Speech Recognition, Natural Language Processing and Blockchain Technology”-Jitendra Mahatpure, Dr Mahesh Motwani, Dr Piyush Kumar Shukla,INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 8, ISSUE 08, AUGUST 2019.
- [5] Suhas R. Mache, Manasi R. Baheti, C. Namrata Mahender, Review on Text-To-Speech Synthesizer, International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 8, August 2015
- [6] Randhir Jagannath Patil1, Dr. S.A.Pardeshi Voice Based Medicine Prescription,2014
- [7] [Removing stop words with NLTK in Python](#) - GFG (Article)
- [8]”[Modi govt wants to define e-prescription format](#)” - ThePrint (Article)

CERTIFICATES

WAYANAMAC EDUCATION TRUST
DON BOSCO INSTITUTE OF TECHNOLOGY
(Affiliated to VTU, Belagavi, Approved by AICTE New Delhi & Recognised by Govt. of Karnataka, Accredited by NBA & NAAC)
Kumbalagodu, Mysore Road, Bangalore - 560074

Approved by
AICTE
New Delhi

Affiliated to
VTU
Belagavi

Recognised by the
Government of
Karnataka

Accredited by
NAAC
Bangalore

Accredited by
NBA
New Delhi

CERTIFICATE

This is to certify that

Abhishek Gowda M V of **K S institute of technology**

Voice based e-prescription

has successfully presented the paper titled “ ”

in the 4th National Conference on “Emerging Trends in Information Technology – NCETIT-2021” organized by Department of Information Science and Engineering, Don Bosco Institute of Technology , Bengaluru held on 19.07.2021 and 20.07.2021.

Dr. Manjula. V
Convener, IS&E
DBIT, Bengaluru

Mrs. Gowramma. B. S
HOD, IS&E
DBIT, Bengaluru

Dr. Hemadri Naidu.T
Principal
DBIT, Bengaluru

WAYANAMAC EDUCATION TRUST
DON BOSCO INSTITUTE OF TECHNOLOGY
(Affiliated to VTU, Belagavi, Approved by AICTE New Delhi & Recognised by Govt. of Karnataka, Accredited by NBA & NAAC)
Kumbalagodu, Mysore Road, Bangalore - 560074

Approved by
AICTE
New Delhi

Affiliated to
VTU
Belagavi

Recognised by the
Government of
Karnataka

Accredited by
NAAC
Bangalore

Accredited by
NBA
New Delhi

CERTIFICATE

This is to certify that

Ashish Amar of **KS Institute of Technology**

Voice Based E Prescription

has successfully presented the paper titled “ ”

in the 4th National Conference on “Emerging Trends in Information Technology – NCETIT-2021” organized by Department of Information Science and Engineering, Don Bosco Institute of Technology , Bengaluru held on 19.07.2021 and 20.07.2021.

Dr. Manjula. V
Convener, IS&E
DBIT, Bengaluru

Mrs. Gowramma. B. S
HOD, IS&E
DBIT, Bengaluru

Dr. Hemadri Naidu.T
Principal
DBIT, Bengaluru

CERTIFICATES



Voice Based E Prescription

by Ashish K Amar

Submission date: 23-Jul-2021 11:16AM (UTC+0530)

Submission ID: 1623011265

File name: Voice_Based_E_Prescription.pdf (1.31M)

Word count: 8351

Character count: 43099

Voice Based E Prescription

ORIGINALITY REPORT

20%

SIMILARITY INDEX

10%

INTERNET SOURCES

2%

PUBLICATIONS

18%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Gitam University Student Paper	4%
2	Submitted to Visvesvaraya Technological University, Belagavi Student Paper	2%
3	Submitted to Higher Education Commission Pakistan Student Paper	1%
4	Submitted to United Colleges Group - UCG Student Paper	1%
5	www.slideshare.net Internet Source	1%
6	Submitted to Berea College Student Paper	1%
7	docplayer.net Internet Source	1%
8	Submitted to Icon College of Technology and Management Student Paper	<1%
