A Project Report

On

Automation of Adverse Drug Helpline

Submitted in partial fulfilment of the requirement of

University of Mumbai

For the Degree of

Bachelor of Engineering

in

COMPUTER ENGINEERING

Submitted by

Ryan Noronha
Ankesh Gaikwad

Supervised by

Ms. Dakshayan

&

Mr. K.V. Subrahmanyam

Department of Computer Engineering

Fr. Conceicao Rodrigues Institute of Technology
Sector 9A, Vashi, Navi Mumbai - 400703

UNIVERSITY OF MUMBAI

2019-2020

APPROVAL SHEET

This is to certify that the project entitled

“Automation of Adverse Drug Helpline”

Submitted by

Ryan Noronha 101636

Ankesh Gaikwad 101668

Supervisors : ____________________

Project Coordinator : ____________________

Examiners : 1. ____________________

2. ____________________

Head of Department : ____________________

Date :

Place :

Declaration

We declare that this written submission for B.E. Declaration entitled "**Automation of Adverse Drug Helpline**" represent our ideas in our own words and where others' ideas or words have been included. We have adequately cited and referenced the original sources. We also declared that we have adhere to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause for disciplinary action by institute and also evoke penal action from the sources which have thus not been properly cited or from whom paper permission have not been taken when needed.

Project Group Members:

1. Ryan Noronha, 101636

2. Ankesh Gaikwad, 101636

Abstract

Medicines play a very crucial role in maintaining day-to-day healthy functioning of the body. Since the invention of modern medicines, countless lives have been saved. Every person young or old is consuming or will consume medicine in some or the other form. However, every medicine manufactured comes with a set of undesirable side effects. In some cases these side effects can have adverse effects on the person. In fact, deaths due to Adverse Drug Reactions (ADRs) have been recognised as one of the leading causes of deaths in the USA.

The number of reported ADRs in India is relatively low even after the the introduction of the Pharmacovigilance Programme of India (PvPI). To enable more patients report such ADRs, we propose a 24x7 voice based helpline. The helpline will help the affected person or the reporter of the patient report an ADR; thus making the reporting of ADRs much more easily accessible to the general public.

Keywords: Automatic Speech Recognition (ASR), Speech to Text, Adverse Drug Reaction (ADR), Pharmacovigilance Programme of India (PvPI)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The World Health Organisation (WHO) defines an Adverse Drug Reaction as: *A response to a drug which is noxious and unintended, and which occurs at doses normally used in man for the prophylaxis, diagnosis, or therapy of disease, or for the modifications of physiological function* [WHO, 1972]. ADRs have been recognised as a leading cause of deaths [4]. Spontaneous reporting of ADRs helps identify rare and serious ADRs [5]. Spontaneous reporting is also a low cost operation and can reach a much wider population [5].

A study conducted in 2014 shows that in India, only 24 out of 23,975 reports (0.016%) were submitted by non health care professionals. The paper suggests that this low percentage could be because of lack of knowledge about where and how to report the ADR, financial incentives, ignorance, lack of time among many other factors. There has been however, a growth in the number of reported ADRs over the past years after the circulation of PvPI newsletters.[4]

1.2 Motivation

With the number of medicines being consumed on the rise, it is natural that the number of people of that would have an ADR will increase in the coming years. A study conducted in 2019 in Croatia [6] shows that there were 126 reported ADRs for non-analgesic over-the-counter (OTC) drugs in 2017 as compared to 88 in 2016. The same study also shows the percentage of ADRs that were caused due to OTC drugs. We present the same in figure 1.1.

The current method in India for voluntary reporting of ADRs involves filling up a form. This method is tedious as it involves people filling up the form and going to the ADR Monitoring Center (AMC) or the National Coordination Center (NCC) to submit it. Additionally, a helpline is also available that operates from 9:00 am to 5:30 pm on weekdays. This helpline uses operators that fill the form for the user and submit it.[2]

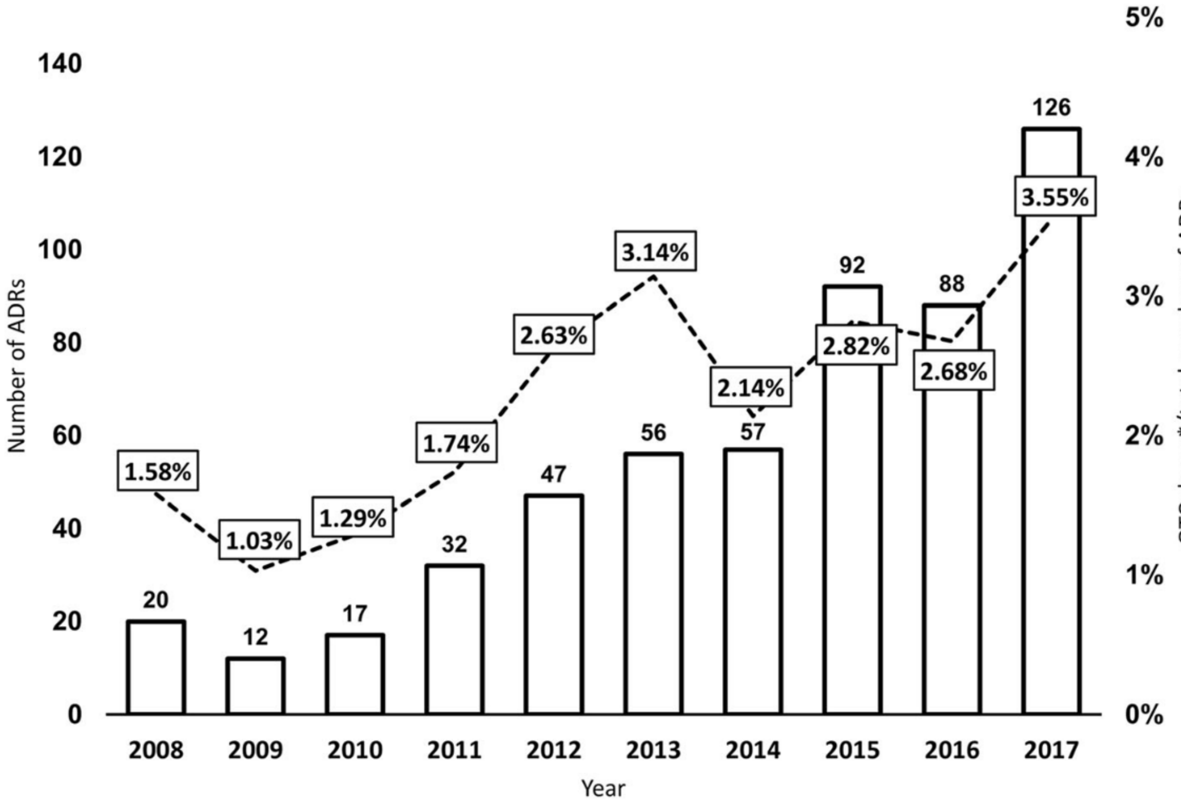


Figure 1.1: Number of ADRs caused due to OTC drugs and the percentage of reported ADRs that were due to OTC drugs

1.3 Mission

"Safeguard the health of the world population by ensuring that the benefits of use of medicine outweigh the list associated with its use."

1.4 Aim and Objectives

The objectives of the system are:

- To encourage reporting culture amongst healthcare professionals (H-CPs) and non HCPs.
- To identify and analyze new signals from the report cases.
- To improve reliability and availability of the process of reporting an ADR.
- To analyze the benefit-risk ratio of market medication.
- To generate evidence based information on safety of medicines.

1.5 Report Outline

This report has been structured as follows:

- Chapter 2 presents how the current system for reporting a suspected ADR works, the pain areas in the current system as well as some background of the technologies used in our project. Additionally, we present some of the work done pertaining to the project in chapter 2.
- In chapter 3, we present the problem statement, the scope of the project, and the proposed system.
- In chapter 4, we present the design of the system using standard UML diagrams, .
- In chapter 5, we present the results of the system using screenshots and some code samples. We also present the results of various testing techniques.
- With chapter 6, we conclude the project report and present the future scope of the system.

Chapter 2

Study Of Automation of Adverse Drug Helpline System

2.1 About the Technique

2.1.1 The Current Process of Reporting an ADR

The current process of reporting an ADR involves filling the form shown in figure 2.1. The same form is applicable to all types of drugs including herbal remedies, prescription drugs, OTC drugs, medical devices, etc. This form is then sent to an AMC or directly to the NCC by the reporter either by post or by email. Additionally, the reporter can also use the helpline that is operational from 9:00 am to 5:30 pm on weekdays to report their ADR(s).[2]

The entire process of reporting an ADR can be summarised using the figure 2.2 [1].

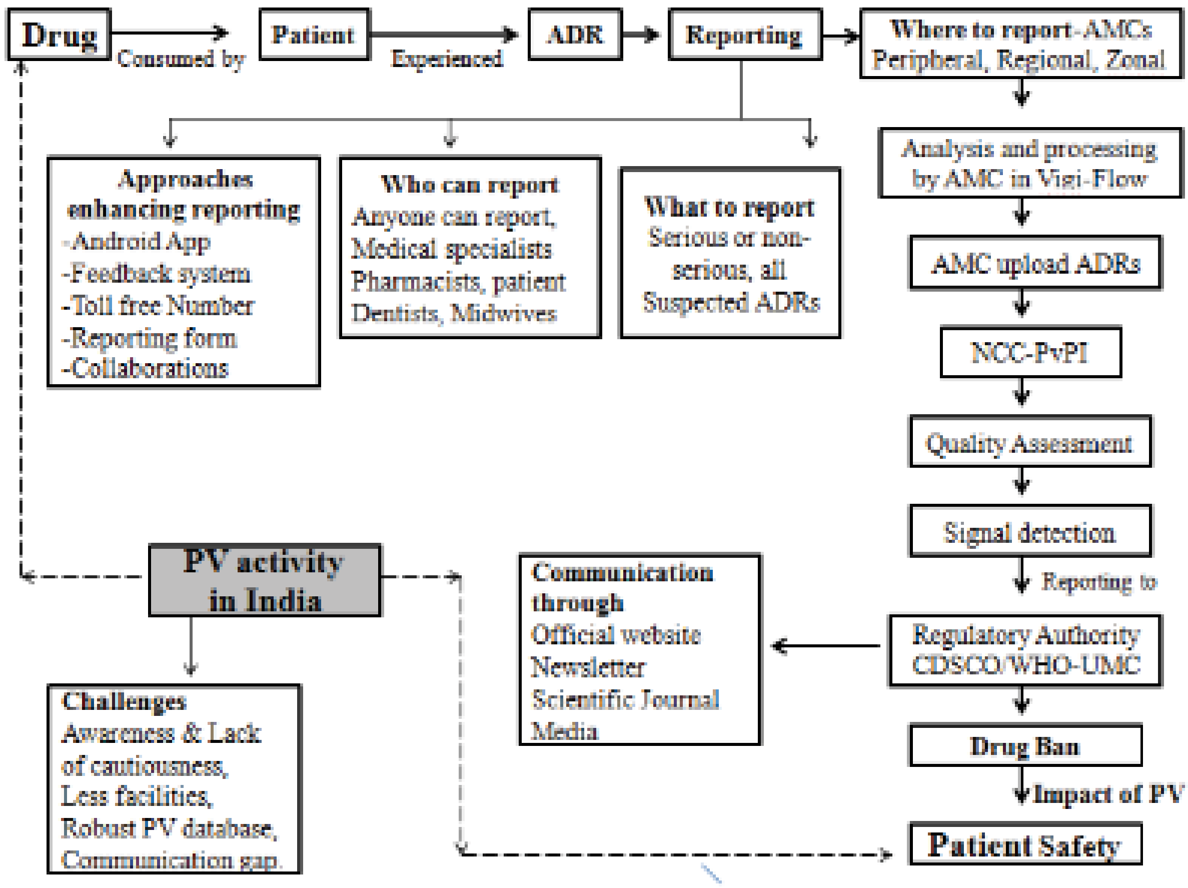
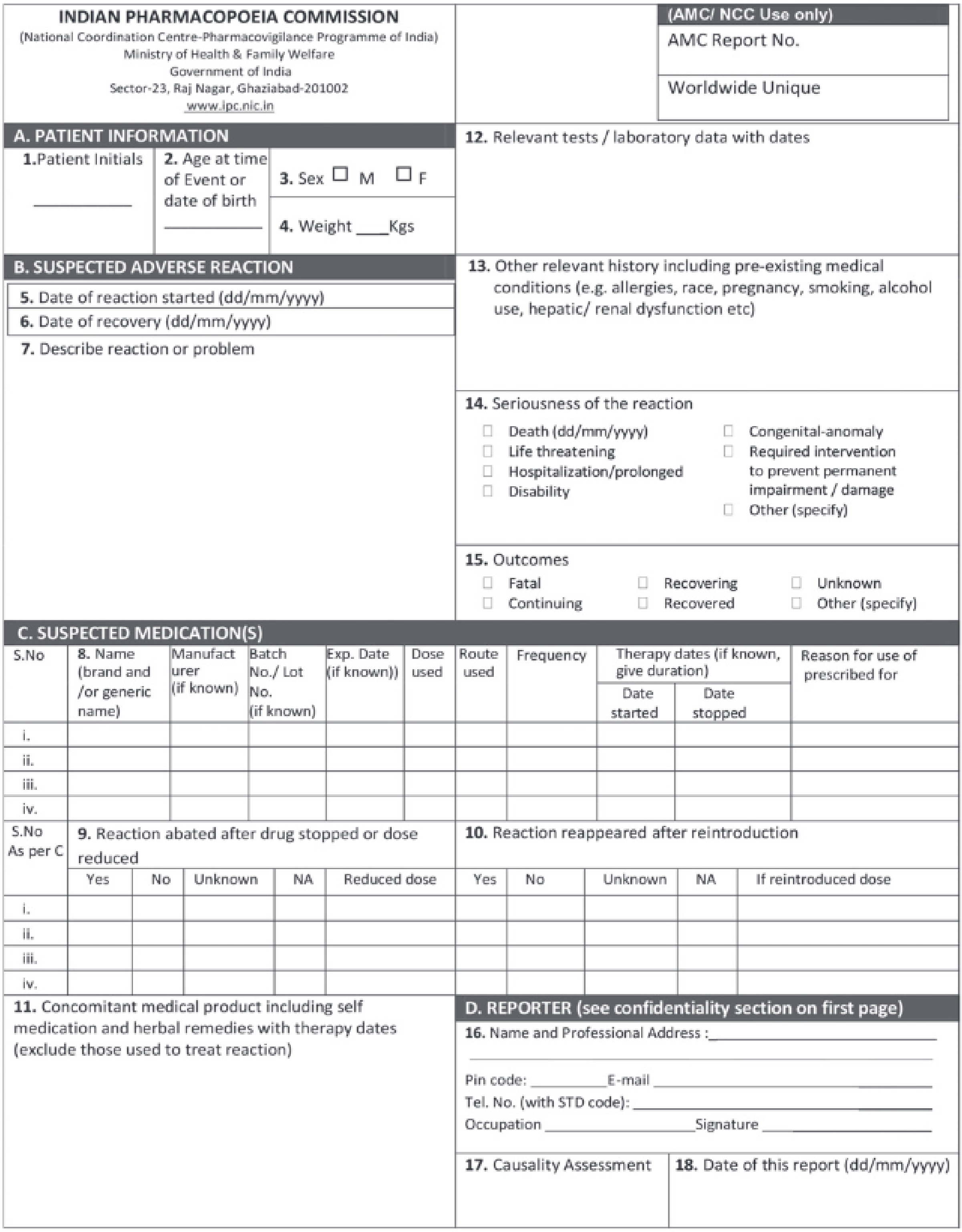


Figure 2.2: Summary of ADR report flow[1]

2.1.2 Pain Areas in the Reporting of ADRs

The most clearly visible pain area currently in the reporting of ADRs is the filling of the form. The form involves filling up multiple fields with great detail and therefore is tedious to perform. If we take a look at the



INDIAN PHARMACOPOEIA COMMISSION
(National Coordination Centre-Pharmacovigilance Programme of India)
Ministry of Health & Family Welfare
Government of India
Sector-23, Raj Nagar, Ghaziabad-201002
www.ipc.nic.in

(AMC/ NCC Use only)
AMC Report No.
Worldwide Unique

A. PATIENT INFORMATION
1.Patient Initials ________
2. Age at time of Event or date of birth ________
3. Sex ☐ M ☐ F
4. Weight ____Kgs

12. Relevant tests / laboratory data with dates

B. SUSPECTED ADVERSE REACTION
5. Date of reaction started (dd/mm/yyyy)
6. Date of recovery (dd/mm/yyyy)
7. Describe reaction or problem

13. Other relevant history including pre-existing medical conditions (e.g. allergies, race, pregnancy, smoking, alcohol use, hepatic/ renal dysfunction etc)

14. Seriousness of the reaction
☐ Death (dd/mm/yyyy)
☐ Life threatening
☐ Hospitalization/prolonged
☐ Disability
☐ Congenital-anomaly
☐ Required intervention to prevent permanent impairment / damage
☐ Other (specify)

15. Outcomes
☐ Fatal ☐ Recovering ☐ Unknown
☐ Continuing ☐ Recovered ☐ Other (specify)

C. SUSPECTED MEDICATION(S)

S.No	8. Name (brand and /or generic name)	Manufacturer (if known)	Batch No./ Lot No. (if known)	Exp. Date (if known))	Dose used	Route used	Frequency	Therapy dates (if known, give duration)		Reason for use of prescribed for
								Date started	Date stopped	
i.										
ii.										
iii.										
iv.										

S.No As per C	9. Reaction abated after drug stopped or dose reduced					10. Reaction reappeared after reintroduction				
	Yes	No	Unknown	NA	Reduced dose	Yes	No	Unknown	NA	If reintroduced dose
i.										
ii.										
iii.										
iv.										

11. Concomitant medical product including self medication and herbal remedies with therapy dates (exclude those used to treat reaction)

D. REPORTER (see confidentiality section on first page)
16. Name and Professional Address :________
Pin code: ________E-mail ________
Tel. No. (with STD code): ________
Occupation ________Signature ________

17. Causality Assessment
18. Date of this report (dd/mm/yyyy)

Figure 2.1: Specimen copy of ADR reporting form[2]

other method of reporting ADRs i.e. the helpline, it involves operators manually filling up the form for the reporter. In this case, it is possible that the operator may forget to ask the reporter the details of a specific field. If this happens, it becomes difficult to follow up with the reporter and get the missing details of the event.

2.2 Available Techniques

2.2.1 An IVR System

IVR stands for Interactive Voice Response. IVR is a technology that automates routine customer service interactions by allowing callers to interact using the dial pad[7]. An example of an IVR application is an automated attendant or voice menu: callers are presented with a recorded menu and respond by selecting a digit or, in some cases, by entering an extension number. The automated attendant eliminates the need for a live operator to handle the call and makes it easy for customers to reach the right agent[7].

The key idea is to automate a routine, repetitive task that would otherwise require the time and effort of an employee[7]. The savings potential gives IVR solutions a very rapid return on investment (ROI), as one server can potentially eliminate multiple live agents.

2.2.2 Asterisk as an IVR System

Asterisk is a private branch exchange (PBX) software that is used in IP PBX and for VoIP purposes[8]. Asterisk includes a wealth of functions that make it a powerful IVR platform: audio playback and recording, digit collection, database, and many more. IVR applications can be built using the Dialplan language or through the Asterisk Gateway Interface (AGI) or more recently Enhanced AGI (EAGI) and can integrate with virtually any external system.

Asterisk is an open source software and hence can be used for free. Asterisk can be used on commodity hardware and can interface with the world using PSTN[8]. This results in a low cost to deploy and maintain an application. As Asterisk is open source, users can modify the source code to suit their needs as opposed to proprietary solutions, that require the vendor to add functionality.

The architecture of asterisk is shown in figure 2.3.

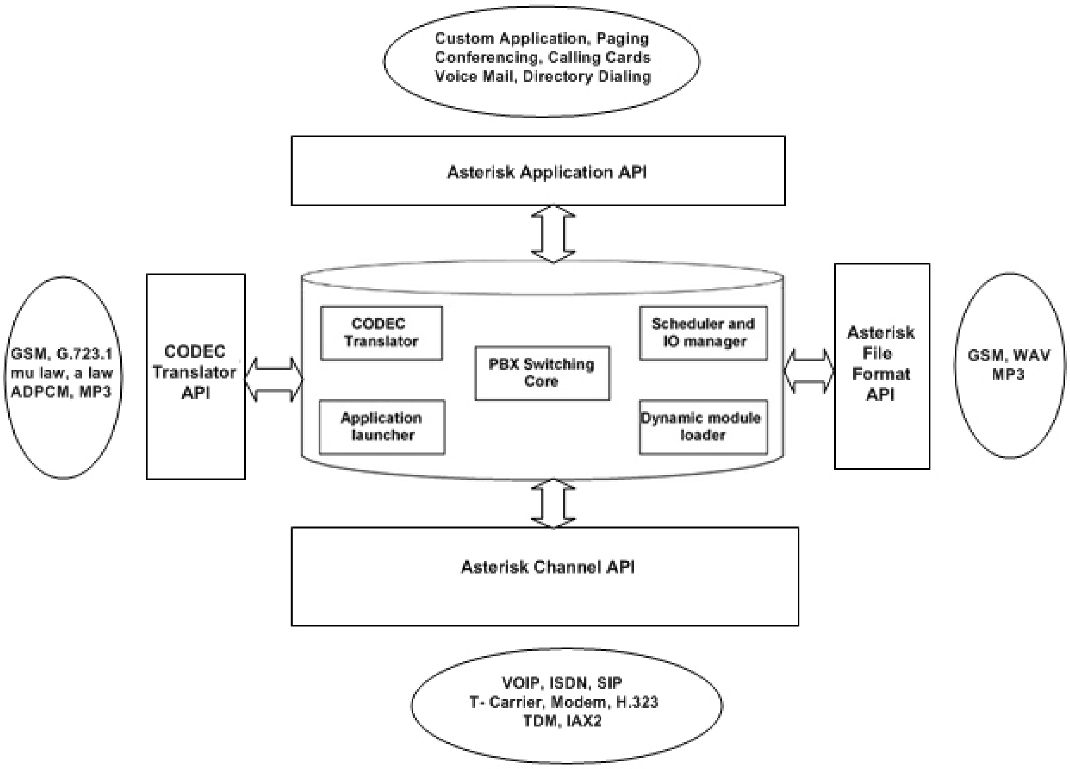


Figure 2.3: Architecture of Asterisk

2.2.3 Speech Recognition using Google Speech-to-Text API

A speech-to-text request can be of 3 types, namely synchronous, asynchronous, and streaming[9]. Each of these types of requests have different uses and characteristics.

A Speech-to-Text API synchronous recognition request (pipeline shown in figure 2.4) is the simplest method for performing recognition on speech audio data. Speech-to-Text can process up to 1 minute of speech audio data sent in a synchronous request[9]. After Speech-to-Text processes and recognizes all of the audio, it returns a response.

A synchronous request is blocking, meaning that Speech-to-Text must return a response before processing the next request.

2.2.4 Django

Django is a high-level Python web framework that allows for development of maintainable and secure websites. Django takes care of all the common web development hassles and allows us to focus on the tasks that are specific to the project.

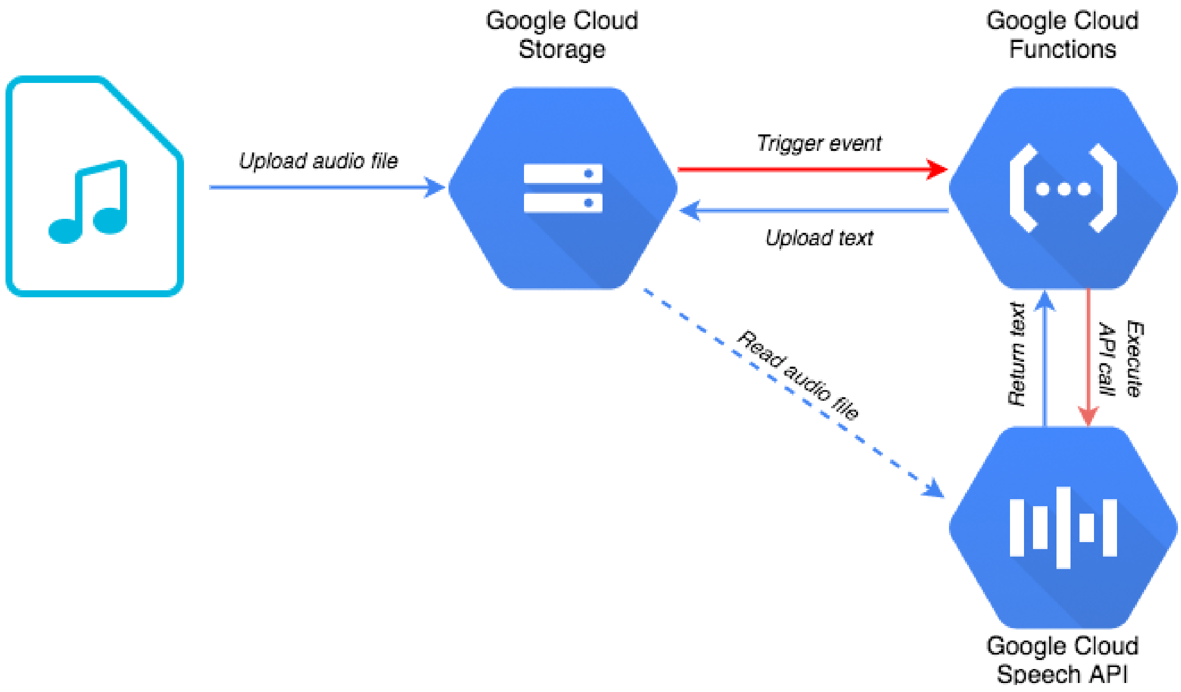


Figure 2.4: Serverless Audio Transcription Pipeline

Django is an open-source framework and has a large library of in-built and third party applications that reduces the amount of code that has to be written.

Why we chose Django?:

- Complete
- Versatile
- Secure
- Scalable
- Maintainable
- Portable

In a traditional data-driven website, a web application waits for HTTP requests from the web browser (or other client). When a request is received the application works out what is needed based on the URL and possibly information in POST data or GET data. Depending on what is required it may then read or write information from a database or perform other tasks required to satisfy the request. The application will then return a response to the web browser, often dynamically creating an HTML page for the browser to display by inserting the retrieved data into placeholders in an HTML template.

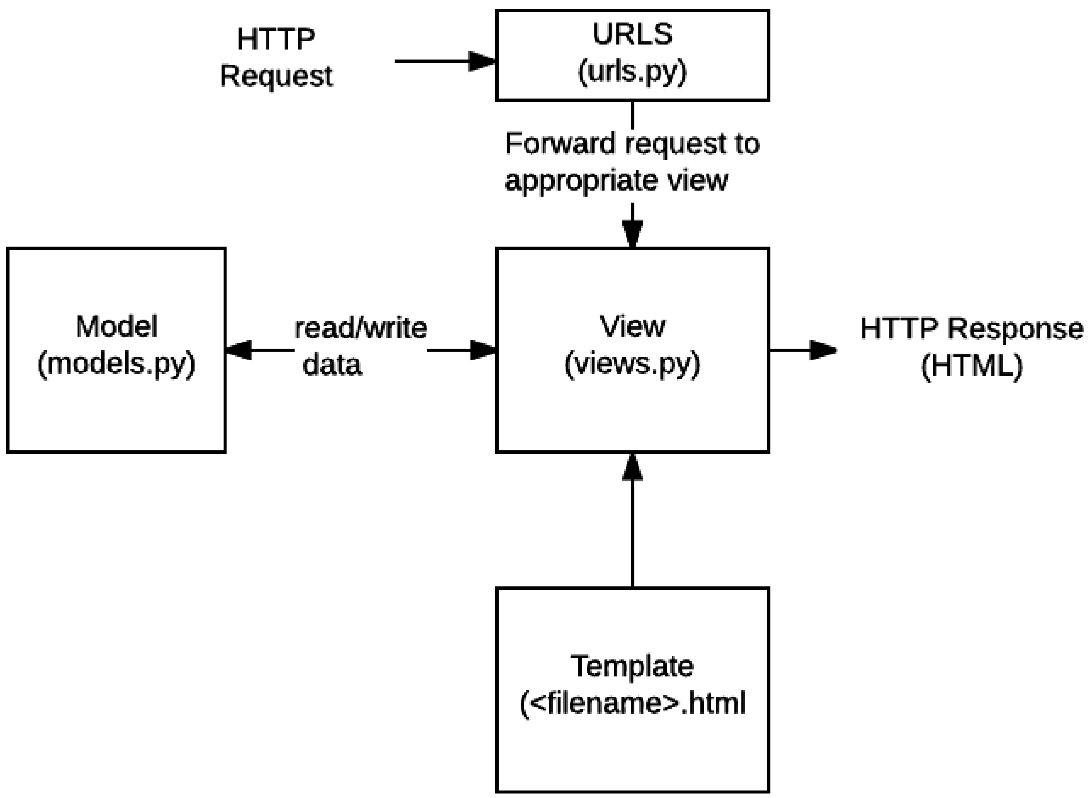


Figure 2.5: File Handling in Django

Django web applications typically group the code that handles each of these steps into separate files:

2.2.4.1 Views

A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via models, and delegate the formatting of the response to templates.

2.2.4.2 URLs

While it is possible to process requests from every single URL via a single function, it is much more maintainable to write a separate view function to handle each resource. A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL. The URL mapper can also match particular patterns of strings or digits that appear in a URL and pass these to a view function as data.

2.2.4.3 Models

Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.

2.2.4.4 Templates

A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A view can dynamically create an HTML page using an HTML template, populating it with data from a model. A template can be used to define the structure of any type of file; it doesn't have to be HTML!

2.3 Related Work

In this section we present some of the research and related work done in the project domain.

2.3.1 A Review on Design and Implementation of IVR System Using Asterisk

Borkar et al.[10] have implemented a standard IVR system using Asterisk on a cloud server that uses CentOS 5. They have used Asterisk to configure a system that comprises of an IVR and voicemail.

This paper also explains the basic terminologies used in the telephony industry and the major files that are used in the Asterisk environment such as `extensions.conf`, `sip.conf`, and `voicemail.conf`

2.3.2 VoIP Implementation Using Asterisk PBX

Rahman et al.[11] have compared the performance of the various codecs used in Asterisk. They have compared the performance of GSM and G711 with and without transcoding on separate machines with the same software configuration but different hardware configurations. They have used SIPp as the stress testing software to plot various graphs of CPU load versus the number of calls.

2.3.3 Speech based dialog query system over Asterisk PBX server

Goel et al.[12] have used Asterisk to recognise digits spoken over the phone. They have configured Asterisk on a Fedora Core 5 system and have used a

Hidden Markov Model (HMM) to perform speech recognition. Their HMM model was trained using HTK Toolkit which allows for building, training, testing, and manipulating HMMs for speech recognition tasks [13].

2.3.4 Customized IVR Implementation Using VoiceXML on SIP (VoIP) Communication Platform

Appari[14] has presented a system wherein they have used Asterisk and VoiceXML (VXML) to implement an IVR. VoiceXML allows for speech synthesis, recording audio, recording DTMFs, recognising audio among various other features[15]. The idea is that after a call has been answered, a simple script transfers control from Asterisk to the VoiceXML browser that handles the call as it has been programmed to.

2.3.5 Consumer reporting of adverse drug reactions: A current perspective

Mukherjee et al.[16] speak about the Pharmacovigilance Programme of India (PvPI) which was initiated in 2010. Under this programme, a National Coordination Center (NCC) has been established in Delhi. This NCC deals with collating, and analysing the data of ADRs and uses the inferences to communicate health risks to healthcare professionals and to the public.

Chapter 3

Proposed System

3.1 Problem Statement

We frame the problem statement as follows:

In today's world, we all take medicines for many reasons. They have become an integral part of our daily lives. Often, these drugs have bad side effects like diarrhoea, nausea, rashes, etc. At present, the method to report such a reaction involves filling a form or optionally using a helpline that operates on weekdays between 9 am and 5 pm. This makes it difficult for the patient to report the reaction. To solve this problem, we propose an automated helpline that can operate 24 hours a day, 7 days a week, so that patients can easily report a reaction.

3.2 Scope

The following points lie in the scope of the project:

1. Reporting of only adverse events (AEs).
2. Language supported is English.
3. No medical assistance will be provided on call.

3.3 Proposed System

In order to facilitate a 24/7 reporting system, we propose the use of an automated helpline that will allow the user to report an ADR at any time of the day, at their convenience.

The patient or a reporter of the patient can call the helpline and describe the event based on the instructions provided by the helpline. These instructions will be designed in compliance with the form shown in figure 2.1. This audio recording will then be used to fill in the respective details of the event in the ADR reporting form as prescribed by the government (show in figure 2.1). This form will be sent to a health centre (hospital, clinic, pathology lab, etc.) that is nearest to the patient's location for further processing.

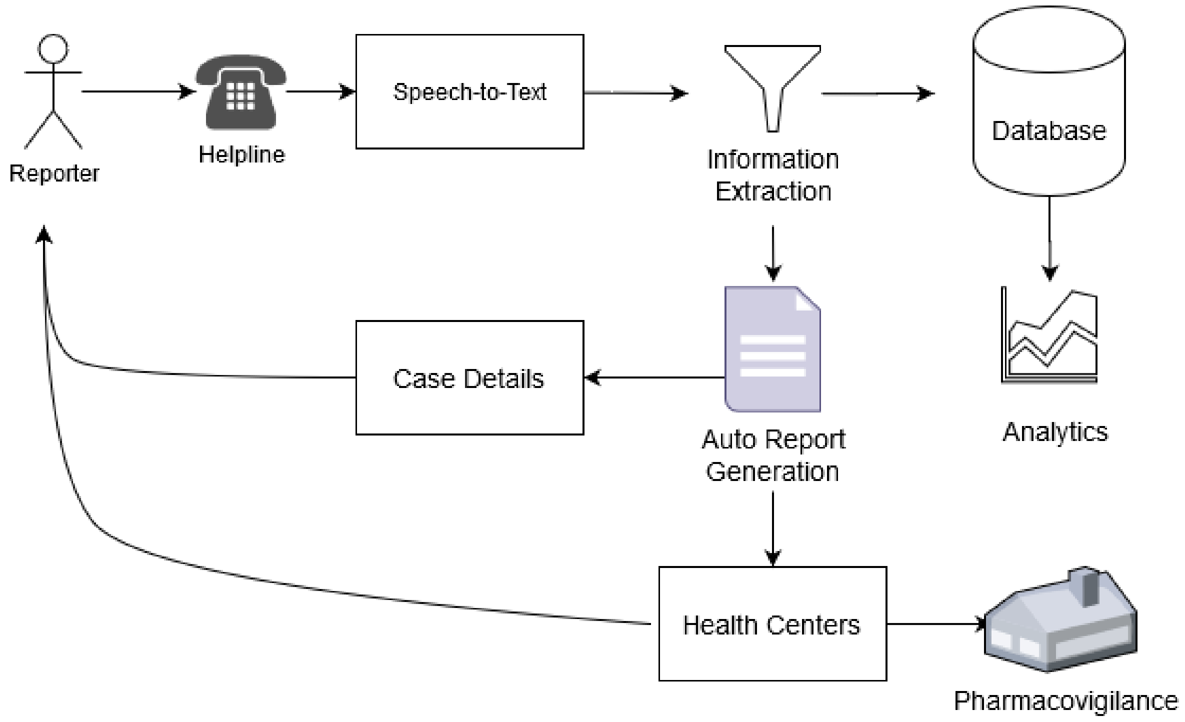


Figure 3.1: Overview of ADH system

Chapter 4

Design Of the System

4.1 Requirement Engineering

4.1.1 Requirement Elicitation

Understanding the requirements of the system is one of the first tasks to undertake. For this, we did the following.

4.1.1.1 Interviews

We first spoke with the members of the company to understand what the system was and what they expected us to build. Upon hearing what they had to say, the requirements of the project were understood but not all the requirements were clear.

4.1.1.2 Brainstorming

The company then organised a meeting with two experts in the field of pharmacovigilance who spoke about how the entire process of researching and developing drugs works and how the reports of adverse events help manufacturers improve the quality of drugs and where and how our project will help this process. They then went on to suggest possible approaches to the project.

One of the approaches discussed was using an IVR. To understand more about IVRs, we spoke to another expert in the field who pointed us to a software that could help us in the project.

4.1.1.3 Domain Analysis

Upon discussing with the experts we could conclude that the project was a combination of two domains, namely, telephony and machine learning.

4.1.2 Software lifecycle model

Choosing the right SDLC model for the project is key to make sure the project satisfies the requirements and also to incorporates feedback from the company. Additionally, to allow for changes in requirements from the company was necessary as the requirements of the project were not clear at the requirements gathering stage.

For our project, we have used the prototyping model. The reason we chose this model was because telephony was a field in which we had no experience and also because the feedback of the company was key.

As the project domain was unknown, we went about trying different technologies and presented them to the company, obtaining feedback from

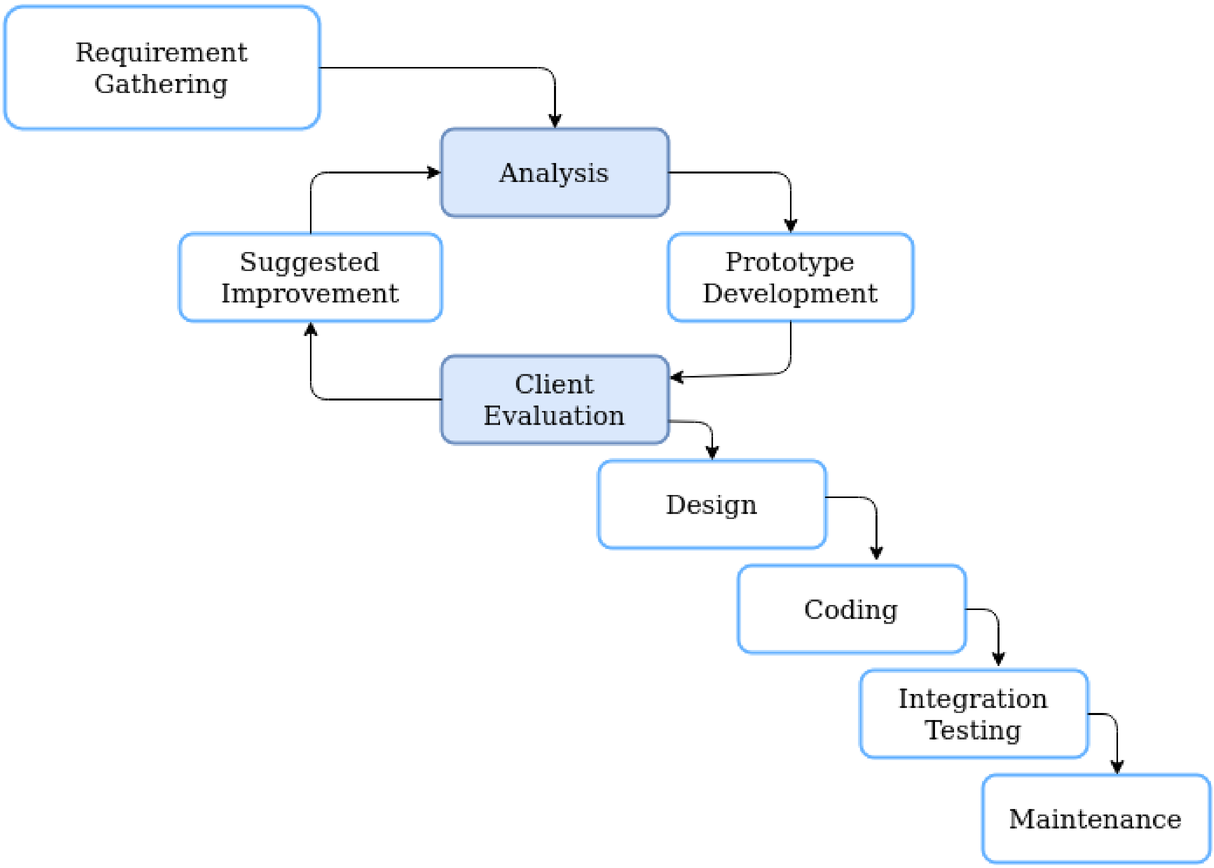


Figure 4.1: Evolutionary prototyping model[3]

them. When their requirement was met, we went ahead and developed the project using that technology.

The SDLC model can be show using the figure 4.1

4.1.3 Requirement Analysis

4.1.3.1 Data flow diagrams

We present the flow of data in the system using data flow diagrams (DFDs) in figures 4.2, 4.3, and 4.4.

Figure 4.2 shows a very high level (level 0) view of the data flowing in and out of the system. The diagram is then refined to capture more and more detail of the data flowing through the system. With the level 2 DFD (figure 4.4), the databases and the flow of data is made clear enough to understand the complete system.

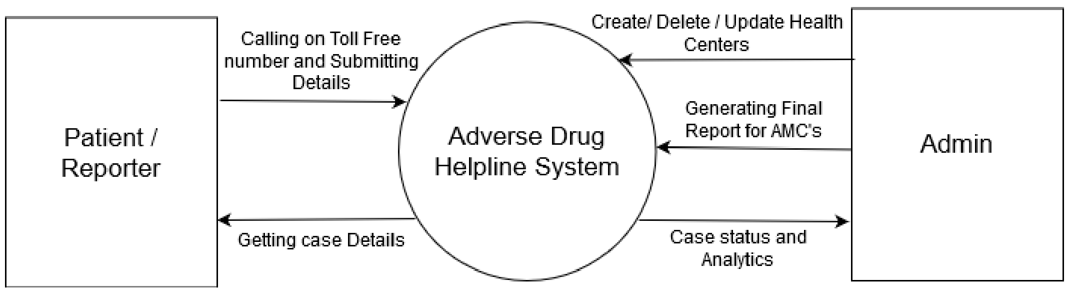


Figure 4.2: Level 0 DFD of ADH system

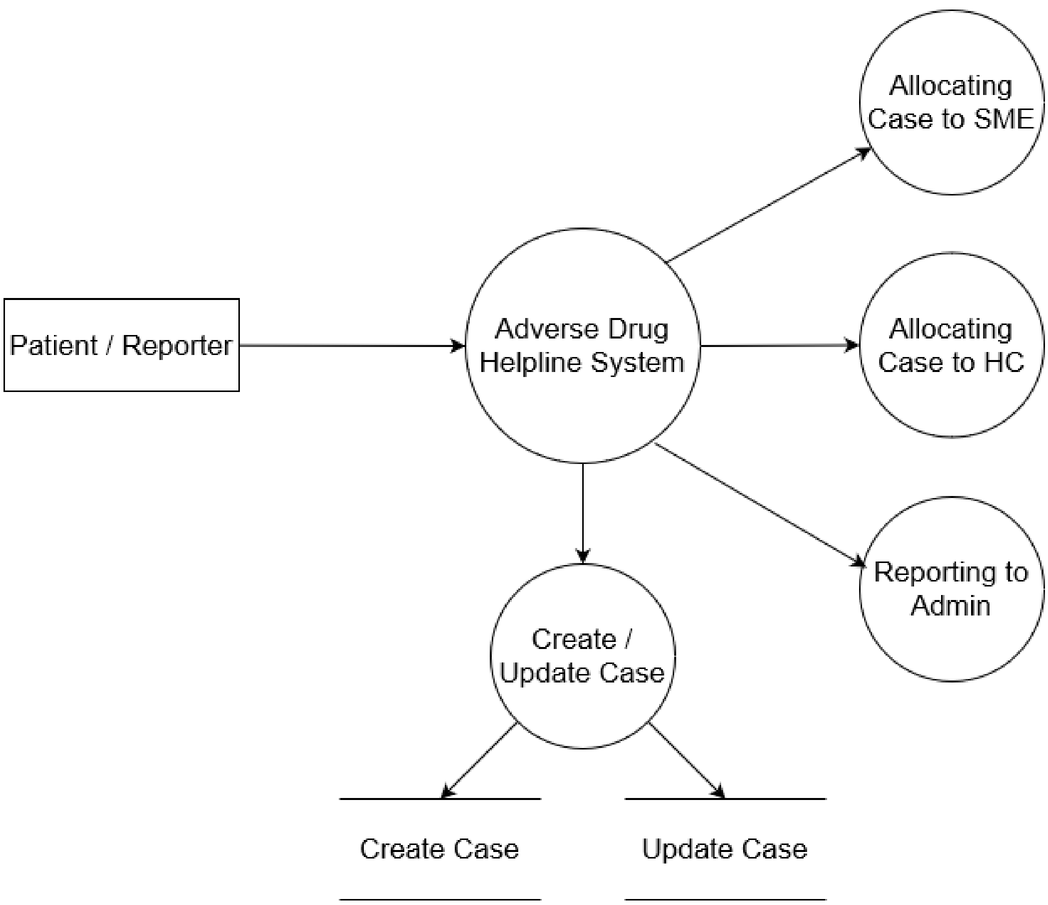


Figure 4.3: Level 1 of DFD ADH system

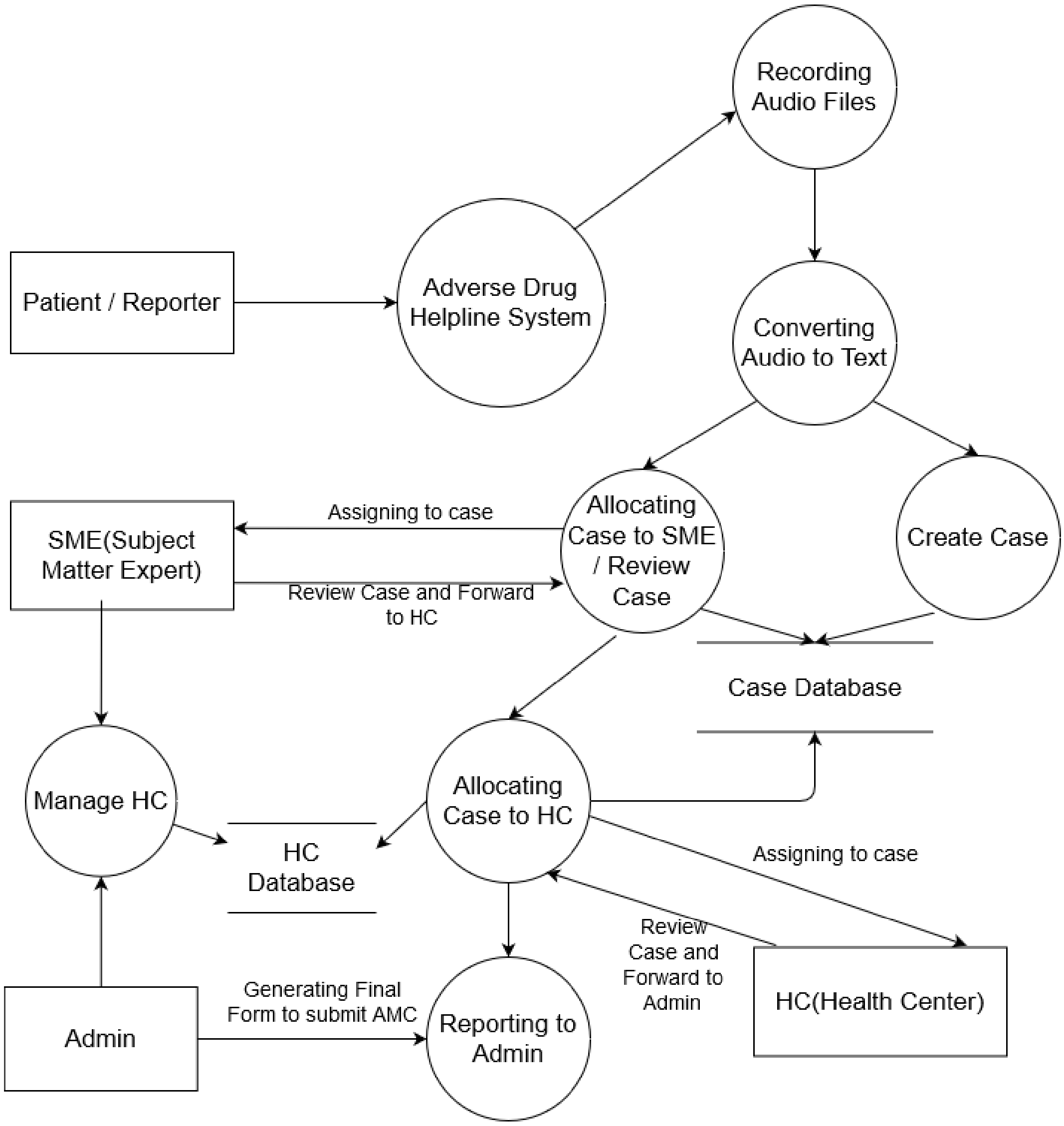


Figure 4.4: Level 2 DFD of ADH system

4.1.3.2 Use Case Diagram

In the figure 4.5, we present the actors, and their roles. As shown in the figure, there are four actors, namely the reporter/patient, subject matter expert(s) (SME), administrator, and health center(s). The health center(s) comprises of hospitals, clinics, pathology labs, etc.

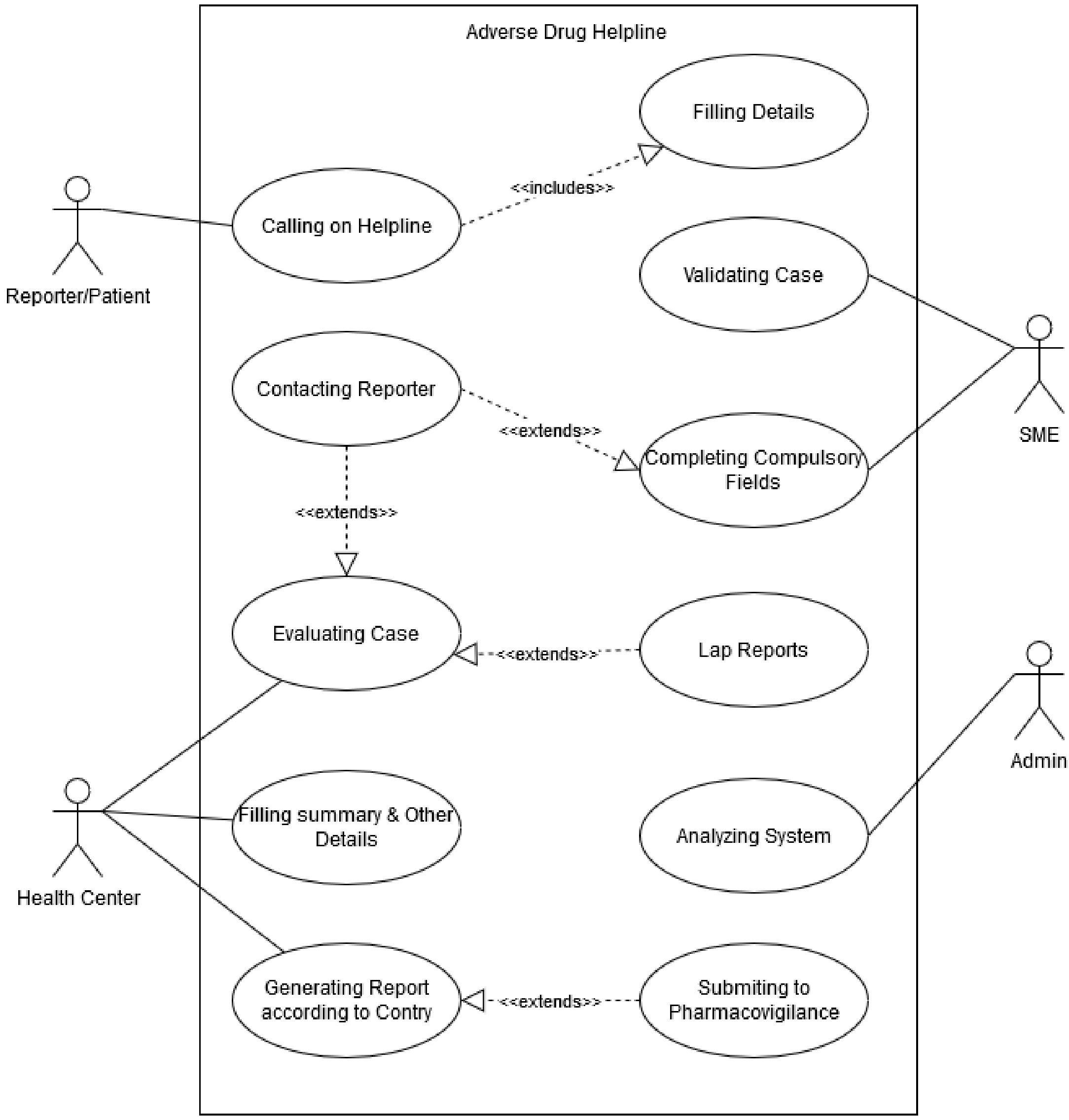


Figure 4.5: ADH use case diagram

4.1.3.3 Cost Analysis

The system uses cloud services to retrieve the coordinates of the health centres and to compute approximate the location of the patient.Additionally, when the health centre has to be allocated, another cloud service is used to get the location of the closest health centre from the patient's current location.

The cloud services that we are using are two Google Maps services, namely, the Geocoding API and the Distance Matrix API. The Geocoding API charges $5 for every 1000 requests and the Distance Matrix API charges $5 for every 1000 elements.[17]

4.1.3.4 Hardware and Software Requirements

The project can be broken down into a client side and a server side. Each of them have different requirements. We present the same as follows.

- **Server side**
  - Software requirements
    1. Python
    2. PostgreSQL
    3. Asterisk
    4. Google Speech-to-Text API
    5. Google Maps API
  - Hardware requirements
    1. 1.8 GHz CPU
    2. 4 GB RAM
    3. Minimum 30 GB Storage
- **Client side**
  - Software requirements
    1. Zoiper softphone
  - Hardware requirements
    1. A device capable of placing a call

4.2 System architecture

4.2.1 UI diagram

4.2.1.1 SME UI diagram

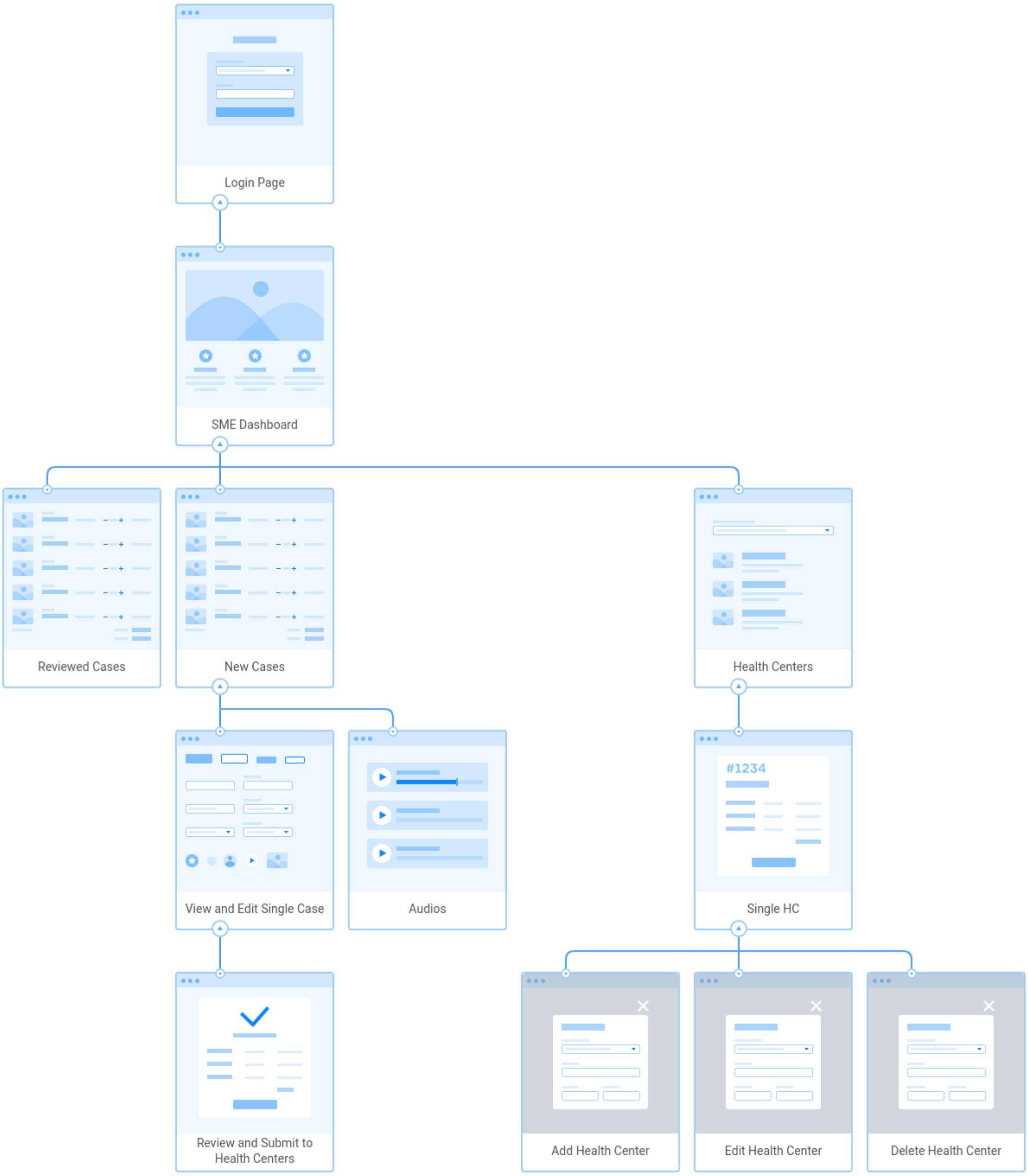


Figure 4.6: SME UI diagram

4.2.1.2 HC UI diagram

Figure 4.7: HC UI diagram

4.2.1.3 Admin UI diagram

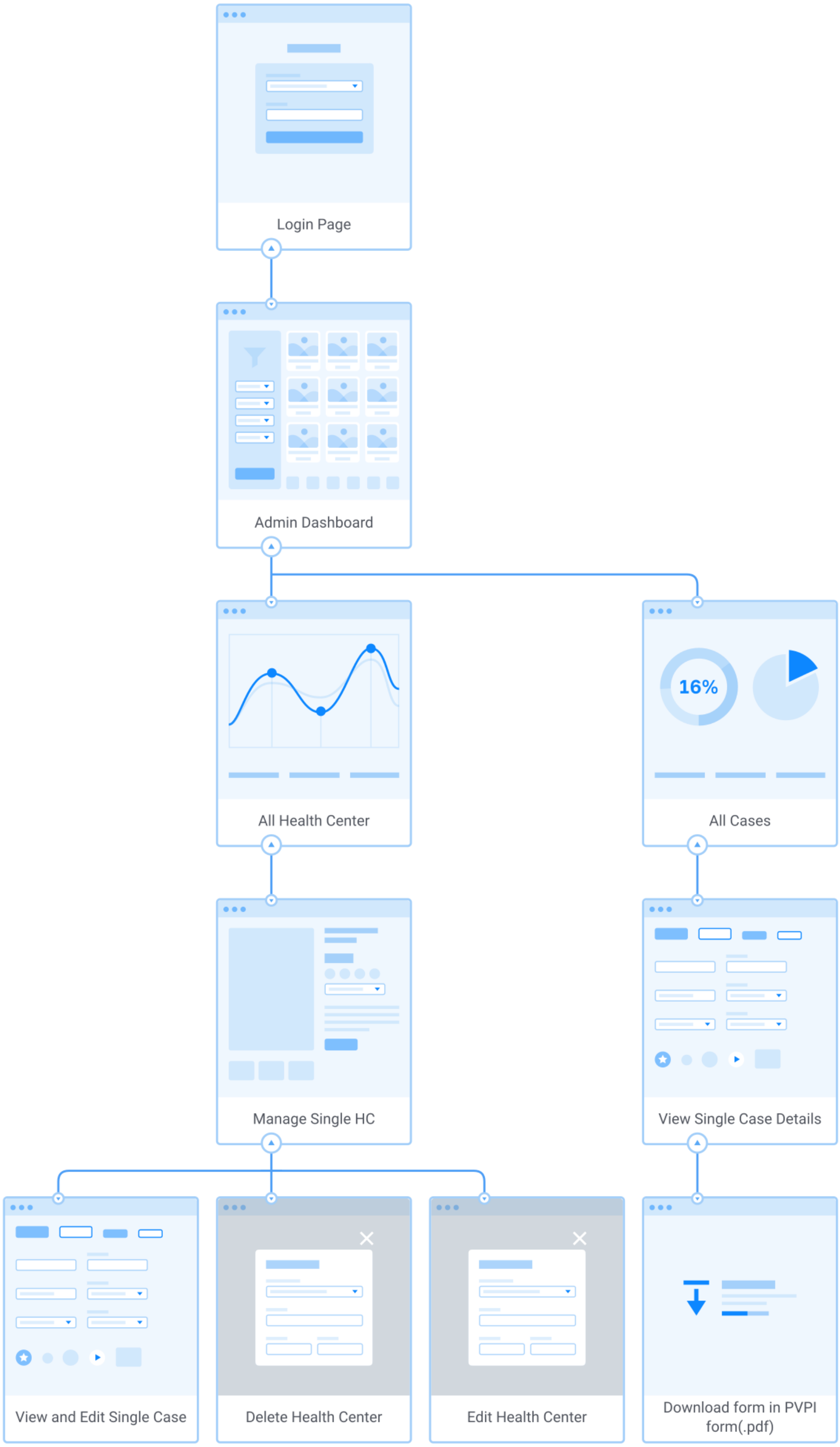


Figure 4.8: Admin UI diagram

4.2.2 UX diagram

4.2.2.1 IVR UX diagram

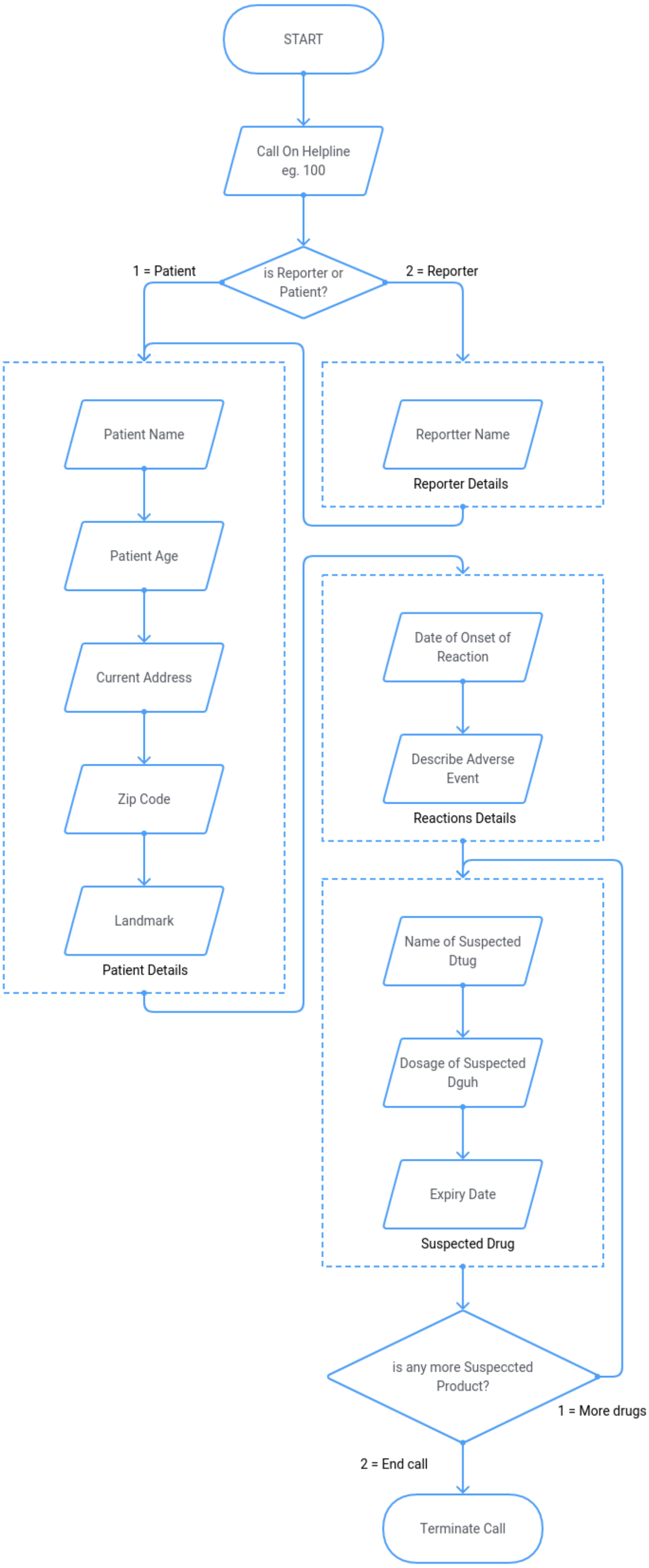


Figure 4.9: IVR UX diagram

4.2.2.2 Portal UX diagram

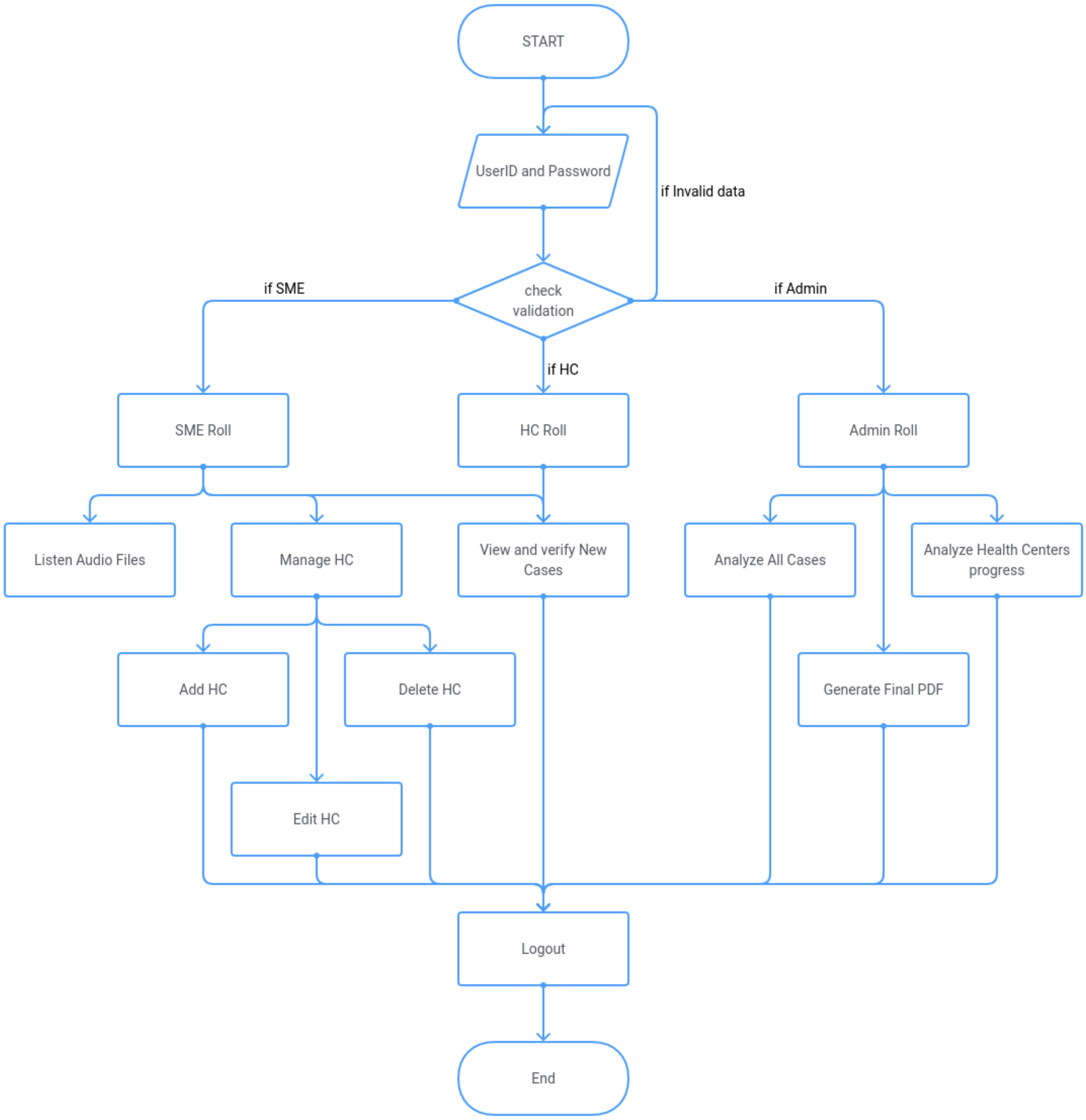


Figure 4.10: Portal UX diagram

4.2.3 Block Diagram

The block diagram of the system is shown in figure 3.1. The working of the system is as follows.

The patient or the reporter of the patient calls the helpline when they experience an ADR. They speak or type the information that is asked over the phone. The IVR system then sends a request to the web server. The request contains the required information to fetch recordings from the IVR.

Once the server receives the request, it parses it and uses the information to fetch the recordings of the call. It uses the speech-to-text API to

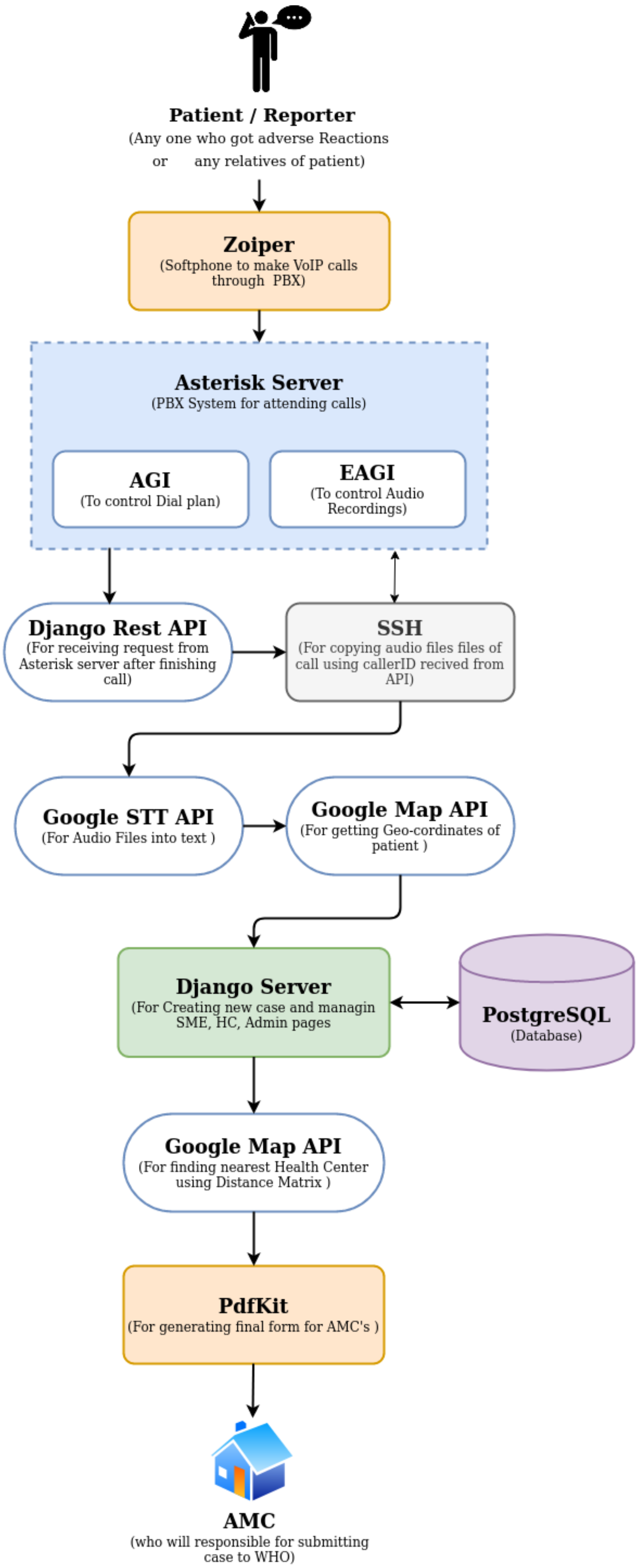


Figure 4.11: Block diagram of ADH system

transcribe the audio. The audio is then archived for future reference and the transcription of the audio is stored in the database. The server then registers a case, using this information and assigns it to the SME.

The SME verifies the inputs for correctness, i.e. they make sure that the transcription of the audio matches the audio. If there is an error, they can listen back to the recording and make the appropriate changes. Additionally, if the SME requires, they can get in contact with the reporter to correct any mistakes. After the SME confirms the validity of the input, a health center is allocated to the case. The health center is chosen such that it is closest to the address of the patient.

After the health center is allocated, they can contact the patient to make sure all the required fields are filled in to register a valid case. The health center can provide their comments on this case and then complete the case. Only once the case is flagged as completed, the final form can be generated. After this, the case can be forwarded to an AMC or the NCC for further action.

The administrator of the system can monitor various parameters such as the number of cases, number of cases assigned to individual health centers, etc. The administrator can also add, edit, and delete health centers. Additionally, the administrator can view the forms generated after the completion of the case.

Chapter 5

Result and Discussion

5.1 Screenshots of the System

5.1.1 Screenshots of the call

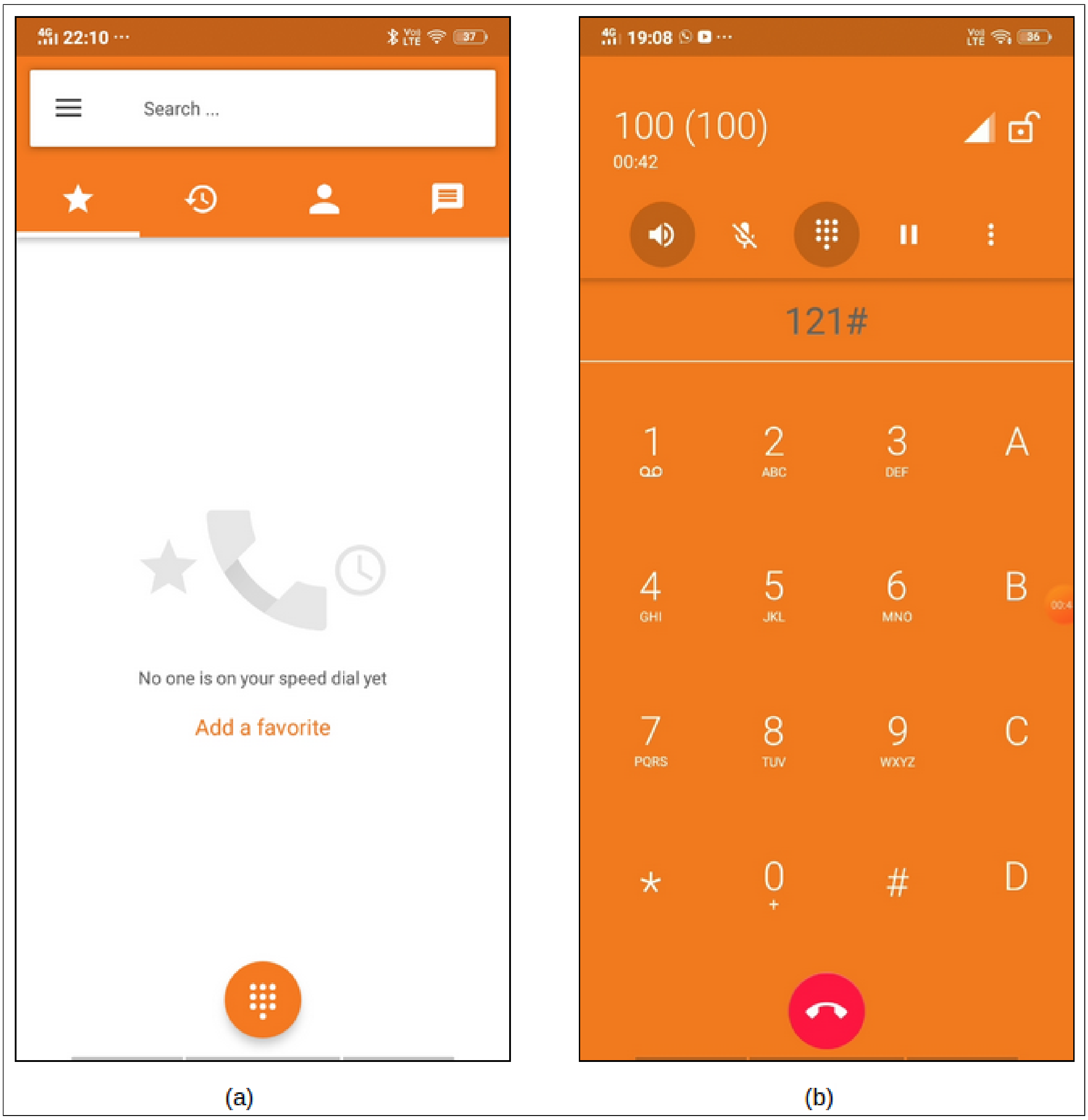


Figure 5.1: The call interface of Zoiper

Figures 5.1 showing the screen of Zopier Application. To call on helpline number 100 as shown in 5.1 we have to register Zoiper at our Asterisk server for eg. 6001@192.168.1.100. After successful registration Zoiper will able to connect call.

5.1.2 Portal Screenshots

5.1.2.1 Login page

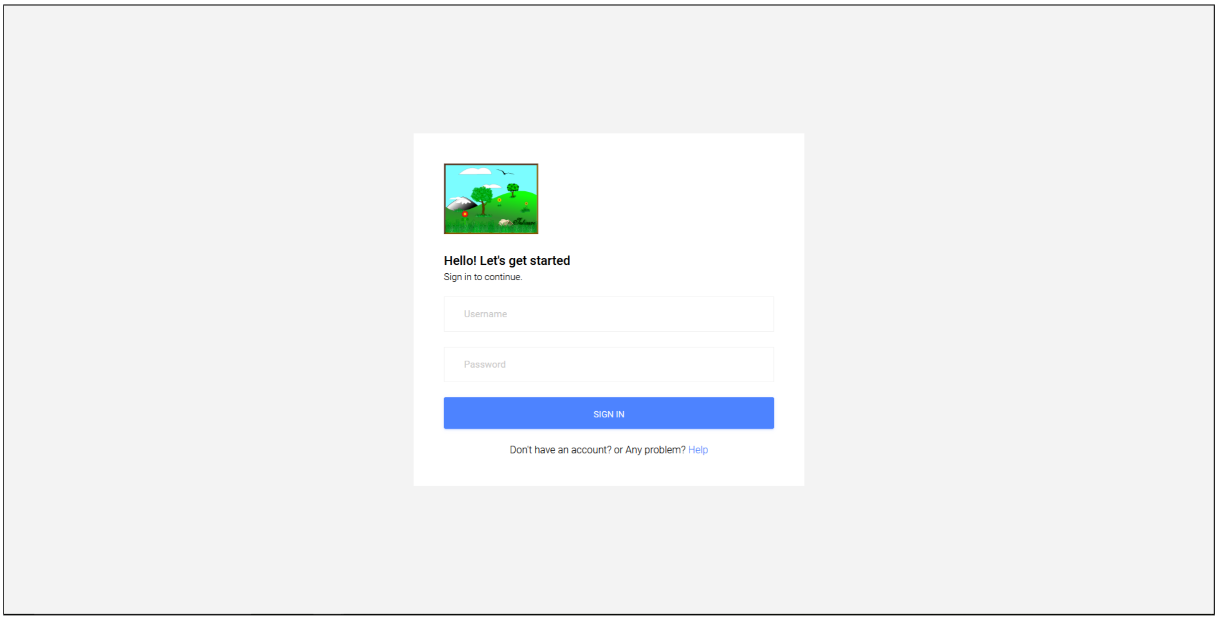


Figure 5.2: Login screen of the portal

5.1.2.2 Subject matter expert related screenshots

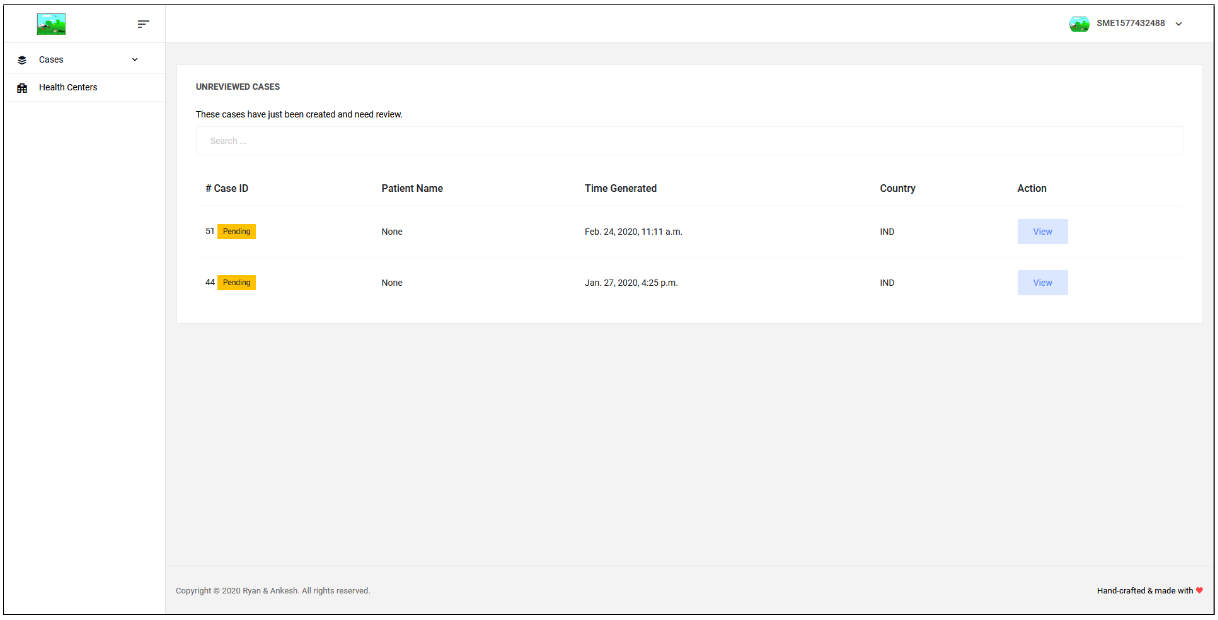


Figure 5.3: SME unreviewed cases page

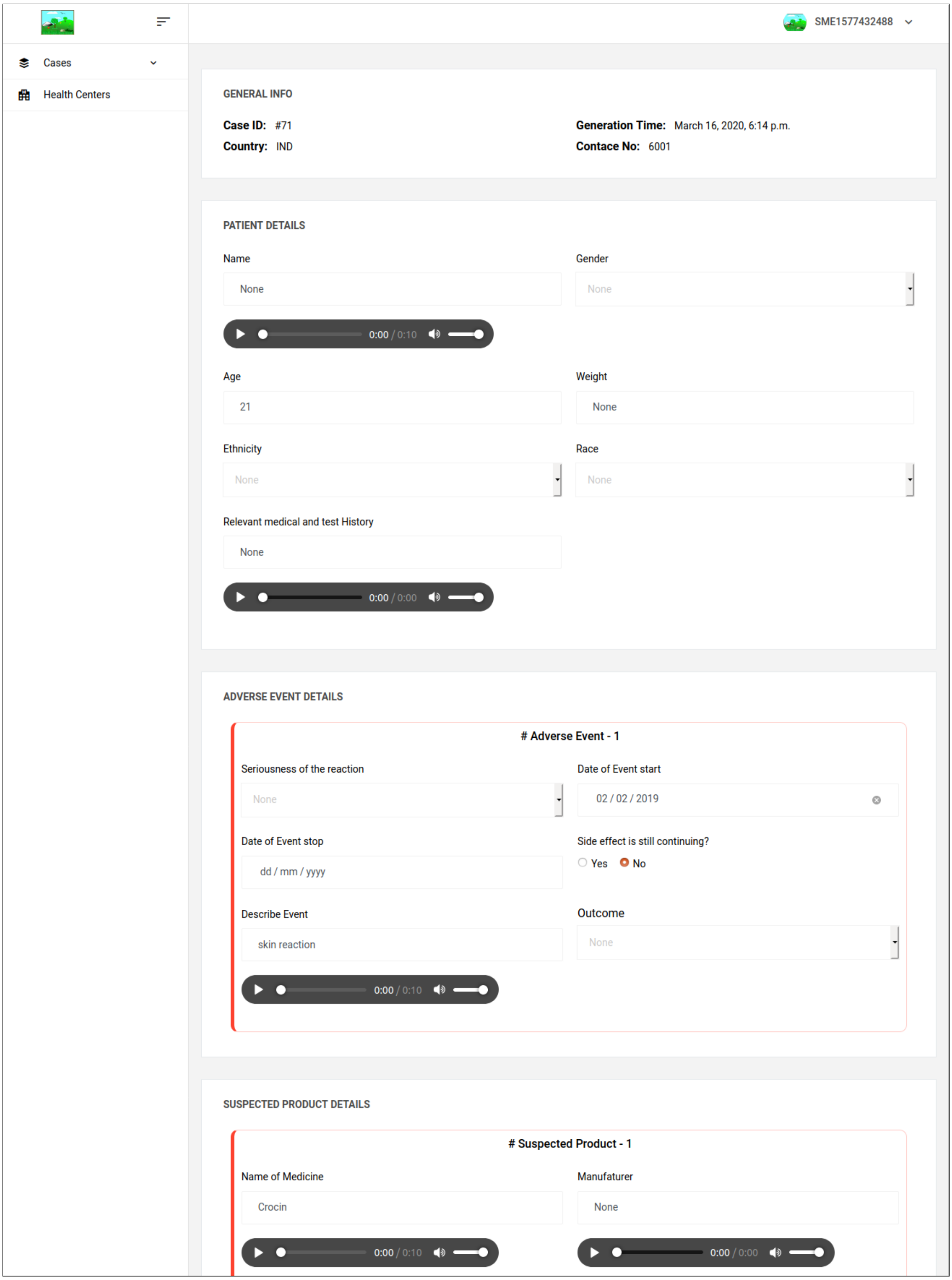


Figure 5.4: Single case after clicking on 'View' button from unreviewed cases (part-1)

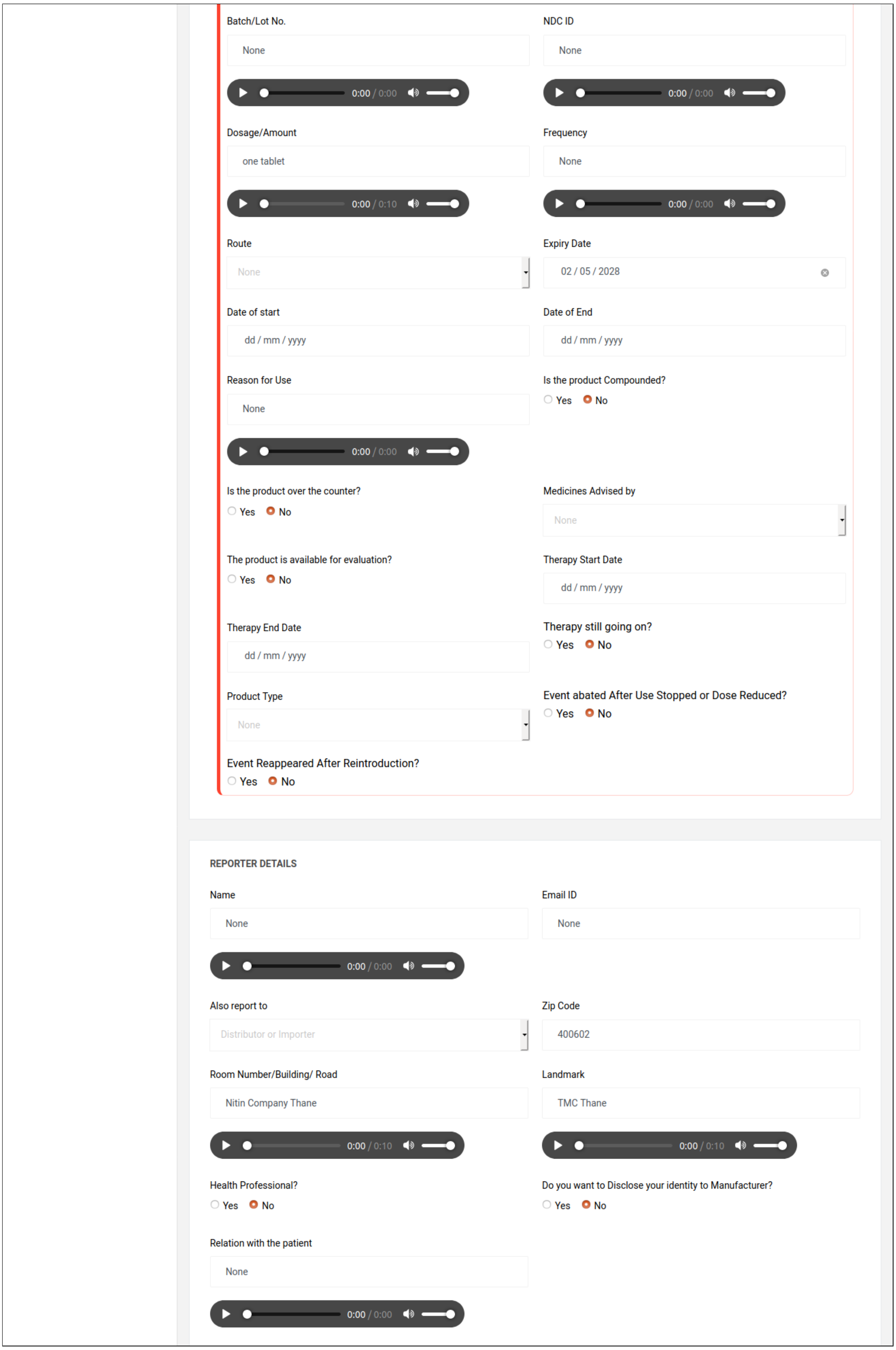


Figure 5.5: Single case after clicking on 'View' button from unreviewed cases (part-2)

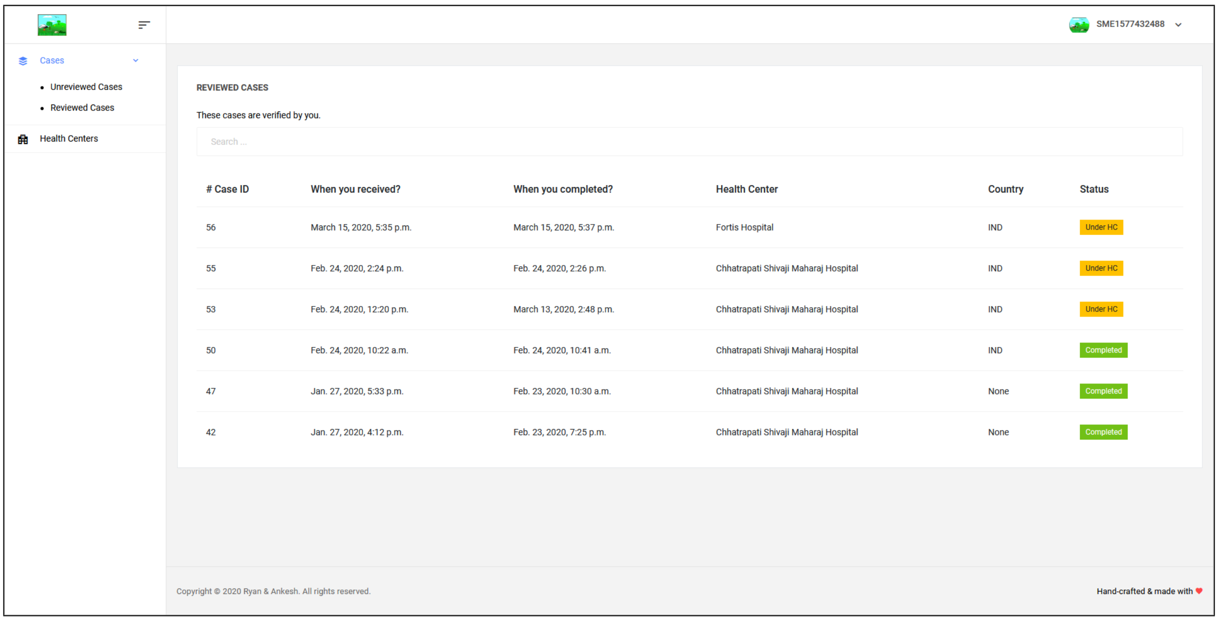


Figure 5.6: SME reviewed cases page

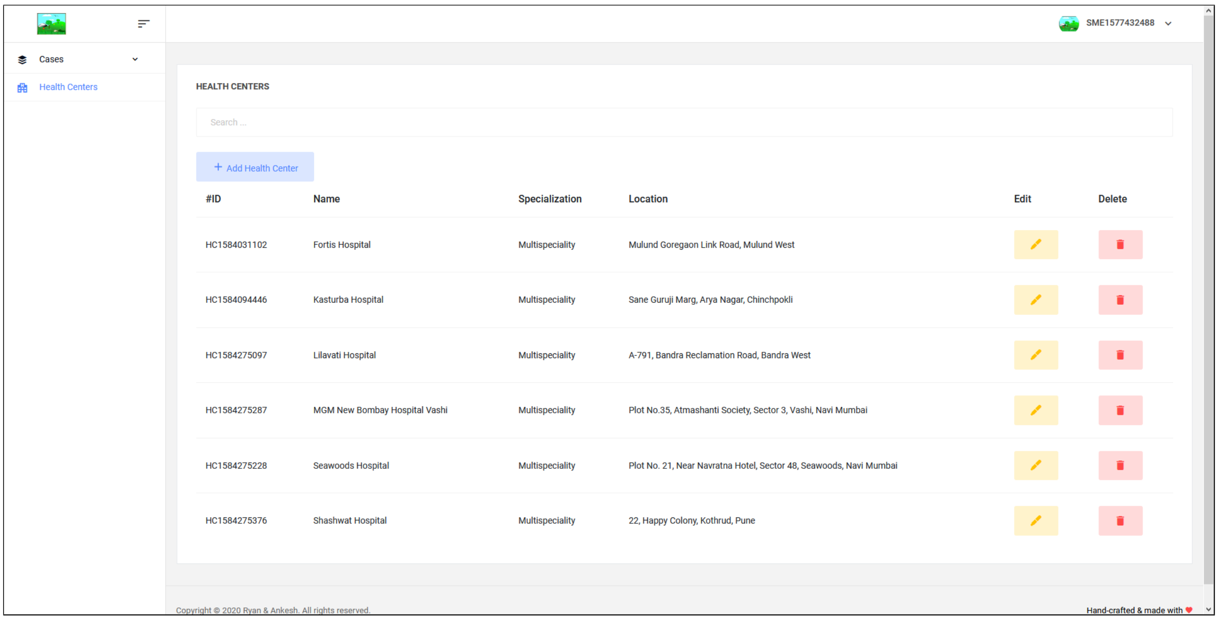


Figure 5.7: The page where the SME can view the health centers

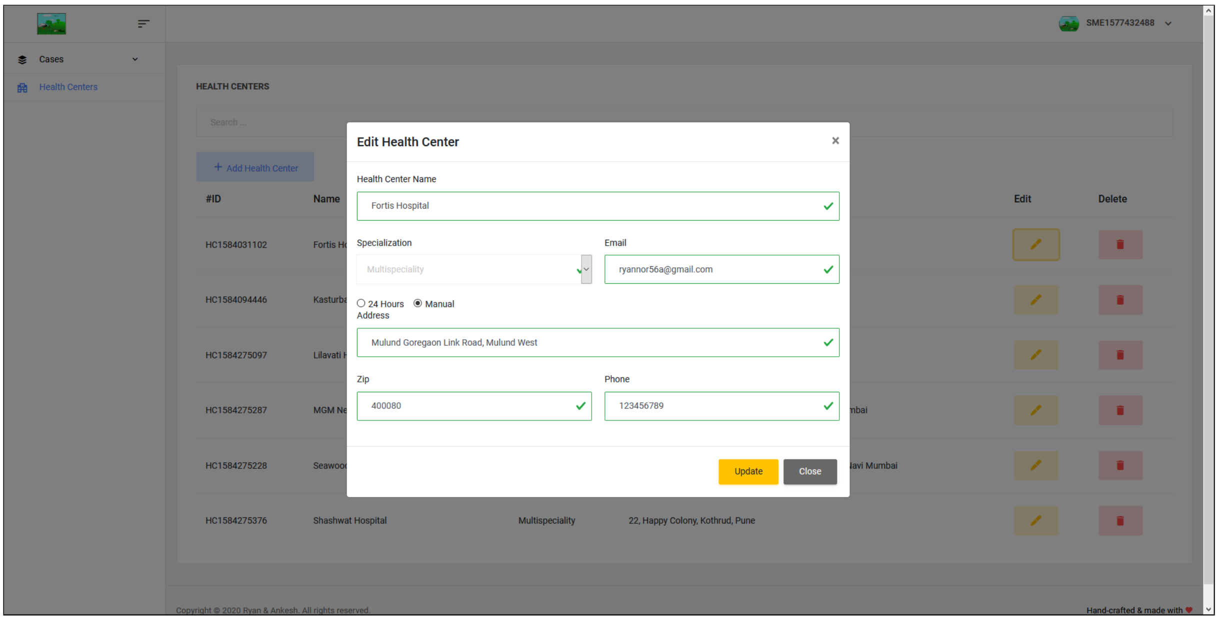


Figure 5.8: SME editing a health center

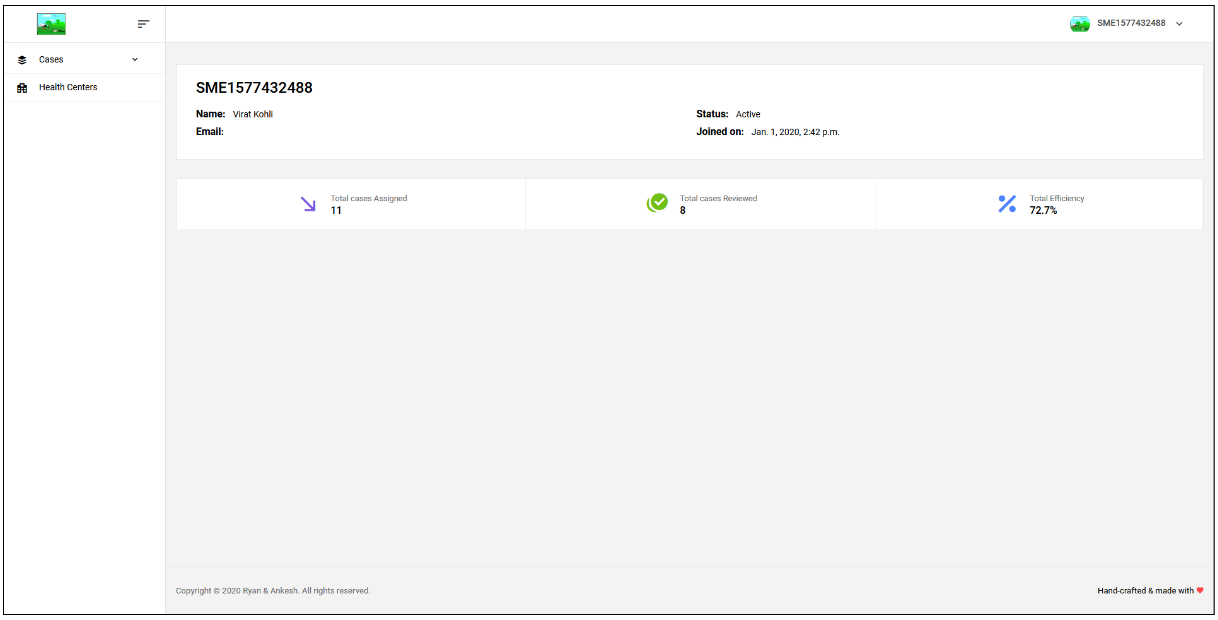


Figure 5.9: The page where the SME can view their statistics

5.1.2.3 Health center related screenshots

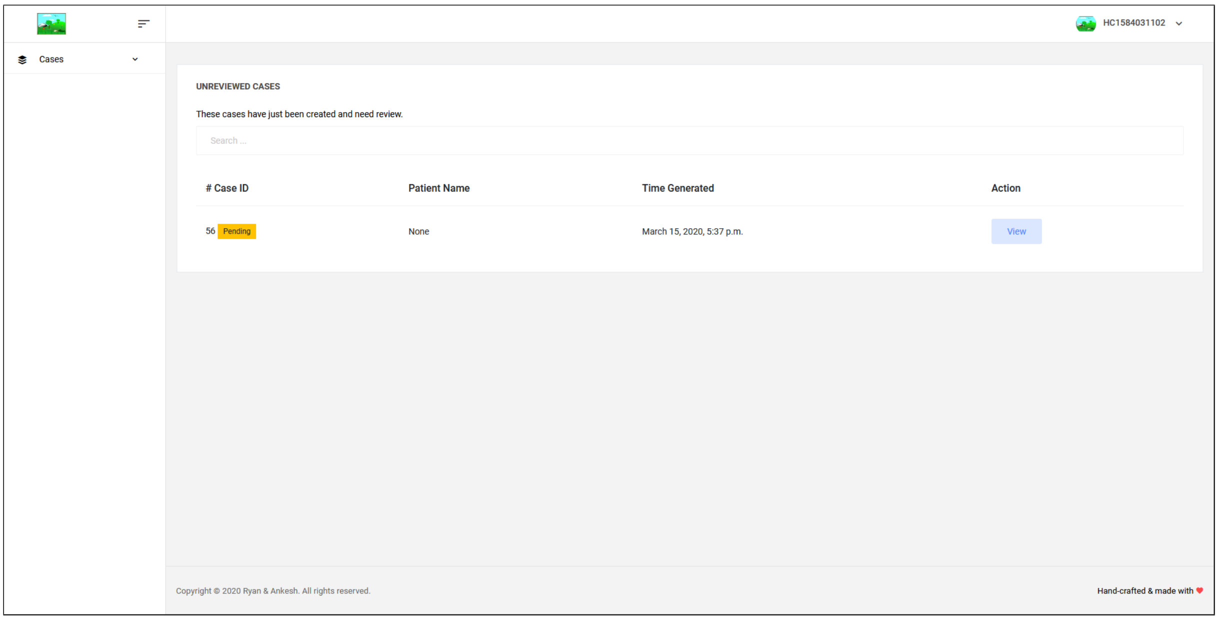


Figure 5.10: HC unreviewed cases page

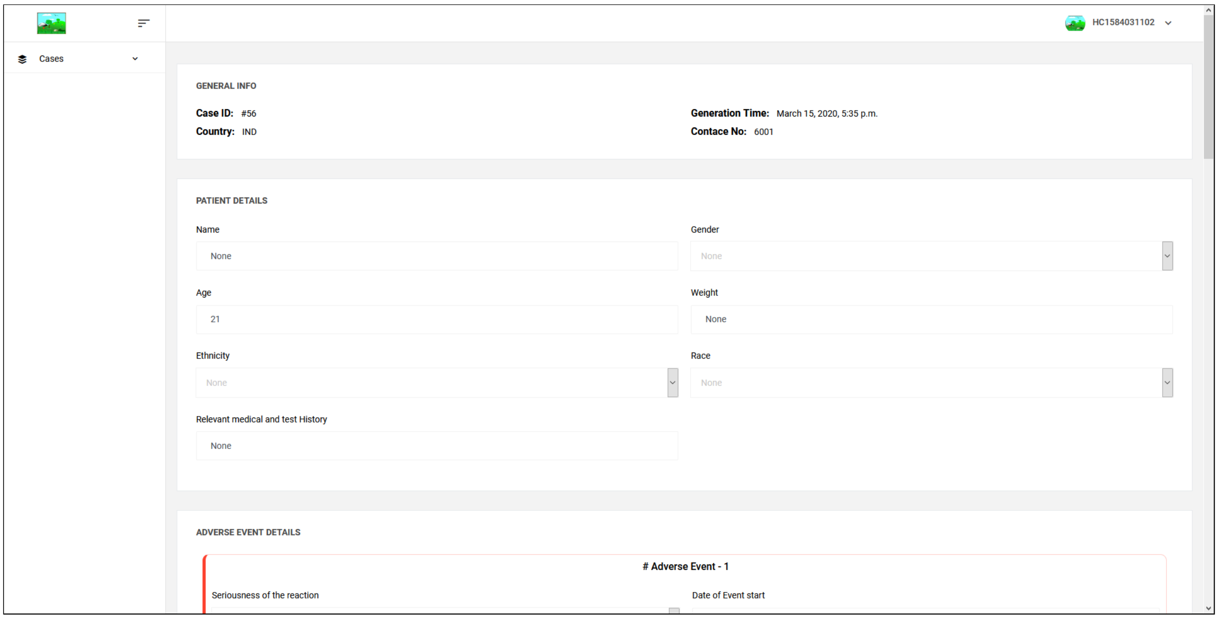


Figure 5.11: HC reviewing a case

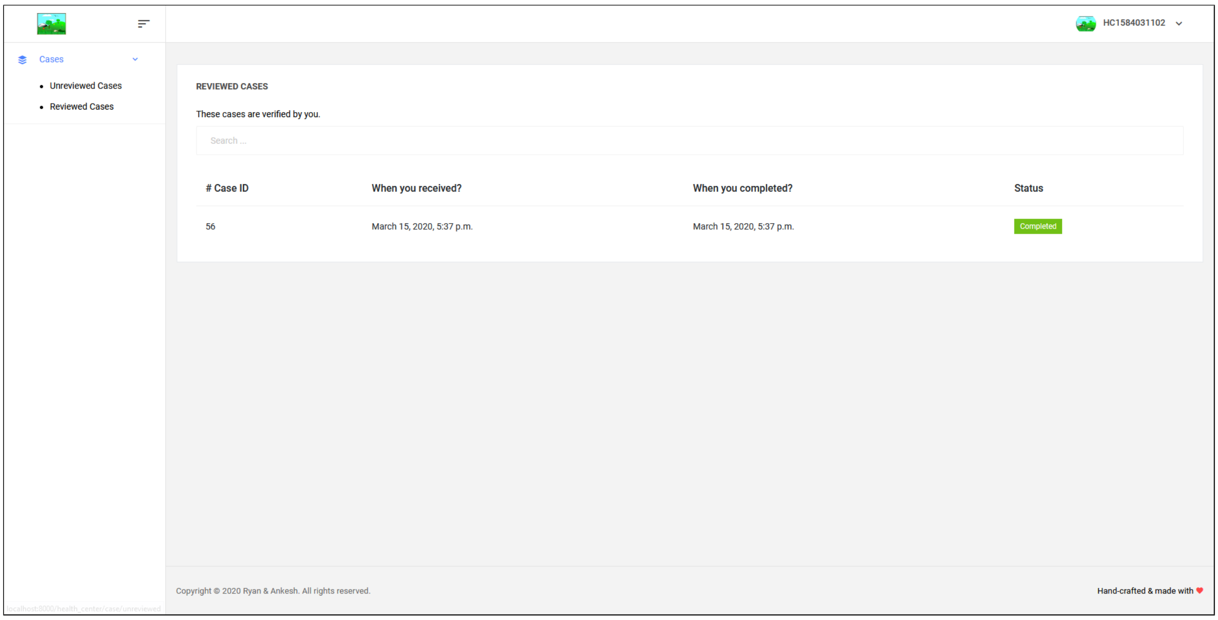


Figure 5.12: HC reviewed cases page

5.1.2.4 Administrator related screenshots

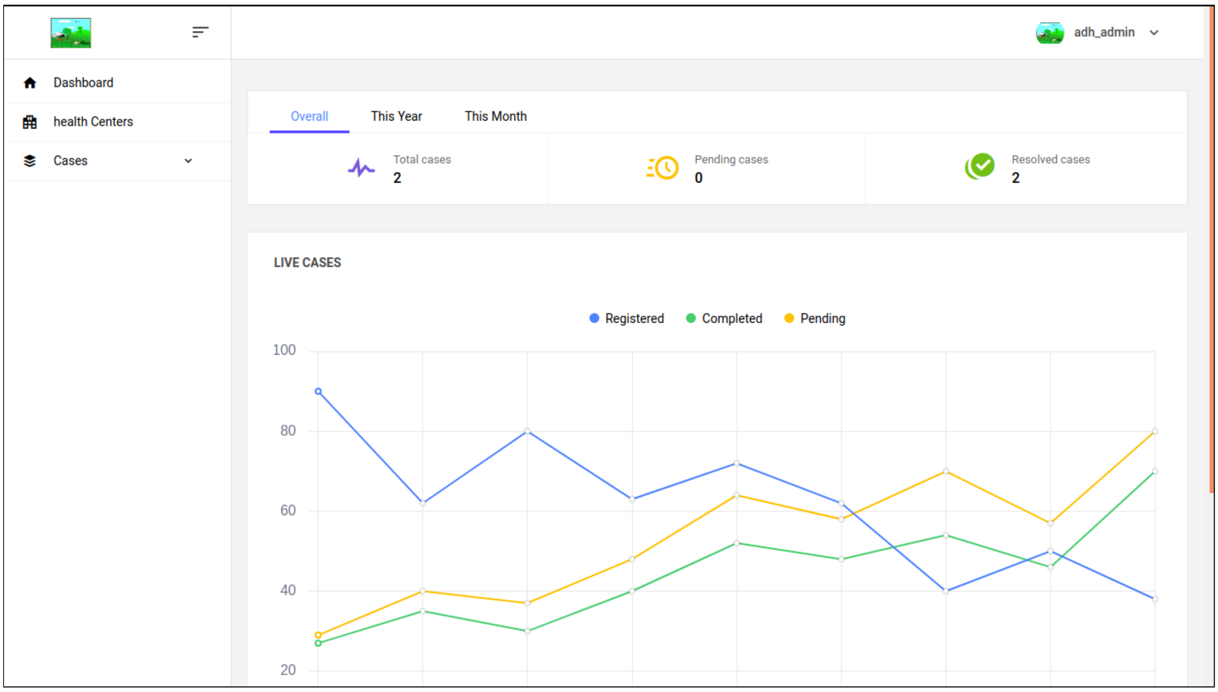


Figure 5.13: The administrator's dashboard

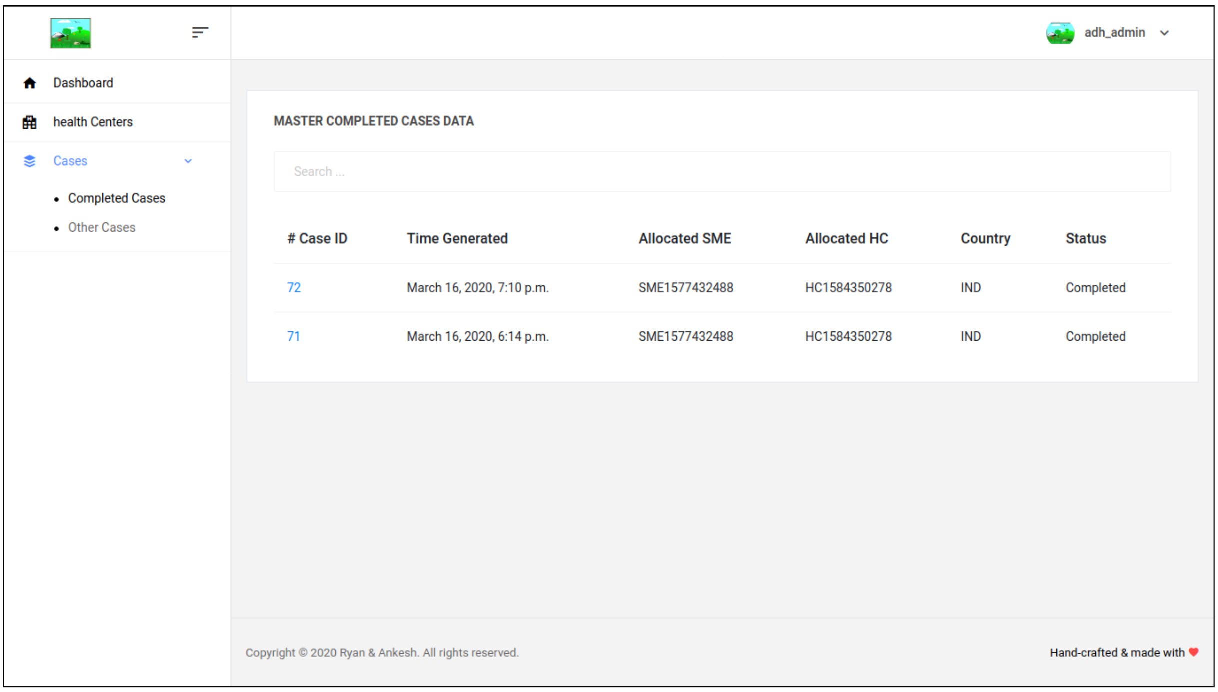
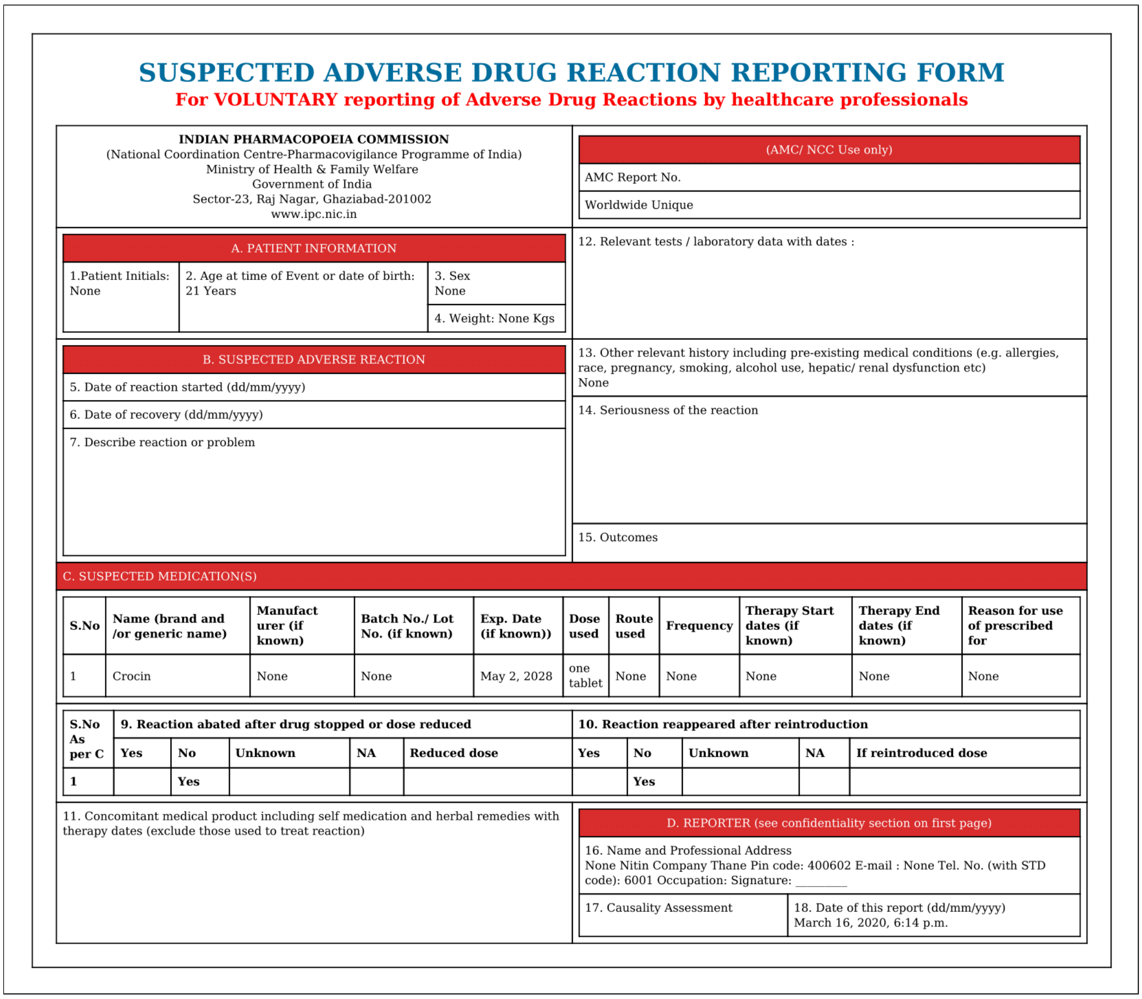


Figure 5.14: All of the completed cases



SUSPECTED ADVERSE DRUG REACTION REPORTING FORM

For VOLUNTARY reporting of Adverse Drug Reactions by healthcare professionals

INDIAN PHARMACOPOEIA COMMISSION
(National Coordination Centre-Pharmacovigilance Programme of India)
Ministry of Health & Family Welfare
Government of India
Sector-23, Raj Nagar, Ghaziabad-201002
www.ipc.nic.in

(AMC/ NCC Use only)
AMC Report No.
Worldwide Unique

A. PATIENT INFORMATION

1.Patient Initials: None	2. Age at time of Event or date of birth: 21 Years	3. Sex None
		4. Weight: None Kgs

12. Relevant tests / laboratory data with dates :

B. SUSPECTED ADVERSE REACTION

5. Date of reaction started (dd/mm/yyyy)
6. Date of recovery (dd/mm/yyyy)
7. Describe reaction or problem

13. Other relevant history including pre-existing medical conditions (e.g. allergies, race, pregnancy, smoking, alcohol use, hepatic/ renal dysfunction etc)
None

14. Seriousness of the reaction

15. Outcomes

C. SUSPECTED MEDICATION(S)

S.No	Name (brand and /or generic name)	Manufacturer (if known)	Batch No./ Lot No. (if known)	Exp. Date (if known))	Dose used	Route used	Frequency	Therapy Start dates (if known)	Therapy End dates (if known)	Reason for use of prescribed for
1	Crocin	None	None	May 2, 2028	one tablet	None	None	None	None	None

S.No As per C	9. Reaction abated after drug stopped or dose reduced					10. Reaction reappeared after reintroduction				
	Yes	No	Unknown	NA	Reduced dose	Yes	No	Unknown	NA	If reintroduced dose
1		Yes					Yes			

11. Concomitant medical product including self medication and herbal remedies with therapy dates (exclude those used to treat reaction)

D. REPORTER (see confidentiality section on first page)

16. Name and Professional Address
None Nitin Company Thane Pin code: 400602 E-mail : None Tel. No. (with STD code): 6001 Occupation: Signature: _______

17. Causality Assessment

18. Date of this report (dd/mm/yyyy)
March 16, 2020, 6:14 p.m.

Figure 5.15: The form generated for the case

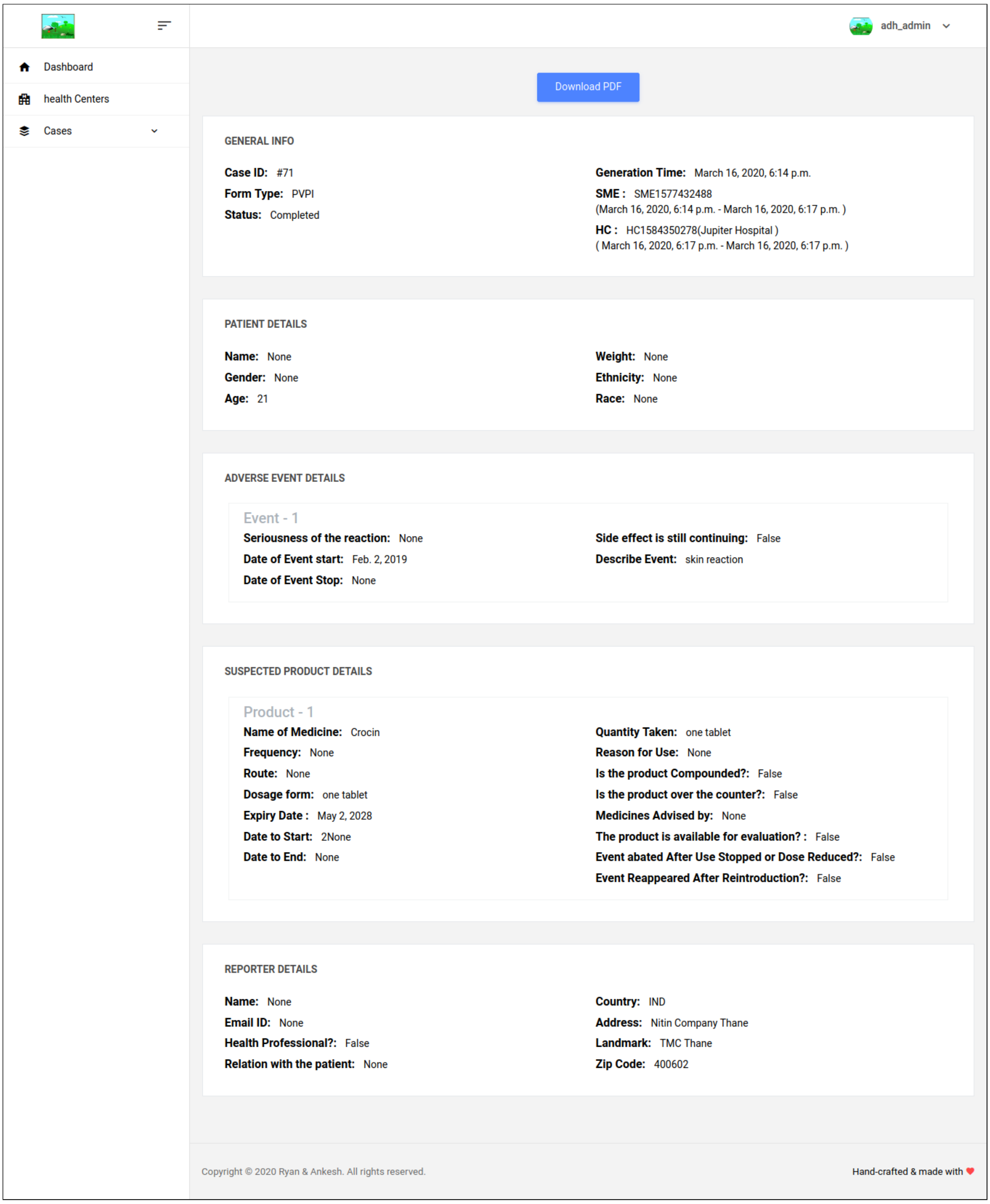


Figure 5.16: Administrator's view of a single case

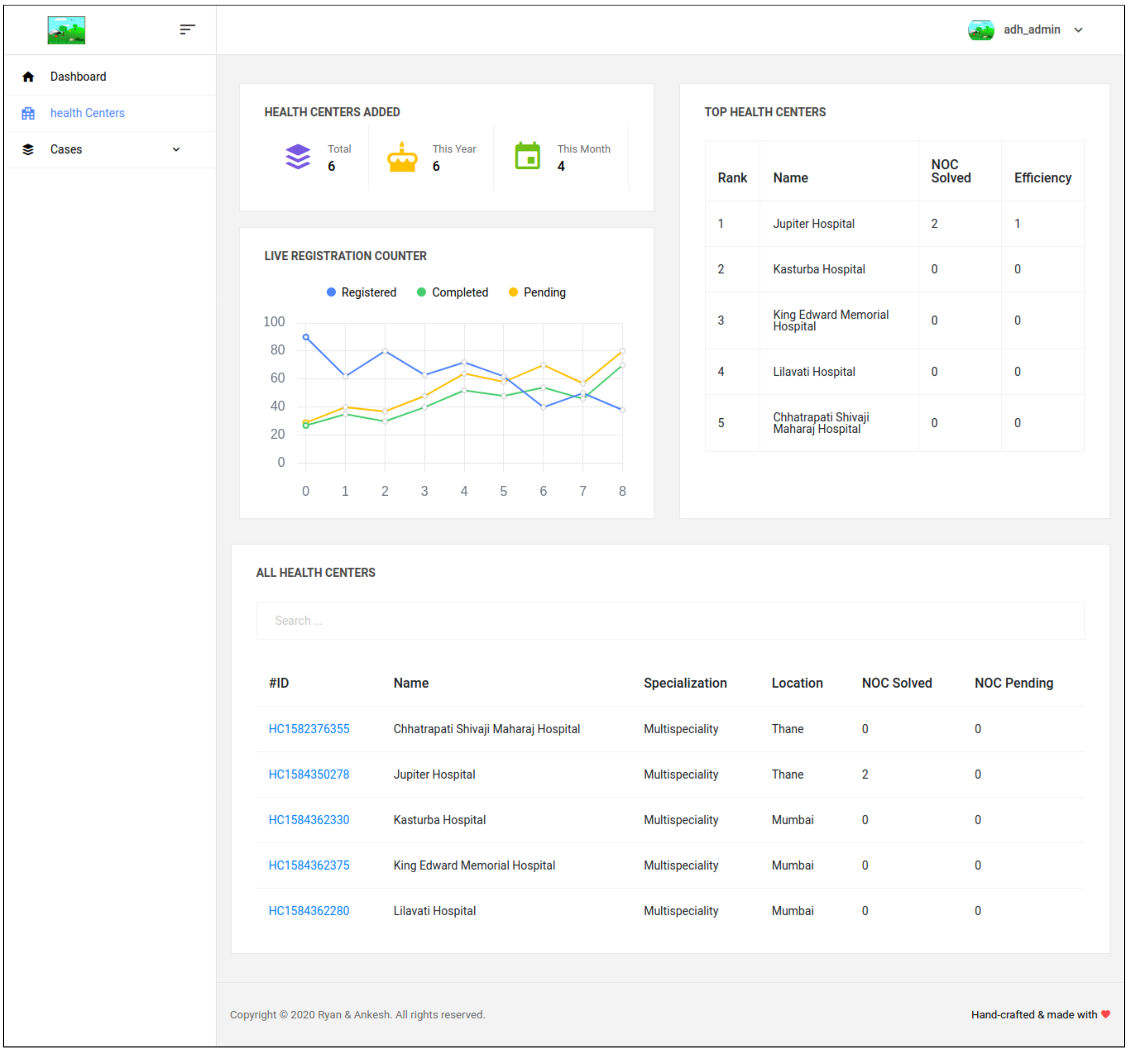


Figure 5.17: Administrator's view of all the HCs registered

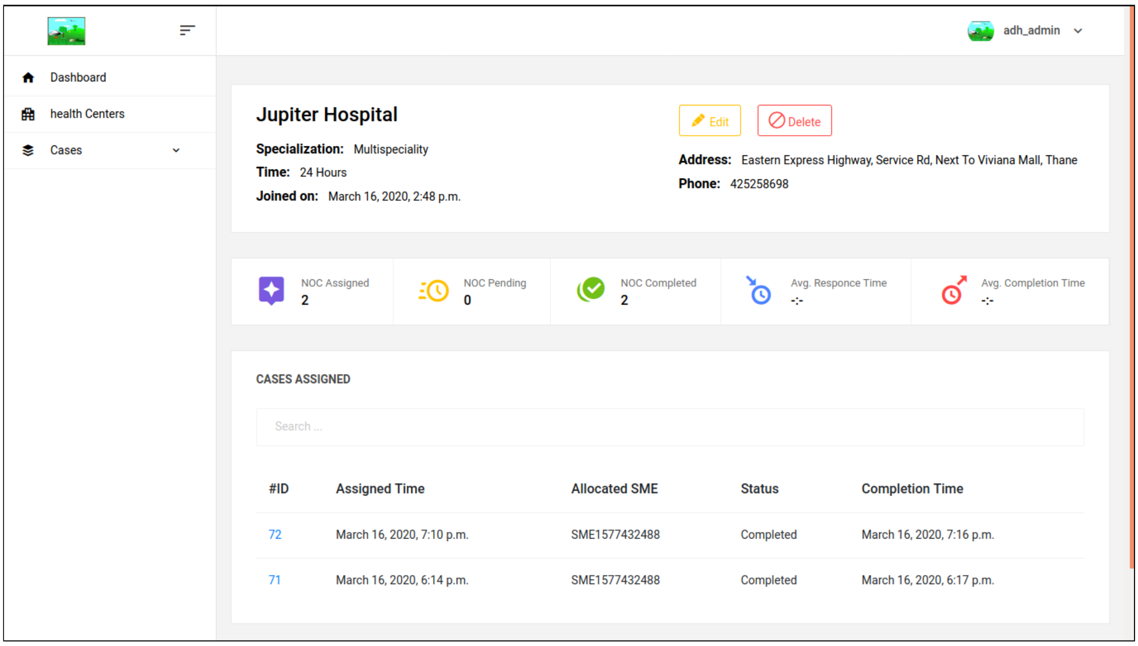


Figure 5.18: Administrator's view of a single HC with all the related information

5.2 Sample Code

5.2.1 Function to get input from the caller

```
def getInputFromCaller(agiHandler: ADHHandler,
                       fileToPlay: str,
                       timeout=4000,
                       numOfDigits=255) -> int:
    r'''
    Get input from the caller after playing the file.

    Arguments
    ---------
    -   fileToPlay => File to be played
    -   numOfDigits => Maximum number of digits to be expected
    -   timeout => Time (in milliseconds) before the timer expires

    Returns
    -------
    Digits entered by the caller if the number is valid, otherwise -1
    '''
    result = agiHandler.get_data(fileToPlay, timeout, numOfDigits)

    try:
        return int(result)
    except ValueError as e:
        return -1
```

5.2.2 Converting RAW file to WAV file

```
# Call a 'sox' subprocess to convert the raw audio to wav
subprocess.run([
    'sox', '--type', 'raw', '--rate',
    str(settings.FREQ_SAMPLING),
    '--encoding', 'signed', '--bits',
    str(settings.BITS_PER_SAMPLE),
    '--channels', '1',
    rawFile, wavFile,
])
```

5.2.3 Copying audio files from Asterisk server

```
postData = dict(request.data)

print("Post data: ",postData)
recordingDir = postData['recordingDir']

audioSource = 'root@192.168.1.254:/var/spool'\
    '/asterisk/recording/adh_recording/wav/{}'.format(recordingDir)
```

```
# Copying files from Asterisk server
subprocess.run([
    'scp',
    '-r',
    audioSource,
    '/var/spool/adh_recordings'
])
```

5.2.4 Transcribing received audio

```
recogniser = sr.Recogniser()

# Iterate over all the audio files received and
# transcribe them using Google's Speech-To-Text.
for audioFile in audioFiles:
    # Since we are using a file instead of a mic, the audio has to be
    # recorded and stored locally.
    with sr.AudioFile(audioFile) as source:
        audio = recogniser.record(source)

    # Try to transcribe the audio.
    # If the audio could not be recognised, the library raises a
    # UnknownValueError
    try:
        transcription = recogniser.recognise_google(audio, language='en-IN')

        print('Audio file contains: ', transcription)
    except sr.UnknownValueError:
        print('Could not recognise the audio')
```

5.2.5 Getting the Geo-coordinates of the patient/reporter

```
gmapsClient = googlemaps.Client(key=settings.GMAPS_KEY)

geocodeResponse = gmapsClient.geocode(
    '{}, Near {}, {}'.format(address, landmark, zipCode)
)

# Source for the distance matrix calculation
geoCoords = (
    geocodeResponse[0]['geometry']['location']['lat'],
    geocodeResponse[0]['geometry']['location']['lng']
)
```

5.2.6 Finding suitable Health Center and assigning the case to it

```
# Get a list of HCs in the same city as the caller
nearbyHc = HealthCenter.objects.all()

# Getting a list of tuple of the coords of the nearby HCs
nearbyHcCoords = list()
for hc in nearbyHc:
    nearbyHcCoords.append((hc.latitude, hc.longitude))

# Query the Distance Matrix API to help find the closest HC
distMatResponse = gmapsClient.distance_matrix(
    patientCoords, nearbyHcCoords
)

# Finding the distance of closest HC from the current location of the
# patient. There will always be a single source and so we can hardcode
# to fetch only the first row in the response.
minDist = 9999999999
hcIdx = -1
for (i, element) in enumerate(distMatResponse['rows'][0]['elements']):
    if element['distance']['value'] < minDist:
        minDist = element['distance']['value']
        hcIdx = i

# Now that a suitable HC is found allocate the case to the HC
suitableHc = nearbyHc[hcIdx].hc.username
print(nearbyHc[hcIdx].name)
```

5.3 Testing

Testing is one of the important phases of our project. It helps us to check whether the actual results match with the expected result and also to help identify errors.

We have performed manual testing by writing test cases. We have done testing in three parts as follows:

5.3.1 Unit testing

In this phase of testing we have broken down our project into small units according to functionality. In total, we have five units as follows:

1. AGI module (table 5.1)
2. EAGI module (table 5.2)
3. REST API module (table 5.3)
4. Subject matter expert role module (table 5.4)
5. HC role module (table 5.5)

5.3.1.1 AGI testing

Table 5.1: AGI testing.

Module No: IVR01		**Date Tested:**16/01/2020	
Module Name:AGI			
Case ID	**Test Case Description**		**Result**
	Prerequisites	**Test Data**	
	Expected Results	**Actual Results**	
IVR0101	No information is entered when prompted		Pass
	Zoiper must be connected to Asterisk.	None	
	Prompt again to enter information	Prompted a total of 3 times before skipping the question	
IVR0102	Entered format of date is wrong		Pass
	Zoiper must be connected to Asterisk.	32/50/0120	
	Entered format of date is wrong	Prompted a total of 3 times after entering wrong date	
IVR0103	Length of Zip code is shorter than expected		Pass
	Zoiper must be connected to Asterisk.	123	
	Prompt again to enter the zip code	Prompted a total of 3 times before skipping the question	

5.3.1.2 EAGI testing

Table 5.2: EAGI testing.

Module No: IVR02		**Date Tested:** 17/01/2020	
Module Name: EAGI			
Case ID	**Test Case Description**		**Result**
	Prerequisites	**Test Data**	
	Expected Results	**Actual Results**	
IVR0201	Recording audio files		Pass
	Zoiper must be connected to Asterisk.	Speech through phone mic	
	Audio file should be save in specified directory on asterisk server	Saved audio file in .wav format in specified directory	

Continuation of Table 5.2			
Case ID	**Test Case Description**		**Result**
	Prerequisites	**Test Data**	
	Expected Results	**Actual Results**	
IVR0202	No speech is provided after beep		Fail
	Zoiper must be connected to Asterisk.	None	
	Prompt to speak again	No prompts from the system and system records noise	

5.3.1.3 REST API testing

Table 5.3: REST API testing

Module No: API01		**Date Tested:** 18/01/2020	
Module Name: REST API			
Case ID	**Test Case Description**		**Result**
	Prerequisites	**Test Data**	
	Expected Results	**Actual Results**	
API0101	Invalid API key in the request header		Pass
	IP address of the web server	Authorization: Token abcd123	
	Decline request due to invalid credentials	same as expected result	
API0102	Missing or incorrect source of recordings		Pass
	Valid API key	recordingDir: "1234567891002"	
	Cannot register a case	same as expected result	
API0103	Missing audio file names		Pass
	Valid API key	"audioFiles": ["patientName", "patientAddress", "eventDescription", "susDrugName", "susDrugDosage"]	
	Audio file will not be available but case will be registered	The patient's landmark audio file will not be available to listen to in the portal as it is not in the list of audio files	

5.3.1.4 Subject matter expert role testing

Table 5.4: SME role testing

Module No: POR01		**Date Tested:** 19/01/2020	
Module Name: SME role			
Case ID	**Test Case Description**		**Result**
	Prerequisites	**Test Data**	
	Expected Results	**Actual Results**	
POR0101	New case should be shown in unreviewed cases.		Pass
	Django server should be running	caseId : 71, Date of reaction : 05/05/2019, suspected medicine : crocine, Adverse event : skin rashes	
	New case should be generated and highlighted in unreviewed cases.	Generated and shown new case in unreviewed cases with 'NEW' tag.	
POR0102	Review case and forward to HC		Pass
	Django server should be running	caseID:71, contact:123 and 26 more parameters	
	Case details should be saved and allocating case to suitable health center	Allocationg case to HC and saved case details.	
POR0103	Adding new HC		Pass
	Django should be running	Name : ABC Hospital, Adrress : Thane 400602	
	New Hc should be added	Same as expected result	
POR0104	Edit HC		Pass
	Django should be running	Name : ABC Hospital, open time : 08 AM, close time : 11 PM, Adrress : Thane 400602	
	HC details should be updated	Same as expected result	
POR0105	Delete HC		Pass
	Django should be running	caseId : 71	
	HC should be deleted	Same as expected result	

5.3.1.5 HC role testing

Table 5.5: HC role testing

Module No: POR02		**Date Tested:** 20/01/2020	
Module Name: HC role			
Case ID	**Test Case Description**		**Result**
	Prerequisites	**Test Data**	
	Expected Results	**Actual Results**	
POR0201	New case should be shown in unreviewed cases.		Pass
	Django server should be running	Save and review case:71 in SME role	
	New case should be generated and highlighted in unreviewed cases.	Generated and shown new case in unreviewed cases with 'NEW' tag.	
POR0202	Review case and forward to Admin		Pass
	Django server should be running	caseID:71, contact:123 and 26 more parameters	
	Case details should be saved and notify to admin	Sav	

5.3.2 Integration testing

In this phase of testing, individual modules as shown in unit testing are combined and tested as a group. This helps us to expose defects in the interfaces and in the interactions between integrated components or systems. We have followed *Big Bang* method where all or most of the units are combined together and tested at one go.

Table 5.6: Integration testing

Date Tested: 21/01/2020 - 10/02/2020			
Case ID	**Test Case Description**		**Result**
	Prerequisites	**Test Data**	
	Expected Results	**Actual Results**	
INT01	On call completion new case should be shown to SME		Pass
	Django server should be running and SSH key should be exchanged	Responding to complete call	
	New case should be generated and highlighted in unreviewed cases	New case generated and highlighted in unreviewed cases with 'NEW' tag in purple color and blinking	

Continuation of Table 5.6			
Case ID	**Test Case Description**		**Result**
	Prerequisites	**Test Data**	
	Expected Results	**Actual Results**	
INI02	Notifying non completed call's to SME		Fail
	Django server should be running and SSH key should be exchanged	Disconnect call before completing	
	SME should get notification	Not showing any notification	
INT03	On SME case review it should be assign to HC		Pass
	Django should be running	caseId:71	
	Allocating case to suitable health center	Allocating case to HC and showing in reviewed case's	
INI04	Login validation with correct parameters		Pass
	Django should be running	userId : SME123456789, password : Root@123	
	Should be login and redirect to SME dashboard	Same as expected result	
INI05	Login validation with incorrect parameters		Pass
	Django should be running	userId : SME123456789, password : 123	
	Should not be login and redirect to login page	Same as expected result	
INI06	Logout		Pass
	Django should be running	Click on logout	
	Should be logout and redirect to login page	Same as expected result	
INT07	PDF generation in AMC format		Pass
	At least one case should be registered	Click on download pdf	
	Pdf should get downloaded according to patients details	Same as expected result	

5.3.3 User acceptance testing

The project was demonstrated to the leadership of the company and then given to them for evaluation of the system. Upon using the system, they found it easy and intuitive to use. A few suggestions were provided by them on a few issues in the system. Most of them could be easily addressed in a matter of days. The only request that could not be completed was to add a facility to end a voice recording by pressing a button such as "#" on the dial pad. This is because Asterisk does not allow us to run two AGI commands in parallel which was required to implement this feature.

Chapter 6

Conclusion & Future Scope

6.1 Conclusion

The Government of India has taken positive steps towards improving drug quality with the introduction if the Pharmacovigilance Program of India (PvPI) under the Indian Pharmacopoeia Commission (IPC). But even after 9 years since the introduction of PvPI in 2010, the number of reported reactions is very low for a country as populous as India. A project like ours aims at improving the awareness of ADR reporting amongst the general public using modern technologies such as Machine Learning. Using simple interfaces such as a phone that can place a call, we aim at making this system easily accessible to a larger population, as a result increasing the number of reported cases and help improve the quality of drugs.

6.2 Future Scope

With this project we have really just scratched the surface. A system like this most definitely has more than what we can offer in the limited time frame. We present some of the improvements and additions that can be made to make this system to make it a lot more helpful.

6.2.1 Help for those affected

Once the report of a patient has been registered, medical help could be sent to the patient's address. Additionally, if the case is severe, a prior appointment can be set up at the nearest emergency room making sure the patient is attended to with minimal delay.

6.2.2 Feedback

People that are interested in knowing the current status of the report and if they wish to provide additional information pertaining to the event could be allowed to do so by introducing a mobile application or a web application.

6.2.3 Additional sources of help

In addition to health centres such as hospitals, clinics, and pathology labs, there could also be communities that are built containing volunteers that can provide first aid for the people affected before medical help can reach them. These communities can be as small as a housing complex.

Once a report has been created volunteers can be notified of the location of the affected person as well as the ailment they are facing and provide first aid accordingly.

6.2.4 Support additional languages

Although English is spoken by a lot of people, to be able to reach a much larger population of people, we must support multiple languages such as Hindi, Tamil, etc.

References

[1] P. N. Amale, D. SA, N. YD, and A. NA, "Pharmacovigilance process in India: An overview," *Journal of Pharmacovigilance*, 2018.

[2] "Guidance document for spontaneous adverse drug reaction reporting," Ghaziabad, India, 2014.

[3] M. Sami, "Software Development Life Cycle Models and Methodologies." `https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/`. Accessed: 10-04-2020.

[4] V. Kalaiselvan, T. Prasad, A. Bisht, S. Singh, and G. N. Singh, "Adverse drug reactions reporting culture in Pharmacovigilance Programme of India." `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4277146/`, 2014. Accessed: 20-10-2019.

[5] M. Schurer, "Spontaneous Reporting." `https://globalpharmacovigilance.tghn.org/articles/spontaneous-reporting/`, 2019. Accessed: 20-10-2019.

[6] J. Bukic, D. Rusic, P. Mas, D. Karabatic, J. Bozic, A. S. Perisin, D. Leskur, D. Krnic, S. Tomic, and D. Modun, "Analysis of spontaneous reporting of suspected adverse drug reactions for non-analgesic over-the-counter drugs from 2008 to 2017," *BMC Pharmacology and Toxicology*, 2019.

[7] S. Geraghty, "What is IVR and 6 benefits of Using One in Your Call Center." `https://www.talkdesk.com/blog/what-is-an-ivr-and-6-benefits-of-using-one/`. Accessed: 10-04-2020.

[8] "Asterisk Project Wiki." Accessed: 10-04-2020.

[9] "Speech-to-Text Basics." `https://cloud.google.com/speech-to-text/docs/basics`. Accessed: 11-04-2020.

[10] S. Borkar and S. Pillai, "A review on design and implementation of IVR system using Asterisk," *International Journal of Advances in Electronics and Computer Science*, 2016.

[11] M. M. Rahman and N. S. Islam, "VoIP implementation using Asterisk PBX," *IOSR Journal of Business and Management*, 2014.

[12] S. Goel and M. Bhattacharya, "Speech based dialog query system over Asterisk PBX server," *ICSPS 2010 - Proceedings of the 2010 2nd International Conference on Signal Processing Systems*, vol. 3, 07 2010.

[13] "HTK Speech Recognition Toolkit." http://htk.eng.cam.ac.uk/. Accessed: 30-03-2020.

[14] K. K. Appari, "Customized IVR implementation using VoiceXML on SIP (VoIP) communication platform," *International Journal of Modern Engineering Research (IJMER)*, vol. 2, pp. 4239–4243, 2012.

[15] "Voice Extensible Markup Language (VoiceXML) Version 2.0." https://www.w3.org/TR/voicexml20/. Accessed: 31-03-2020.

[16] S. Mukherjee, S. Sen, V. Kalaiselvan, and S. Tripathi, "Consumer reporting of adverse drug reactions: A current perspective," *International Journal of Green Pharmacy*, vol. International Journal of Green Pharmacy • Jul-Sep 2016 • 10 (3) — 136, 09 2016.

[17] "Google Maps Platform Pricing." https://cloud.google.com/maps-platform/pricing. Accessed: 10-04-2020.

Acknowledgement

Success of a project like this involving high technical expertise, patience and massive support of guides, is possible when team members work together. We take this opportunity to express our gratitude to those who have been instrumental in the successful completion of this project. We would like to show our appreciation to **Ms. Dakshayani** for their tremendous support and help, without them this project would have reached nowhere. We would also like to thank our project coordinator **Mrs. Rakhi Kalantri** for providing us with regular inputs about documentation and project timeline. A big thanks to our HOD **Dr. Lata Ragha** for all the encouragement given to our team. We would also like to thank our principal, **Dr. S. M. Khot**, and our college, **Fr. C. Rodrigues Institute of Technology, Vashi**, for giving us the opportunity and the environment to learn and grow.

Project Group Members:

1. Ryan Noronha, 101636

2. Ankesh Gaikwad, 101668

Appendix A

Timeline Chart

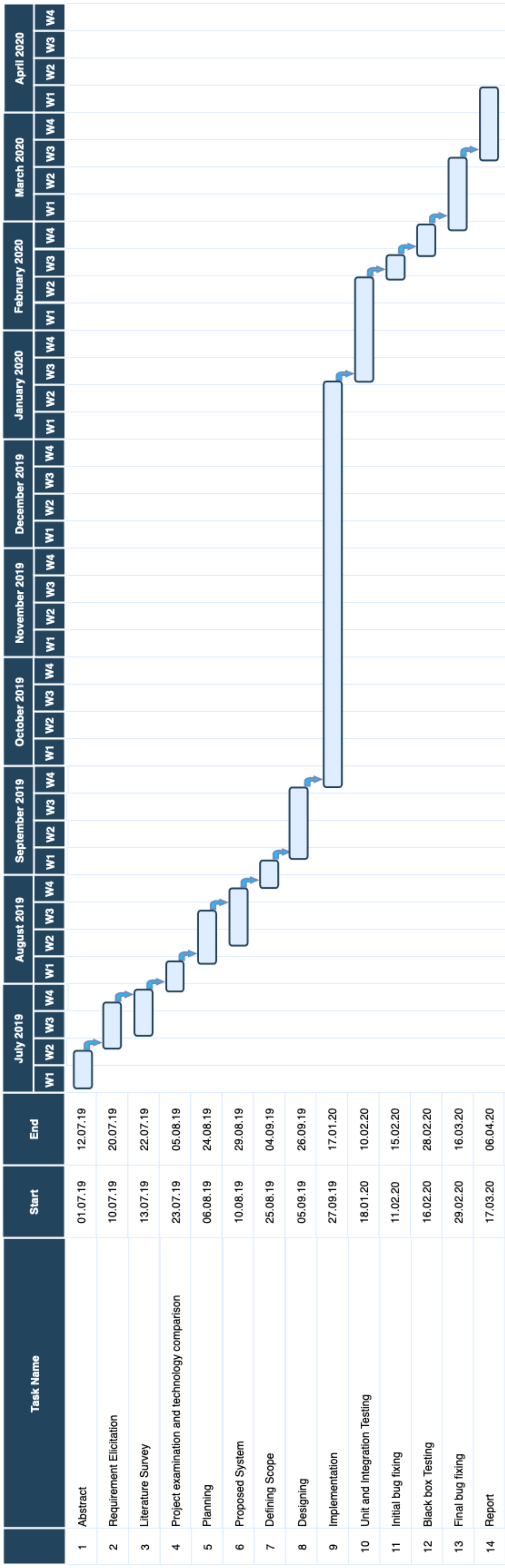


Figure A.1: Timeline chart of ADH System