A PROJECT REPORT

ON

"MEDICATION ADHERENCE MONITORING WITH TRACKING AUTOMATION & EMERGENCY ASSISTANCE"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

Shaikh Shadab Ali Murad Ali Minatunnisa	**17DCO74**
Kazi Obaid Abdul Aziz Zaibunnisa	**17DCO69**
Ansari Mohd Adnan Azimuddin Shehnaz	**17DCO63**
Shaikh Romaan Usman Shaheen	**16DCO77**

UNDER THE GUIDANCE OF
PROF. Irfan Jamkhandikar

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 & 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2019-2020

AFFILIATED TO

UNIVERSITY OF MUMBAI

A PROJECT II REPORT
ON

"MEDICATION ADHERENCE MONITORING WITH TRACKING AUTOMATION & EMERGENCY ASSISTANCE"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN COMPUTER ENGINEERING

BY

Shaikh Shadab Ali Murad Ali Minatunisa	**17DCO74**
Kazi Obaid Abdul Aziz Zaibunnisa	**17DCO69**
Ansari Mohd Adnan Azimuddin Shehnaz	**17DCO63**
Shaikh Romaan Usman Shaheen	**16DCO77**

UNDER THE GUIDANCE OF
PROF. Irfan Jamkhandikar

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 & 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2019-2020
AFFILIATED TO

UNIVERSITY OF MUMBAI

Anjuman-I-Islam's Kalsekar Technical Campus

Department of Computer Engineering
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 & 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

CERTIFICATE

This is certify that the project entitled

"MEDICATION ADHERENCE MONITORING WITH TRACKING AUTOMATION & EMERGENCY ASSISTANCE"

submitted by

Shaikh Shadab Ali Murad Ali Minatunnisa	**17DCO74**
Kazi Obaid Abdul Aziz Zaibunnisa	**17DCO69**
Ansari Mohd Adnan Azimuddin Shehnaz	**17DCO63**
Shaikh Romaan Usman Shaheen	**16DCO77**

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engineering) at ***Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai*** under the University of MUMBAI. This work is done during year 2019-2020, under our guidance.

Date: / /

(Prof. Irfan Jamkhandikar)
Project Supervisor

(Prof. Kalpana R. Bodke)
Project Coordinator

(Prof. Tabrez Khan)
HOD, Computer Department

DR. ABDUL RAZAK HONNUTAGI
Director

External Examiner

Acknowledgements

We would like to take the opportunity to express our sincere thanks to our guide **Prof. Irfan Jamkhandikar**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout our project research work. Without his kind guidance & support this was not possible.

We are grateful to him for his timely feedback which helped us track and schedule the process effectively. His time, ideas and encouragement that he gave has helped us to complete our project efficiently.

We would like to express deepest appreciation towards **DR. ABDUL RAZAK HONNUTAGI**, Director, AIKTC, Navi Mumbai, **Prof. Tabrez Khan**, Head of Department of Computer Engineering and **Prof. Kalpana R. Bodke**, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped us directly or indirectly during this course of work.

Shaikh Shadab Ali Murad Ali Minatunissa
Kazi Obaid Abdul Aziz Zaibunnisa
Ansari Mohd Adnan Azimuddin Shehnaaz
Shaikh Romaan Usman Shaheen

Project I Approval for Bachelor of Engineering

This project entitled ***"MEDICATION ADHERENCE MONITORING WITH TRACKING AUTOMATION & EMERGENCY ASSISTANCE"*** by ***Shaikh Shadab Ali Murad Ali Minatunissa, Kazi Obaid Abdul Aziz Zaibunnisa, Ansari Mohd Adnan Azimuddin Shehnaaz, Shaikh Romaan Usman Shaheen*** is approved for the degree of ***Bachelor of Engineering in Department of Computer Engineering.***

Examiners

1. .....
2. .....

Supervisors

1. .....
2. .....

Chairman

.....

Declaration

We declare that this written submission represents 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 declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Shaikh Shadab Ali Murad Ali Minatunisa
17DCO74

Kazi Obaid Abdul Aziz Zaibunnisa
17DCO69

Ansari Mohd Adnan Azimuddin Shehnaz
17DCO63

Shaikh Romaan Usman Shaheen
16DCO77

ABSTRACT

In today‘s busy world, people are not able to track/monitor the medication events of their dear ones who are suffering from diseases appropriately. There are times when a patient “forgets” to have or “doesn‘t take” the prescribed medicine at a given schedule. Such scenarios in medical term is known as Medication Non-adherence (MNA).

The MNA to prescribed treatment is thought to cause at least 100,000 preventable deaths and $100 billion in preventable medical costs per year. The reason for Non-adherence shows 63% for forgetfulness.[5] A study kempegowda institute of medical sciences & research center, Bangalore, India depicts over 21% of MNA problem in hypertensive patients.[9] The technology that could help in improving the MNA related problem is over 28% combining phone call (10%), live chat (3%), SMS (9%), Mobile applications (5-8%).[11]

Despite, The medical profession largely ignores MNA or sees it as a patient problem and not a physician or health system problem.[5] This is a great loss to society considering the health effects and causes being generated from this issue.

This report highlights various mechanisms that can be thought to integrate in order to improve the adherence of a patient.[1] Unlike already existing applications, Our Application (MAMTE) will guide a patient through automated calls, maintenance & tracking of log of events, & will use state of the art technologies such as Google Cloud Vision A.I and Deep learning neural networks for text and handwriting detection-recognition and extracting medication events through images.

Keywords: Medication Non-Adherence, handwriting recognition, google cloud vision A.I, deep learning neural networks, medication events tracker.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Medication adherence (M.A) often refers to as whether a person suffering from diseases have taken their prescribed medicines on time. Not doing so (MNA) will lead to adverse results and expensive medication care as the medical condition could be much bad. While presenting an appropriate solution, let us first take a look that what are the reasons of non-adherence to medication events. What are the drastic concerning parameters that leads a patient to follow its prescribed medication by the respective doctors? World Health Organization conveys reason of non-adherence to medication events can be grouped in 5 dimensions.[8]

Figure 1.1: 5 Dimensions of Non-adherence. (Source adapted from [8])

1.1 Purpose

Given a doctor's prescription and/or providing appropriate medication details, providing with such a platform that can automate the notification process of scheduled medication events reminder through automated calls, maintain a transparency record of events to the respective parental authority and also keeping track of the patient medical history with appropriate analysis and emergency assistance like speed computation, diet recommendation, current location finder & contacting a doctor.

1.2 Project Scope

Our strategy to tackle this MNA related issue is to develop Web and mobile application that can be useful for three main roles those are doctors, patients & the mentor. As soon as the medication events are available in the system, all of these roles will get proper facilities & notifications in a timely manner. Also, Our application would assist emergency assistance facilities with proper scheduling and tracking mechanisms via the Geo-location, Google Cloud Vision API's.

1.3 Project Goals and Objectives

1.3.1 Goals

- To ensure that effective measures has been taken for the MNA related issues being tackled properly, irrespective of the cases such as intentional skipping of prescribed dosage of medicines or unintentionally skipping it.
- To increase the healthy life style of a patient, doctor, and the mentor by representing quick and reliable forms of authentic data related to the medication activities.

1.3.2 Objectives

- To make use of the available state of the art technologies to adequately suit the end users and target them with utmost flexibility.
- To facilitate a user in the time of need by giving features such as timely notification reminders, tracking through log of events, computing travelling histories and current speed ratio.
- To ensure the reliability of the data, and provide a transparent ecosystem among all the stake holders.

1.4 Organization of Report

This report is organized as follows: The introduction is given in Chapter 1. It describes the fundamental key aspects used in this project. It motivates to study and understand the different techniques used in this work. This chapter also presents the outline of the objective of the report.
The Chapter 2 describes the review of the relevant various techniques in the literature survey. It describes the pros and cons of each technique.
The Chapter 3 presents the Project planning. It describes the managerial approaches taken during the project development Also, assumption & constrained with associated roles and responsibilities and formulated schedule, each member capabilities along with the project budget and timeline.
The Chapter 4 presents the Detail Theory. It describes the product in detail through software requirement specification.
The Chapter 5 talks about the System Design. It focuses on technical and non technical requirements of the product encompassing the proposed system.
The societal and technical applications are mentioned in Chapter 6.
The Chapter 7 follows the system testing. It describes the assurance and reliability of the developed application.
The Chapter 8 depicts the pictorial demonstration of the developed application through necessary screenshots.
The summary of the report is presented in Chapter 9.

Chapter 2

Literature Survey

2.1 A Medication Adherence Monitoring System for People with Dementia

Due to forgetfulness, people with dementia need assistive technologies for managing medications. In this paper we present a Kinect-based smart system for unobtrusive medication adherence monitoring of people with memory-degrading conditions. Unlike existing pill dispensers, our system not only reminds a patient on time of medicine intake and provides the corresponding medication dose, but also vocally guides the patient through the steps of medication intake, controlling correctness and completeness of his actions and alerting the caregiver if problems occur. The experimental evaluation of prototype system shows that it effectively monitors medication adherence and is very easy to use.[2]

2.1.1 Advantages of Paper

a. The paper gives ample of information needed to build an application that could suit the patient suffering from a disease called as dementia.

b. The techniques mentioned are innovative and can help the illiterate people as well through text to speech assistance.

c. Easy to use application.

2.1.2 Disadvantages of Paper

a. Maintenance of the application may be dramatically costly.

b. Not suitable for the patients having a eye-sight problem.

c. Is still a prototype and not a stable application.

2.1.3 How to overcome the problems mentioned in Paper

a. More sophisticated technologies can be integrated so that the scalability of the application is increased.

b. Open source technologies can be used in order to reduce the maintenance cost.

2.2 The Unmet Challenge of Medication Non-adherence

Medication non adherence for patients with chronic diseases is extremely common, affecting as many as 40% to 50% of patients who are prescribed medications for management of chronic conditions such as diabetes or hypertension. This non adherence to prescribed treatment is thought to cause at least 100,000 preventable deaths and $100 billion in preventable medical costs per year. Despite this, the medical profession largely ignores medication non adherence or sees it as a patient problem and not a physician or health system problem. Much of the literature on non adherence focuses on barriers to adherence, with the assumption that appropriate adherence is the normal course of events and non adherence is an aberration. This approach minimizes and oversimplifies the problem. It is not easy for humans to change their behavior, even for what many physicians see as a minor change such as taking prescription medications. Improving medication adherence has not been well studied, but a Cochran review shows that multi-factorial interventions are more effective. In at least one integrated health care system, Kaiser Permanent Northern California, a combination of approaches centered on the electronic health record has improved medication adherence rates to above 80%. Using similar elements would be feasible in other health care systems but would require motivation and planning. Effective change will not happen until key players decide to take on this challenge and reimbursement systems are changed to reward health systems that improve medication adherence and chronic disease control[5]

2.2.1 Advantages of Paper

a. Gives in detail analysis for the potential barriers causing the non-adherence.

b. Showcases real statistics and causing effects of the MNA that are still not yet fulfilled and tackled properly.

2.2.2 Disadvantages of Paper

a. This paper discusses a lot about MNA, but fails to grasp the end - user environment practically.

b. The applications discussed in this paper, would be only applicable to hospitals that too if given a good investment infrastructure.

2.2.3 How to overcome the problems mentioned in Paper

a. The application proposed, can be developed on a smaller scale in order for the end user adaption.

b. Different state of the art technologies can be used for proposing a relevant application.

2.3 Gamification of Medication Adherence in Epilepsy

Adherence to medication regimens is a crucial factor in seizure-freedom and well-being for people with epilepsy. In contrast, taking medication inconsistently increases the risk of not only seizures and their adverse effects, but drug side-effects and unnecessary modifications to treatment plans. Epilepsy is prevalent across all age groups and we have been slow to utilise both the technologies and psychologies derived from computer gaming. Gaming has broken through to the mainstream and is no longer the preserve of younger males, mirroring the adoption of smart-phones. 'Gamification' motivates users into engaging in an activity with a higher intensity and duration. Introducing gaming elements into a non gaming context has the potential to transform routine tasks into more enjoyable and motivating experiences. This has been exploited by marketing executives, but also has clear uses in a healthcare setting too. We discuss how previously published frameworks could be employed to help people with epilepsy adhere to medication regimens to create a patient-focused, modifiable and fun experience.[2]

2.3.1 Advantages of Paper

a. Talks about a crucial aspect that can gain user attractions quite easily.

b. Theoretically, mentioned adequate steps necessary for developing a successful application.

2.3.2 Disadvantages of Paper

a. Addiction : It makes the people addicted to mobile devices, which may lead to harmful effects on human eyes or swelling as the light rays gets radiated from the devices. Lack of parental authority indulgences.

b. No practical proposed system available.

2.3.3 How to overcome the problems mentioned in Paper

a. Appropriate parental mechanisms can be implemented with proper scheduling & tracking facilities.

2.4 Emerging IT for Medication Adherence

Medication adherence is one of the most complex and persistent challenges in healthcare, but emerging IT can play a major role in improving it. The authors discuss five major technologies for medication adherence. [1]

2.4.1 Advantages of Paper

a. Gives five major ways on how an Anti-MNA sort of application can be developed. including hardware solutions and software solutions.

b. The proposed system ensures a robust environment.

2.4.2 Disadvantages of Paper

a. Will require the need of the core technical team for making such proposed system to work.

b. The grasp and adaptation of the application might require a bit of training infra as well.

c. Hardware indulgence will lead to costly setup.

2.4.3 How to overcome the problems mentioned in Paper

a. Can accommodate by focusing on only one solution that are in trend keeping the cost factor as well and portability aspect.

b. Ensuring that the end product is user friendly.

2.5 EAST: An Efficient and Accurate Scene Text Detector

Previous approaches for scene text detection have already achieved promising performances across various benchmarks. However, they usually fall short when dealing with challenging scenarios, even when equipped with deep neural network models, because the overall performance is determined by the interplay of multiple stages and components in the pipelines. In this work, we propose a simple yet powerful pipeline that yields fast and accurate text detection in natural scenes. The pipeline directly predicts words or text lines of arbitrary orientations and quadrilateral shapes in full images, eliminating unnecessary intermediate steps (e.g., candidate aggregation and word partitioning), with a single neural network. The simplicity of our pipeline allows concentrating efforts on designing loss functions and neural network architecture. Experiments on standard datasets including ICDAR 2015, COCO-Text

and MSRA-TD500 demonstrate that the proposed algorithm significantly outperforms state-of-the-art methods in terms of both accuracy and efficiency. On the ICDAR 2015 dataset, the proposed algorithm achieves an F-score of 0.7820 at 13.2fps at 720p resolution.[5]

2.5.1 Advantages of Paper

a. Gives accurate text detection.

b. Works on the real life use cases and noise full real scenes.

c. Gives detail about the text detection algorithms, and proposed system.

2.5.2 Disadvantages of Paper

a. Not suitable for text placed in a tilted fashion.

b. Only encompasses detection & not recognition.

c. The real life application will require an adequate front end to be built for the end user adaptations.

2.5.3 How to overcome the problems mentioned in Paper

a. Training the model with more use cases for the tilted images and tweaking an algorithm a bit.

b. Can encompass recognition algorithm as well to better suit each text.

2.6 Technical Review

Each techniques mentioned have had their pros and cons. After analysing, it is to be noted nobody has ever thought to integrate different different advantages to make a full-fledged reliable application. Which can also ensure the portability, adaptability aspects. We found there are still some features that can be introduced further more for the betterment of the MNA related applications. Those introductory key features could be as follows

- Medication Reminders (Automated calls, SMS, App notification).
- Medication event scheduling through NLP.
- Medication events live tracking with effective monitoring.
- Emergency Assistance using mobile native sensors.

Features	MAMTE App	Remote patient monitoring (RPM)	MyTherapy	Medisafe
Medication Reminders (Automated calls, SMS, App notification).	✓	✗	✗	✗
Medication event scheduling through NLP.	✓	✗	✗	✗
Medication events live tracking with effective monitoring	✓	✗	✓	✗
Patient Location live tracking & Alert with diet recommender.	✓	✗	✗	✗
TTS sys assisting illiteracy & dementia, Alzheimer patient	✓	✗	✗	✗
Report Inventory & Accurate Analytics	✓	✗	✓	✓
Instant messaging with respective doctor.	✓	✗	✗	✗
Emergency Assistance using mobile native sensors.	✓	✗	✗	✗
Gamification based Reward System.	✓	✗	✗	✗

Table 2.1: Competitive Analysis

2.6.1 Advantages of Technology

a. Ensuring broader audience and end user ratio.

b. To make the system more reliable and capable enough to bear the real life loads.

c. Maintenance of effective log usages.

d. Quick and easy way to get the glimpse of the sample available data.

2.6.2 Reasons to use this Technology

a. Google Cloud Vision AI - Fast, accurate, Tested reliable platform to recognize text both computer generated and handwritten forms. So to facilitate a user for quick interaction through the system input.

b. Nexmo API with SMS Bulk Services - To ensure the reminder are not only bounded with any smartphone users. Each end user will be able to get reminder.

c. AWS DynamoDB - For live cloud storage facility and immediate fetching of records.

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Shadab Shaikh	Backend, Front end
2	Obaid Kazi	Backend, Front end
3	Mohd Adnan Ansari	Front end
4	Shaikh Romaan	UI & UX Designs

3.2 Work Breakdown Structure

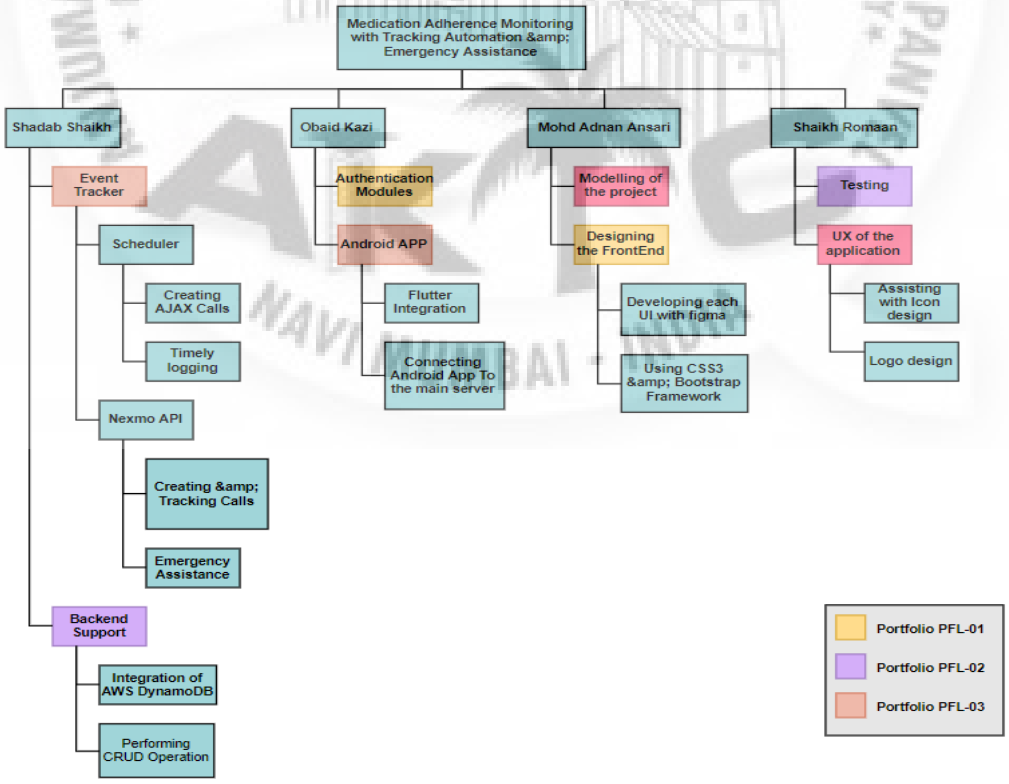

Figure 3.1: Work Break Down structure for each members.

3.3 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Shadab Shaikh	Team Leader	Handling the Backend Logic
2	Obaid Kazi	Backend Developer	Interacting front end with the backend
3	Mohd Adnan Ansari	Front End Developer	Creating UI for the application
4	Shaikh Romaan	UX & Software Tester	Assisting in Icon design, Validation

3.4 Assumptions and Constraints

It has been assumed that due to the cheapness in the android smartphone industry, almost about 85%+ users will be having a device with at least a camera, mic, and that can bear an active internet connection. Still for the users availing Non-Smartphone based facility can accommodate with the sms & calling reminder services. The application mainly concerns with the senior citizens and the immature kids their ages has been presumed as above the age of 50 years and below the age 15.
The project modules encompassing the text detection, database integration, integrating sms & calling api's, generating log of events, depicting data in an extensive manner through various charts should be achieved and developed within first month of the development phases. Various technologies have certain pricing policies so in order to run freeware the limitations to the api request has to be ensured.

3.5 Project Management Approach

This project follows the scrumm based agile methodology. Each key modules have been allocated to respective members with a particular time frame. For an instance, if for any reason that module isn't completed on time it would be then put forward to Backlogs. Meanwhile the task would be carried out in sprints each sprint reflects a particular module. As and when a sprint is into its development phase it would be then put forward to In progress Lane, followed with Testing Lane, Code review lane, Done lane and the final deployment lane. Before starting the current sprint if there are any sprint in the backlog lanes the module will be allocated with appropriate time frames. For facilitating with such requirement and methodology, Trello a web-based Kanban-style list-making application which is a subsidiary of Atlassian to keep track of everything, from the big picture to the minute details online tool has been used. Once a particular sprint is gone through each of the lane and is currently under the deployment lane, the next sprint comes in following the iterative approach of waterfall model after each interval of execution as and when necessary to accommodate changes if any for the betterment and smooth conduct of the project phases.

3.6 Ground Rules for the Project

- The project has to be developed using the scrumm based agile methodologies meaning each of the members will know at any moment of time what is being carried out at current time frame.
- To maximize the throughput, the members having specialization of a particular domain will be responsible and would be categorized broadly in the backend, frontend & the validation phases.
- For any exceptional situations, a proper form of communication has to be submitted so that necessary delegations can be made.
- For transacting, logging activity separate email id would be used and each of the developed use cases will be made available on the cloud infra.
- The project will be broken down into smaller stubs, and will be developed likewise. After the stub has shown desirable output for any given input, the integration would be done.
- Respective versioning should be maintained via the git management tools.
- During the development, the api request limit should also be respected and the end product should be tested before bringing into the integration phases.

3.7 Project Budget

Table 3.3: Table of computed budget

SR. No	Technologies/Human Resource	Breakdown	Summary
1	Nexmo Voice Calling API	1.64 INR/minute	10 euro free credit
2	SMS Gateway/Services	12 p/sms	100 Demo SMS
3	AWS DynamoDB	80.11 INR/request	1200 INR
4	Google cloud vision A.I API	107.29 INR/50k request	Free 1 year service
5	Time spent by each member	8 hours	240 Hours/Member

3.8 Project Timeline

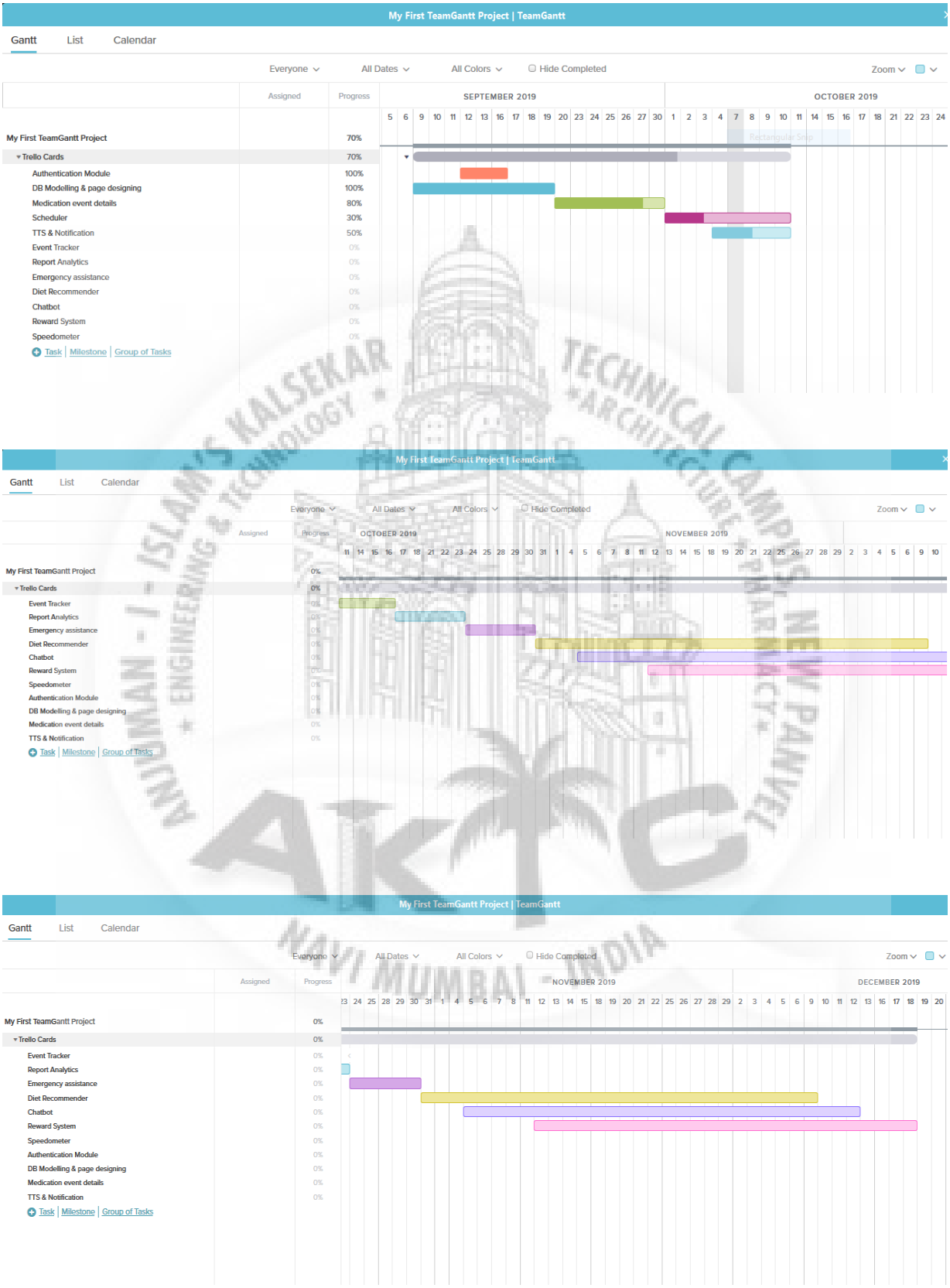

Figure 3.2: Project Timeline describing completion of the project module wise.

Chapter 4

Software Requirements Specification

4.1 Overall Description

4.1.1 Product Perspective

The product is supposed to improve the adherence of the patient. It is a Web and mobile application that can be useful to avail proper facilities & notifications in a timely manner. Project MAMTE would assist emergency assistance facilities with proper scheduling and tracking mechanisms via the Geo-location, Google Cloud Vision API's.

4.1.2 Product Features

Medication Reminders (Automated calls, SMS, App notification), Medication event scheduling through NLP, Medication events live tracking with effective monitoring, Patient Location live tracking & Alert with diet recommender, TTS sys assisting illiteracy & dementia patient, Report Inventory & Accurate Analytic, Emergency Assistance using mobile native sensors.

4.1.3 User Classes and Characteristics

There are three main roles who will be directly using this application those are mentors, patient & doctors which will act as the end users. At the same time there would be an admin panel where the technical team will be assisting the queries for the end users. The mentor will have the authority to create the patient and assign the registered doctors with the patient associated. The doctor will have the facility to avail all the reports of the medication adherence activity of the patient. The patient can trigger the sos alarm system to get the emergency assistance.

4.1.4 Operating Environment

This is a web based application along with mobile application and hence will require the operating environment for a client and server based architectural hierar-

chies. For web application this software highly depends on HTML5 browser support to get the full enriching features. For mobile application, this software relies on the native sensors of the devices such as geolocation and gyroscopic sensors.

4.1.5 Design and Implementation Constraints

The database used in this application are AWS DynamoDB along with sqlite3 and requires the django framework version 2.5 to be in a running condition at the server side. The usage of api's such as Nexmo, Google Cloud Vision, SMS Services have had their own restrictions to the daily hit limit count so that has to be respected.

4.2 System Features

4.2.1 Medication Reminders

Description and Priority

Unlike others, our application also concerns with illiteracy factors & wide open/friendly for such users as well by ensuring reminders through automated dynamic calls, SMS, and system notification with TTS enabled engine. This feature is the crux of this application and has an high priority.

Stimulus/Response Sequences

The mentor gets registered and creates a new patient and associates the medication details of the newly created patient. The system in response delivers reminders through sms, calling, and in-app notification in an accurate and timely manner.

Functional Requirements

- REQ-1 : The system must contain the time of the medication event of the patient.
- REQ-2 : The system must only trigger respective patient phone call and ensure accuracy and transparency.
- REQ-3 : The system should leave the event details of the patient who are not opted for particular time zones.

4.2.2 Medication event scheduling through NLP

Description and Priority

Using our very own deep neural network technologies, and Google Cloud Vision API by means of image processing a user would be able to feed the medicine details just by giving medicines images to our system. This feature has a medium priority.

Stimulus/Response Sequences

The mentor places the medicine image in front of the application and in response the system extracts the medicine name and set to the medicine allocation form field.

Functional Requirements

- REQ-1 : The mentor should have a device with camera enabled facility and an image having the medicine name mentioned on it.
- REQ-2 : If the camera is not available, the manual entry option should be available for the selection of user.
- REQ-3 : In the case of unsuccessful extraction for any exceptional reason, cropping a particular portion of image should be made available in the follow up screen.

4.2.3 Medication events live tracking with effective monitoring

Description and Priority

Users will be able to get daily summarize reports with proper analytical mechanisms. This feature has high priority.

Stimulus/Response Sequences

All of the roles doctors, patient, mentor after logging in request for the patient medication data, in response the system gives a pictorial representation through various charts .

Functional Requirements

- REQ-1 : The patient should have had at least got a notification for once via the system.
- REQ-2 : A seperate logging facility and infra disk space must be available.

4.2.4 Patient Location live tracking & Alert

Description and Priority

With the usage of A.I, Geo-location API's & Deep learning driven technologies a user will be able to track patients speed of travelling, current location, diet plan and would be able to get alert notification through system native sensors. This feature has high priority.

Stimulus/Response Sequences

The patient should login via the mobile application, and allow the necessary permissions as and when prompted, in return as soon as patient changes the lattitude & longitude it get reflects to the system with details like time, place, speed.

Functional Requirements

- REQ-1 : The internet connection should be reliable enough.
- REQ-2 : A device with GIS sensors.
- REQ-3 : Application must not proceed any further if not given necessary permissions.

4.3 External Interface Requirements

4.3.1 User Interfaces

This application has been developed for various resolution standard such as 320px, 360px, 375px, 410px, 425px, 512px, 576px, 768px, 980px, 1024px, 1280px, 1440px, 1680px, 1920px. The UI elements have been responsively developed for different resolution as mentioned. At 1024px or higher resolution the elements have been arranged for high range displays such as computer/laptop system and for 768px or 980px resolution the element have been arranged for medium range displays such as tablets. Lastly, the remaining resolution in that the element have been arranged keeping in mind the low range displays such as smartphones.

4.3.2 Hardware Interfaces

The hardware which is expected to be in the usage of this application are camera devices and the printer. The camera can be in both internal web cam or external web cam form whereas the printer can be the ps/2 or usb/wireless form.

4.3.3 Software Interfaces

Boto3 SDK handles the interface between the AWS dynamoDB server & database with this application. Django framework and python interpreter 3.*+ handles the interface of the sqlite3 databases. Various api have been used such as Google Cloud Vision, Nexmo, SMS API's which have their respective sdks and packages. Whenever user hits a request it is first server by the backend django server and then the data is transferred to the respective sdk's and packages in return the response is fetched in the form of JSON.

4.3.4 Communications Interfaces

For the web application, the https protocol along with ssl security certificate is a must to establish a connection between the client and the server. The usage of HTML5 Browser is expected for satisfactorily user experience. Where as in concern to Mobile application, the mobile initiates an http request to the server and the backend server respond accordingly in JSON form.

4.4 Nonfunctional Requirements

4.4.1 Performance Requirements

The system is intact with a bunch of api request and many variables devices such as smartphones, symbian phones, laptop/computer devices. The system must be capable enough to bear the loads at any given point of time, The system at which the server is being hosted should have ample of processing speed along with compatible clocking hertz to suit the need.

4.4.2 Safety Requirements

While in the registration phase, the qr code is generated. This qr must be handled with utmost care as with the help of this any user would be able to access the system. The logging and pdf report should be showcased only to the respective roles of the users and must not encompass any public references.

4.4.3 Security Requirements

The password credential of the end user should be enclosed and encrypted so that none of the persons such as system admin can access any user credential & perform any malicious conduct. The user has to be given the right to the data and can view the data with any given point of time.

4.4.4 Availability Requirements

Appropriate cloud storage facilities must be used to facilitate the end user for the storage and availability. For any given cases if a user forgets his/her credential appropriate measures must be provided to gain back the available resources to the end user. The data must be available to the cloud storage and can be accessed by the end user as and when required simultaneously, assisting the transparency and accuracy of logging events with respect to chart representations.

Chapter 5

System Design

5.1 System Requirements Definition

The system must facilitate all the end user roles namely doctor, patient & the mentor and serve the purpose of tackling the Medication Non adherence related issues. It should form the basis to reduce the Non adherence ratio of the patient after giving necessary input to the system, it should accommodate such a platform that can automate the notification process of scheduled medication events reminder through automated calls, maintain a transparency record of events to the respective parental authority and also keeping track of the patient medical history with appropriate analysis and emergency assistance like speed computation, diet recommendation, current location finder & contacting a doctor.

5.1.1 Functional requirements

- The system automatically validates and prompt necessary information when a username is not available while registration of a new end user.
- It should generate the QR based upon the sign up input given by the end user and allow logging in with the generated QR.
- The system must first encrypt the password credential before storing into the database in order to fulfill the privacy of the end user.
- The system must facilitate text detection and state of the art technologies for smooth and effective user experience while filling of the forms.
- The logging of events should be maintained as and when the time zones are reached for a particular patient medication event
- The system must dynamically be responsive enough for standard resolutions.
- The system must depict the chart representation whenever requested by the end user.

Use-case Diagram

Group Code: -1
Medication Adherence Monitoring With Tracking Automation & Emergency Assistance

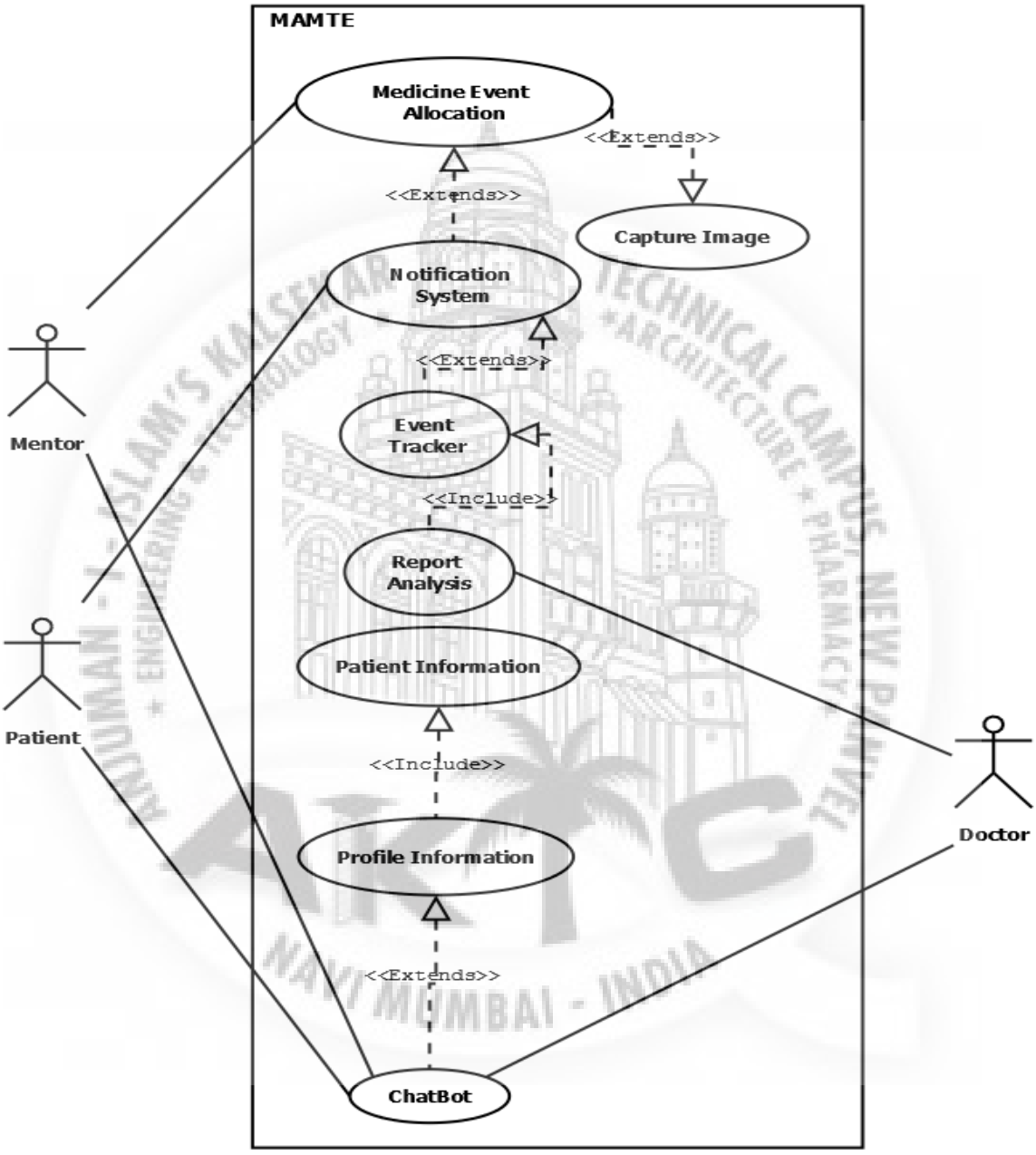

Figure 5.1: Use Case Diagram for MAMTE.

Use case diagram depicting how patient, doctor & mentor are interacting to the system. Also, how several processes are using each other functionalities via include & extends.

Data-flow Diagram

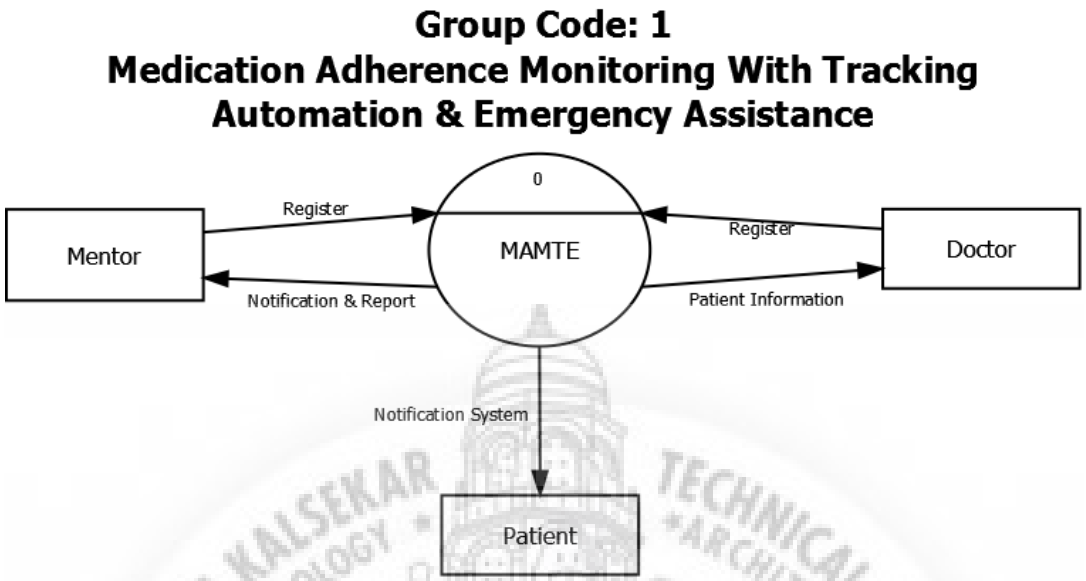

Figure 5.2: DFD Level 0 for MAMTE

DFD Level 0 depicting main outcome of the system, mentor gets the report of a patient, doctor can avail report and have chat with their patient, patient will get reminder.

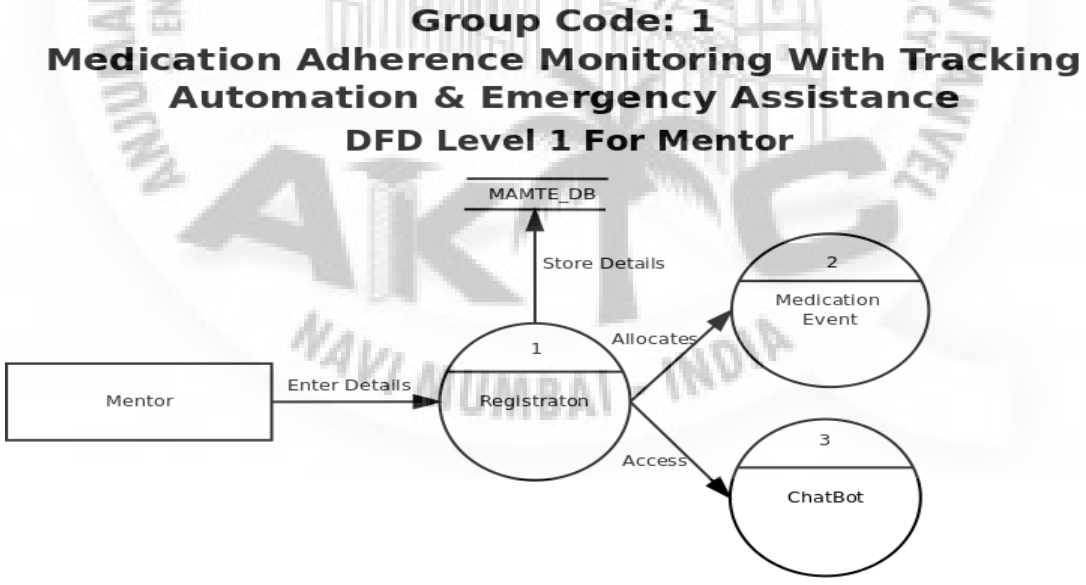

Figure 5.3: DFD Level 1 for Mentor

DFD Level 1 for mentor showing their main process flow in the system via registration details being stored into Database and getting access to chat bot system & medication event system

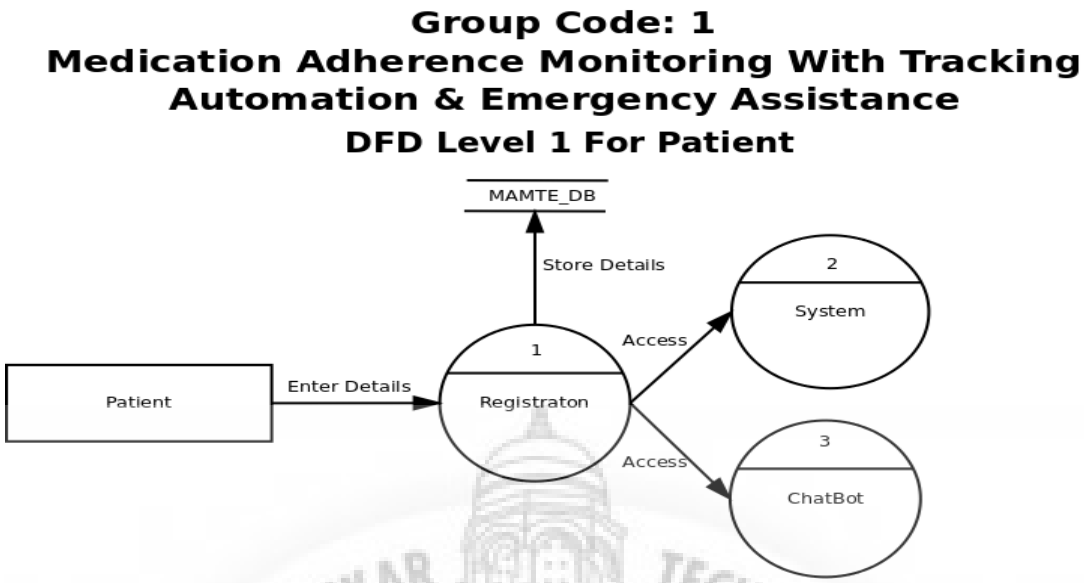

Figure 5.4: DFD Level 1 for Patient

DFD Level 1 for patient showing their main process flow in the system via registration details being stored into Database and getting access to chat bot system & reminder system facilitites.

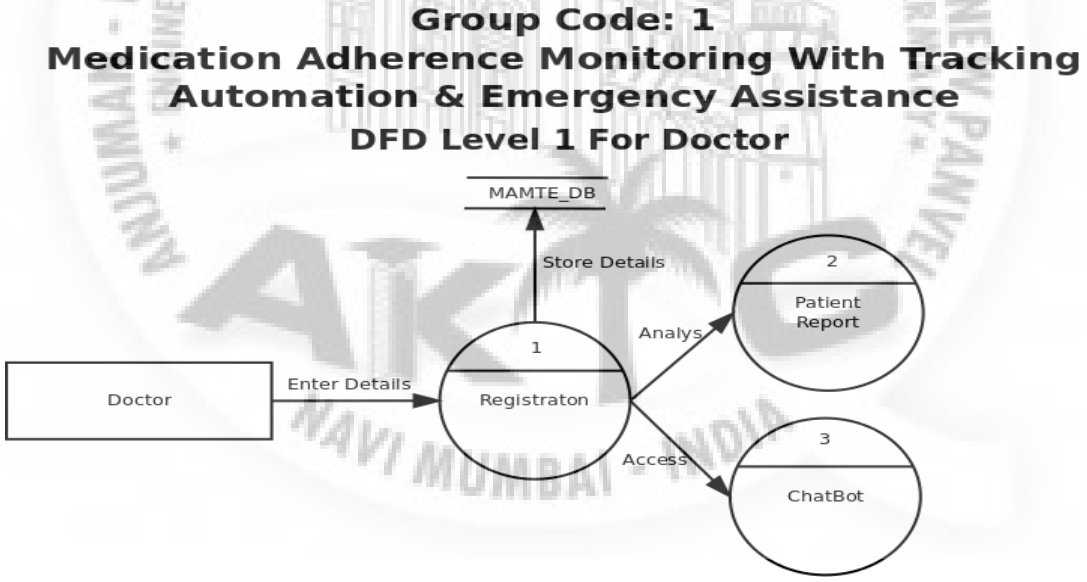

Figure 5.5: DFD Level 1 for Doctor

DFD Level 1 for doctors showing their main process flow in the system via registration details being stored into Database and getting access to chat bot system & patient report facilities.

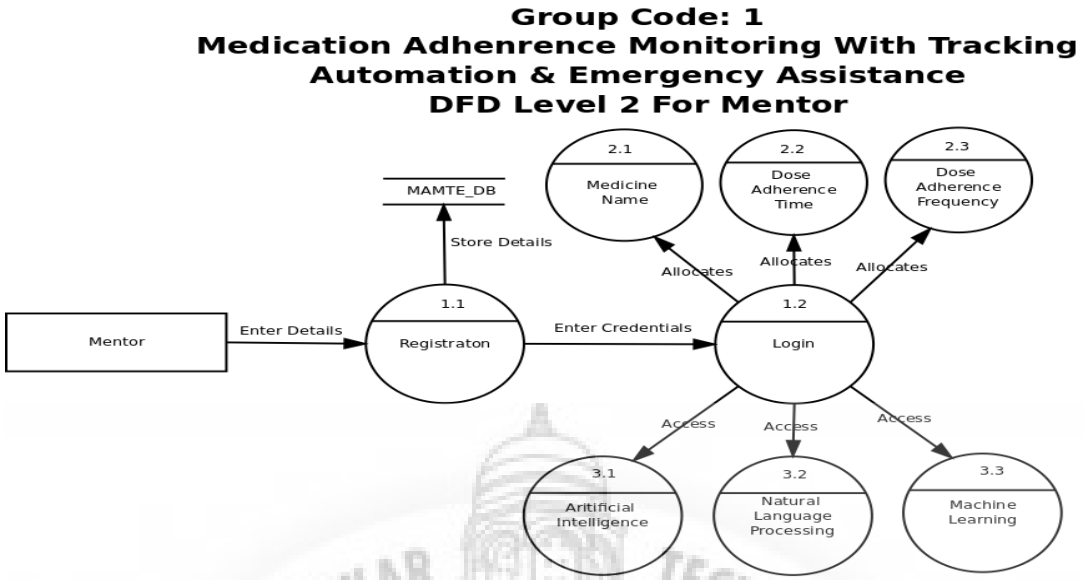

Figure 5.6: DFD Level 2 for Mentor

DFD Level 2 for mentors showing their detail login process flow in the system via authentication modules, entering medication event & chat bot internal mechanisms.

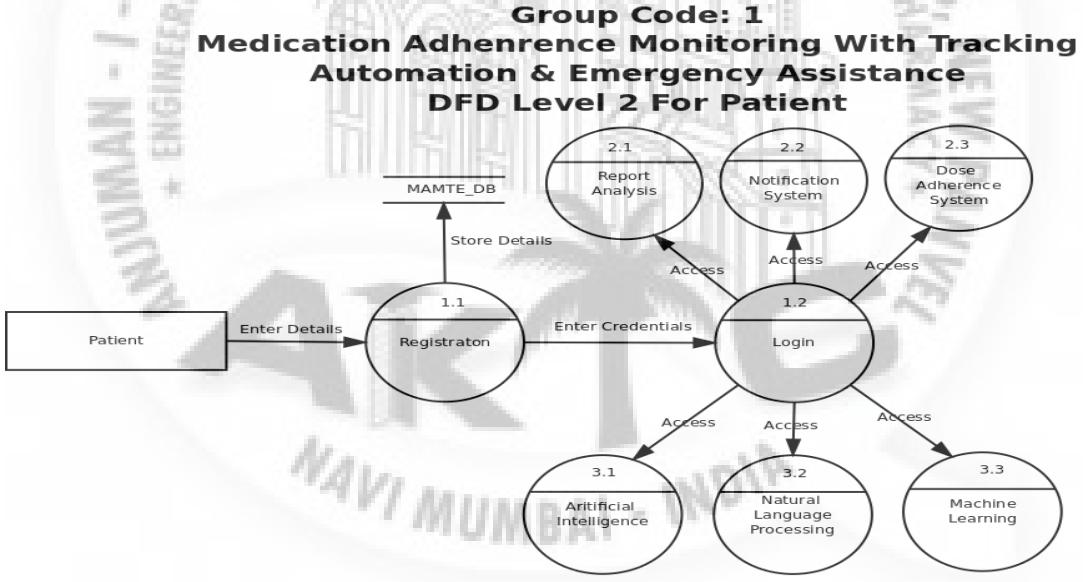

Figure 5.7: DFD Level 2 for Patient

DFD Level 2 for patient showing their detail login process flow in the system via authentication modules, report mechanisms, notification reminders & chat bot internal mechanisms.

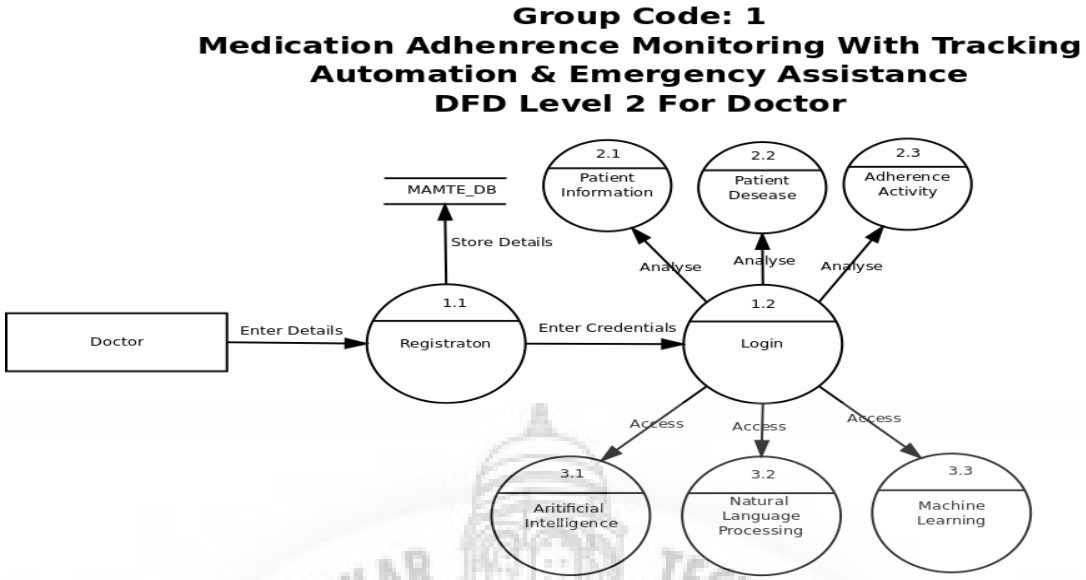

Figure 5.8: DFD Level 2 for Doctor

DFD Level 2 for doctors showing their detail login process flow in the system via authentication modules, getting patient information, adherence activity & chat bot internal mechanisms.

5.1.2 System requirements (non-functional requirements)

- The system is intact with a bunch of api request and many variables devices such as smartphones, symbian phones, laptop/computer devices. The system must be capable enough to bear the loads at any given point of time, The system at which the server is being hosted should have ample of processing speed along with compatible clocking hertz to suit the need.
- While in the registration phase, the qr code is generated. This qr must be handled with utmost care as with the help of this any user would be able to access the system. The logging and pdf report should be showcased only to the respective roles of the users and must not encompass any public references.
- The password credential of the end user should be enclosed and encrypted so that none of the persons such as system admin can access any user credential & perform any malicious conduct. The user has to be given the right to the data and can view the data with any given point of time.
- Appropriate cloud storage facilities must be used to facilitate the end user for the storage and availability. For any given cases if a user forgets his/her credential appropriate measures must be provided to gain back the available resources to the end user. The data must be available to the cloud storage and can be accessed by the end user as and when required simultaneously, assisting the

transparency and accuracy of logging events with respect to chart representations.

- The system must provide appropriate automatic backup procedures for the availability and reliableness of the data and should be located at several locations on an online cloud servers. For an exceptional cases, such as failure, database hit & miss write ratio the recovery measures must be implemented.

Algorithm for the detection & recognition of text from a sample images.

1. Converting the image into base64 format and sending it to the backend server.
2. Storing and converting back the base64 format into png format images using ndmap array concept.
3. Reading the newly stored image, Grayscaling the image, creating a bounding boxes around the regions and slicing them.
4. Using the google cloud vision AI text detection api, receving the response through JSON format
5. Comparing each recognized text with the name available into the database.
6. If any match found going to step 7, else allowing the user to crop the region of image and going back to step 1
7. returning the data in the JSON format.

Database Schema/ E-R Diagram

5.2 System Architecture Design

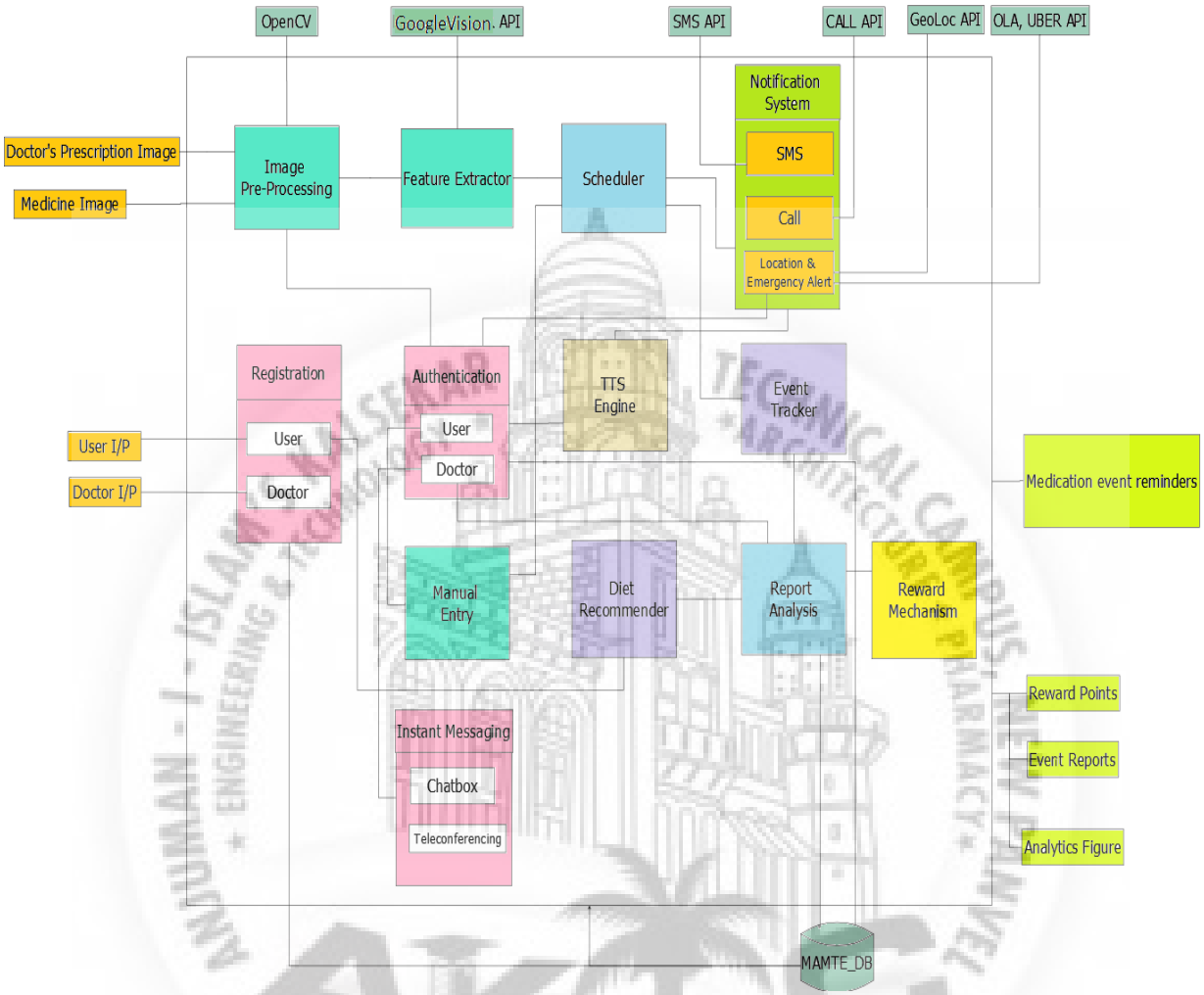

Figure 5.9: System architecture for MAMTE

The system architecture reflects the each and every module present in the application in detail. The input to the system has been placed to the left side of the system boundary. Various API & library are available at the top, stubs at the bottom and output can be seen at the extreme right side. The elements present inside the system boundaries are the key modules which act as an input to the other inter-coupling.

5.3 Sub-system Development

The system broadly contains module such as Image extraction & medication event input unit, Registration & authentication module unit, Notification system unit, Event tracker unit, Report analysis and the output unit.

5.3.1 Module 1 - Registration & authentication

A user gets enroll to the system via a quick sign up, after profile completion and validity of a genuine user a user would also get a qrcode generated with the username & password enclosed. A user then enters the details, the passwords are stored in sha256 encrypted hash with 12000 rounds and 32 saltsize ensuring the security aspects. With the help of QR generated or username & password a user will be able to gain access to the system. The user roles are divided into three section namely doctors, mentor acting as a guardian for the patient and the patient who is currently suffering from a disease.

Module 1 Block Flow Diagram

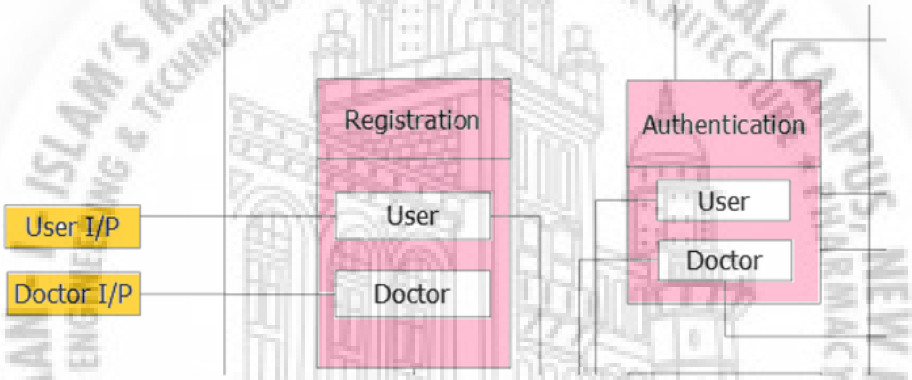

Figure 5.10: Registration & authentication module

5.3.2 Module 2 - Image pre-processing, feature extraction & manual entry

For the betterment of the user and due to the advancement in technology, a user need not be made to fill a tedious forms, application provides a take on snap feature of the medicine & prescription given by authorized doctors. Our system will detect & recognize text to extract necessary medication event details using Deep learning custom build neural network model & A.I technology driven Google Cloud Vision, For this the Pytesseract OCR along with OpenCV library has been used. For a noisy case, the manual entry with gamification UX is also present. The crucial challenge arises over here is to predict the handwriting recognition and that too of a professional medical experts. The accuracy of the google cloud vision model is there for the rescue resulting into above 85% accuracy given a noisy environment and a 99% confidence ratio given a manageable predefined fonts. The two basic details which are gonna target and the system will try to recognize are medicine name, and dosage time. The input for this module would be the image samples of the prescription & medicines and the output will be carry forwarded by the scheduler.

Module 2 Block Flow Diagram

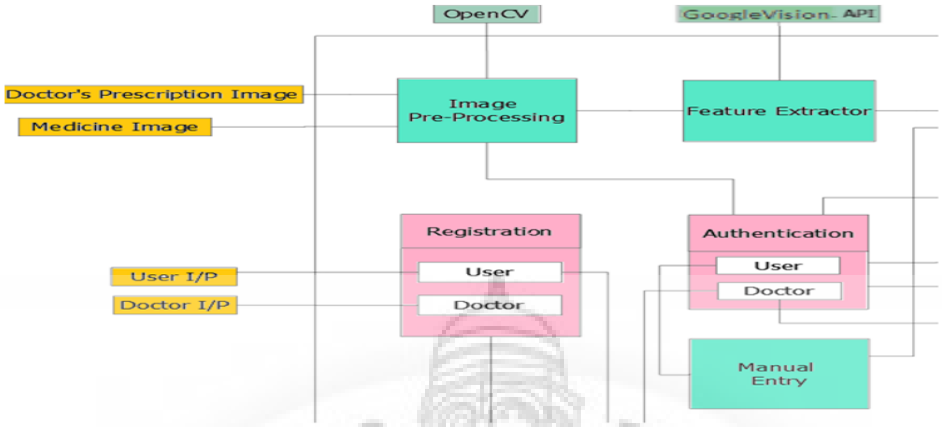

Figure 5.11: Image pre-processing, feature extraction & manual entry

5.3.3 Module 3 - Scheduler

After the successful medication event details extraction through various mechanisms, the system will take care of scheduling and handle the time management process and will act as a medium for triggering the event. For this either the threading mechanism can be thought of implementation or else the micro services. The input of this module will be Extraction module and the output of this module will be Event Tracker and also Notification & alert system. Scheduling a live application considering the stake of a patient health is not an easy task, the server will have to take care about the time zones, their medication times, and dispatch the request in an efficient manner. For this Django services in collaboration with ajax job-queue scheduler are integrated.

Module 3 Block Flow Diagram

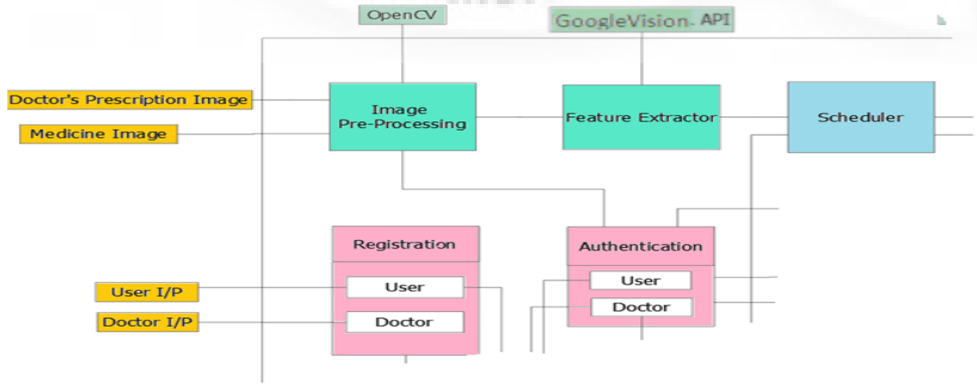

Figure 5.12: Scheduler

5.3.4 Module 4 - Event Tracker

This would be having all the logs of events for the patient medication adherence activity day wise and accordingly reward mechanism will be determined. This would act as a base for timely report generation of a patient activity and can help in the productivity of the sustainable growth of the application. For dumping of such huge amount of log involves high response time and a highly secure database server so for that amazon AWS DynamoDB will be serving to tackle the response related aspects.

Module 4 Block Flow Diagram

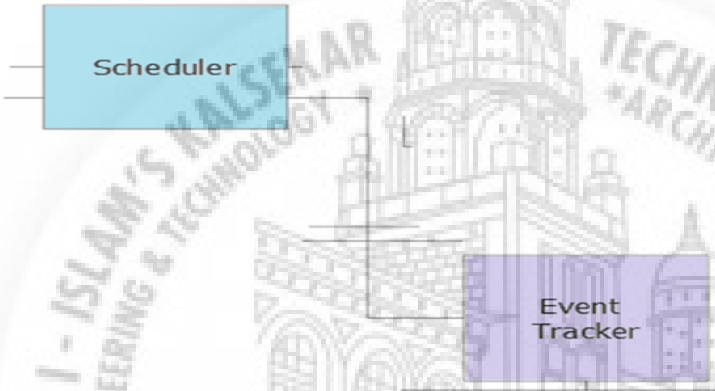

Figure 5.13: Event Tracker

5.3.5 Module 5 - TTS Engine & Notification system

With the help of various Geo-location & Calling API's , a user would get dynamic generated calls reminding about the doses that a patient needs to take with custom made emergency facilities such as Speedometer where a patient travelling speed is computed, Current location finder, Alerting using native sensors. The automated calling feature is the crux USP of this application considering the short survey taken by us regarding the ratio of smart phone users, and old java or symbian phones. The MNA related issues drastically hit the senior citizens or the children's, most probably they are non-smart phone users so an app only notification solution is not ideal. Automated calling functionality increases the target audience as well as it will become a great cause to serve the society. For ensuring such functionality, The nexmo api will be integrated within the application.

Module 5 Block Flow Diagram

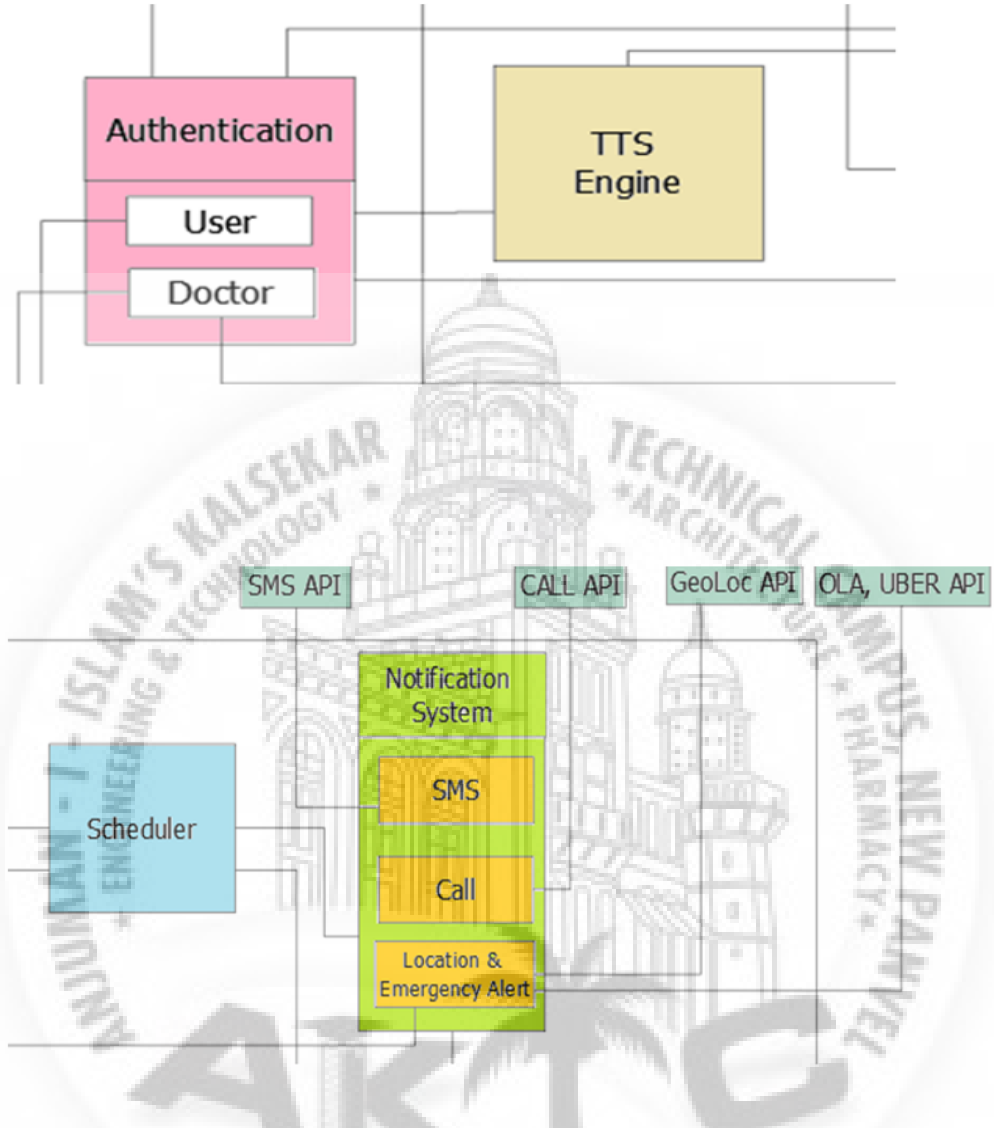

Figure 5.14: TTS Engine & Notification system

5.3.6 Module 6 - Report Analysis & Diet recommender

After event tracking module, accordingly the analysis of report will be showcased daily & also our system will use A.I driven deep learning technologies to predict various patterns among infected patients, their behaviours, depending upon the diseases diet plan will be recommended as a suggestion. This would solve the problem for a patient going into the long queues just to get a consultation from doctor and the user will also have the sigh of relief for a patient health.

Module 6 Block Flow Diagram

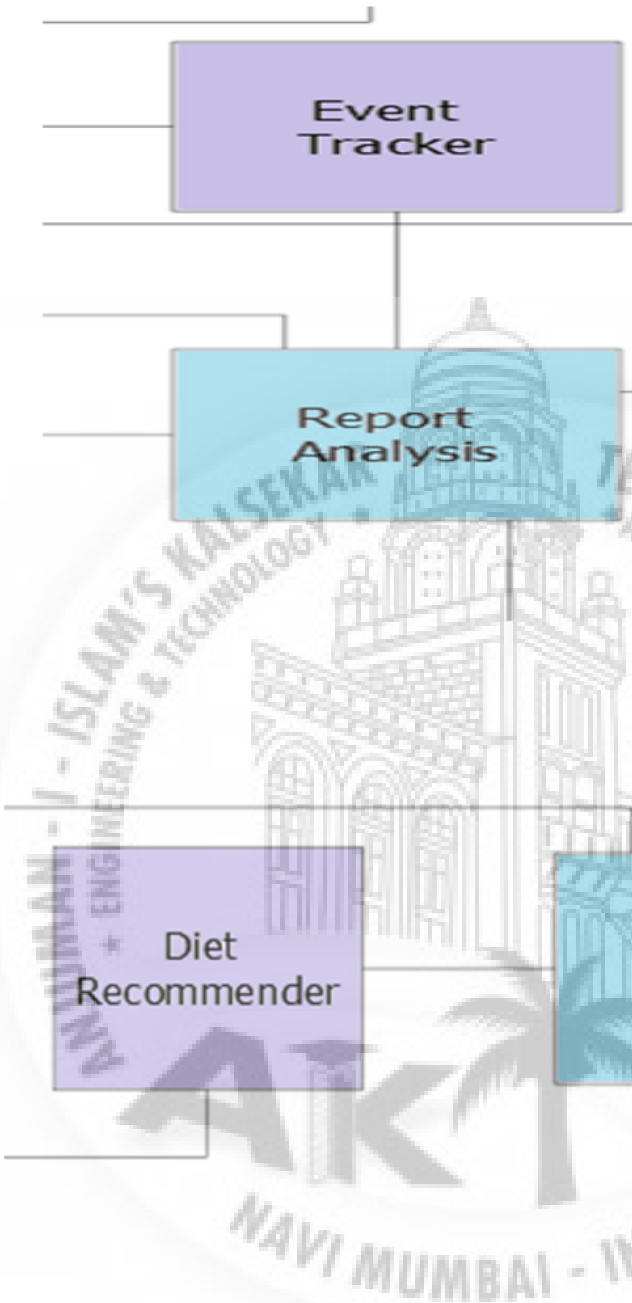

Figure 5.15: Report Analysis & Diet Recommender

5.3.7 Module 7 - Instant Messaging

We approached to some of the doctors and prepared a questionnaire regarding patient, their common behaviours. There was also one of the prime question asked what are things that makes a bit challenging for a medical professional. The response was the doctors life is very busy, they can get a call even at mid night and they have to console the patient. Meanwhile there is also a concern for a large portion of doctor that is, the contact sharing. A patient have to contact, but depending upon scenarios a doctor must also have the right to privacy. while some organization

provide basic contact office support while some don't. Concerning to the doctors privacy breaching issues, the doctors will have the facility to directly avail all the notification & reports of patient. Patient will have direct access to contact doctors for respective situations.

Module 7 Block Flow Diagram

Figure 5.16: Report Analysis & Diet Recommender

5.4 Systems Integration

The system has been developed using the scrumm based agile methodology and have encompassed the stubs and drivers concept. We have modularized the whole project into smaller units for fast and elegant development. Various SDK such as boto3 for AWS DynamoDB, Nexmo, Google Cloud Vision, SMS Request packages have been used extensively module wise. After the module showed ample of satisfactorily result it has then been integrated to make a complete fledged application.

5.4.1 Class Diagram

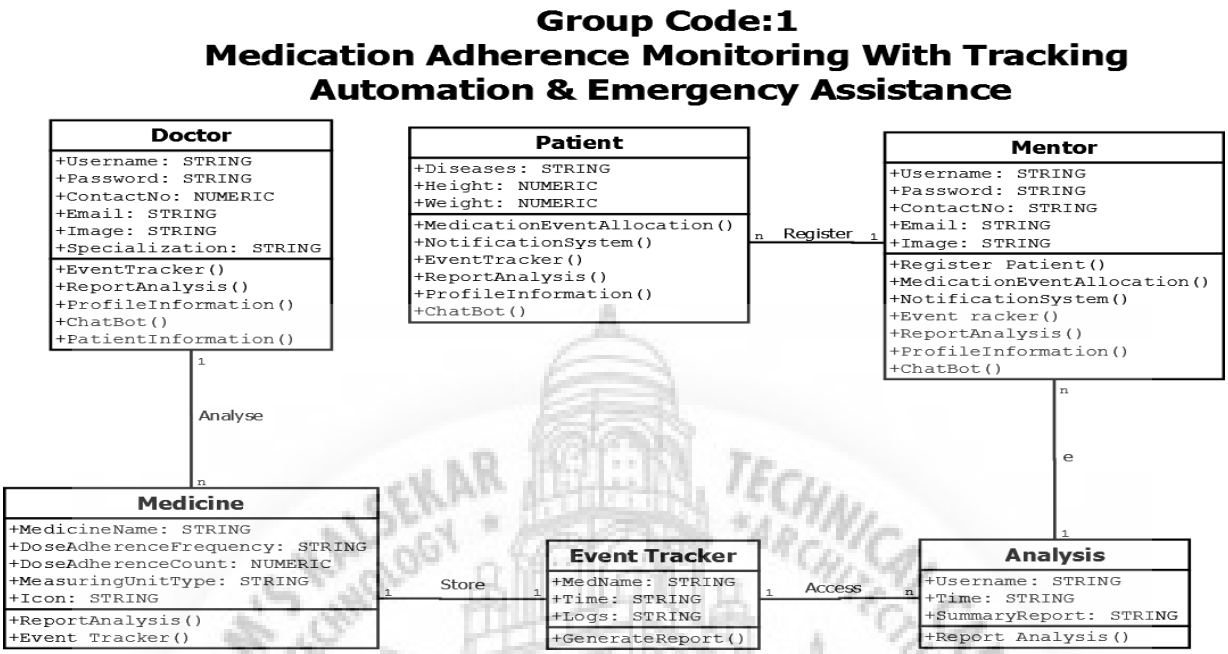

Figure 5.17: Class Diagram for MAMTE

Class diagram showing each modules interconnection and relation between how one class with its characteristics is interacting with the other.

5.4.2 Sequence Diagram

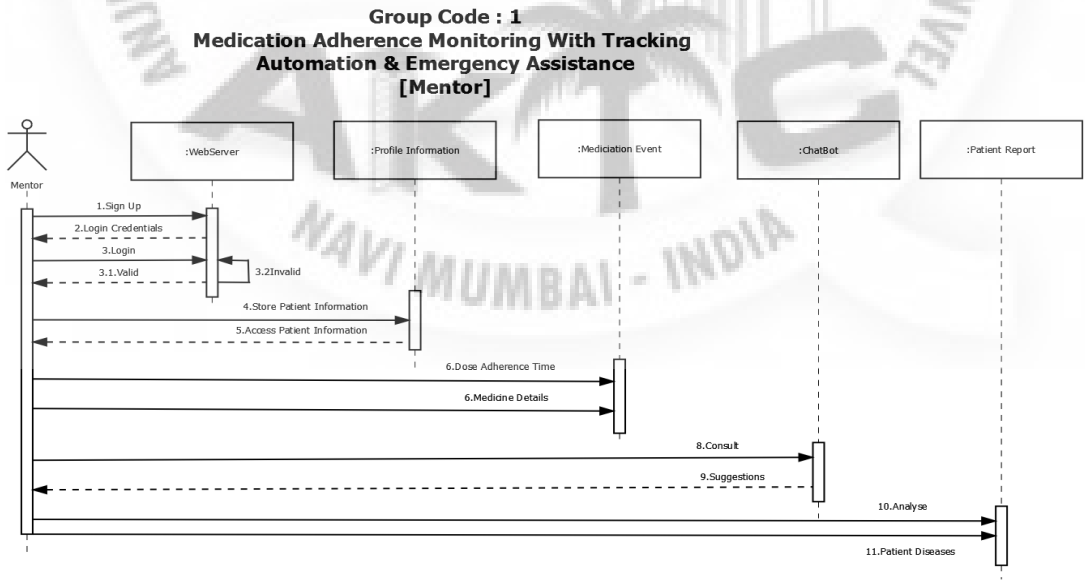

Figure 5.18: Sequence Diagram of Mentor

Sequence diagram for mentor representing how this user will be able to access what type of resources and at which time intervals.

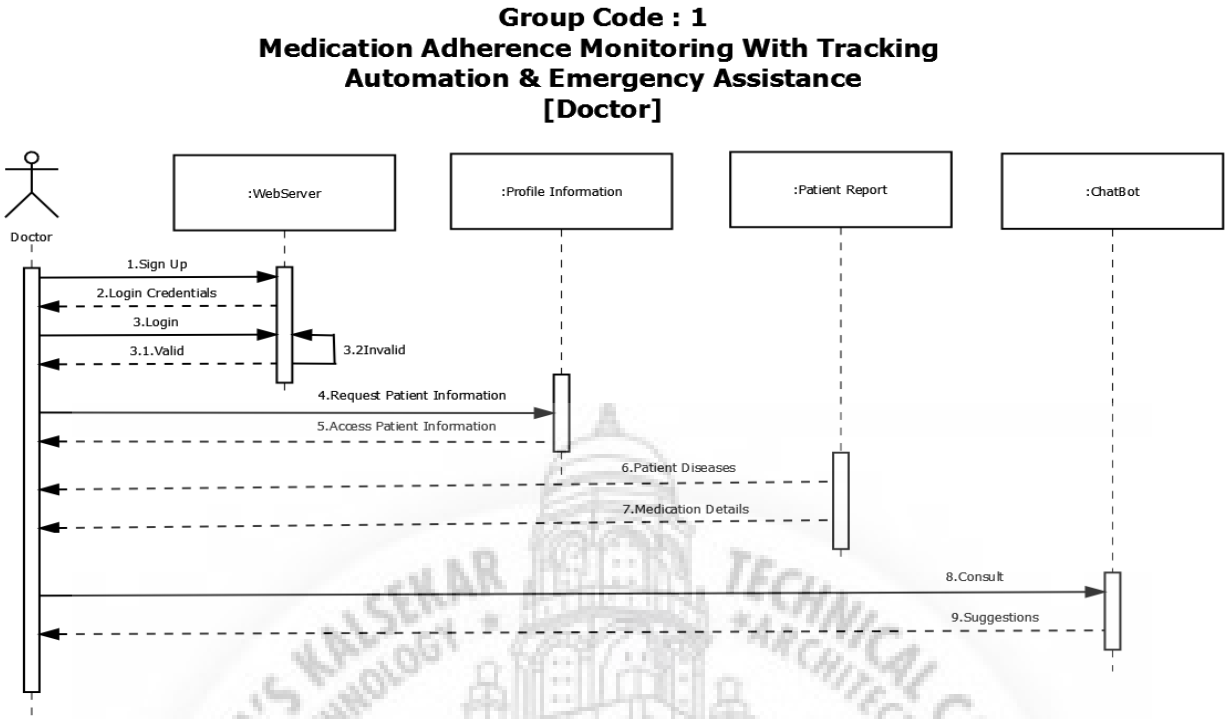

Figure 5.19: Sequence Diagram of Doctor

Sequence diagram for Doctor depicting how this user will be able to access what type of resources and at which time intervals.

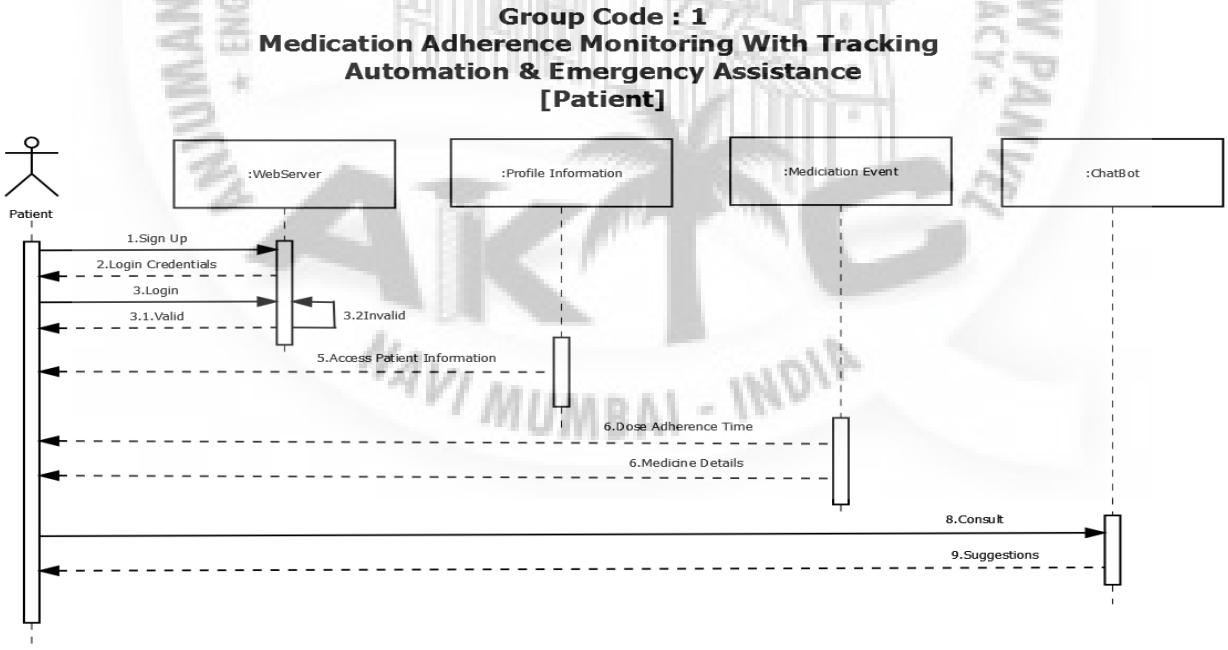

Figure 5.20: Sequence Diagram of Patient

Sequence diagram for Patient reflecting how this user will be able to access what type of resources and at which time intervals.

5.4.3 Component Diagram

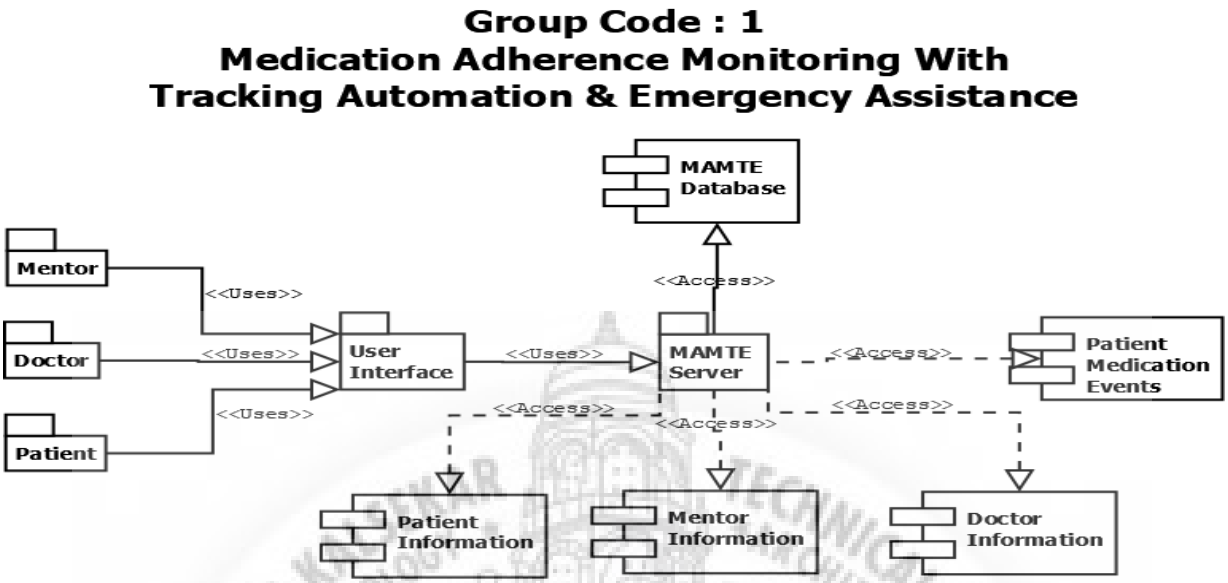

Figure 5.21: Component Diagram for MAMTE

Component Diagram describing the organization and wiring of the physical components in the system.

5.4.4 Deployment Diagram

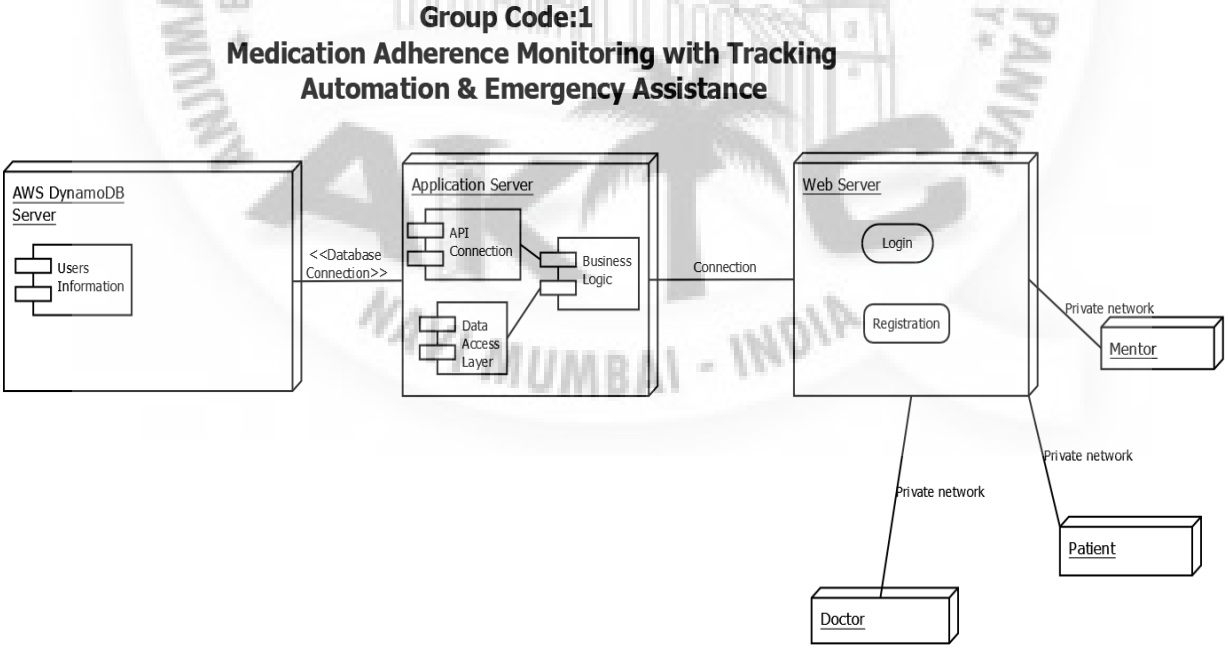

Figure 5.22: Deployment Diagram

Deployment Diagram describing the static deployment view of the system & visualize the topology of the physical components of the system, where the software components are deployed.

Chapter 6

Implementation

6.1 Call Notification Through Nexmo Api

Nexmo connects our apps directly to carriers around the world. Integrate SMS and Voice messages using one simple API. Use Nexmo Verity to register users, verify transactions, and implement two factor authentication. It is a tool in the Voice and SMS category of a tech stack.

With the help of this module we are able to facilitate the end user with timely notification through calling and it gives us the potential benefit of increasing the target audiences.

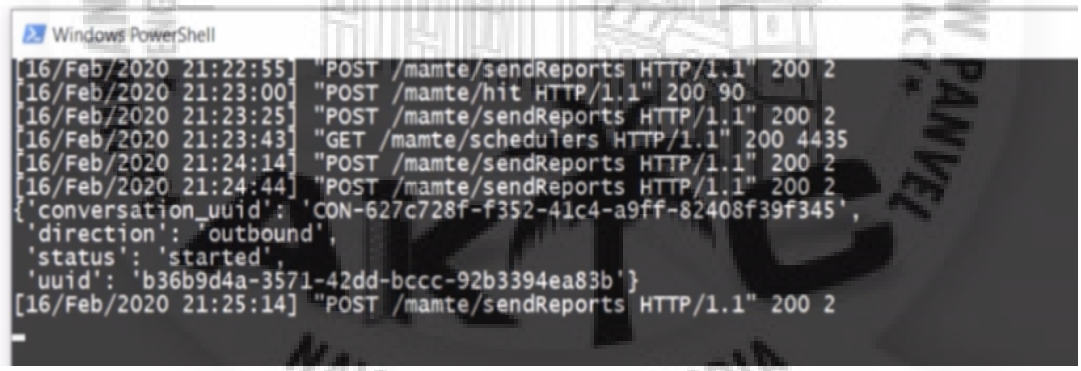

Figure 6.1: Sample JSON Response of Nexmo API

```
import nexmo
from pprint import pprint
client = nexmo.Client(
  application_id='f55af314-c1aa-46a8-8f8c-b87cb0109a5d',
  private_key='private.key',
)
medname='Dolo 650'
ncco = [
  {
    'action': 'talk',
    'voiceName': 'Aditi',
    'text': "<speak><prosody volume='loud'>"+"rahul"+"ne"+"crocin. "+" ka
       reminder skip kiya hai. Thankyou! </prosody></speak>"
  }
```

```
]
response = client.create_call({
  'to': [{
    'type': 'phone',
    'number': '91989xxxxxx'
  }],
  'from': {
    'type': 'phone',
    'number': '1201xxxxxx'
  },
  'ncco': ncco
})
pprint(response)
```

6.2 Text Detection And Recognition Through Google Vision Api

The Vision API can detect and extract text from images. There are two annotation features that support optical character recognition (OCR).

This module helps the mentor to not just go with the manual entry like forms rather establishes an innovative way of communication where a mentor is expected to place the image which has medicine name and it would be extracted automatically.

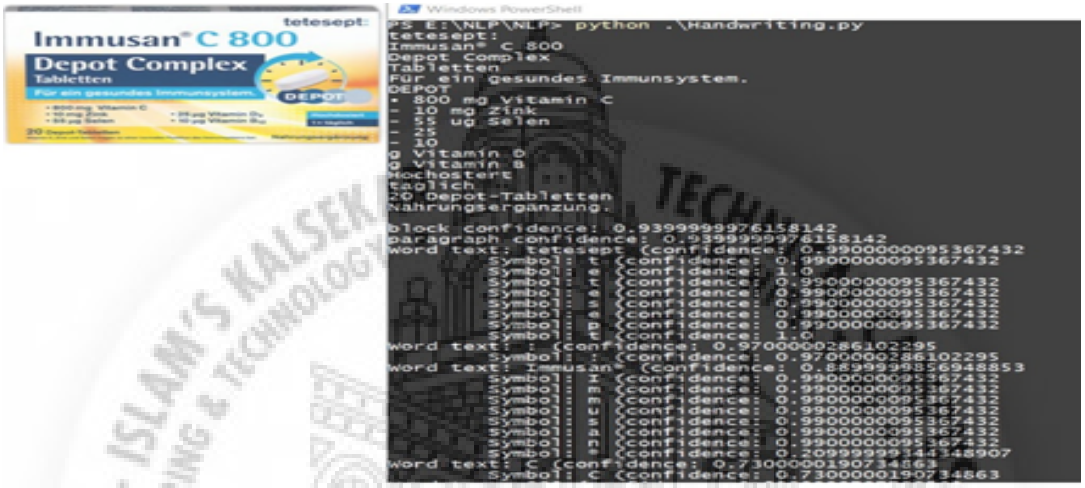

Figure 6.2: Sample image provided and medicine name detected as immusane

```
import os,io
from google.cloud import vision
from google.cloud.vision import types
import pandas as pd
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = r'ServiceAccountToken.json'
client = vision.ImageAnnotatorClient()
FILE_NAME = 'cartfit.jpg'
#FOLDER_PATH = r'C:\Users\shaik\Desktop\HT\stubs\NLP\img\tests'
FOLDER_PATH = r'/media/yash/SHADABSK/NLP/NLP/img/tests/'
def detectText(img):
    with io.open(img,'rb') as image_file:
        content = image_file.read()
    image = vision.types.Image(content=content)
    response = client.text_detection(image=image)
    print(response)
    texts=response.text_annotations
    df=pd.DataFrame(columns=['locale','description'])
    for text in texts:
        df=df.append(
            dict(
                Locale=text.locale,
                description=text.description
            ),
            ignore_index=True
        )
    return df
print(detectText(os.path.join(FOLDER_PATH,FILE_NAME)))
```

6.3 QR Code Generation

The QR Code is a two-dimensional version of the bar code, known from product packaging in the supermarket.The QR Code has found its way into mobile marketing with the widespread adoption of smartphones. ”QR” stands for ”Quick Response”, which refers to the instant access to the information hidden in the Code.

The output of this module allows the end user to gain access to the system without having the requirement to manually inserting the credentials.

Figure 6.3: Sample generated QR code of a mentor

```
import qrcode
import os

qr = qrcode.QRCode(
   version=1,
   box_size=15,
   border=5
)

username="MAMTE"
password="MAMTA"

directory='folder/'

if not os.path.exists(directory):
    os.makedirs(directory)

data = (username,password)
qr.add_data(data)
qr.make(fit=True)
img = qr.make_image(fill='black', back_color='white')
img.save(directory+username+'.png')
```

6.4 Contact verification, Emergency Assistance through bulk sms gateway/services

An SMS gateway or MMS gateway allows a computer to send or receive Short Message Service (SMS) or Multimedia Messaging Service (MMS) transmissions to or from a telecommunications network. Most messages are eventually routed into the mobile phone networks. Many SMS gateways support media conversion from email and other formats.

By using such services we ensure, that the end user is not only authenticated but authorized as well. At the same time we try to engage the users irrespective of the user having a smartphone or a normal phone.

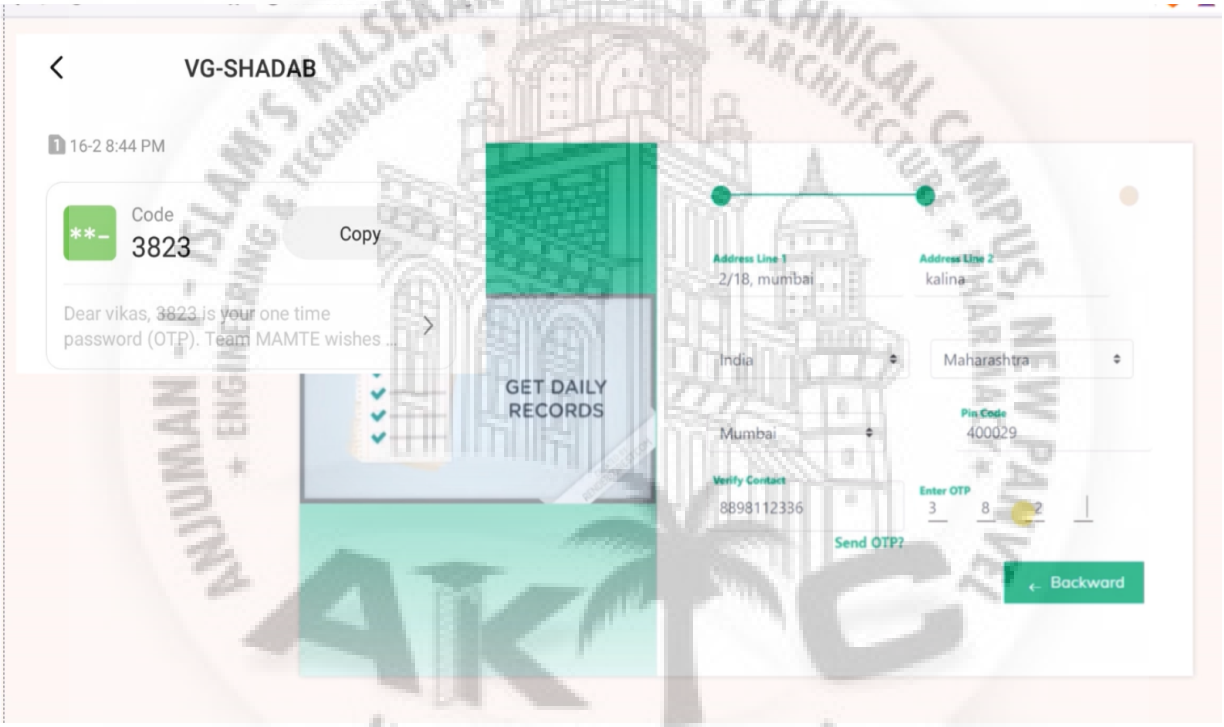

Figure 6.4: A user entering the otp sent through the system for verification

```
import requests
urlink = 'http://bulk.sagartech.net/http-tokenkeyapi.php?'
authentic_key = 'authentic-key=<TOKEN KEY HERE>'
sender_id= 'senderid=SHADAB'
route = 'route=2'
number='number=9892327169'
message='message=hello there'
result = requests.get(urlink+"&"+authentic_key+"&"+sender_id+"&"+route+"&"+
    number+"&"+message)
print(result)
if(result.status_code == 200):
  print("OK")
```

6.5 Generating Canvas Charts for reporting & analytics

Any Numerical Data can be plotted on the Canvas as Graph using Data Plots. Data Plot is plotted on a grid representing the co-ordinate axis. Multiple colors can be used for different data sets inorder to make the distinction clear.

To reflect appropriate analysis and data in an intuitive manner, we have used apex chart.js to render the medication event data representing in pictorial chart form.

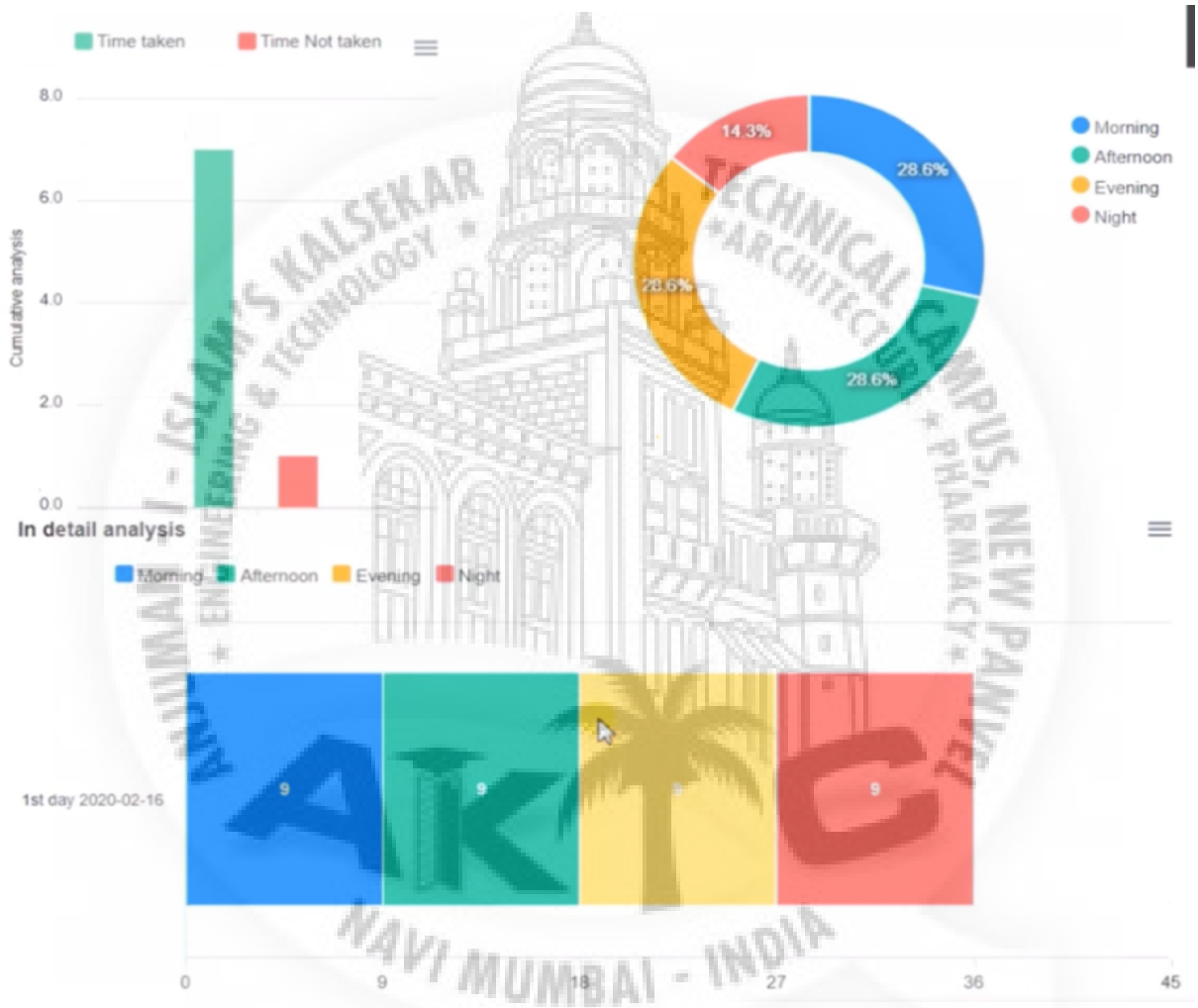

Figure 6.5: Usage of doughnut charts, vertical bar chart & stacked bar chart

```
<!DOCTYPE html>
<html>
    <head>
      <title></title>
      <script src="https://code.highcharts.com/highcharts.js"></script>
        <script src="https://code.highcharts.com/modules/exporting.js"></script>
        <script src="https://code.highcharts.com/modules/export-data.js"></
            script>
        <script src="https://code.highcharts.com/modules/accessibility.js"></
            script>

        <style type="text/css">
```

```
            .highcharts-figure, .highcharts-data-table table {min-width: 320px;
                max-width: 660px;
                 margin: 1em auto;}

            .highcharts-data-table table {font-family: Verdana, sans-serif;
                border-collapse: collapse; border: 1px solid #EBEBEB; margin: 10
                px auto; text-align: center; width: 100%; max-width: 500px;
            }
            .highcharts-data-table caption { padding: 1em 0; font-size: 1.2em;
                color: #555;
            }
            .highcharts-data-table th {
              font-weight: 600;
        </style>
    </head>
    <body>

        <figure class="highcharts-figure">
            container <div id="container"></div>
            <p class="highcharts-description">
                Messages
            </p>
        </figure>

        <script type="text/javascript">
          // Radialize the colors
            Highcharts.setOptions({
                colors: Highcharts.map(Highcharts.getOptions().colors, function
                    (color) {
                    return {
                        radialGradient: {
                            cx: 0.5,
                            cy: 0.3,
                            r: 0.7
                        },
                        stops: [
                            [0, color],
                            [1, Highcharts.Color(color).brighten(-0.3).get('rgb'
                                )] // darken
                        ]
                    };
                })
            });

            // Build the chart
            Highcharts.chart('container', {
                chart: {
                    plotBackgroundColor: null,
                    plotBorderWidth: null,
                    plotShadow: false,
                    type: 'pie'
                },
                title: {
                    text: 'Report Analytics'
                },
                tooltip: {
                    pointFormat: '{series.name}: <b>{point.percentage:.1f}%</b>'
                },
                accessibility: {
                    point: {
                        valueSuffix: '%'
```

```
                    }
                },
                plotOptions: {
                    pie: {
                        allowPointSelect: true,
                        cursor: 'pointer',
                        dataLabels: {
                            enabled: true,
                            format: '<b>{point.name}</b>: {point.percentage:.1f}
                                %',
                            connectorColor: 'silver'
                        }
                    }
                },
                series: [{
                    name: 'Share',
                    data: [
                        { name: 'Chrome', y: 61.41 },
                        { name: 'Internet Explorer', y: 11.84 },
                        { name: 'Firefox', y: 10.85 },
                        { name: 'Edge', y: 4.67 },
                        { name: 'Safari', y: 4.18 },
                        { name: 'Other', y: 7.05 }
                    ]
                }]
            });
        </script>
    </body>
</html>
```

6.6 Registration Module

A user gets enroll to the system via a quick sign up, after profile completion and validity of a genuine user a user would also get a qr code generated with the username & password enclosed. A user then enters the details, the passwords are stored in sha256 encrypted hash with 12000 rounds and 32 salt size ensuring the security aspects.

With the help of QR generated or username & password a user will be able to gain access to the system. The user roles are divided into three section namely doctors, mentor acting as a guardian for the patient and the patient who is currently suffering from a disease.

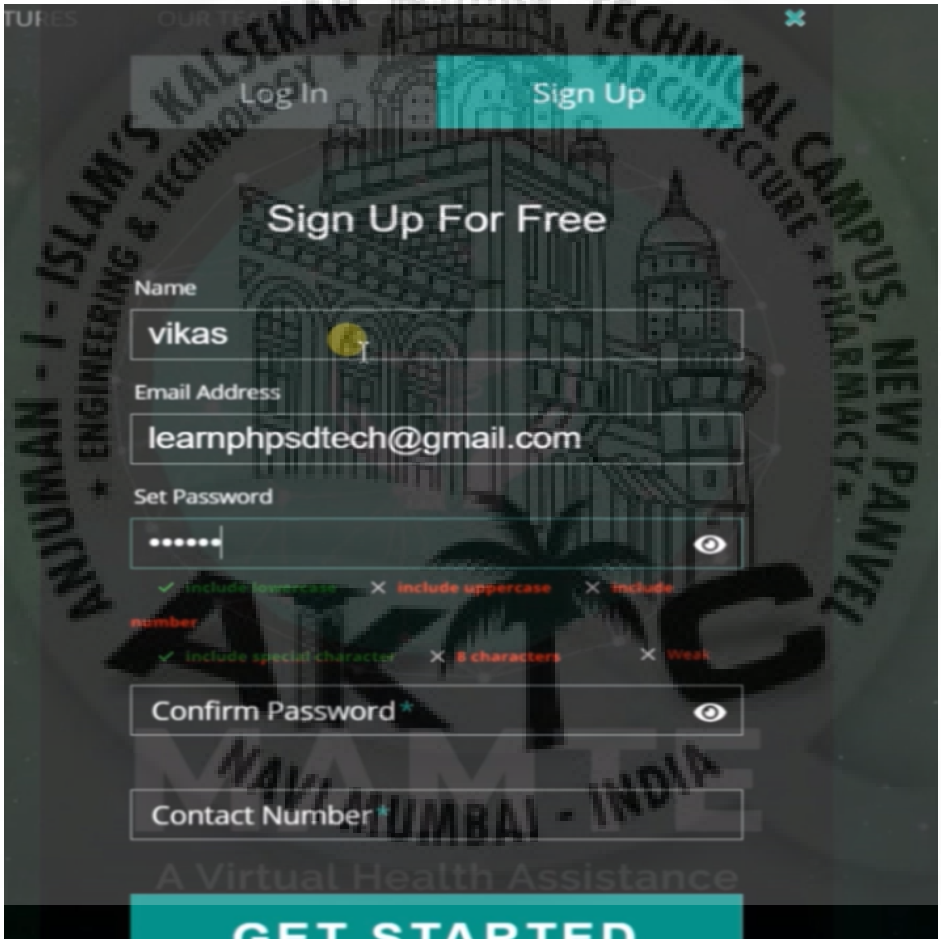

Figure 6.6: Sample key input to registration module.

```
def registerUser(request):
    if request.method == "POST":
        creatime=time.ctime()
        username=request.POST['uname']
        password=request.POST['upass']
        password=pbkdf2_sha256.encrypt(password,rounds=12000,salt_size=32)
        email=request.POST['email']
        contact=request.POST['number']
        usertype=request.POST['user']
```

```
        db = boto3.resource('dynamodb')
        tb=db.Table("Users")
        qr = qrcode.QRCode(
                version=1,
                box_size=15,
                border=5
            )
        qr_data=(username,password)
        qr.add_data(qr_data)
        qr.make(fit=True)
        img = qr.make_image(fill='black', back_color='white')
        directory = 'healthassistance/templates/static/img/'+username+'/'
        if not os.path.exists(directory):
            os.makedirs(directory)
        img.save(directory+username+'qr.png')
        response=tb.put_item(
            Item={
                'Username': username,
                'Password': password,
                'EmailID': email,
                'ContactNo': contact,
                'GroupName': usertype,
                'CreationTime': str(creatime),
                'Qr_det': directory+username+'qr.png',
                'Address':'None',
                'Age':'None',
                'Answer':'None',
                'DOB':'None',
                'Fname':'None',
                'Gender':'None',
                'Image':'None',
                'LastModified':'None',
                'Lname':'None',
                'Mname':'None',
                'Quest':'None',
                'LastLogin': 'None'
            }
        )
        if(response['ResponseMetadata']['HTTPStatusCode']==200):
            if(usertype=="Doctor"):
                tb2=db.Table("Doctor")
                response2=tb2.put_item(
                Item={
                    'DUsername': username,
                    'CreationTime': str(creatime)
                    }
                )
                if(response2['ResponseMetadata']['HTTPStatusCode']==200):
                    data={'True': 'true'}
                    return JsonResponse(data)
            else:
                data={'True': 'true'}
                return JsonResponse(data)
        else:
            data={'None': 'none'}
            return JsonResponse(data)
    else:
        return render(request, 'register.html' ,{})
```

6.7 QR Login module

This module has a scanner which makes use of the internal camera or any external camera plugged into the system. The QR image is stored into base64 format and then passed onto the backend server for further processing.

At the background pyzbar algorithm works, to scan and extract the username and password which is encrypted ensuring the reliability and privacy aspect of the user.

Figure 6.7: Generated QR shown to the scanner for logging in to the system

```
def qrurl(request):
    if request.method == "POST":
        data={}
        qr_data=request.POST['qrcheck']
        image_path = 'healthassistance/templates/qr_det/'

        if not os.path.exists(image_path):
            os.makedirs(image_path)

        base64_data = re.sub('^', '', qr_data)
```

```
        img = Image.open(BytesIO(base64.b64decode(base64_data)))
        img.save(image_path+'test.png', "PNG")
        im=cv2.cv2.imread(image_path+'test.png')
        im=cv2.cv2.resize(im,(500,500))
        decodedObjects = pyzbar.decode(im)
        str1=""
        username=""
        password=""

        try:
            str1=str((decodedObjects[0][0]))
            str1=str1.lstrip('b\'\"(\'').rstrip('\'\")\'')
            str1=str1.split(',')
            username=str1[0].rstrip('\'')
            password=str1[1].lstrip(' \'')

            db = boto3.resource('dynamodb')
            tb=db.Table("Users")
            Username = tb.get_item(
            Key={
                'Username': username
            })


            if(Username['Item']['Username']==username):
                if (Username['Item']['Password']==password):
                    request.session['usname'] = Username['Item']['Username']
                    data['true']='true'
                    data['usname']=request.session['usname']

                    return JsonResponse(data)
                else:
                    data['false']='false'
                    return JsonResponse(data)
        except:
            return JsonResponse(data)
    else:
        return render(request, 'login.html' ,{})
```

6.8 Scheduler & Event Tracker

After the successful medication event details extraction through various mechanisms, the system will take care of scheduling and handle the time management process and will act as a medium for triggering the event. For this either the threading mechanism can be thought of implementation or else the micro services. The input of this module will be Extraction module and the output of this module will be Event Tracker and also Notification & alert system. Scheduling a live application considering the stake of a patient health is not an easy task, the server will have to take care about the time zones, their medication times, and dispatch the request in an efficient manner. For this Django services in collaboration with Ajax job-queue scheduler are integrated..

This would be having all the logs of events for the patient medication adherence activity day wise and accordingly reward mechanism will be determined. This would act as a base for timely report generation of a patient activity and can help in the productivity of the sustainable growth of the application. For dumping of such huge amount of log involves high response time and a highly secure database server so for that amazon AWS DynamoDB will be serving to tackle the response related aspects.

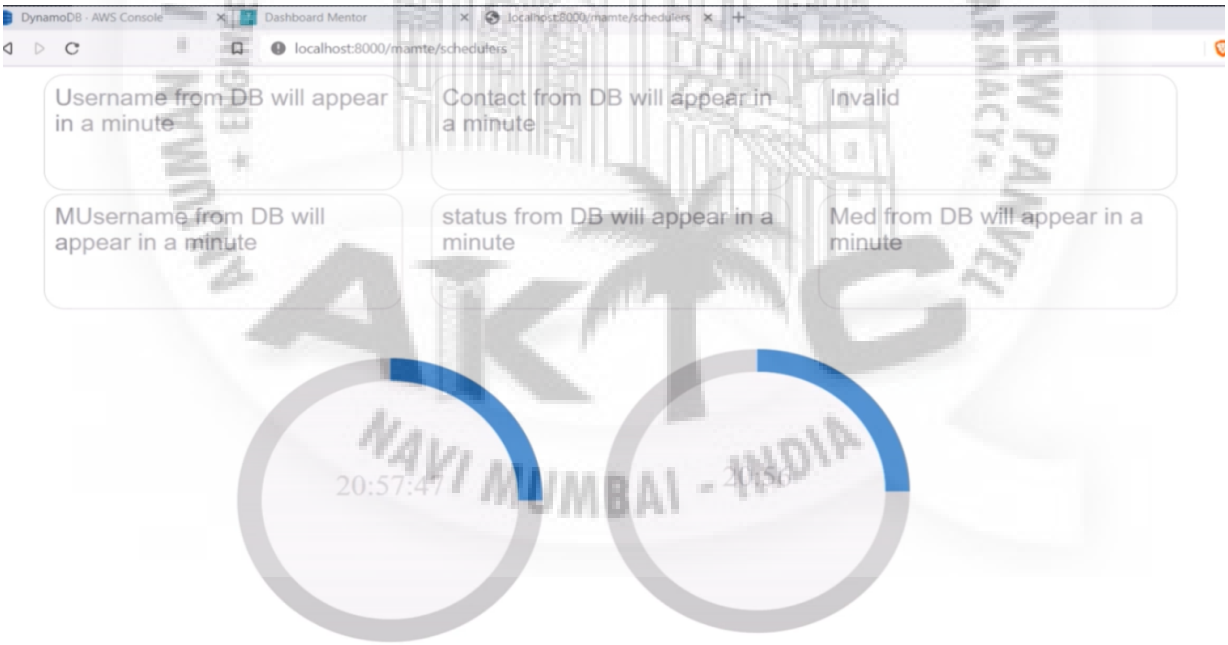

Figure 6.8: Admin panel with the glimpse of scheduler working on time zones for respective patients & logging in each and every activity.

```
def sample_job_every_2s(request):
    if request.method == 'POST':
        notifylistUsername=[]
        notifylistPUsername=[]
```

```
        notifylistMUsername=[]
        notifylistContact=[]
        notifylistTimeAM=[]
        notifylistTimeNoon=[]
        notifylistTimeOther=[]
        notifylistTimePM=[]
        notifylistTimeMed=[]
        notifylistPdisease=[]
        notifylistPdoctor=[]
        status=""
        db = boto3.resource('dynamodb')
        data={}
        tb=db.Table("Customization")
        DbTimeList = tb.scan(
        TableName='Customization',
        AttributesToGet=[
             'PUsername',
             'AMNotifyTimeStart',
             'NoonNotifyTimeStart',
             'OtherNotifyTimeStart',
             'PMNotifyTimeStart',
             'MedName'
             ]
        )

        notify=DbTimeList['Items']
        i=0
        cnt=len(notify)-1
        while(i<=cnt):
            notifylistUsername.append(notify[i]['PUsername'])
            notifylistTimeAM.append(notify[i]['AMNotifyTimeStart'])
            notifylistTimePM.append(notify[i]['PMNotifyTimeStart'])
            notifylistTimeNoon.append(notify[i]['NoonNotifyTimeStart'])
            notifylistTimeOther.append(notify[i]['OtherNotifyTimeStart'])
            i+=1

        Curtime=time.ctime()
        Curtime=Curtime[11:16]
        t1=time.ctime()
        t1=t1[11:13]

        if(int(t1)<12):
            for i in range(len(notifylistTimeAM)):
                if(Curtime==notifylistTimeAM[i]):
                    notifylistTimeMed.append(notify[i]['MedName'])

                    tb2=db.Table("Users")
                    Username = tb2.get_item(
                    Key={
                        'Username': notifylistUsername[i]
                    })

                    tb3=db.Table("Patient")
                    MUsername = tb3.get_item(
                    Key={
                        'PUsername': notifylistUsername[i]
                    })

                    notifylistContact.append(Username['Item']['ContactNo'])
                    notifylistPUsername.append(Username['Item']['Username'])
                    notifylistMUsername.append(MUsername['Item']['MUsername'])
```

```
                    notifylistPdisease.append(MUsername['Item']['Disease'])
                    notifylistPdoctor.append(MUsername['Item']['DUsername'])
                    status='am'
    for x in range(len(notifylistPUsername)):
            PRUsername = tb.get_item(
            Key={
                'PUsername': notifylistPUsername[x]
            })


            if(status=="am"):
                client = nexmo.Client(
                    application_id='f55af314-c1aa-46a8-8f8c-b87cb0109a5d',
                    private_key='healthassistance/private.key',
                )

                calluuid=PRUsername['Item']['AMNotifyTimeEnd']

                callinfo = client.get_call(calluuid)
                time.sleep(3)

                logpath="healthassistance/templates/static/logs/"+
                    notifylistMUsername[x]+"/"+notifylistPUsername[x]+"/"

                if(int(callinfo['duration'])>3):
                    file1 = open(logpath+notifylistPUsername[x]+"-"+status+".txt
                        ","a")
                    file1.writelines(str(time.ctime())+"#"+calluuid+"#"+callinfo
                        ['to']['number']+"#"+callinfo['duration']+"#"+
                        notifylistMUsername[x]+"#"+notifylistPUsername[x]+"#"+
                        notifylistPdoctor[x]+"#"+notifylistPdisease[x]+"#"+" OK
                        "+"\n")

                    tbl=db.Table("Customization")
                    tbl.update_item(
                        Key={
                            'PUsername': notifylistPUsername[x],
                        },
                        UpdateExpression='SET AMNotifyTimeEnd = :uuid',
                        ExpressionAttributeValues={
                            ':uuid': callinfo['duration']
                        }
                    )
        }
```

Chapter 7

System Testing

The application and the project as a whole has been implemented by using the core software engineering principles in mind. The whole application was divided into smaller modules and more smaller modules so to accommodate with the quick and bug less coding practises. Many a times a module would require a higher dependent module in that case driver concept was used and many a times a module would require the input from the lower dependent module in that case stubs concept were used. After the smaller modules were created the integration testing were performed adequately and tested thoroughly handling even the minute exceptional cases also.

7.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Registration Module	Giving username input	The system alerts when the username is already taken by someone else, and leaves when the username is available	The system should reflect if the username is not available and is taken by someone else already
T02	QR-Code generation	Registering as a new user with valid inputs	System generates a qr code in the backend of the respective username directory	The system must generate a directory with the specified username newly registered along with qr code residing in that.

T03	Contact Verification	Inserting the OTP recieved through the system	The system enables user to proceed	The system must not allow user to proceed if the otp entered is not correct.
T04	Login	Valid credentials	The system lets the user signin	The system must not prompt appropriate dialog if detected as wrong credentials and must facilitate with resetting the password.
T05	Forgot Password	Inserting the registered email	The system generates a unique url for the requested user and allows to reset the password	The system should discard the newly generated url after the user clicks on it making it a one time session and should reset the requested user credentials.
T06	Text Detection	Giving medicine image	The system recognized the text and extract the necessary medicine name	The system must generate the exact medicine name or should provide most likely name or allow user to crop certain portion of the image if nothing works then it should allow the user to go for the manual entry.
T07	Saving patient report analytic	User clicking on save this button	The system capture the generated charts and represent it in a pdf form	The system must be capable enough to bear responsive resolution and convert the report to pdf format.

7.2 Sample of a Test Case

Title: Text Detection – Appropriate medicine name extracted from the given image

Description: A registered user should be able to get the medicine name as the output being set on the medicine allocation form.

Precondition: the user must already be registered, and the user must be a mentor with a patient associated.

Assumption: a supported HTML5 browser or compatible platform for the android app is being used.

Test Steps:

1. Registering with as a mentor and completing the procedures.
2. Logging in with valid credentials.
3. Creating a patient with appropriate details.
4. Clicking on Medicine entry tab
5. Selecting Yes i have a camera option when prompted
6. Showing image in front of the camera which has medicine name
7. Clicking on capture button

Expected Result: The prompt and the camera being closed, and the medicine name from the image is extracted and set to the medicine entry form 1st input.

Actual Result: After the mentor places the image in front of the camera opened and clicked on the capture button, the prompted dialog was closed and depending upon the internet speed connection after 1 second the new form was showed up asking the medicine entry details. The first input was already filled with the medicine name which was extracted by the text detection algorithm via the system.

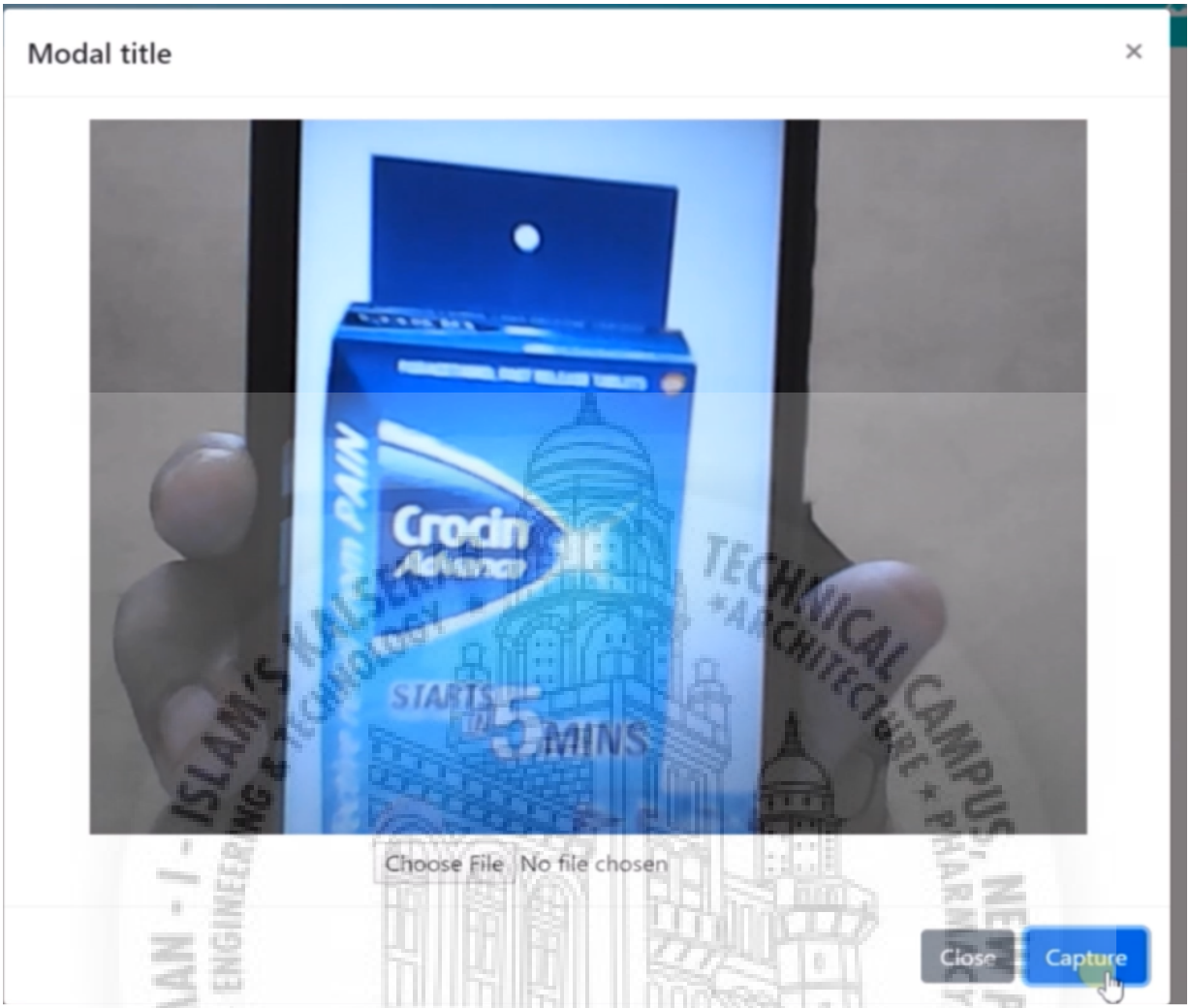

Figure 7.1: Actual result: Showing the image containing the medicine name.

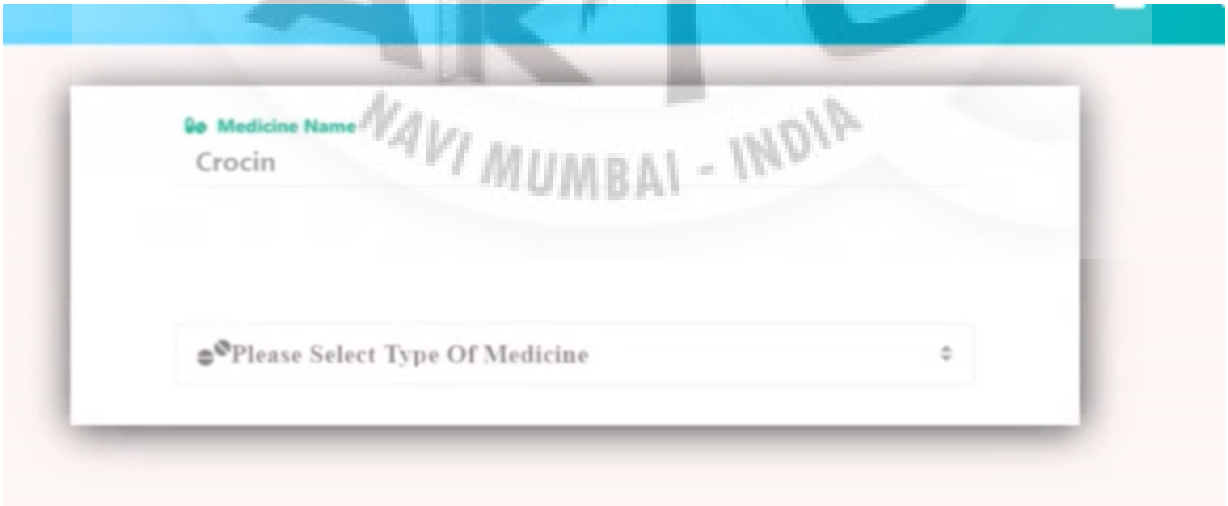

Figure 7.2: Actual result: The extracted medicine name set on the medicine name input of the medicine allocation form.

7.2.1 Software Quality Attributes

- Usability - The system can be used with more efficient manner concerning to the illiteracy factor we have ensured TTS based mechanisms, and non smartphone bases solutions to better the healthy life being and tackle the MNA related issues.
- Adaptability - The system has been integrated with proper guidance mechanism and introductory learning videos so that the end user quickly grasp the root of the application in mere a minute.
- Security - The system encompass the Sha256 algorithm with 12000 rounds and 32 salt size ensuring the security aspect and handles the session with one time unique generation of urls.
- Scalability - The system can be scaled of upto bearing even the smartphone, hardware based IOT solutions and simultaneously run the batch processing of scheduling work with ease.
- Reliability - The system is reliable enough to bear the load of as many as 1000 user request at the same time and ensures the accuracy of data with cloud backup support.
- Reusability - The system uses various frameworks and coding structure that are in the current trend. Such framework allows the system to meet the MVT architecture and ensure coding standards in such a way that appropriate key modules and function can be utilised thoroughly following the DRY concepts.
- Maintainability - The application and its internal mechanism are readable, extensible and testable to the extent that even the naive user will have a clearer glimpse of what the coding pattern are performing. Proper documentation, quoting and commenting has been taken care of in most of the scenarios and use cases. Throughout to ensure readability and testability the consistency factor has been the key factor been into consideration.

Chapter 8

Screenshots of Project

8.1 Homepage/Landing page of the application

Figure 8.1: Initial screen along with logo for MAMTE

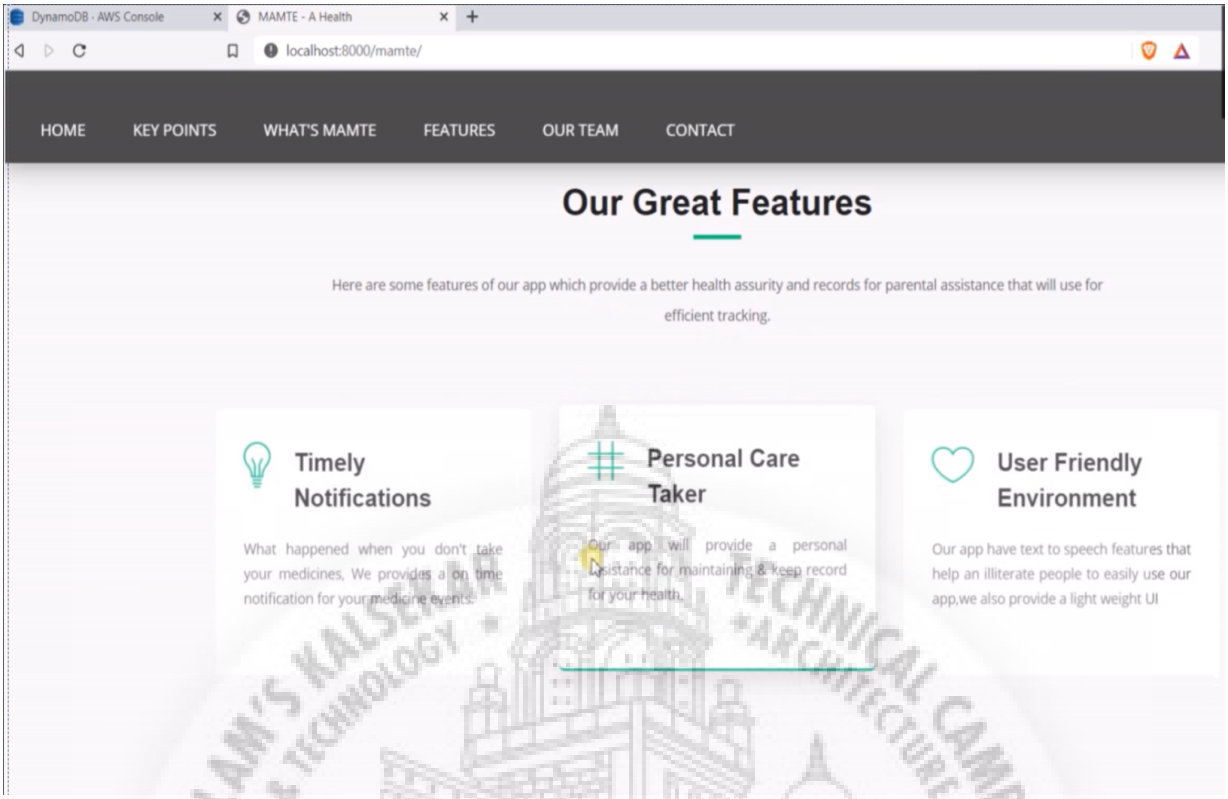

Figure 8.2: Homepage describing the key benefits to use our application

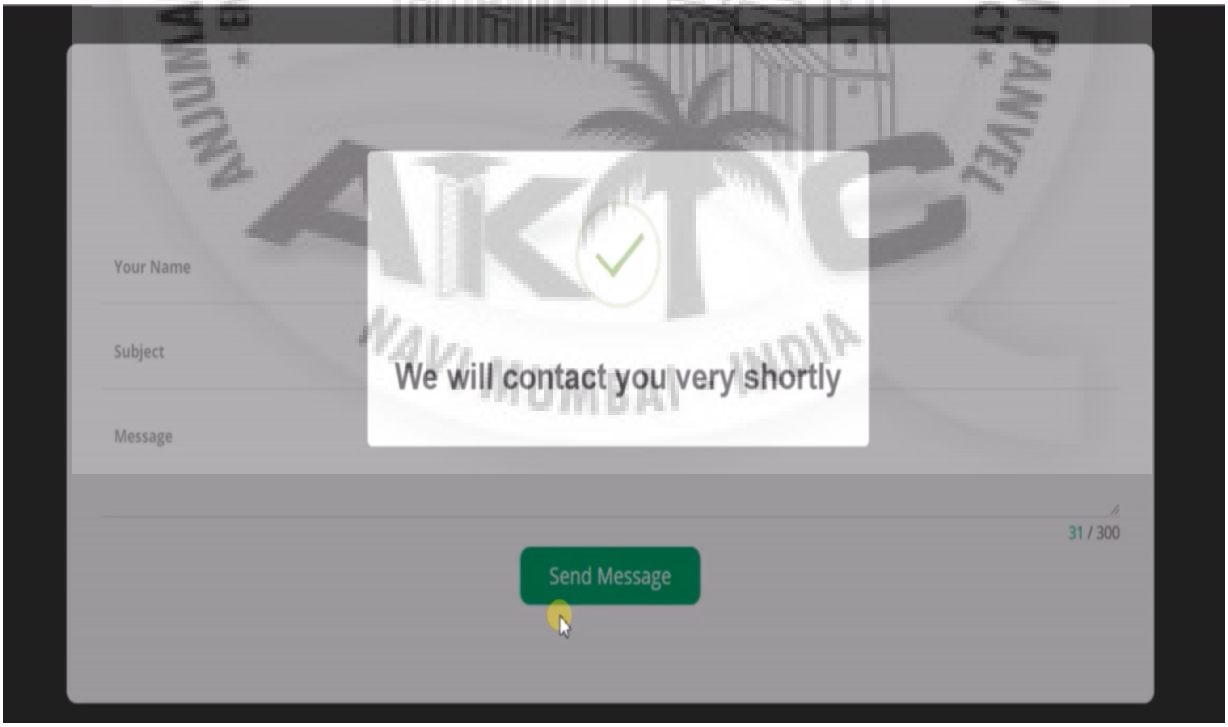

Figure 8.3: Contact form

8.2 Registration

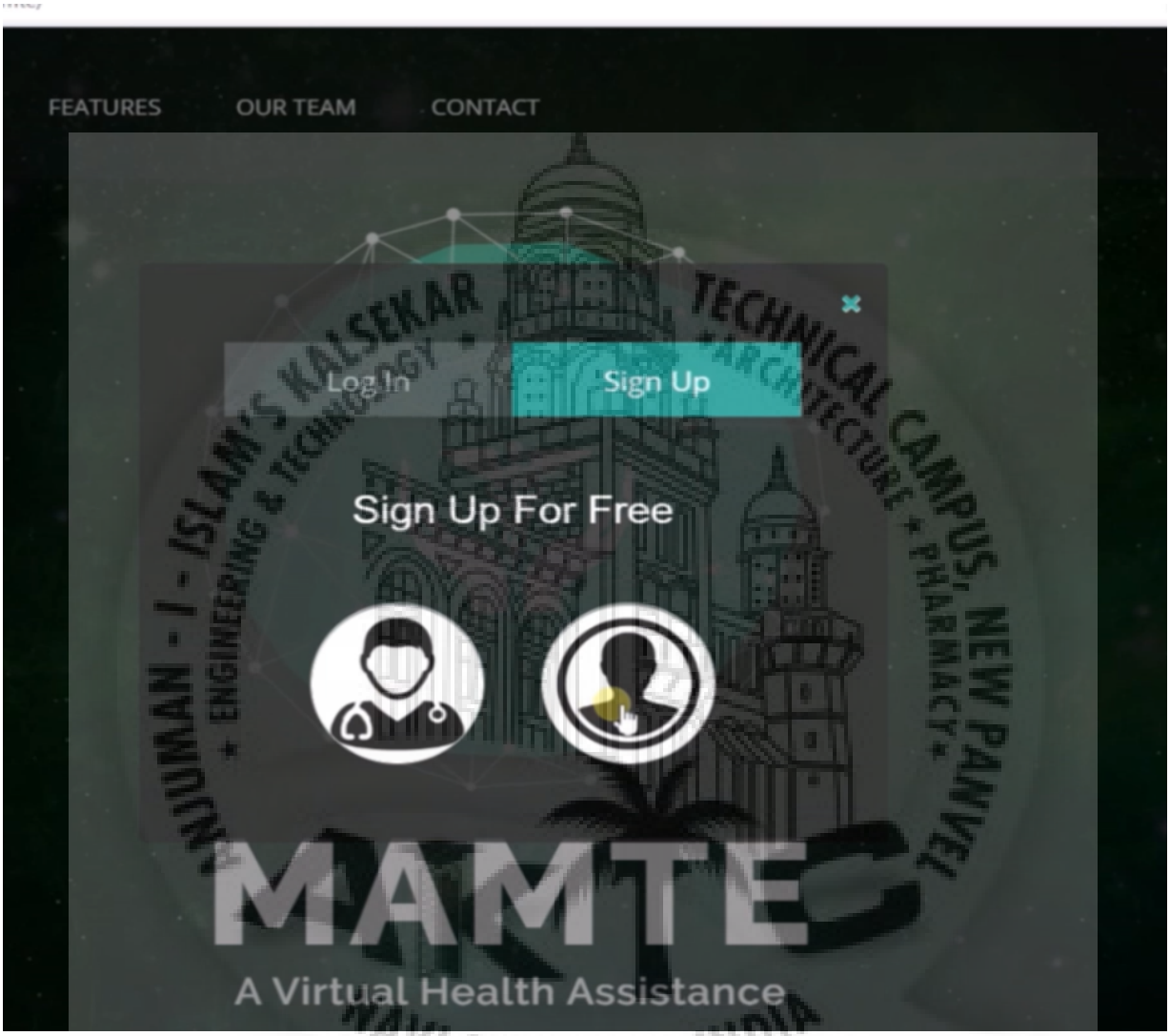

Figure 8.4: Selecting Mentor to signup

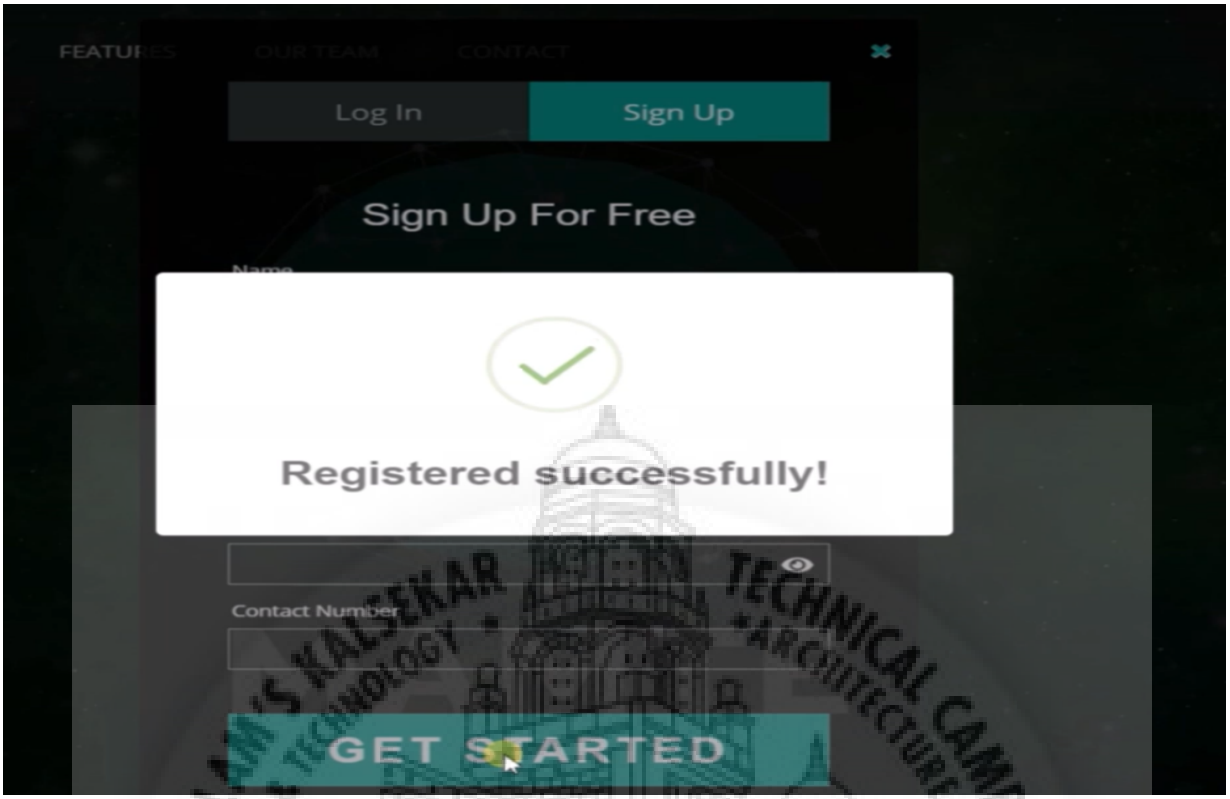

Figure 8.5: Registration Successful

Figure 8.6: Internally QR generated at the backend

8.3 Login

Figure 8.7: Logging in through QR and gaining access of the system

8.4 First Login

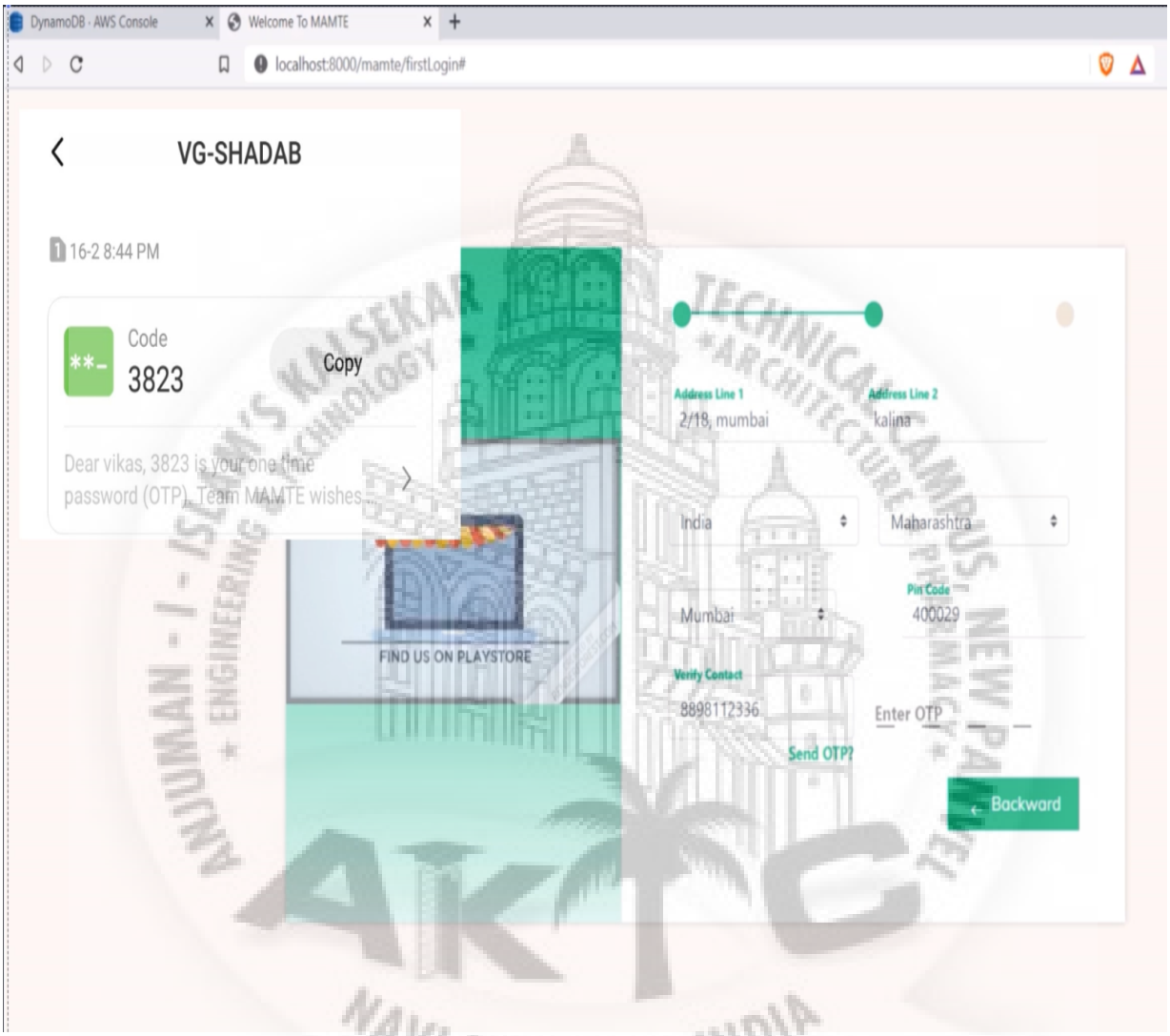

Figure 8.8: OTP sent via the system during first login

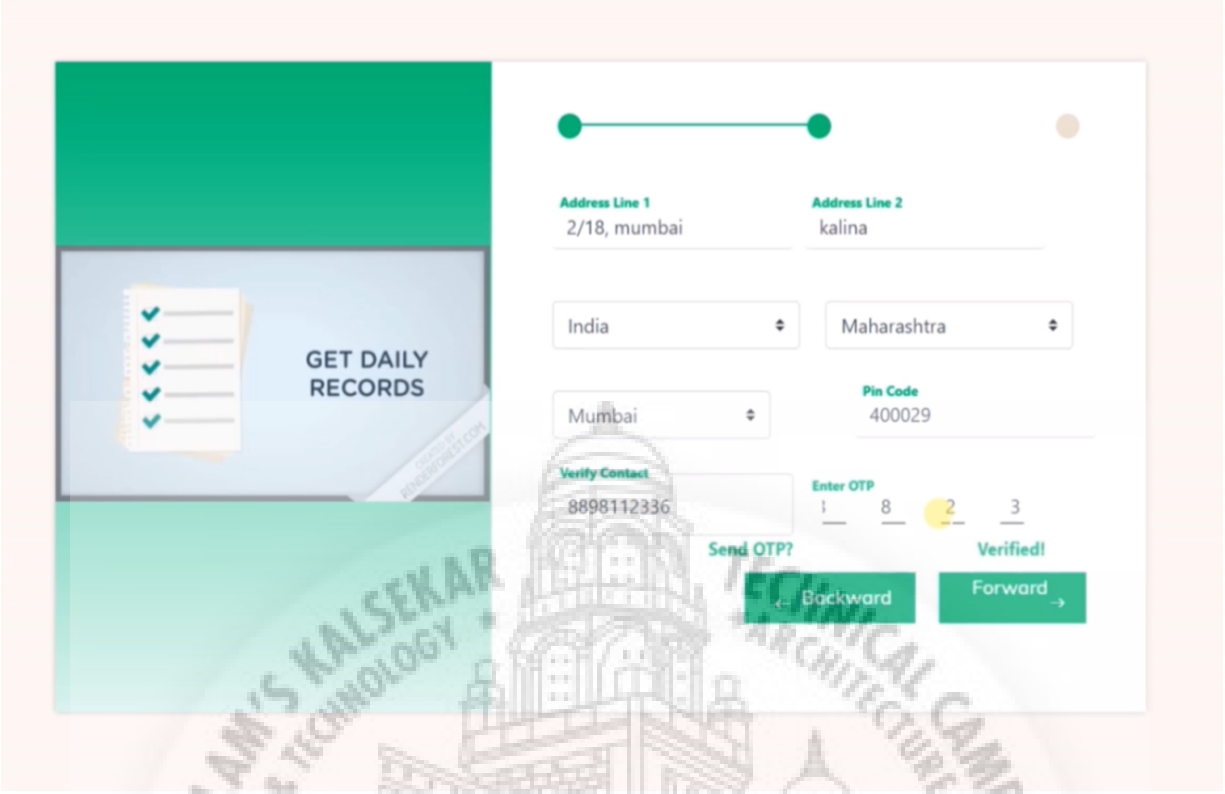

Figure 8.9: Verified Successfully

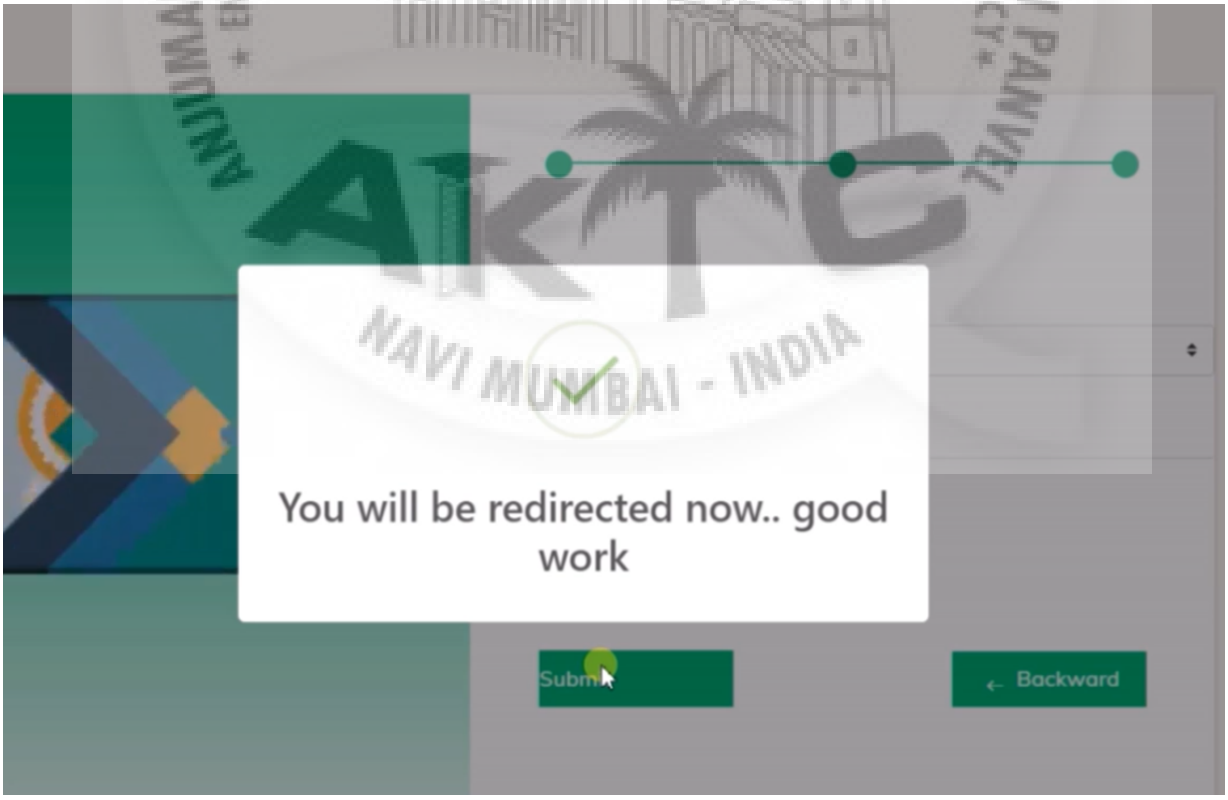

Figure 8.10: First login completed successfully

8.5 Adding new patient

Figure 8.11: New patient added by the mentor successfully

8.6 Text Detection & Recognition

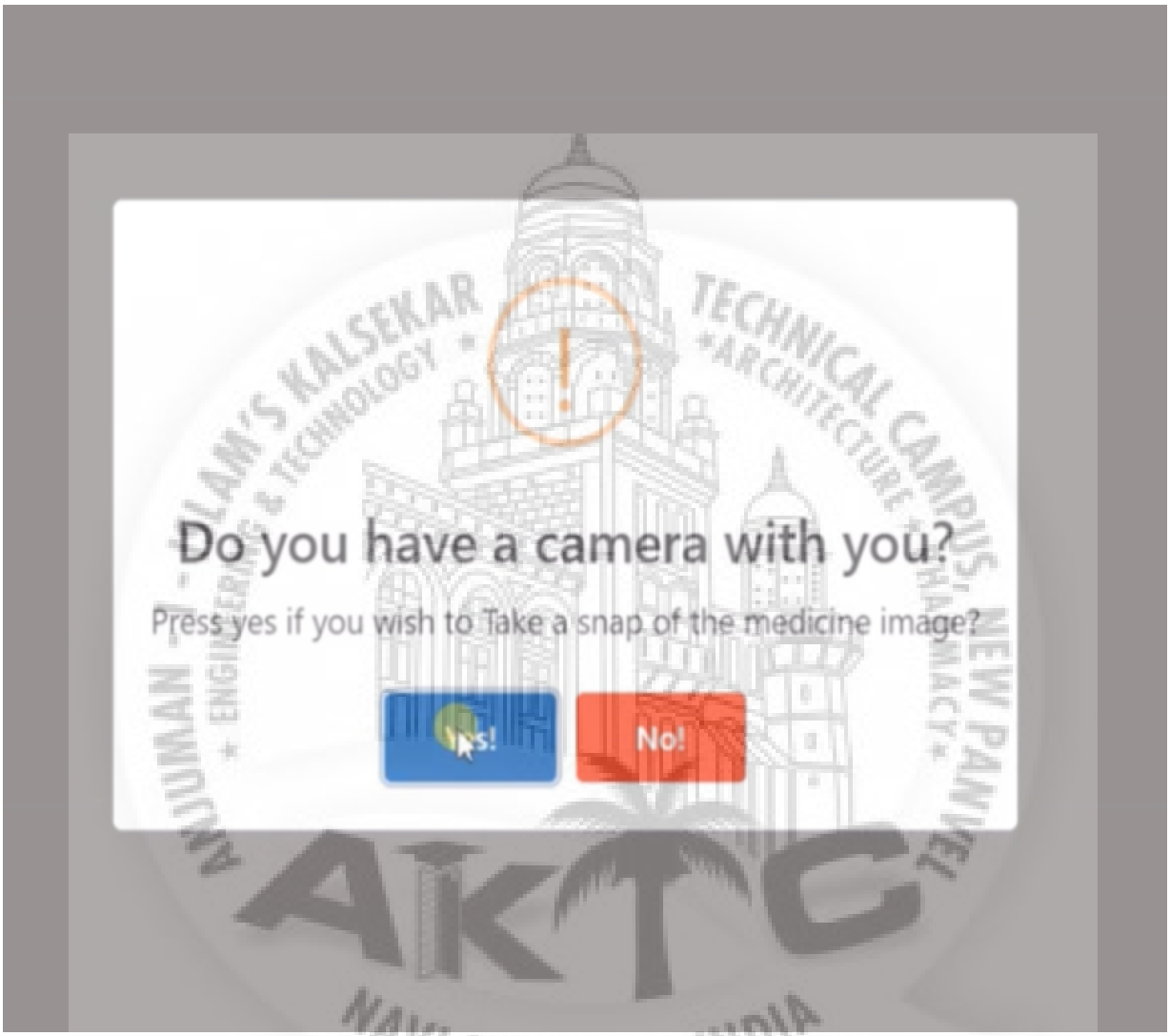

Figure 8.12: Clicking on yes to open the device camera

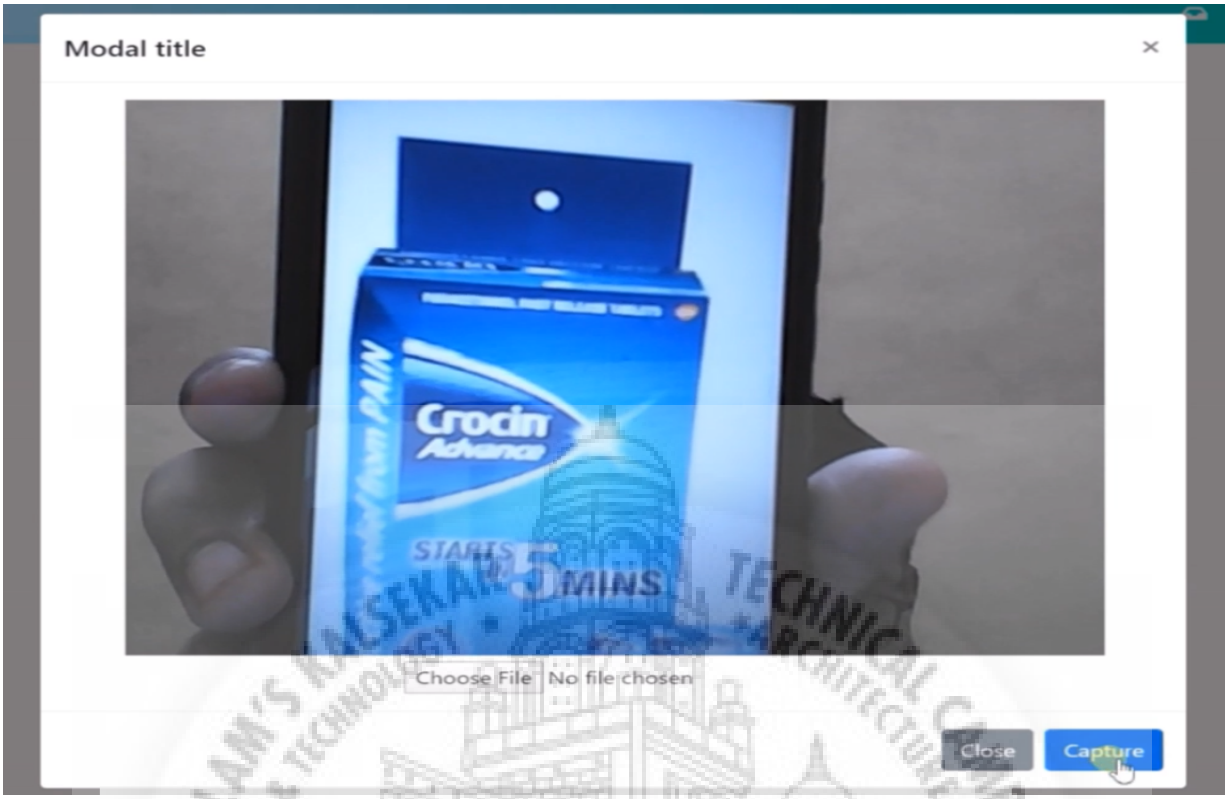

Figure 8.13: Placing the image sample which has medicine name in front of the camera

Figure 8.14: Medicine name recognized and set on the medicine entry form element

8.7 Scheduler & Event Tracker

Figure 8.15: Giving timezone input to the scheduler via medicine entry form

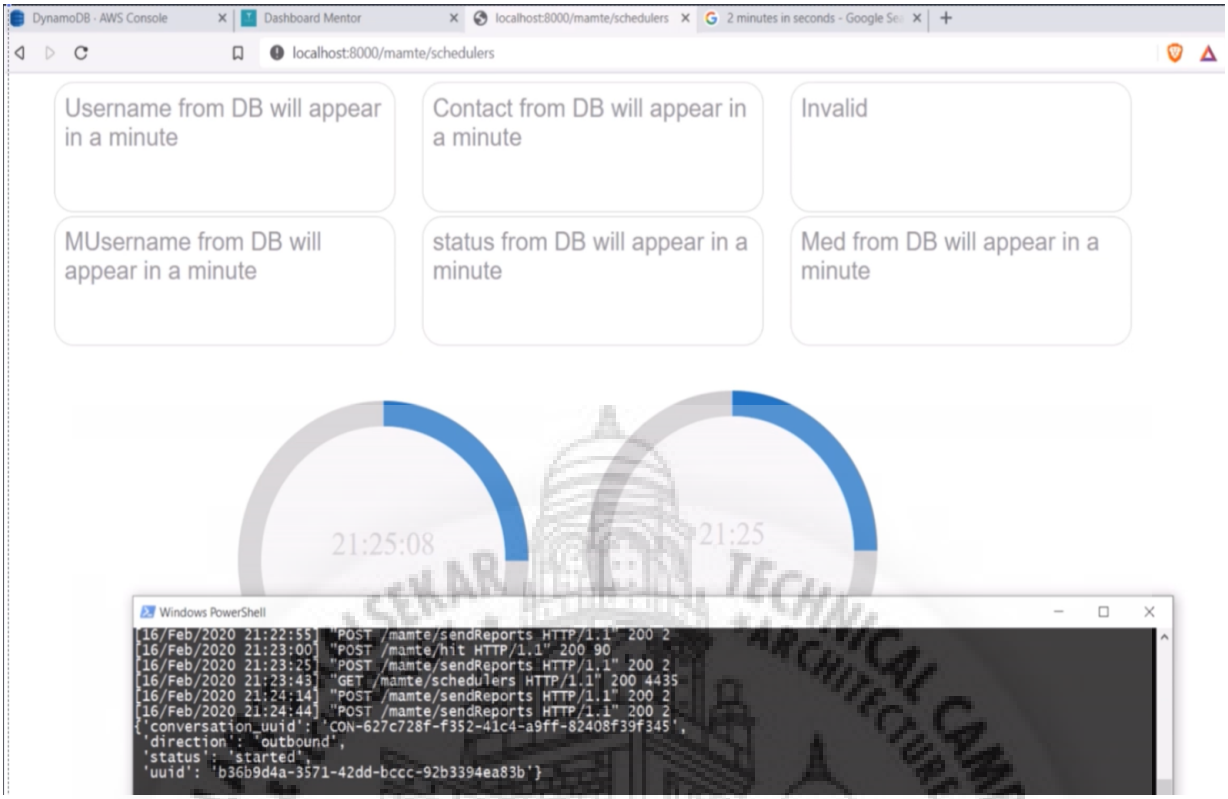

Figure 8.16: Scheduler panel along with Calling notification through JSON Response

Figure 8.17: Event Tracker maintaining and tracking log of events internally

8.8 Report inventory & Analytics

Figure 8.18: Reflecting the medication data through various charts

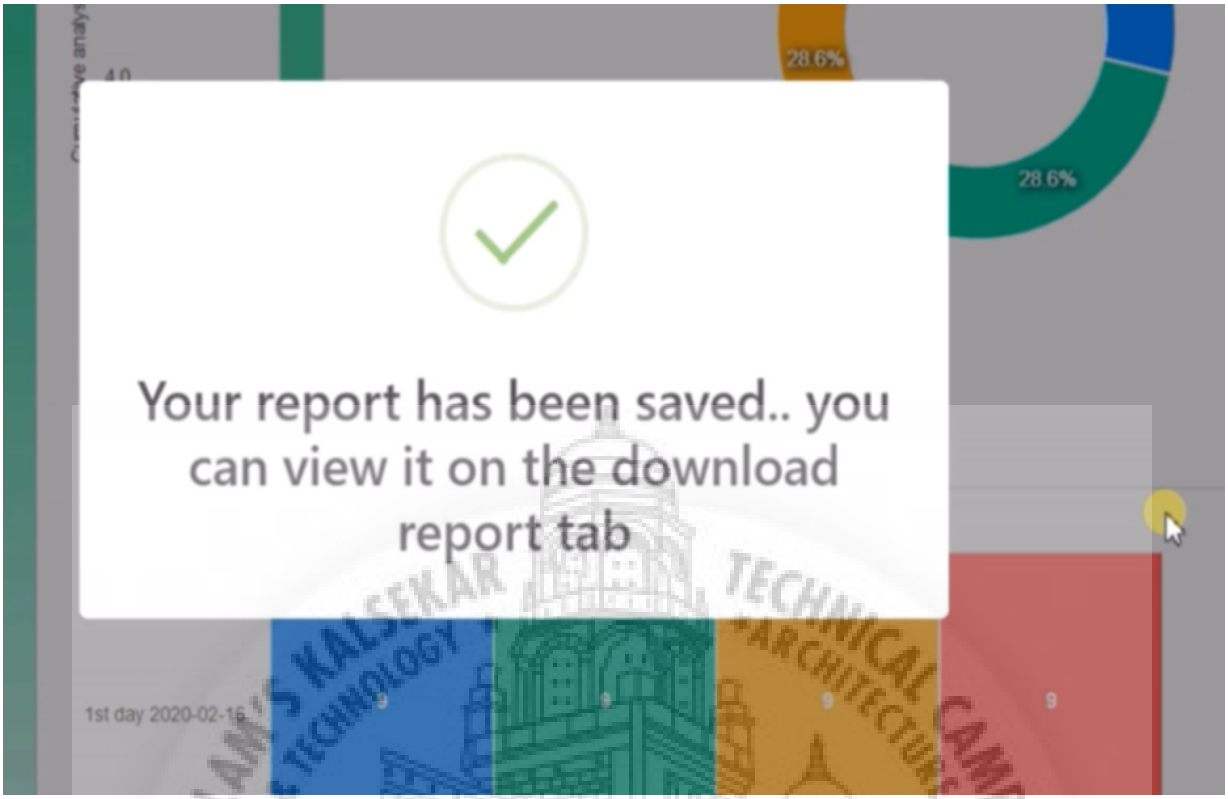

Figure 8.19: Saving the report in the pdf format

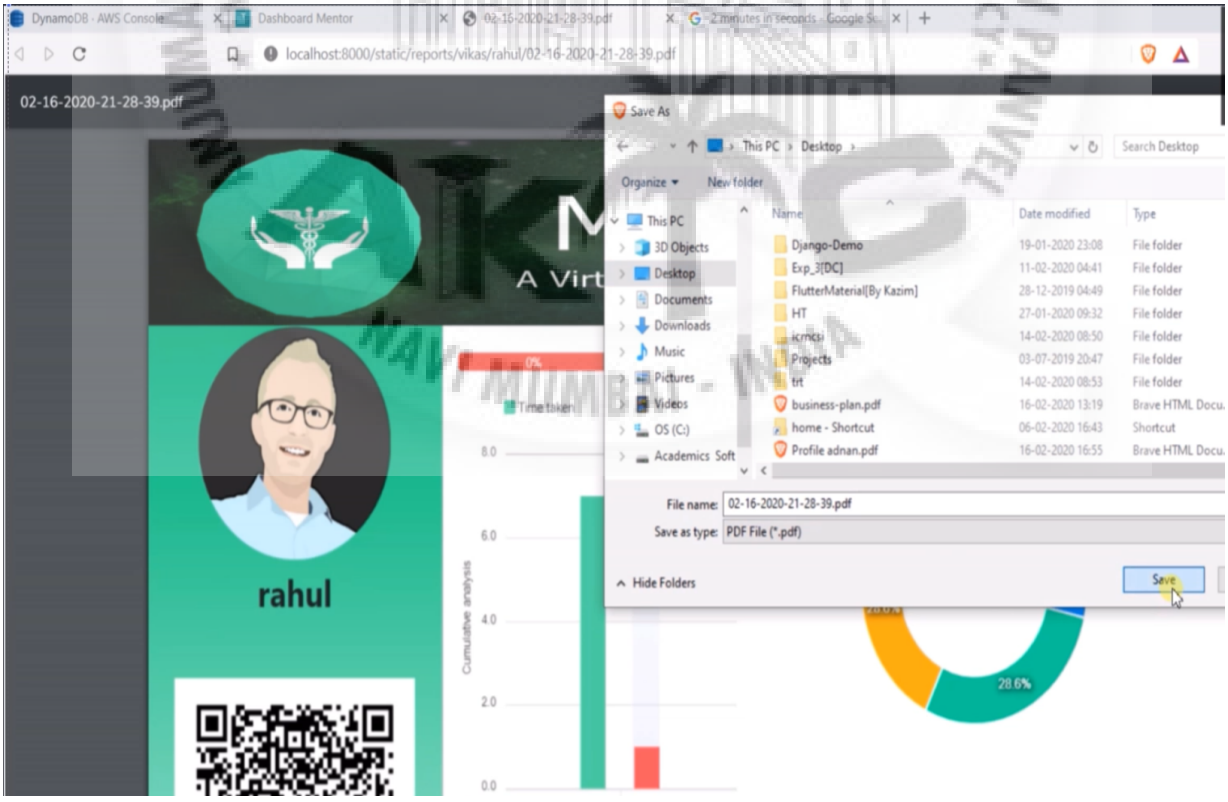

Figure 8.20: Downloading the saved report

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

Medication Adherence Monitoring with Tracking automation & Emergency assistance (MAMTE), is an health-care oriented cross platform digital solution to tackle the medication non-adherence related issues. The causes of MNA are huge, its affected people are increasing, the need of the hour is to have such platforms that could help patient to take care of their medication event needs, for parents/mentor track & automate their dear ones daily routine activity. Our application strives to be the top contender given that latest need of the hour technologies such as deep learning, A.I, Web-REST API's being present in our system. The key module being automated dynamic calling reminder of dosage, daily generation of report, Geo-location operation such as current location finder, Speedometer, NLP based medication details extractor from images, A.I Driven chat-bots, live consultancy from doctor, QR-login etc.

9.2 Future Scope

IOT Device technologies can also be integrated which can come in handy where in if for a given case an accident has occurred,

- Introducing SOS system with the movement tracking of a running vehicle.
- IOT sensors will timely notify and emergency assistance will be provided to the victim.

References

[1] *Emerging IT for Medication Adherence*; Gregory Gimpel , Upkar Varshney, Punit Ahluwalia, IT Professional, May/June, 2016.

[2] *A Medication Adherence Monitoring System for People with Dementia*; Vasily Moshnyaga, Masaki Koyanagi, Fumiyuki Hirayama, Akihisa Takahama, Koji Hashimoto, Cybernatics, October, 2016.

[3] *EAST:An Efficient and Accurate Scene Text Detector* ; Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang , IEEE Computer Society Conference on Computer Vision and Pattern Recognition. , July, 2017.

[4] *Gamification of Medication adherence in epilepsy*; Mohammad Izzat Abdul Rahima, Rhys Huw Thomas, ELSEVIER, September, 2017.

[5] *The Unmet Challenge of Medication Nonadherence* ; Fred Kleinsinger, The Permanente Journal , May, 2018.

[6] Medication adherence: its importance in cardiovascular outcomes. *ncbi.nlm.nih.gov*; Bryson CL, Rumsfeld JS. `https://www.ncbi.nlm.nih.gov/pubmed/19528344`, October, 2019.

[7] Pill Trakk: medication reminder - UX case study *medium.com*; Julie Shukla `http://tiny.cc/qk7xnz`,October, 2019.

[8] Pill Reminder App *medium.com*; Zameer Raza `https://medium.com/@razazameer53/pill-reminder-app-a5e8ad48a1d8`, November, 2018.

[9] A Study of Medication Compliance in Geriatric Patients with Chronic Illnesses at a Tertiary Care Hospital.) *ncbi.nlm.nih.gov*; Shruthi R1, Jyothi R2, Pundarikaksha HP3, Nagesh GN4, Tushar TJ1. https://www.ncbi.nlm.nih.gov/pubmed/28208878, October, 2019.

[10] OpenCV Text Detection (EAST text detector) *pyimagesearch.com*; Adrian Rosebrock https://www.pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector/, October, 2019.

[11] 15 Frightening Stats on Medication Adherence) *pillsy.com*; Jeff L https://www.pillsy.com/articles/medication-adherence-stats, October, 2019.

[12] Pill Reminder – Medication Tracker App Development Key Features & Cost) *brsoftech.com*; Admin (Brsoftech) http://tiny.cc/vn7xnz, October, 2019.

[13] Medication Reminder Apps That Think Outside The (Pill)Box) *brsoftech.com*; Lynsey Gray https://cutt.ly/Gys2bOc, October, 2019.

[14] 5 APPS THAT IMPROVE MEDICATION ADHERENCE) *blog.evisit.com*; Teresa Iafolla https://blog.evisit.com/top-apps-improve-medication-adherence, October, 2019.

[15] The Top Medication Reminder Apps for Patients) *pharmacytimes.com*; Tyler C. Dalton https://www.pharmacytimes.com/contributor/christina-tarantola/2017/12/the-top-medication-reminder-apps-for-patients, October, 2019.

Achievements

1. Publications
   (a) *Medication Adherence Monitoring with Tracking automation & Emergency assistance*; Shadab Ali Shaikh, Obaid Kazi, Mohd Adnan Ansari, and Romaan Shaikh, International Conference on Mobile Computing and Sustainable Informatics - EAI/Springer Innovations in Communication and Computing, January, 2021. (https://www.springer.com/gp/book/9783030497941)
2. Conferences
   (a) *Medication Adherence Monitoring with Tracking automation & Emergency assistance*; Shadab Ali Shaikh, Obaid Kazi, Mohd Adnan Ansari, and Romaan Shaikh, International Conference on Mobile Computing and Sustainable Informatics (ICMCSI 2020) , January, 2020. (Venue : HOTEL HIMALAYAS 2141, Sahid Sukra Marg, 10, Lalitpur, Nepal.)
3. Project Competitions
   (a) *Medication Adherence Monitoring with Tracking automation & Emergency assistance*; Shadab Ali Shaikh, Obaid Kazi, Mohd Adnan Ansari, and Romaan Shaikh, 6th National Level Project Exhibition cum Poster Presentation , March, 2020. (Venue : Universal College of Engineering, Vasai, Mumbai.)