RESPIRATORY AILMENT DETECTION AND RECOVERY APP

Final Year Project Report

Submitted by

Rohit Bhatia (I006)
Deepak Chaudhary (I010)
Osama Dalwai (I012)
Varad Gandhi (I014)

Under The Guidance Of

Prof. Dhanashree Huddedar

In fulfillment for the award of the degree of

MBA TECH.

INFORMATION TECHNOLOGY

At

Department of Information Technology
Mukesh Patel School of Technology Management & Engineering
NMIMS (Deemed –to-be University)
JVPD Scheme Bhaktivedanta Swami Marg,
Ville Parle (W), Mumbai-400 056.

April, 2023

CERTIFICATE

This is to certify that the project entitled "**Respiratory Ailment Detection and Recovery App**" is the bonafide work carried out by Rohit Bhatia (I006), Deepak Chaudhary (I010), Osama Dalwai (I012) and Varad Gandhi (I014) of B Tech (IT), MPSTME, Mumbai, during the VII Semester of the academic year 2022-23, in fulfillment of the requirements for the award of the degree of Bachelors of Technology as per norms prescribed by NMIMS. The project work has been assessed and found to be satisfactory.

(Signature of Internal Mentor)
Name: Prof. Dhanashree Huddedar
Designation: Assistant Professor,
Dept. of IT

(Signature of External Examiner)
Name:
Designation:

HOD (IT)
(Dr. Ketan Shah)

Dean
(Dr. Alka Mahajan)

DECLARATION

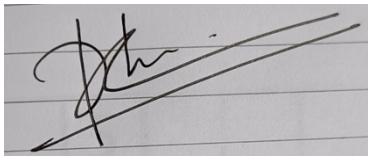
We, Rohit Bhatia, Deepak Chaudhary, Osama Dalwai and Varad Gandhi roll numbers: I006, I010, I012 and I014 respectively, understand that plagiarism is defined as any one or the combination of the following:

1. Uncredited verbatim copying of individual sentences, paragraphs or illustrations (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.
2. Uncredited improper paraphrasing of pages or paragraphs (changing a few words or phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did or wrote what. (Source: IEEE, The Institute, Dec. 2004)

We have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of our work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.

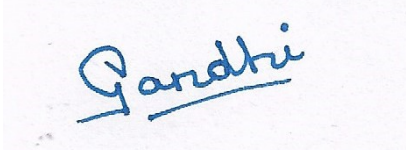
We affirm that no portion of our work can be considered as plagiarism and we take full responsibility if such a complaint occurs. We understand fully well that the guide of the seminar/project report may not be able to check for the possibility of such incidences of plagiarism in this body of work.

Signature:
Name: Rohit Bhatia
Roll No.: I006
Date: 1/4/2023



Signature:
Name: Deepak Chaudhary
Roll No.: I010
Date: 1/4/2023

Signature:
Name: Osama Dalwai
Roll No.: I012
Date: 1/4/2023



Signature:
Name: Varad Gandhi
Roll No.: I014
Date: 1/4/2023

ACKNOWLEDGEMENT

We would like to express our gratitude towards the members of the IT division, whose guidance throughout the project journey has been important for us to complete the same. The acknowledgement of all faculties who have been helpful throughout our journey and to whom we have approached to.

Most importantly we express our sincerest gratitude to our mentor professor Prof. Dhanashree Huddedar for her guidance, constant advice, meaningful insights and information, patience, and most importantly her belief in us which inspired us to keep pushing further in our research and development of our entire project. Her mentorship has been a vital aspect for the success of our project for which we are grateful for and we thank you for the way you lead us and had trust in us.

Abstract

In the wake of COVID, healthcare took a new perspective. People are now more concerned about their health and the possible effects of different diseases than ever before. But informal communication channels and unverified information circulating all around has increased the circumstances where people are relying more than these or first on these than on certified medical professionals. It has never been more important to integrate the overall healthcare resources such that people are no longer afraid and no longer have to look for other avenues for medical guidance.

Through this report, we wish to propose an application that combines the traditional healthcare offering with modern features that enable 24x7 care without creating burden on the existing healthcare infrastructure. It can be considered as a way of optimizing healthcare for a wider and more cost-effective reach while giving patients a greater control over their health and wellbeing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Overview

3.2 Project Specifications

COVID was the most recent pulmonary ailment that had a global impact in terms of lives and livelihoods lost. COVID-19, caused by the SARS-CoV-2 coronavirus was not the first mass respiratory disease that affected the world (with history being witness to large outbreaks like SARS and MERS in the past). However, it still created a global level panic and fear that was fueled by uncertainty and lifestyle restrictions. Amidst all of this, the healthcare industry was working ever so hard to figure out this new strain and its treatment, treat patients who had contracted it and on developing the vaccine, convince the world to take it. On the other end, people who were scared, in most cases rightfully so, fearing what would happen if they got COVID, whether they would be able to identify it, whether they could trust hospitals in respect of treatments, whether they could themselves follow treatment like social media suggested (Was it that simple? Could just taking steam be enough to prevent COVID?) etc.

Healthcare struggled not because of capabilities or lack thereof but because of lack of communication, fear and misplacement of trust (reliance on false or insufficient or inefficient claims and social media instead of authentic medicinal treatments). Thus, there is a great need for a platform that brings together all the elements of the healthcare infrastructure, providing maximum reach to existing resources while eliminating geographical, educational and communication barriers.

1.2 Objectives and scope

The overall objective of the project is to create a streamlined and effective platform through which individuals can be more knowledgeable of their anatomy to identify symptoms on time, get the appropriate medical attention and thus create a synchronized healthcare system.

To do this, the scope of the project will be as follows:

a. Creation of a **secure user profile** for patients, doctors and other medical professionals to encourage participation without confidentiality concerns;
b. To include a **symptom monitor** through checklists and questionnaires to guide early action;
c. **Reminders and follow-ups** will facilitate better healthcare at home where users can provide their medical conditions and records and they will be prompted about their medicines, appointments, diagnostics, follow-ups etc. at appropriate intervals;
d. Doctor written and doctor approved **medical blogs** that give users authentic, reliable and medically sound suggestions;
e. **Geolocation based analysis** to identify potential mass spread of diseases in an area and accordingly suggest precautionary care; and
f. **Medicine substitution** recommendation based on symptom evaluation and historical medical records of patients.

1.3 Literature Survey

a. A. K. Singh, "A Comparative Study on Disease Classification using Machine Learning Algorithms," SSRN Electron. J., no. August, 2019, doi: 10.2139/ssrn.3350251
One of the primary features of the application will be the use of algorithms to identify symptoms and predict ailments. The paper mentioned above talks about liver disease detection and thus similar principles can be easily translated for any other dataset. It discusses Fuzzy Logic (providing an overall accuracy of 58.8%) and Fuzzy Neural Networks (providing an overall accuracy of 91%) as means of doing so, giving a partial conclusion that neural networks may provide a better prediction accuracy.

b. G. C. Hansen, K. H. Falkenbach, and I. Yaghmai, "Voice recognition system.," Radiology, vol. 169, no. 2, p. 580, 1988, doi: 10.1148/radiology.169.2.3175016
This research paper explores the use of the Hidden Markov Model for building models for mapping letters from processed audio signals. It also identifies the Mel Frequency Cepstral Coefficients (MFCC) for feature extraction. Such research will facilitate us in development and deployment of speech-to-text and text-to-speech features in our app which are aimed at eliminating the communication and accessibility constraints.

c. J. K. Kairi and A. K. Sharma, "Substitutes of prescription medicines – A review of concerns relevant to doctors and patients," J. Clin. Diagnostic Res., vol. 11, no. 8, pp. FE01–FE05, 2017, doi: 10.7860/JCDR/ 2017/28552.10439
The paper identifies substitute medicines as being either of generic and therapeutic nature and the care to be taken in suggesting substitutes. Generic substitutions are solely based on generic brand properties and basic information provided by the institutions. Therapeutic substitutions face the issue of ambiguity due to the differences in the side-effects on the patient. Patients often find that they are unable to get the prescribed medicine due to reasons beyond their control and they end up buying a substitute for the medicine without consultation of their doctor (to judge effectiveness, suitability etc.). Therefore, the above paper provides guidance on the specificity of generic and therapeutic medicines. Further, such suggestions have to be made carefully by understanding the biology of the patient and maintaining the precision in the process to provide a safe and permanent solution.

d. B. Pandipati and R. P. Sam, "Speech to text Conversion using Deep Learning Neural Net Methods," vol. 12, no. 05, pp. 2037–2042, 2021
Accessibility is a major feature introduced in our proposed application that eliminates communication barriers by enabling people understand their conditions better and thus facilitate them in their communications with medical professionals. Many a times, the inability to communicate properly with the doctor discourages people from seeking advice as they feel embarrassed or misunderstood. Thus, we use this research paper to understand how technology can be leveraged to deploy hands-free speech-to-text feature. The Hidden Markov model has particularly enabled effective development of the feature. Deep neural network has been identified to have a remarkable effectiveness in understanding the spoken text with a 94% accuracy.

e. C. Huang et al., "Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China," Lancet, vol. 395, no. 10223, pp. 497–506, 2020, doi: 10.1016/S0140-6736(20)30183-5

COVID has triggered the need for understanding our anatomies better and is the recent event that has motivated this project. Moreover, it is among the pulmonary conditions for which we intend the app to be useful to users. Thus, we identified this paper which covers the identification of COVID in people in Wuhan, China. It covers aspects such as testing, effects of the disease, the statistics of the infected people and their demographics, the symptoms, complications, effect on biology (such as concentration of cytokines) etc. It also covers a sufficiently detailed analysis of the methods and procedures used in the testing.

1.4 Expert Discussions, Opinions and Feedbacks

The motivation behind the project is rich in human emotion and the aim is to help people get better access to healthcare while not adding a burden on the existing healthcare infrastructure. Thus, it was critical that we understand firsthand from medical professionals how to achieve this. We conducted interviews with Dr. Akshaya Rane (M.D Ayurvedic, BKC Covid Centre) and Dr. Jasmin Dalwai (BHMS, MBA Healthcare Management, Pune) to understand what difficulties doctors and medical professionals face while treating patients, what are the typical fears and skepticisms that obstruct a smooth treatment process and how treatment is impacted by external factors and outside influences.

The idea and proposed elements of this project were discussed with them and based on their feedback, the following points were agreed:

a. The application can help doctors treat a larger number of patients across a wider geographic area that would typically be out of their reach;
b. The doctors can get a pre-assessment of patient conditions, streamline treatment process and improve efficiency of healthcare resources that would otherwise be typically congested;
c. The application can help doctors and patients communicate more effectively, eliminating a lot of communication barriers;
d. Real-time monitoring of patient data will help in improving and modifying treatment plans on time; and
e. The doctors believe that patients will genuinely and largely benefit from authentic medical blogs written and verified by doctors and will received much better instructions about self-care measures.

The general feedback was that the idea for this project has the potential to improve communication and interactions between doctors and patients, improve access to healthcare infrastructure and optimize treatment processes within the existing healthcare infrastructure and free up a lot of resources to serve a wider range of public. Moreover, it was identified that the project could have future potential to cover a larger part of healthcare services.

1.5 Comprehension and Culminating Efforts

The initial project scope work and background studies performed were critical in the complete comprehension of the full extent of features through which the application will be able to serve all users. This will thus define the scope of the project and the component tasks and modules in the actual project. These are discussed in Chapter 2, Section 2.1 "**Objectives, Scope and Selection Criteria of Tasks**".

Chapter 2: Analysis and Design

2.1 Study and Analysis

To understand the main problem of the public (i.e., patients, families etc.), we tried to use empathizing techniques (such as profiling and interviews) to understand common pain points of these prospective users. The following were noted:

a. Patients are often skeptical about getting medical attention because they have a genuine fear of hospitals or lack trust in the process;
b. Patients generally rely on social media or some information medium for self-care remedies as a first course of action, prior to seeking professional medical advice;
c. Patients find that accessing the appropriate professional is difficult due to geographic or communication barriers;
d. Patients often find it difficult to communicate with medical professionals and this discourages them from seeking medical advice;
e. Patients find it a long and cumbersome process to get treated for severe conditions and their trust in the treatment decreases as its duration increases.

Thus, understanding the existing problems at hand, we categorized the potential stakeholders of the applications for whom our offering would serve as a tipping point towards a trustable source of medical infrastructure. The stakeholders are categorized into:

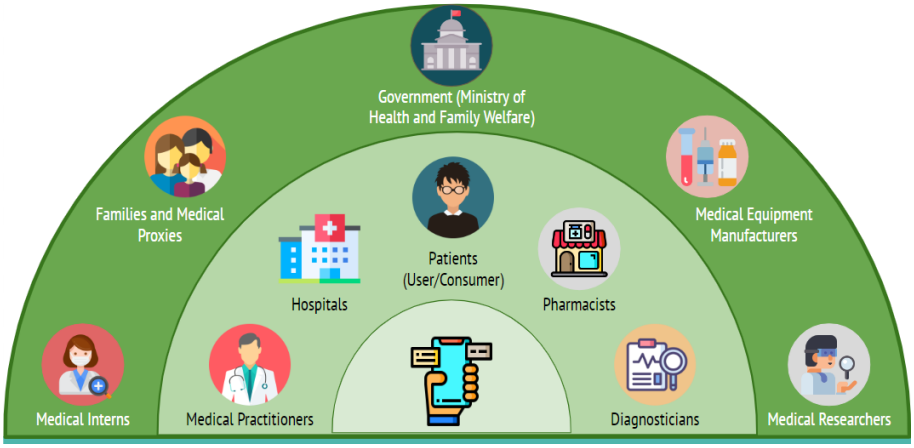


Figure 1: Stakeholder Canvas

a. Primary stakeholders: These are the direct population, that are concerned and will be heavily impacted by the application that enable them to have quicker access to resources, better infrastructure and are the closer ones to the application as shown in the above figure.
b. Secondary stakeholders: Who are relatively concerned with the availability of the data to work upon such that they are up to date with what goes and as such can be prepared on what to expect from the data available to them.

An initial summarized understanding, in the form of a user profile canvas, based on an interview conducted is given below:

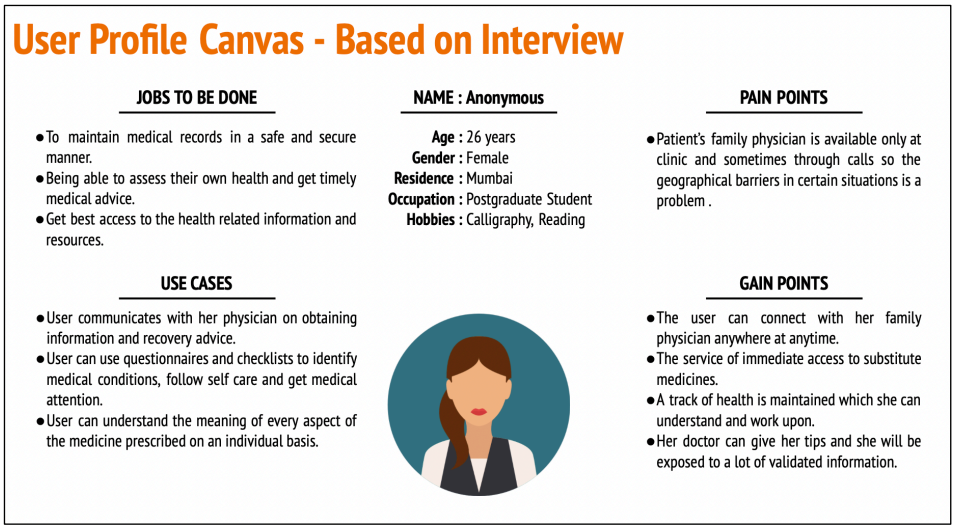


Figure 2: User Profile Canvas

Based on the activities and tasks to be performed from the user through the prototype given, we then understood the interactions of a potential user with the touchpoints, noted down their actions and experience to jot down their emotions all mapped to a user journey map to understand how they go through with all phases of the application

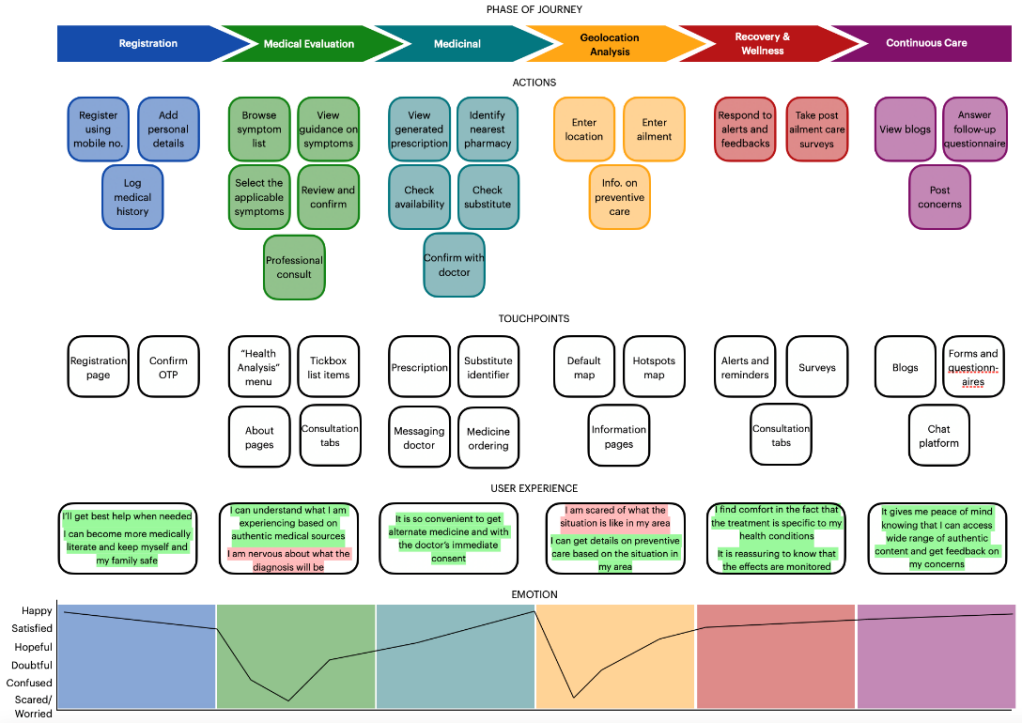


Figure 3: User Journey Map

2.2 Feasibility and Prosperity

Discussions with medical professionals, the above mentioned aspects and a comparative analysis with existing related services highlighted the following potential key success factors for the project:

a. Scope of existing applications is majorly focused on finding doctors and hospitals nearby, appointment booking, and online medicine ordering. Our application focuses on an end-to-end healthcare provision ensuring an effective treatment is monitored and post-ailment care is observed.
b. Our application also focuses on streamlining and optimizing existing healthcare infrastructure to relieve their burden while ensuring that everyone gets the care they deserve.
c. The application will not replace traditional pharmacies but will rather develop and integrate with existing pharmacies to create a supply chain that also supports and motivates them.
d. The application is focused on treatment effectiveness and efficiency and aims to disseminate medical knowledge and medical care under the best supervision and with the highest levels of authenticity.

We have also conducted the following feasibility study too to understand the technical, design and the market feasibilities to get an understanding of the best suited aspects of our application.

Technical Feasibility

- The android mobile application "Radar" is easy to develop using the dart language with the Flutter framework with the feasibility of deploying on a cross-platform basis suitable to initially Android platforms – particularly Android smartphones irrespective of their dimensions.
- The development of this application would require Google collab for Machine Learning and NLP translations implementations, Flask for API, Android emulators for testing the mobile application, MongoDB for database services, Postman for testing and debugging of the APIs.
- The front end has been first developed using Figma and then the same has been carried out on Flutter and with the help of Flask APIs we have made the backend to frontend connectivity to display the complete application.
- Once, the functionalities being concurrently done are completed we would then proceed on towards their integrations and then would test these for overall performance.

Design Feasibility

- Our focus would be on simplifying with a subtle the interface such that the user has a convenient experience where we would be providing visual cues to the users for navigation purposes and the feature of voice to text commands – allowing both audio and text controls, such that they have no issues in understanding the flow of the application.
- While designing the interface, making sure that the interface is consistent in terms of the color, fonts, and arrangement of content.

- Making sure that the icons, fonts and colors are easy to see to the users such that the user has no difficulty in understanding their purpose, flow and that it becomes easier to navigate through the application

Operational and schedule feasibility

- With respect to the initial plan of respiratory ailments – "RADAR", it can be said with confidence that it would require a span of roughly 8 months to fully develop, with initial months based on conceptualization, understanding requirements etc. and the latter months for the development and testing.
- The operations of the application are straightforward with each functionality having their respectable cues available, reducing the confusion and time taken for them to complete a walkthrough. Also, backend fetches and computations are well within estimated time parameters ensuring a smooth flow of the application

Market Feasibility

With studies claiming that in 2021, the health and fitness category of digital applications was one of the top 5 most growing mobile app categories and thus with our unique offerings we intend to extend on this growing branch offering a mechanism for the doctors and users to reduce the level of panic and urgencies.

2.3 Frameworks and Process Flow

In line with the objectives and scope mentioned in this document, this section documents various diagrammatic representations of different aspects of the project.

System Architecture

The system architecture depicted below has the different modules pertaining to the functionalities which we intend to provide through our application. The datastore will act as a common store for patient/doctor activity and information although every component will have its own dataset for training the modules.

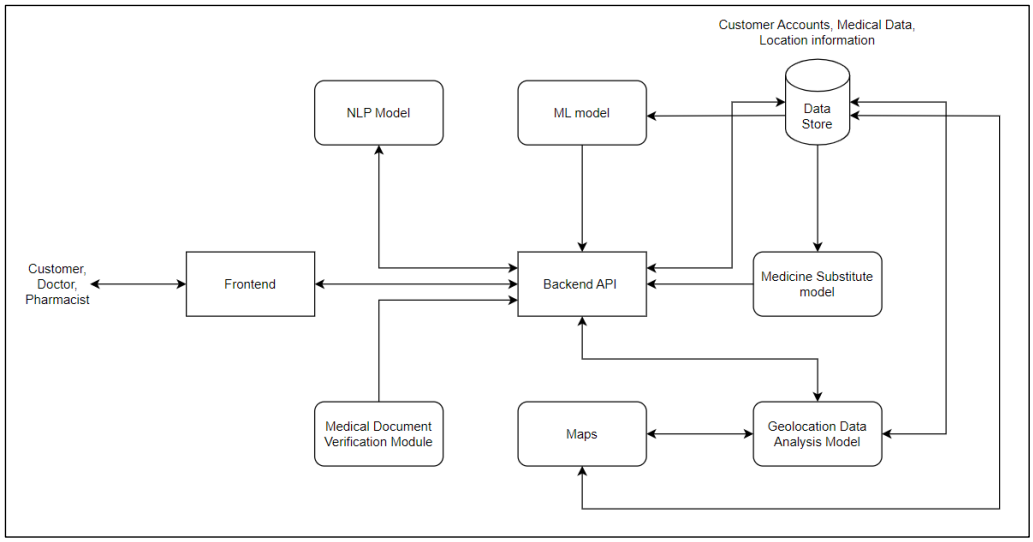


Figure 4: System Architecture Comprising of Users, Front-End, Back-End and Components

Sequence Diagram for Signup/Login

This is the first step in the user's interaction with the application. The user tries to access the application by either checking in the existing account or creating a new account. The Flask API helps in communication between the user and the database for this procedure. Every time the validity of the user is verified so that no person can have two accounts.

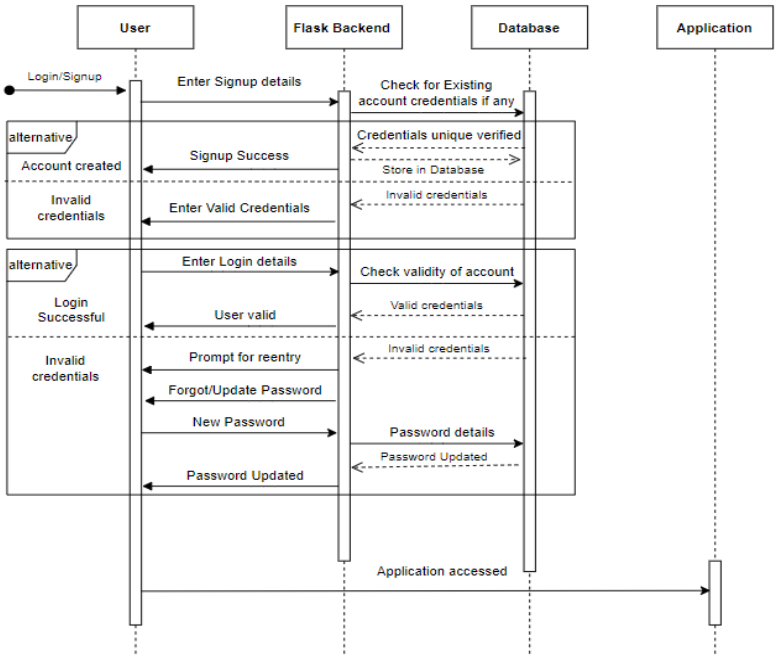


Figure 5: Sequence Diagram for Signup/Login

Sequence Diagram for Ailment Detection

Every interaction in the above sequence takes place through the database since it will act as a common repository with all the information related to the application and its

stakeholders stored there. An alternative case wherein the user may want to view their history is also highlighted in order to provide a base for further treatment procedures.

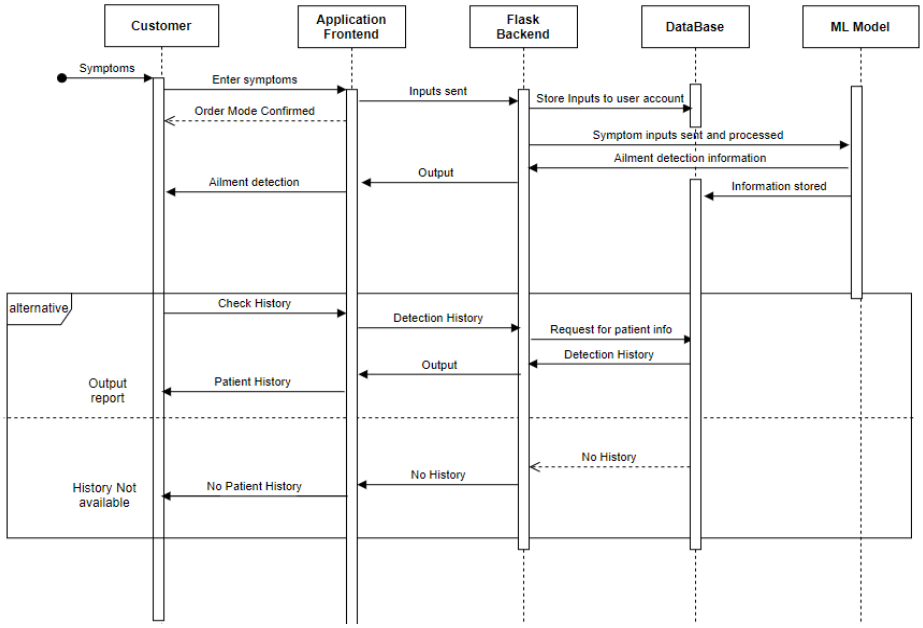


Figure 6: Sequence Diagram for Ailment Detection

Sequence Diagram for Medicine Substitution

The substitute medicine feature module has a separate datastore for all the compositions and their therapeutic effects stored in a single repository. Database will store the entire information to create a history regarding the past medicines searched by the patient for reviewing.

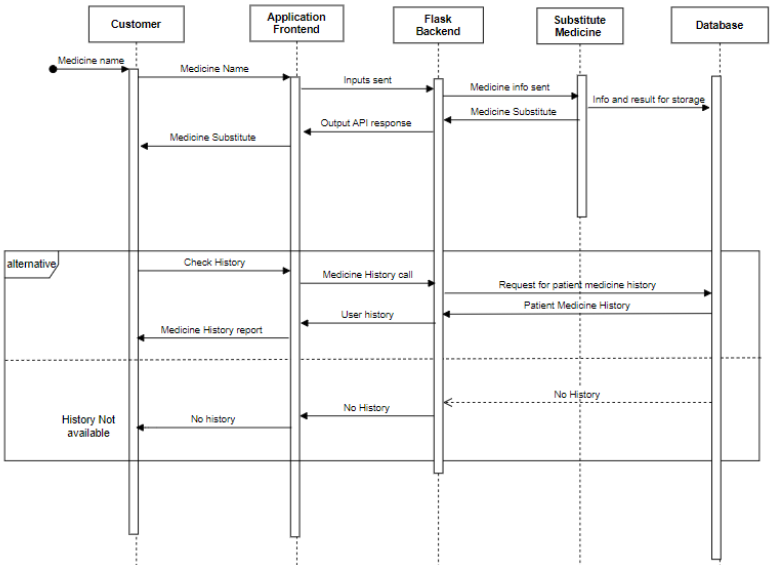


Figure 7: Sequence Diagram for Medicine Substitution Feature

Class Diagram for Major Components

The following class diagram consists of the major components to be present in our application. Every entity is denoted as a separate class with attributes to uniquely identify

the entity. The user class is at the centre of every interaction since all the queries and processes take place on User commands.

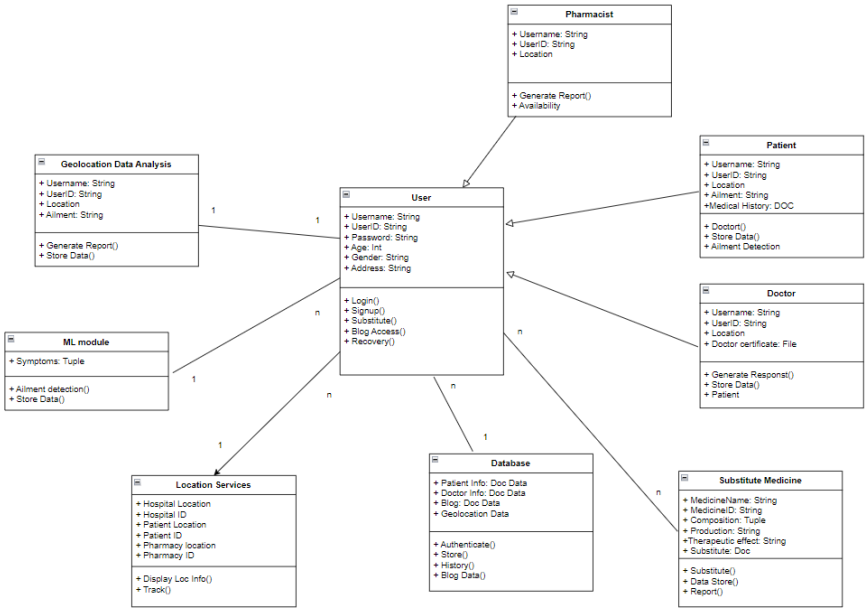


Figure 8: Class Diagram for Major Components

Data Flow diagrams

The data flow diagram provides the entire workflow as to how the data will flow through the system and be processed by the system.

DFD Level 0:

The Level 0 diagram provides an overall view of all the information providers and also the overall processed elements that are going to be processed by the system and provided to the user.

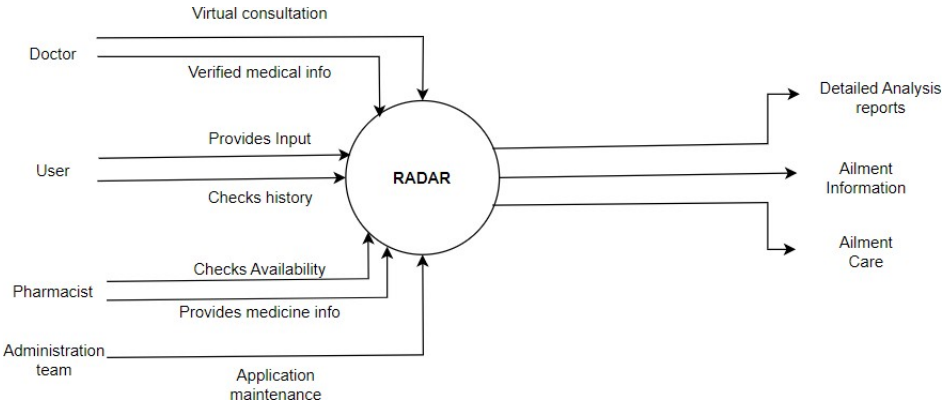


Figure 9: DFD Level 0

DFD Level 1:

The given DFD provides a deeper view into how the data is processed by each component in the system and how the user interacts with the different components

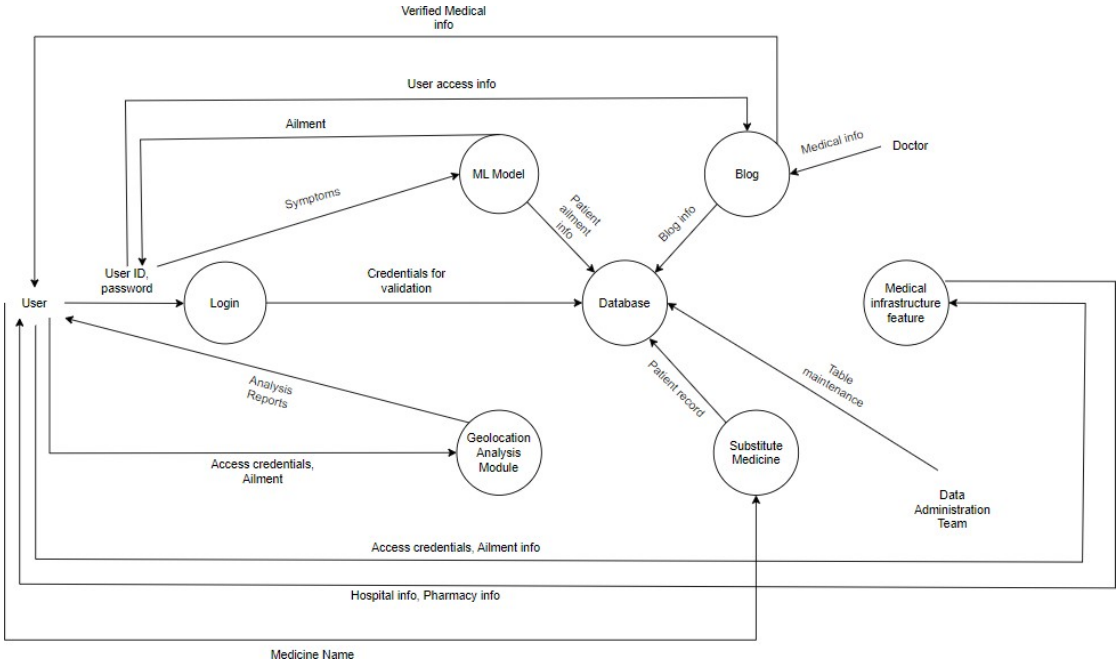


Figure 10: DFD Level 1

Use Case Diagram for Medicine Feature

The above use diagram specifies the overall command interaction between the user and the substitute medicine model wherein the database acts as an intermediary for providing patient medical history to filter out the medicines accordingly.

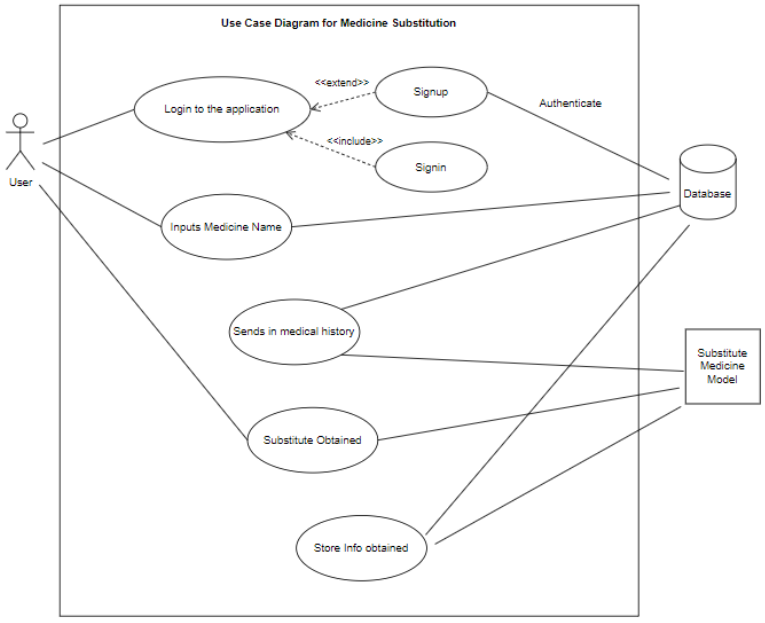


Figure 11: Use Case Diagram for Medicine Feature

Use Case Diagram for Geolocation Analysis

The following use case diagram is about the analysis that we intend to provide to the user based on the severity of the ailment. The presence of certain respiratory factors will be taken into consideration which will be stored by Geolocation Analysis Module and with the help of Maps feature, will filter out a personalized report for the Patient regarding the ailment.

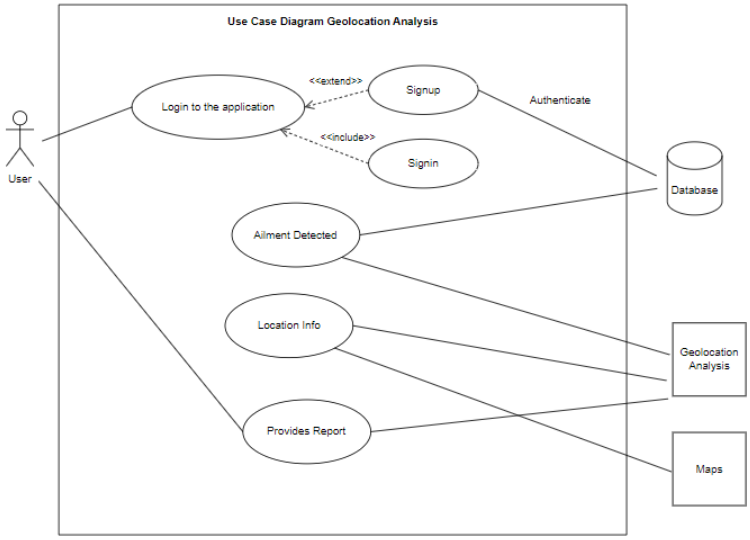


Figure 12: Use Case Diagram for Geolocation Analysis

Use Case Diagram for Ailment Detection

The following Use case diagram shows in how the user interacts with the ML model developed for ailment detection. The database acts as a repository for storing the information provided by the user and the output provided by the ML model which can be viewed later as a history for applying better recovery methodologies.

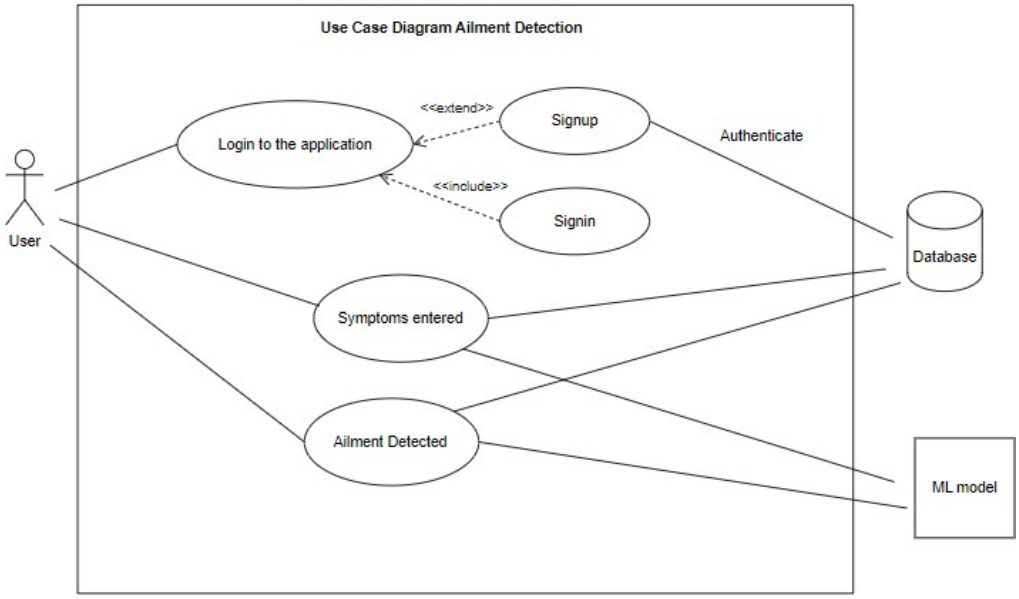


Figure 13: Use Case Diagram for Ailment Detection

2.4 Technologies and software details

Based on the following goals set above, the tools required and the plans to be carried out with these are identified below:

- Android application: The offering provided is a mobile application. While initially being focused on Android applications, the perk of developing through the flutter framework of deploying for cross-functional platforms would be key for increasing the reach of the application.
- FLASK: Provides a web application approach in order to integrate ML classifier codes that we intend to build for our disease classification feature.
- Maps: Will be extensively used for geolocation analysis and also for medicine substitute features in our application.
- Database: MongoDB is the application currently being chosen as the base for creating the database which will store the information related to the ailments and also be useful for storing the user information. Since our application will run on live analysis, accessibility is the key in our project which Mongo DB has the capability to provide.
- Machine Learning Techniques: The different classifiers will be used for our prediction model and decision-making process on which the entire Ailment discovery feature is based upon. Also, certain decision making procedures regarding the ailment care will also be assisted by machine learning in order to make better decisions.
- API: These will be used for integrating the different components altogether to create a chain (network) of services for exchanging data, securing it and also maintain the integrity of the application.

For carrying out the same implementations we have used the following applications:

Flutter

Flutter is a popular mobile app development framework that we used to develop app. With Flutter, we were able to create a beautiful and responsive user interface, complete with custom widgets, animations, and other features. The inbuilt features such as Notification panels and Server State Management along with HTTP Server management have been a critical component considered while selecting Flutter since it is of absolute importance in our medical application.

Flask

Flask is a popular Python web framework that is used to build machine learning models and serve them as APIs. Flask provides a simple and flexible way to create web applications and RESTful APIs, which has been used to deploy machine learning models and make them accessible to users. Flask also consists of the SQL Alchemy datastore which helped to store the server-side state management data for the patient.

To use Flask in a Flutter app, we did following steps:

- Creating a Flask app: This involves defining the Flask app and setting up the routes and endpoints that will be used to receive and respond to requests from the Flutter app.
- Defining API endpoints: This involves defining the API endpoints that will receive requests from the Flutter app and return predictions from the machine learning model.

- Serializing the model: This involves saving the trained machine learning model in a format that can be loaded and used by the Flask app.
- Serving the model: This involves starting the Flask app and serving the machine learning model using the defined API endpoints.
- Integrating the API into the Flutter app: This involves sending HTTP requests from the Flutter app to the Flask app and receiving responses containing predictions from the machine learning model.

APIs have been created for the following features-

- Blogs: Posting and viewing Blog Information
- Geolocation Analysis: Receiving Analysis
- Detection: Receiving Analysis
- Substitution: Getting the medicine information from the datastore

MongoDB

MongoDB is a popular NoSQL database used for storing and managing large volumes of unstructured and semi-structured data. In the context of storing a database of doctors, MongoDB is used to store information about doctors such as their name, contact information, specialty, education, and other relevant details. Along with the doctors, it is also used to create a Patient's database for storing the patient credentials.

Google Maps API

Google Maps API is a powerful tool for building location-based applications that make use of Google Maps. We used Google Maps API is to locate pharmacies or other medical facilities near a user's current location. Several markers and polyplot lines to show the route from the user to pharmacies is also important.

Google Colab

Google Colab is a free online platform that provides a Jupiter notebook environment for data science and machine learning tasks. It provides access to powerful computing resources, including GPUs and TPUs, which can be used to speed up computationally intensive tasks. We used Google collab to test our machine learning code which is the medicine substitution method, and identify ailments. Along with the ML and substitution medicine code, it is also used to develop the python backend for the Geolocation Analysis for accessing the datastore created with all the sectorial health information.

Docker

The application is an open-virtualization service used for providing containerized solutions. The application has been used to create a container for the database which is developed in Mongo DB as a storage medium for the Patient's and Doctor's data.

Postman

Postman is used for building and testing APIs. The FLASK APIs used for building the application backend are tested on the platform by sending and receiving in the JSON requests. The methods tested on the platform are POST, GET, PUT, PATCH based on different modes of the application and different APIs generated.

Microsoft Power BI

Microsoft Power BI is an analytics platform used for performing EDA on the data. For our application, Power BI is used extensively to analyze and clean the datastore for the Machine Learning dataset, Geolocation Information dataset and the Medical Store dataset in order to prepare the data for integrating with our application

2.5 Project Planning

The following are the list of all the activities that have been accomplished till date along with the time taken for their completion. We have also made a rough estimate of further implementations with the times we believe will be required.

Sr No.	Task	Start Date	End Date	Predecessors
1.	Project proposal	15-Jul-2022	23-Jul-2022	-
2.	Initial feasibility discussions	24-Jul-2022	31-Jul-2022	Project proposal
3.	Problem definition, scope and feasibility study	24-Jul-2022	20-Aug-2022	Project proposal
4.	Modules discussions with experts and guides	21-Aug-2022	04-Aug-2022	Problem definition and scope
5.	Literature review on modules	25-Aug-2022	19-Sep-2022	Problem definition and scope
6.	Tools and technology research	19-Sep-2022	28-Sep-2022	Literature review on Module
7.	Skills development	29-Sep-2022	02-Nov-2022	Tools and technology research
8.	Analysis of competitors and unique features	29-Sep-2022	10-Oct-2022	Tools and technology research
9.	UI design research and wireframe	11-Oct-2022	25-Oct-2022	Analysis of competitors and unique features
10.	Design document and future plan work	26-Oct-2022	02-Nov-2022	UI designs
11.	Data acquisition	20-Oct-2022	20-Nov-2022	UI designs
12.	Symptom dataset preparation and validation	21-Nov-2022	20-Dec-2022	Data acquisition
13.	Front end development	1-Dec-2022	10-Jan-2023	UI designs
14.	ML code for detection specifics	06-Jan-2023	25-Jan-2023	Symptom dataset preparation and validation
15.	Substitute medicine data prepare	26-Dec-2022	20-Jan-2023	Data acquisition
16.	Substitute medicine code	16-Jan-2023	05-Feb-2023	Substitute medicine data prepare
17.	Database auth with MongoDB	21-Jan-2023	05-Feb-2023	Front end development
18.	Blog API features	06-Feb-2023	05-Mar-2023	Front end development
19.	Geolocation analysis	26-Feb-2023	10-Mar-2023	Front end development

20.	Pharmacy module	06-Mar-2023	20-Mar-2023	Substitute medicine code, Front end development
21.	Recovery module	06-Mar-2023	20-Mar-2023	Front end development
22.	Integrations	21-Mar-2023	25-Mar-2023	Completion of all modules
23.	Testing	26-Mar-2023	31-Mar-2023	Integrations

Table 1: Project Planning Schedule

Based on the above-mentioned activities and their estimated durations, the following Gantt and PERT charts have been made to get an accurate timeline of the activities, both planning and the technical ones.

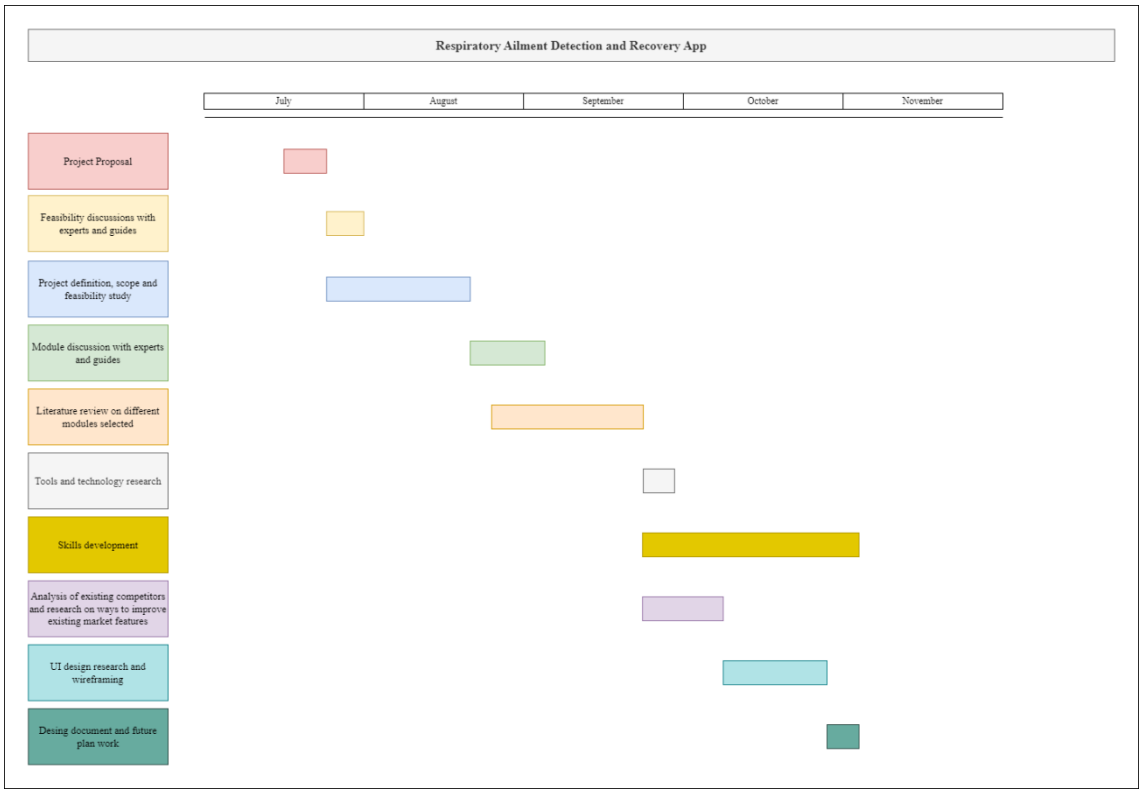


Figure 14: Gantt Chart for the Planning Phase Activities

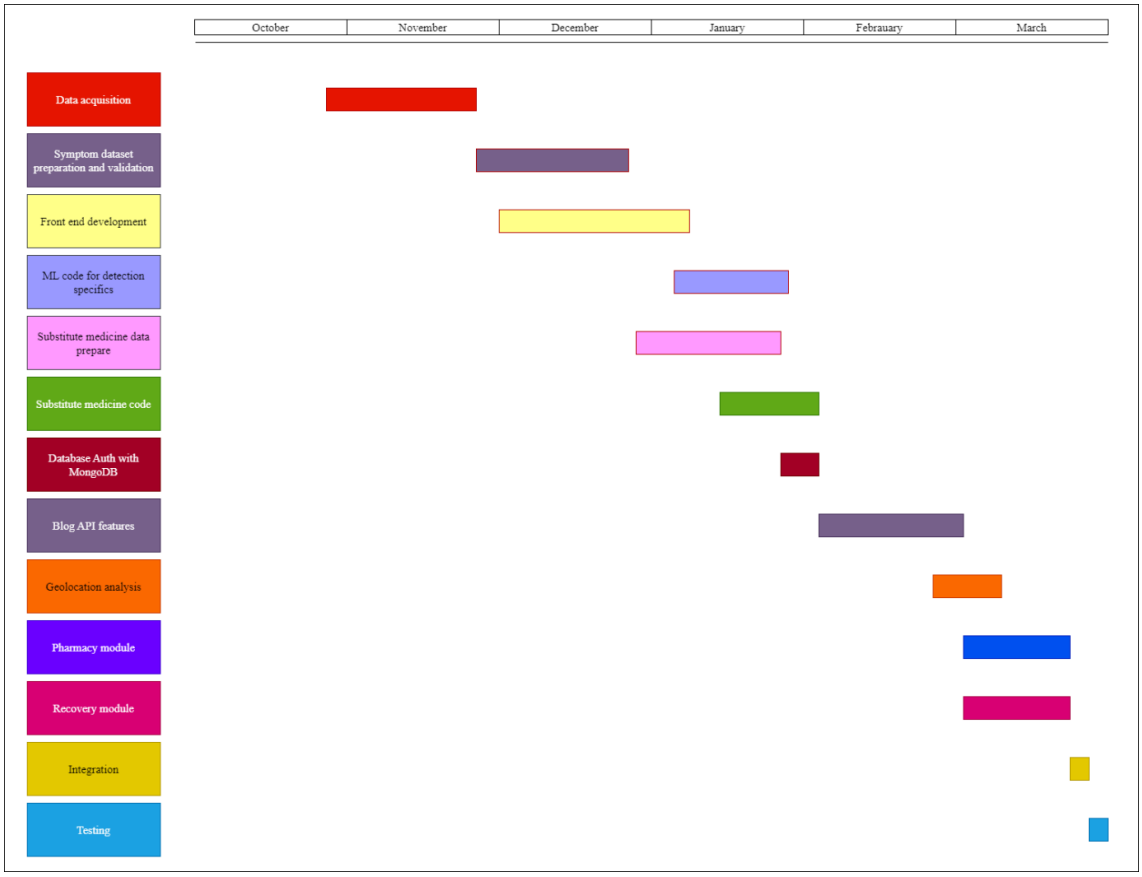


Figure 15: Gantt Chart for Technical Activities

The scheduling of these activities, with their timelines and durations for each of them have been mapped out in a flow in a PERT chart given below

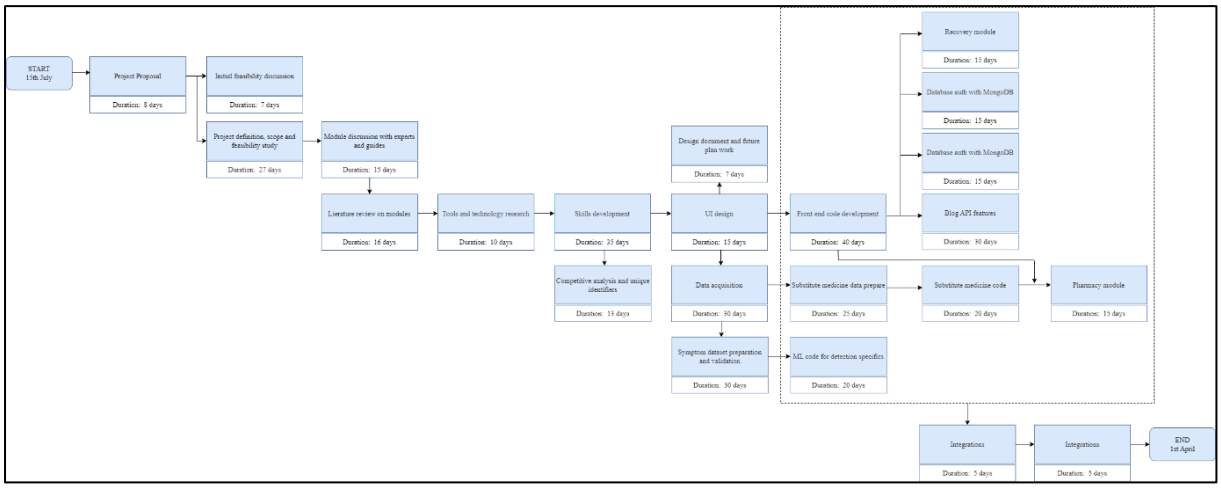


Figure 16: PERT Chart

Chapter 3: Project Description

3.1 Critical tasks selection

Critical tasks, that contribute to the uniqueness of the application, and the criteria for their selection as such are given below:

a. **Geolocation based analysis:** It provides a detailed analysis of cases reported in and around the user's location to provide an analytical report citing the reasons as to how the patient might have contracted the ailments. Through digital maps, we aim to provide a more sophisticated understanding of how the patient might have the problems and would they be recurring and what measures he/she needs to take for extra protection.

b. **POST-ailment care:** A critical and unique feature of our application which rests upon creating a more digitalised (automated) way to provide a post-ailment care to improve the health condition of the patient. Using an automated structural flow to study the entire recovery mechanisms and creating repositories for the procedures will be essential to gather from the verified doctors to carry this forward. Focus will lie on building the immunity following health practices by creating a separate health data record for the patient and follow-ups will be taken in order to keep a health check on the patient.

c. **Medical Blogs:** The feature is completely unique and also demands authenticity of the information being posted due to which authenticating the doctors and their profiles is a top priority which will be done through image processing capabilities. This will be a key feature towards our goal as ultimately we tend to improve the confidence among users for maximizing self-help which with the aid of resources to them becomes more accurate.

d. **Medicine Substitution:** Medicine substitution will consider several factors such as symptoms, evaluation, anatomy and genetics, medical history, etc. An in-depth database of medicines, their compounds, effects and side-effects etc. will be essential to the identification of the right substitute. Moreover, the application will send push notifications to doctors whose patients have to resort to substitute medicines so that they can appropriately monitor or alter dosage and treatment plans.

e. **Health analysis:** This feature involves the addition of a symptom monitor through checklists and questionnaires to guide early action where through Machine Learning algorithms we tend to provide the best course of action is suggested through the detection specifics being provided, where the confidence of data predicted is the weighted average of multiple models.

f. **Pharmacy:** The Pharmacy feature is all about bringing in the information regarding the different medicines available at different medical locations in order to provide a smart and easier way to understand the availability of medicines present in the locality of the user. Instead of creating our own inventory of medicines, we aggregate medicine availabilities to the users so that they can check and buy medicines accordingly.

3.3 Project workflow

Our Project is a complete end-to-end application starting from Authentication of the stakeholder (Patient or Doctor) right to the endpoint of different features present in our app thus providing a wholesome experience to the stakeholder.

The diagrammatic representation of the workflow below is an entire overview on the information which flows through the application. The user authenticates himself/herself in the application and then the App state moves forward to the Home screen wherein separate features, if selected, process User inputs and provide the output before terminating.

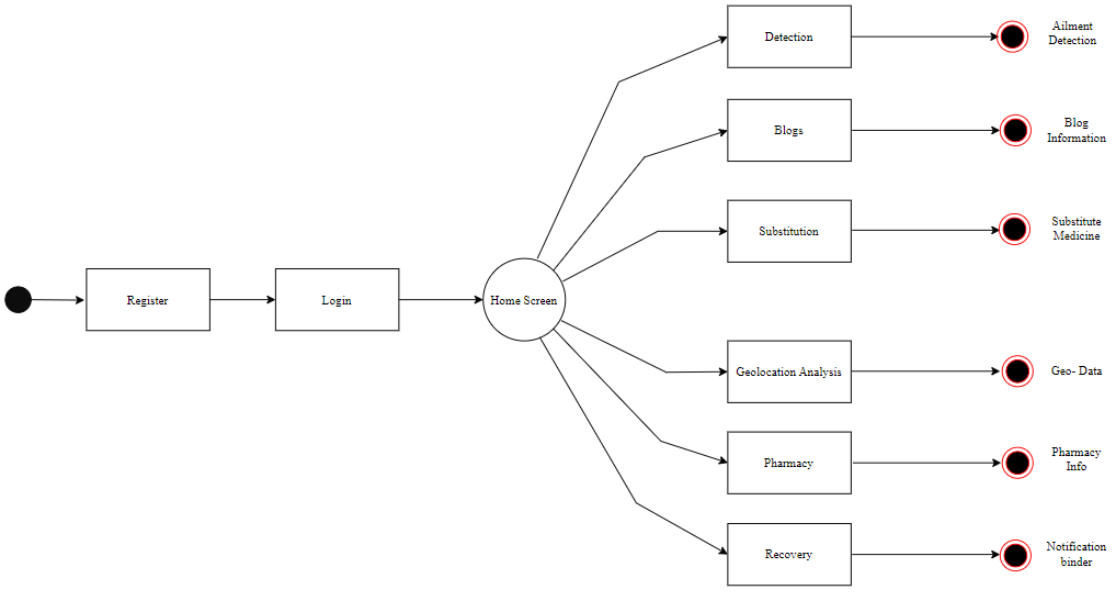


Figure 17: Complete Project Workflow

The overall workflow of every feature present is as follows

Detection:

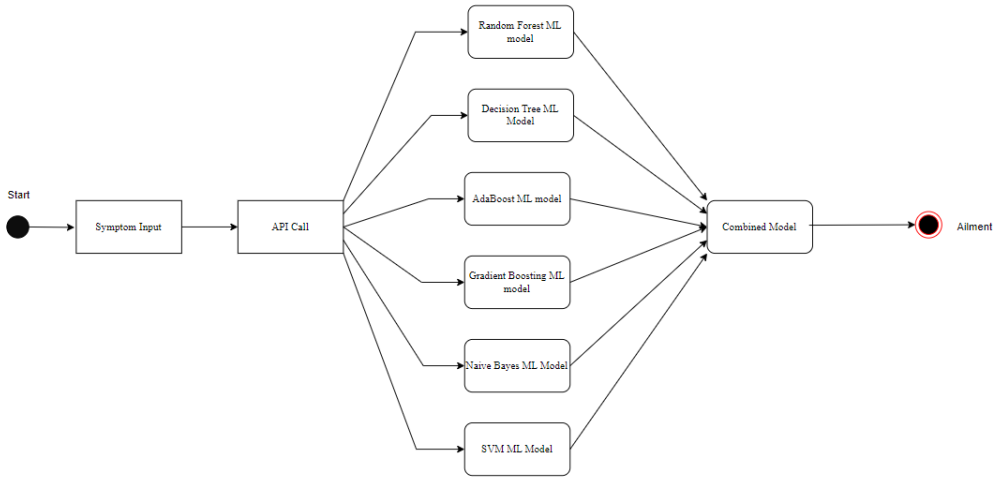


Figure 18: Detection Model Workflow

The User from the Home Screen accesses the Detection Module wherein he/she fills in the symptoms which are fed to the ML models individually, then combined within a single model to provide the Detected Ailment to the user.

Substitution:

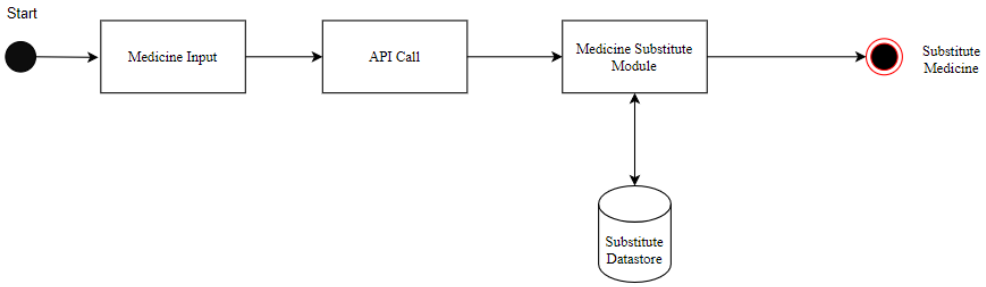


Figure 19: Substitute Medicine Module Workflow

In this module, the User inputs the medicine name after which an API is called which calls in the Medicine Substitute Module which searches through the datastore and provides the Substitute Medicine to the User.

Blogs:

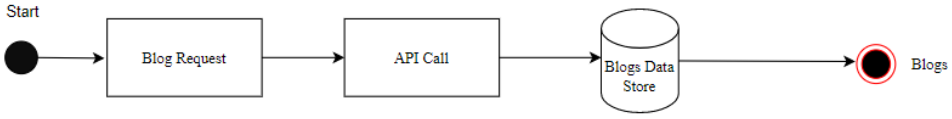


Figure 20: Blogs Module Workflow

In this module the User directly calls for viewing all the blogs which calls for an API which then processes the request and sends all the blogs to the frontend of the application.

Geolocation Analysis:

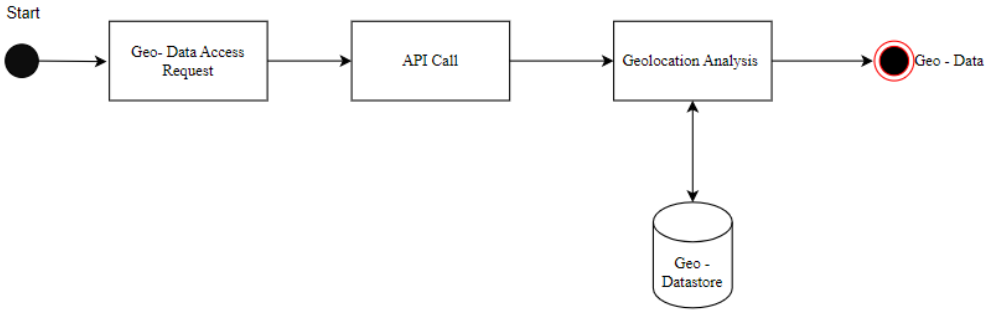


Figure 21: Geolocation Analysis Module

In the above module, the User Initiates the Geo – Data Access request which is further converted into an API call which calls in the Geolocation Analysis Function connected to the datastore which provides the Analysis information to the user.

Pharmacy Data:

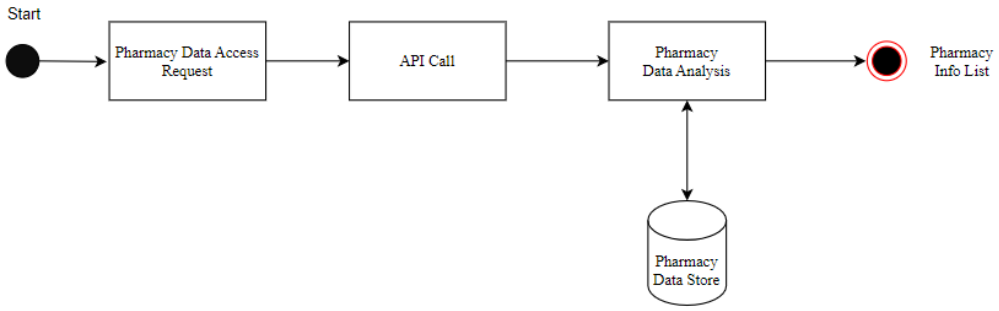


Figure 22: Pharmacy Data Module

In the above module, the Request for the Pharmacy Data is initialized by the user, then converted into an API call for accessing the database and providing the pharmacies information along with the availability of medicine.

Recovery:

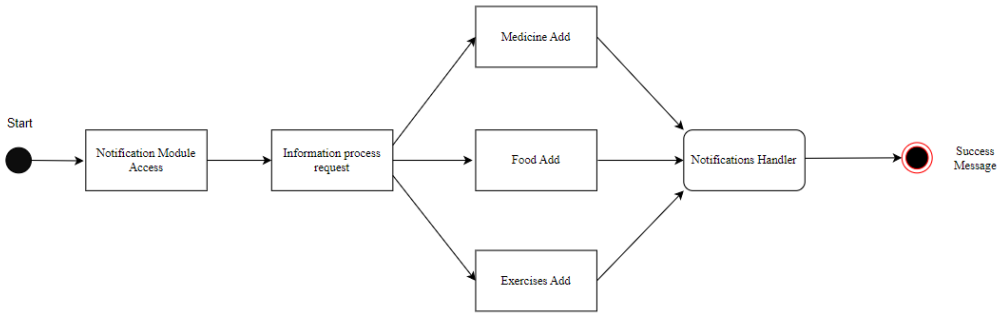


Figure 23: Recovery Module Workflow

The Recovery Module takes in the User input as the first step in order to initialize the functionality wherein once the information fed by the user, the Notifications Handler is called which further processes and sets the timers for activities as per the user requirements.

3.4 Methodologies and principles used

1. Design Thinking approach
   To develop an application that factors and takes into consideration the pain that a user goes through with the absence of the needed requirements and to understand how we could potentially cater and make our offering the one they desire, we have followed the design thinking approach, which having iterative steps takes place as a part of the following phases:

- Empathize: Where we initially tried to understand the problem, did a user research, and mapped all the concerned stakeholders involved, such that we then moved on towards making their profile canvas' to understand their pain points and working on it to move towards developing of new ideas catering to them.
- Define: Here, based on collecting all the pain points and clustering them into separate problem categories faced by different users, we try to come up with definitive problem statements that we aim to tackle.
- Ideate: Having brainstorming sessions and random word technique methods to try and come up with the best solutions based on each techniques implications and significance.
- Prototyping: We tried to understand in this phase how a user would go through the application and tried to understand their experience with the same, by developing high fidelity prototypes and understand the constraints and desires with respect to both the technical feasibilities and the users such that we focus on towards the final deployment.
- Test: The prototypes once developed are tested with the stakeholders involved (patients along with doctors) to understand the different parameters related to the usability and functionality of the application.

2. Agile methodology: Scrum framework
For the development of the application, we have devised that we are going to conduct sprints every week wherein we have specified a leader (scrum master) to lead the project to keep a track of all the developments in the application. Also, the external stakeholders (Doctors & Patients) will be included in the meeting every 3 weeks to understand the requirements and take in the review of all the stakeholders involved. Review sessions would be organized before the sprint was over, and tasks for iterations would be added to the backlog to be completed during the following sprint.

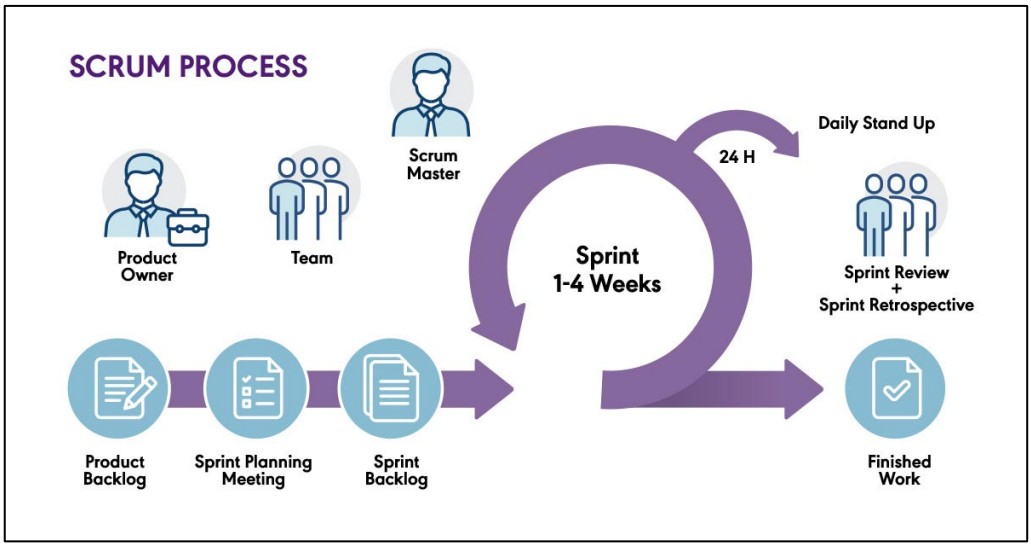


Figure 24: Scrum framework

YAGNI
YAGNI stands for 'You Aren't Gonna Need It' is a software development principle which we are following in our development approaches. We have divided the functionalities into separate modules since there is a lot of functionality dependency

present, thus completing one step at a time is critical for the success of our application as we intend to check the stability of every software module before implementing it.

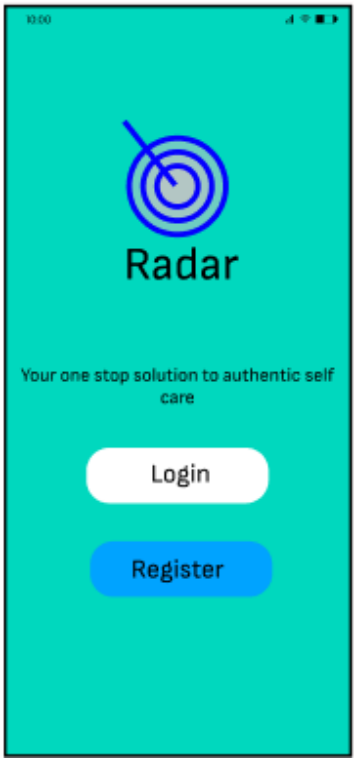
3. Software design principles

Principle 1	The reason it all exists The whole application and its features have been designed to deliver value to the users at different stages of medical attention, from general precautionary care and awareness to symptom identification, treatment and post-treatment care, with features identified and determined on the basis of the value they can add during these stages.
Principle 2	Keep it simple, stupid The application, though focused on a complex matter of medical conditions and their identification and treatment, will be visually simple, less cluttered and aimed to get the most information will less and precise questions. Different features will have distinct markings and sections that will be clearly visible to users.
Principle 3	Maintain the vision The vision has been focused to identification and treatment of respiratory ailments to be able to develop a well-functioning application that will also serve as proof of concept and inspiration to improve overall healthcare infrastructure in the future.
Principle 4	What you produce, others will consume We understand that the application will have users in various forms, from general audience to patients, doctors, other medical practitioners and professionals, pharmacists etc. The design of the application and its components will be prioritized to meet the specific needs of these stakeholders in a manner that mutually benefits all, for instance, instead of developing a new online platform selling medicines, we have decided to develop the network of local pharmacies to support their businesses too.
Principle 5	Be open to the future While the focus for the present is highly specific, we acknowledge the future potential for the application and the growth opportunities before us. While we are working on developing the present objectives, we also are obtaining understanding from relevant stakeholders about future potential for the application.
Principle 6	Plan ahead for reuse In light of our understanding of the potential future uses of the application, we intend to design codes that can be similarly reproduced or duplicated to provide effective analysis for other ailments in the future. For instance, the program for symptom classification can similarly be reproduced with specific changes to conditions for heart conditions specially to monitor health of people with hereditary conditions.

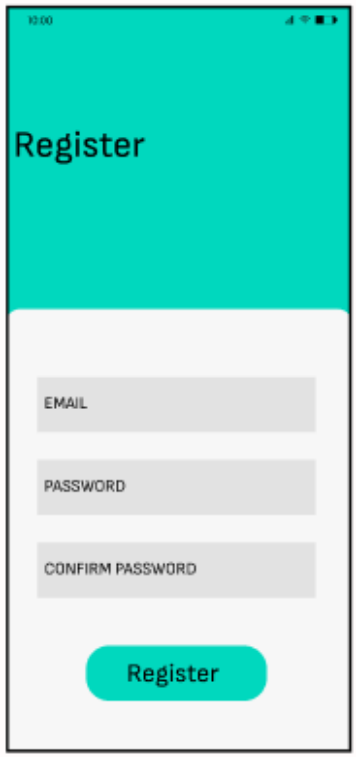
Table 2: Software design principles

Chapter 4: Project Implementations

4.1 Wireframe configurations



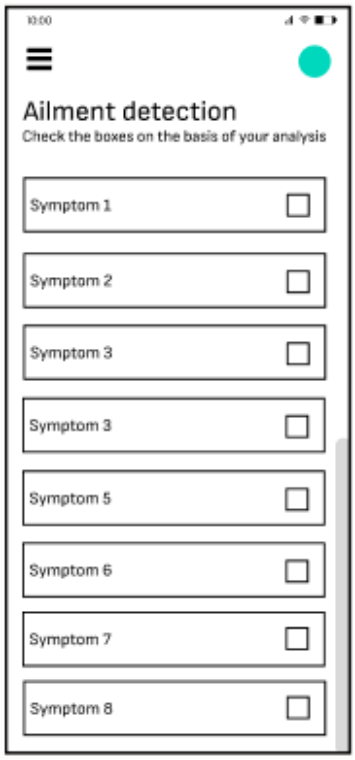
Application home page



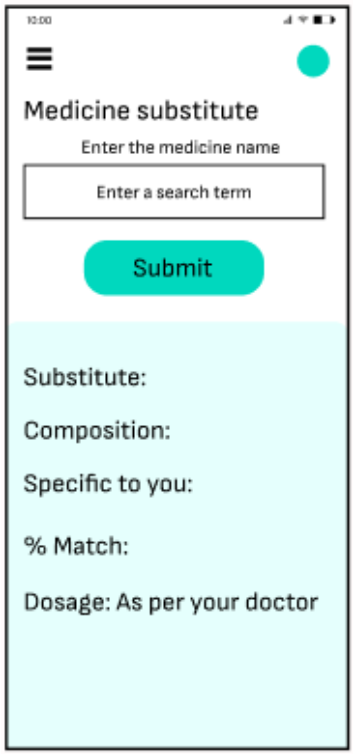
Registration page

Login page

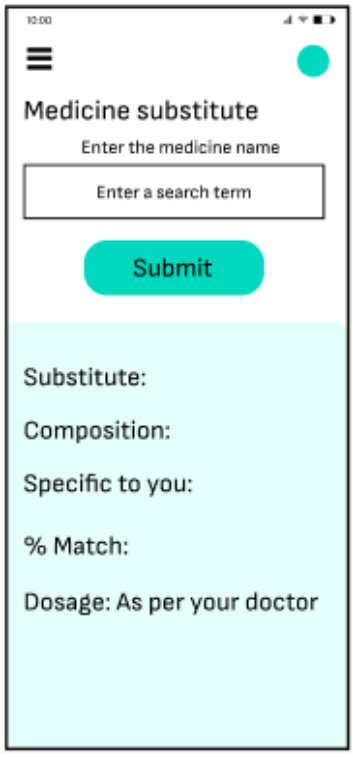
Main user dashboard



Ailment checklist for symptom inputs



Output page for the ailment detection



Medicine substitution functionality

Figure 25: Wireframes

4.2 Postman configurations for all the services

Detection: This API is created with a 'GET' functionality, where the different symptoms are fed in to call the Machine Learning models. The output is provided in the JSON format where the different models provide different predictions all equated into a single equation.

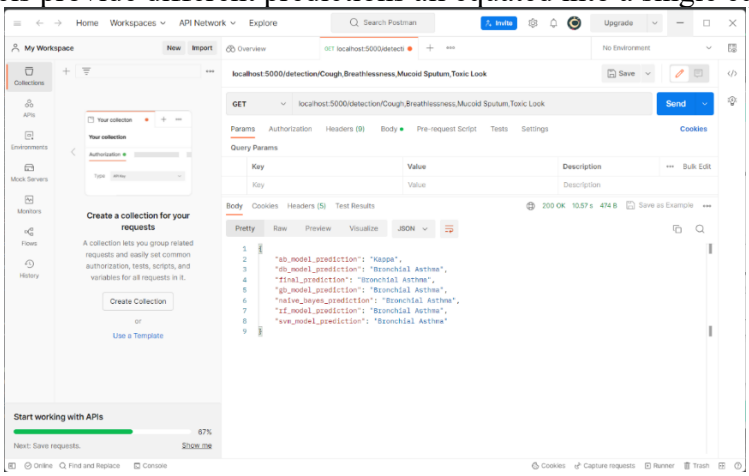


Figure 26 a: Detection

Substitution: The API created below for the module takes in the Medicine name and provides a substitute through a datastore and with a match between the compositions of two medicines being made to provide a strong substitute along with their percentage matches for the same.

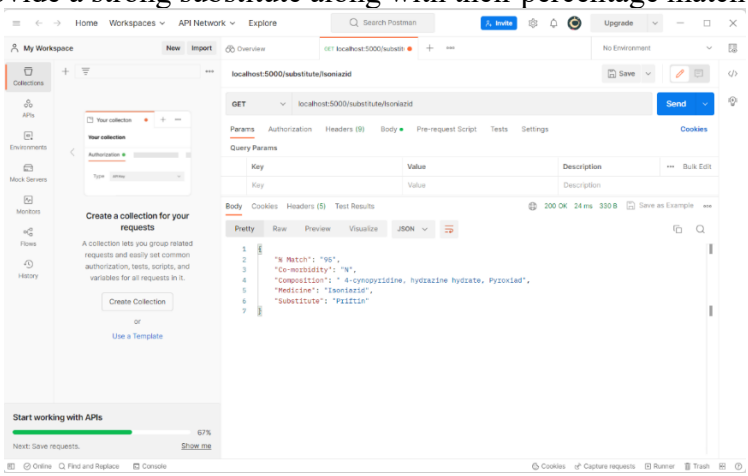


Figure 26 b: Substitution

Geolocation: The API created would take in the Live location of the user and the ailment he/she wants to search and provide an analysis of the geographical area regarding the cases and facilities present of that particular ailment to the user.

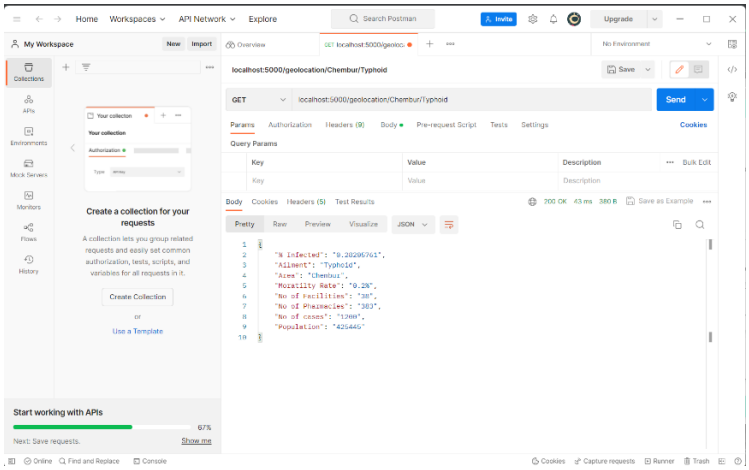


Figure 26 c: Geolocation

Blogs: The API created for this function would provide all the information regarding the different blogs posted by the doctors which are enabled to the users through the 'GET' method.

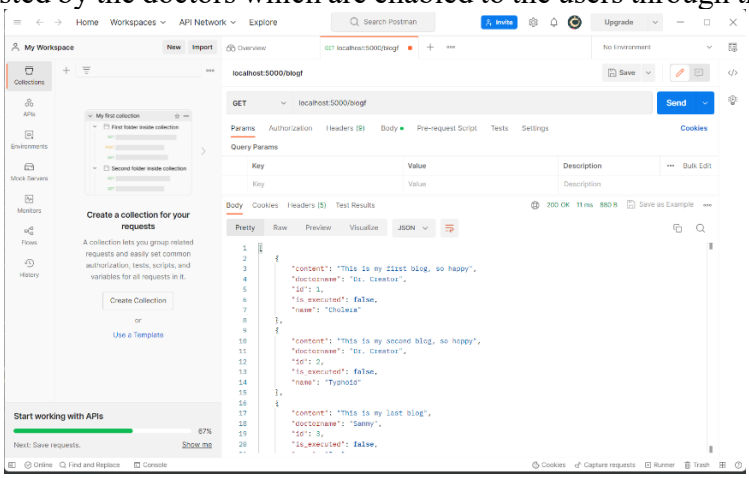


Figure 26 d: Blogs

Pharmacy Stores: Pharmacy Stores medicine availabilities is provided as backend service wherein the different pharmacies containing the medicine requested by the user are shown to the user through the GET API.

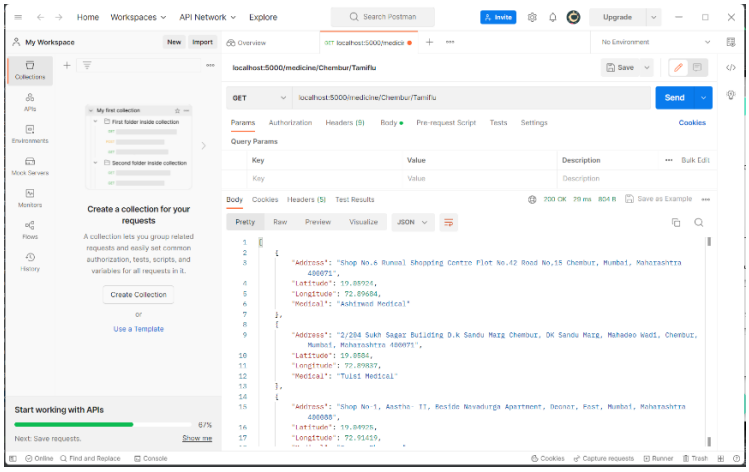


Figure 26 e: Pharmacy stores

4.3 Dataset preparations

Different datasets have been developed for detection and Substitute medicine purposes along with the translation data.

The snippet below is the Ailment detection dataset which consists of 41 different types of symptoms and 10 different types of ailments developed through different combinations of these symptoms. We had collected more than 200 samples per respiratory ailment and validated from the subject matter experts. The Prognosis column is in string format for getting in the ailment based on a combination of 0s and 1s each corresponding to a certain symptom. This data is then fed to the Machine Learning model.

	throat_irri	redness_o	sinus_pres	runny_no	congestio	chest_pai	weakness_	fast_heart	loss_of_sn	toxic_look	muscle_pa	belly_pain	mucoid_s	rusty_spu	blood_in_	fever	body_ach	taste_loss	hearing_lc	sore_eyes	lower_bac	prognosis
2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
4	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
6	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
7	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bronchial Asthma
9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
14	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
15	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
16	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
17	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bronchial Asthma
19	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
20	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
21	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
22	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
23	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
24	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
25	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
26	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
27	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Bronchial Asthma
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bronchial Asthma

Figure 27 a: Ailment detection dataset

The snippet below is about building up a dataset which consists of the medicines specific to the ailments and their substitutes with the compositions and percent match between the two medicines.

	Medicine Name	Composition	Substitute	Percent Mat	Co-morbidity validation
2	Rifadin	Rifampin, Sodium formaldehyde sulfoxate, sodium hydroxide	Nitazaoxanide	90	Y
3	Isoniazid	4-cynopyridine, hydrazine hydrate, Pyroxiad	Priftin	95	N
4	Ethambutol	Gelatin, Hydroxypropide Methylcellulose, Magnesium Stearate, Sorbitol, Stearic acid, Sucrose, Titanium dioxiide	Linezolid	95	Y
5	Rifatar	Isoniazid, rifampicin, pyrazinamiad	Avelox	99	Y
6	Crocin	Caffeine, Paracetamol	Paracin	100	Y
7	Benadryl	Diphenhydramine, Ammonium chloride and Sodium citrate	Claritin	100	Y
8	Cetrizin	Pregelatinized starch, lactose, maize starch, povidone, magnesium stearate, macrogol 6000, basic polymethacrylate, titanium dioxide	Loratadine	100	Y
9	Sirturo	Lactose, Bedaquiline fumurate	Pretomanid	100	Y
10	Zyvox	corn starch, microcrystalline cellulose, hydroxypropylcellulose, sodium starch glycolate, magnesium stearate, hypromellose, polyethylene glycol, titanium dioxide, and	Tygacil	95	Y
11	Glucocorticoids	Steroids, Mineralcorticods	Hydrocort	100	N
12	Augmentin	Amoxicillin trihydrate, Potassium cluvunate	Cipro	90	Y
13	Zithromax	Azithromycin trihydate, Lactose anhydrous, sulphur dioxide	Clindamycin	100	Y
14	Kelfex	Cephlaxinemonohydrate, dimethicone, carboxymethylcellulose, F D&C Blue No, F D&C Yellow No	Biocef	100	Y
15	Cipro	Monohydrochloride monohydrate salt of quinilecarbofluoro acid	Bactrim	99	Y
16	Moxatag	Amoxicillin trihydrate, Potassium cluvunate	Trimox	99	Y
17	Moxilin	Amoxicillin trihydrate	Periostat	88	Y
18	Robafen	Guiafensin, Dexthromethorphan trihydrate	Bidex 400	99	Y
19	DexAlone	Dexthromethorphan	Babee syrup	99	Y
20	Delsym	Dexthromethorphan polisterix, extrerel syrup, sodium	Robitussim	77	Y
21	Advil	Ibuprofen, lecithin, Sorbitol, polythene glycol, potassium hydroxide	Naproxen	80	N
22	ConZip	Tradamol HCl	Fluka 150	100	Y
23	Motrin	Ethanol, Ibuprofen	Arnica	100	Y
24	OxyContin	Opium alakloid, thebanine	Fentanyl	100	Y
25	Roxicodon	Opium alakloid, thebanine	Tylenol	100	Y
26	Ritonavir	Ethanol, water, polyoxyl castor oil, propylene glycol, anhydrous citric acid, saccharin sodium, peppermint oil	Lopinavir	92	Y
27	Remdesivir	Carboxylic esters, nitrile, pyrollotraizine	Fabiflu	91	Y
28	Molnupiravir	Croscarmellose sodium, Hydroxypropyl cellulose, Magnesium stearate and Microcrystalline cellulose and purified water	Molxvir	90	Y

Figure 27 b: Medicine dataset

The data has been transformed using Power BI tool to understand the data and removing any mistakes in the data. Further we have used Data cleaning methods in Python to further customize the data according to the models which will be developed in ML.

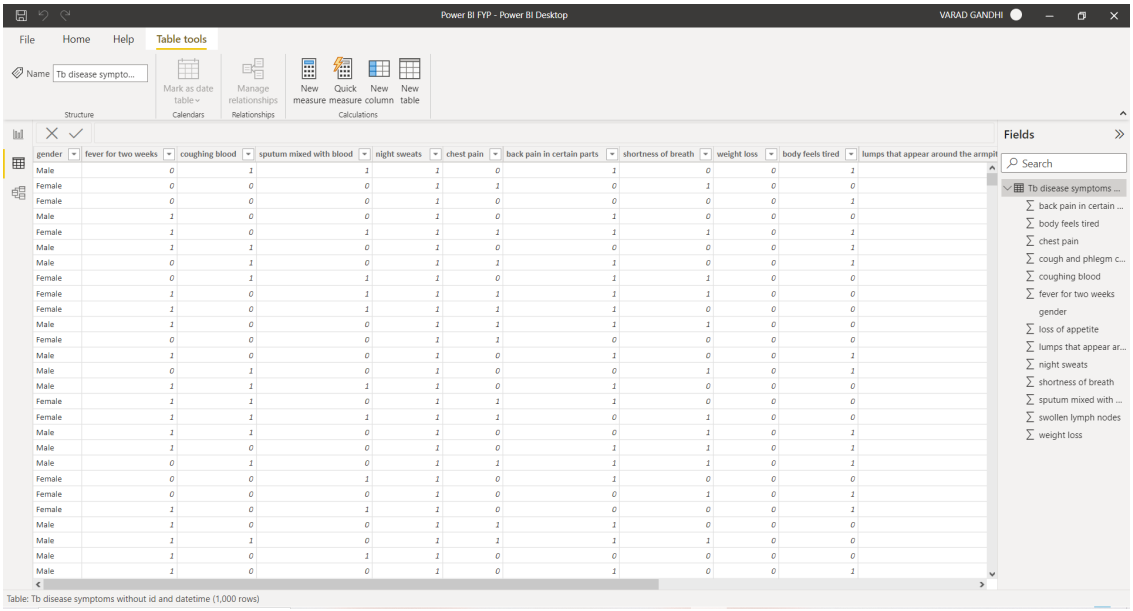


Figure 27 c: Data transformation and cleaning

The snippet below is all about providing a geolocation analysis wherein certain area analysis is provided based on different ailments wherein different parameters such as population and the statistics related to the ailment and area are provided

	A	B	C	D	E	F	G	H	I	J
1	Area	City	Pincode	Ailment	Population	No of Cases	No of Faciliti	No of Pharmacies	% Infected	Mortality Rate
2	Chembur	Mumbai	400071	Tuberculosis	425445	77000	38	383	18.09869666	5.16
3	Chembur	Mumbai	400071	Typhoid	425445	1200	38	383	0.28205761	0.2
4	Chembur	Mumbai	400071	Pneumonia	425445	600	38	383	0.141028805	18
5	Chembur	Mumbai	400071	Bronchial Asthma	425445	2700	38	383	0.634629623	27.9
6	Chembur	Mumbai	400071	Common Cold	425445	No estimates	38	383	No estimates	No estimates
7	Chembur	Mumbai	400071	Alpha	425445	4000	38	383	0.940192034	2.47
8	Chembur	Mumbai	400071	Delta	425445	3200	38	383	0.752153627	8.56
9	Chembur	Mumbai	400071	Delta Plus	425445	2300	38	383	0.54061042	22.47
10	Chembur	Mumbai	400071	Kappa	425445	40	38	383	0.00940192	1.67
11	Chembur	Mumbai	400071	Omicron	425445	2000	38	383	0.470096017	3.04
12	Chembur	Mumbai	400071	BA.1	425445	120	38	383	0.028205761	0.4
13	Vile Parle West	Mumbai	400056	Tuberculosis	242216	72000	120	197	29.72553423	5.16
14	Vile Parle West	Mumbai	400056	Typhoid	242216	1400	120	197	0.577996499	0.2
15	Vile Parle West	Mumbai	400056	Pneumonia	242216	700	120	197	0.288998249	18
16	Vile Parle West	Mumbai	400056	Bronchial Asthma	242216	400	120	197	0.165141857	27.9
17	Vile Parle West	Mumbai	400056	Common Cold	242216	No estimates	120	197	No estimates	No estimates
18	Vile Parle West	Mumbai	400056	Alpha	242216	6000	120	197	2.477127853	2.47
19	Vile Parle West	Mumbai	400056	Delta	242216	900	120	197	0.371569178	8.56
20	Vile Parle West	Mumbai	400056	Delta Plus	242216	240	120	197	0.099085114	22.47
21	Vile Parle West	Mumbai	400056	Kappa	242216	60	120	197	0.024771279	1.67
22	Vile Parle West	Mumbai	400056	Omicron	242216	2200	120	197	0.908280213	3.04
23	Vile Parle West	Mumbai	400056	BA.1	242216	310	120	197	0.127984939	0.4

Figure 27 d: Geolocation analysis

The snippet below is about providing information regarding different pharmacies and the availability of different medicines and through this dataset, only the pharmacies where the medicine is available will be displayed.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Area	City	Pharmacy Name	Pharmacy Latitude	Pharmacy Longitude	Pharmacy Address	Rifadin	Isoniazid	Rifatar	Crocin	Benadryl	Cetrizin	Sirturo	Zyvox	Tamiflu	Priftin	Tygacil
2	Chembur	Mumbai	Ashirwad Medical	19.05924	72.89684	Shop No.6 Runwal Shopping Centre Plo	2	0	16	25	10	0	7	0	1	2	2
3	Chembur	Mumbai	Tulsi Medical	19.0584	72.89837	2/204 Sukh Sagar Building D.k Sandu M	4	0	15	30	12	2	2	0	1	3	2
4	Chembur	Mumbai	P & G Medico	19.05327	72.91572	Shop n.19/20,Gr floor, Neelkanth Gard	4	2	23	23	8	2	2	3	0	4	4
5	Chembur	Mumbai	Balaji Medical	19.0626	72.90021	70/D, Uday Building, Shop No. 3 M D S	0	12	22	22	0	7	1	4	0	2	6
6	Chembur	Mumbai	Deonar Pharmacy	19.04925	72.91419	Shop No-1, Aastha- II, Beside Navadur	1	14	10	21	1	8	0	1	5	2	7
7	Vile Parle West	Mumbai	Amar Medical Store	19.01591	72.8369	Shop Numner 1 R.N. Cooper Hospital (	9	22	1	2	4	1	25	0	2	2	4
8	Vile Parle West	Mumbai	Rajni Medical	19.10329	72.83372	Shop no 11, Vaikunthlal Mehta Rd, nea	4	12	0	2	4	0	25	0	6	1	8
9	Vile Parle West	Mumbai	Saraswati Medical	19.11063	72.83988	1390 IRLA TANK, Swami Vivekananda F	5	3	9	6	8	0	26	4	6	1	9
10	Vile Parle West	Mumbai	Diamond Medical	19.10856	72.83805	1, 1C, Irla Rd, Navpada, Irla, Vile Parle '	7	7	9	7	1	0	42	5	8	5	9
11	Vile Parle West	Mumbai	Sai Kripa Medical	19.10665	72.83945	V M Road No 5, Kartik Chawl,, Nehru N	7	0	0	0	0	3	8	6	0	7	9

Figure 27 e: Pharmacies

4.4 NLP code development

The entire process for the Natural Language Processing module for the translation of texts is shown as given below:

- Data Collection method: Compile a sizable corpus of English translations of Hindi texts. This will be used as your translation model's training set of data.
- Pre-processing: Tokenizing text data into sentences and words, eliminating stop words and punctuation, and conducting stemming or lemmatization to standardize the text are all examples of pre-processing.
- After that each sentence has been marked with Start and end to mark where sentence starts and ends in Hindi corpus.
- In the context of developing code for translating text from Hindi to English, encoding and decoding is done and used in the following ways:
  Encoding: The Hindi text is first encoded into a machine-readable format, such as a sequence of numerical values or a series of tokens that represent the words in the text. This is typically done using a process called tokenization, where each word is assigned a unique code or token. The encoded data can be used as input to a neural network or other machine learning model.
  Decoding: Once the model has generated the English translation of the input text, the output is encoded back into a human-readable format, such as a string of characters or words in the English language. This is typically done by reversing the encoding process, where each token or numerical value is mapped back to its corresponding word or character.
- Language Model: Using a neural network-based method, such as LSTM to train a language model on the Hindi corpus. This model can produce text in Hindi while preserving the language's context.
- Train a translation model that takes English text as input and generates Hindi text as output. This model should be able to translate English sentences to Hindi while preserving the meaning and context of the text.

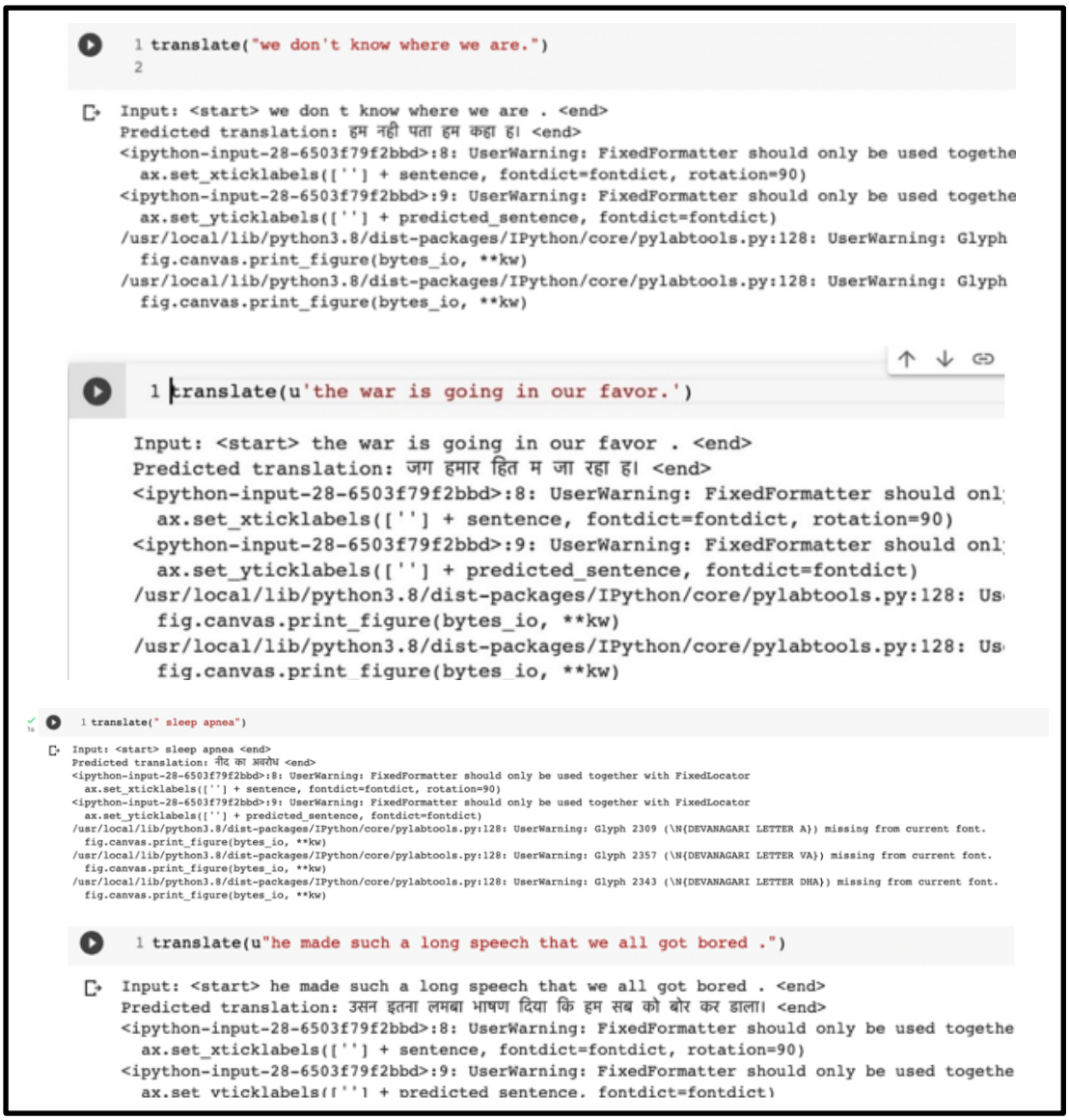


Figure 28: NLP Code

4.5 User authentications and database connectivity

The user authentication model developed consists of only two fields mainly Username and Password for authentication purposes. The schema developed consists of 2 fields, that is Username and password.

The index.js is critical in terms of creating a route for the API to work.

```
index.js > ...
const express=require('express')
const app=express()
const port =8080 || process.env.port
const cors=require('cors')
const bodyParser=require('body-parser')
const mongoose=require('mongoose')
mongoose.connect("mongodb://localhost:27017/mydb",{useNewUrlParser:true,useUnifiedTopology:true})

app.use(cors())
app.use(bodyParser.urlencoded({extended:true}))
app.use(bodyParser.json())
app.use('/',require('./routes/user.route'))

app.listen(port,()=>{
    console.log('port running on'+port)
})
```

Figure 29: index.js

The model developed then needs to connect to the frontend using the route.js file logic along with the testing framework that we have developed below where the data being fed are to be checked for existence with required fields else we give the conditions for errors depending on the situation of either sign-in or sign-up. Finally, the module created would be exported to create the entire user database.

```
routes > user.route.js > router.post('/signup') callback > User.findOne() callback
router.post('/signup',(req,res)=>{
    User.findOne({username:req.body.username},(err,user)=>{
        if(err){
            console.log(err)
            res.json(err)
        }else{
            if(user==null){
                const user=User({
                    username:req.body.username,
                    password:req.body.password
                })
                user.save()
                .then((err)=>{
                    if(err){
                        console.log(err)
                        res.json(err)
                    }else{
                        console.log(user)
                        res.json(user)
                    }
                })
            }else{
                res.json({
```

Figure 30: route.js

4.6 Ailment Detection code development

The use of Machine Learning classification algorithms has been critical in our analysis because of their capability to detect the ailment based on different combination of ailments and that the final model accuracy would be based on the weighted average of the accuracies of each model.

In the snippet below, all the Machine Learning libraries related to the Model are imported and the dataset is read. A total of 41 different combination of symptoms and 10 different types of respiratory ailments are present in the dataset.

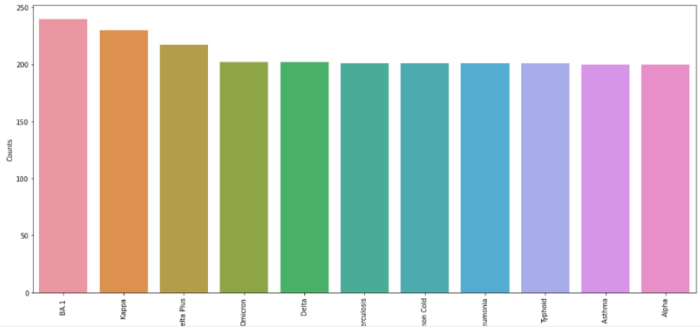
```
import numpy as np
import pandas as pd
from scipy.stats import mode
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier

%matplotlib inline
```

```
[ ] df=pd.read_csv('Detection.csv').dropna(axis=1)
    df.head()
```

	continuous_sneezing	chills	muscle_wasting	vomiting	fatigue	weight_loss	cough	high_fever	breathlessness	sweating	...	mucoid_sputum	rusty_sputum	blood_in_sputum	fever	boc
0	0	0	0	0	1	0	1	1	1	0	...	1	0	0	0	
1	0	0	0	0	0	0	1	1	1	0	...	1	0	0	0	
2	0	0	0	0	1	0	0	1	1	0	...	1	0	0	0	
3	0	0	0	0	1	0	1	0	1	0	...	1	0	0	0	

Roughly 200 instances are present of each ailment in the Training dataset which are fed to the classification models used in our implementation.

```
[ ] ailments = df["prognosis"].value_counts()
    temp_data = pd.DataFrame({
        "Ailments": ailments.index,
        "Counts": ailments.values
    })

    plt.figure(figsize = (18,8))
    sns.barplot(x = "Ailments", y = "Counts", data = temp_data)
    plt.xticks(rotation=90)
    plt.show()
```



The snippet below displays the classification models used for building up the ailment detection feature. A total of 6 models have been used namely SVC, Gaussian Naïve Bayes, Decision Tree, Random Forest, Adaboost, Gradient Boosting

```
[ ] type(df['prognosis'])

    pandas.core.series.Series
```

```
[ ] encoder = LabelEncoder()
    df["prognosis"] = encoder.fit_transform(df["prognosis"])
```

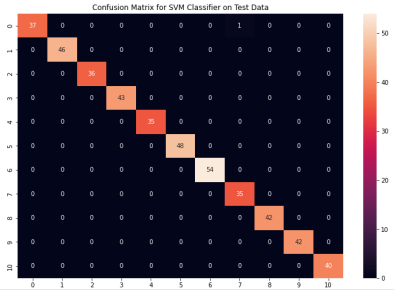
```
[ ] X = df.iloc[:,:-1]
    y = df.iloc[:, -1]
    X_train, X_test, y_train, y_test =train_test_split(
      X, y, test_size = 0.2, random_state = 24)

    print(f"Train: {X_train.shape}, {y_train.shape}")
    print(f"Test: {X_test.shape}, {y_test.shape}")

    Train: (1836, 41), (1836,)
    Test: (459, 41), (459,)
```

```
[ ] def cv_scoring(estimator, X, y):
        return accuracy_score(y, estimator.predict(X))


    models = {
        "SVC":SVC(),
        "Gaussian NB":GaussianNB(),
        "Random Forest":RandomForestClassifier(random_state=24),
        "Decision Tree":DecisionTreeClassifier(random_state=24),
        "AdaBoost":AdaBoostClassifier(n_estimators=20,random_state=0),
        "GradientBoosting":GradientBoostingClassifier(random_state=24)
    }
```

The SVC Model provides a 100% accurate classification on the training data and a 99.78% accuracy when applied on test data.

```
[ ] svm_model = SVC()
    svm_model.fit(X_train, y_train)
    preds = svm_model.predict(X_test)

    print(f"Accuracy on train data by SVM Classifier\
    : {accuracy_score(y_train, svm_model.predict(X_train))*100}")

    print(f"Accuracy on test data by SVM Classifier\
    : {accuracy_score(y_test, preds)*100}")
    cf_matrix = confusion_matrix(y_test, preds)
    plt.figure(figsize=(12,8))
    sns.heatmap(cf_matrix, annot=True)
    plt.title("Confusion Matrix for SVM Classifier on Test Data")
    plt.show()
```

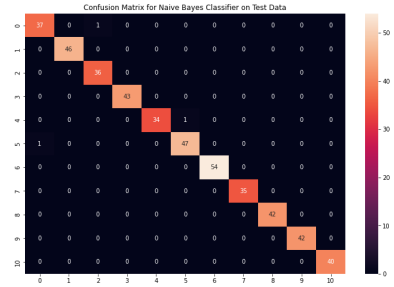
```
Accuracy on train data by SVM Classifier: 100.0
Accuracy on test data by SVM Classifier: 99.78213507625271
```



The Gaussian Naïve Bayes model provides a 99.67% accuracy on training data while a 99.34% accuracy on test data.

```
[ ] nb_model = GaussianNB()
    nb_model.fit(X_train, y_train)
    preds = nb_model.predict(X_test)
    print(f"Accuracy on train data by Naive Bayes Classifier\
    : {accuracy_score(y_train, nb_model.predict(X_train))*100}")

    print(f"Accuracy on test data by Naive Bayes Classifier\
    : {accuracy_score(y_test, preds)*100}")
    cf_matrix = confusion_matrix(y_test, preds)
    plt.figure(figsize=(12,8))
    sns.heatmap(cf_matrix, annot=True)
    plt.title("Confusion Matrix for Naive Bayes Classifier on Test Data")
    plt.show()
```

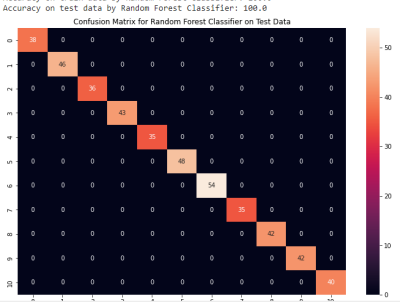
```
Accuracy on train data by Naive Bayes Classifier: 99.67320261437908
Accuracy on test data by Naive Bayes Classifier: 99.34640522875817
```



The Random Forest model is the best prediction model since it provides accurate results on the training as well as testing data.

```
[ ] rf_model = RandomForestClassifier(random_state=24)
    rf_model.fit(X_train, y_train)
    preds = rf_model.predict(X_test)
    print(f"Accuracy on train data by Random Forest Classifier\
    : {accuracy_score(y_train, rf_model.predict(X_train))*100}")

    print(f"Accuracy on test data by Random Forest Classifier\
    : {accuracy_score(y_test, preds)*100}")

    cf_matrix = confusion_matrix(y_test, preds)
    plt.figure(figsize=(12,8))
    sns.heatmap(cf_matrix, annot=True)
    plt.title("Confusion Matrix for Random Forest Classifier on Test Data")
    plt.show()
```

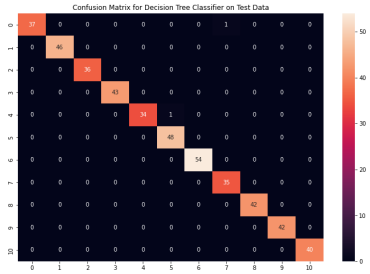
```
Accuracy on train data by Random Forest Classifier: 100.0
Accuracy on test data by Random Forest Classifier: 100.0
```



The decision tree model is only 0.5% away in the difference between the prediction on the training and the testing data.

```
[ ] dt_model = DecisionTreeClassifier(random_state=24)
    dt_model.fit(X_train, y_train)
    preds = dt_model.predict(X_test)
    print(f"Accuracy on train data by Decision Tree Classifier\
    : {accuracy_score(y_train, rf_model.predict(X_train))*100}")

    print(f"Accuracy on test data by Decision Tree Classifier\
    : {accuracy_score(y_test, preds)*100}")

    cf_matrix = confusion_matrix(y_test, preds)
    plt.figure(figsize=(12,8))
    sns.heatmap(cf_matrix, annot=True)
    plt.title("Confusion Matrix for Decision Tree Classifier on Test Data")
    plt.show()
```

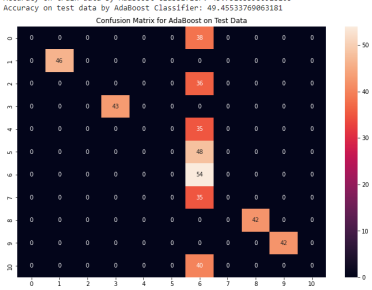
```
Accuracy on train data by Decision Tree Classifier: 100.0
Accuracy on test data by Decision Tree Classifier: 99.56427015250546
```



The Adaboost algorithm used provides a lower rate of accuracy in the prediction due to its ability combine various weak learners thus bringing in a lot of noise in its training and testing

```
[ ] ab_model = AdaBoostClassifier(n_estimators=20,random_state=0)
    ab_model.fit(X_train, y_train)
    preds = ab_model.predict(X_test)
    print(f"Accuracy on train data by AdaBoost Classifier\
    : {accuracy_score(y_train, ab_model.predict(X_train))*100}")

    print(f"Accuracy on test data by AdaBoost Classifier\
    : {accuracy_score(y_test, preds)*100}")

    cf_matrix = confusion_matrix(y_test, preds)
    plt.figure(figsize=(12,8))
    sns.heatmap(cf_matrix, annot=True)
    plt.title("Confusion Matrix for AdaBoost on Test Data")
    plt.show()
```

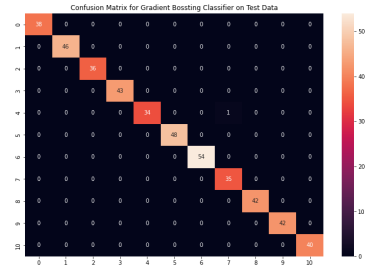
```
Accuracy on train data by AdaBoost Classifier: 45.751633986928105
Accuracy on test data by AdaBoost Classifier: 49.455337690631181
```



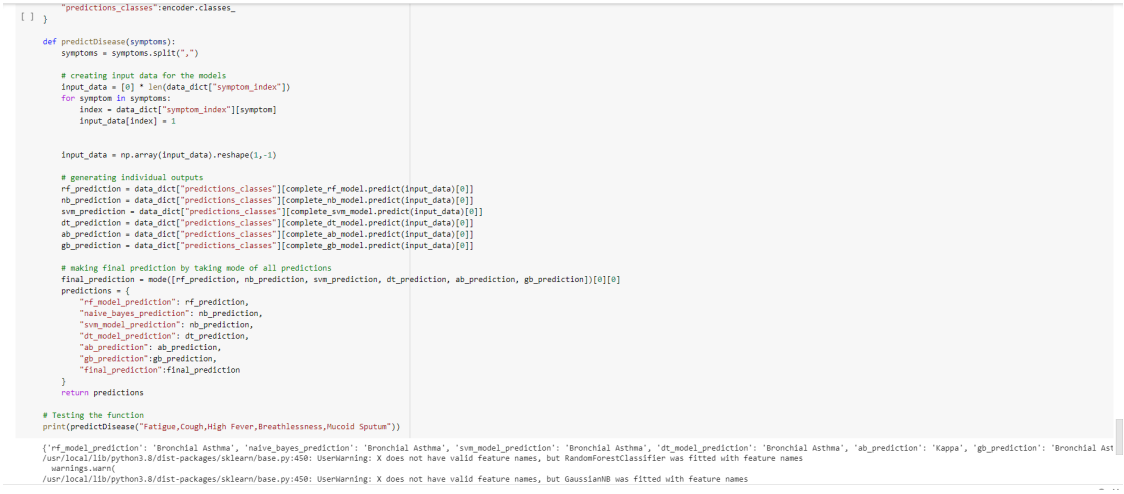
The gradient boosting algorithm is a strong algorithm which doesn't have any biases from the original outputs in the dataset.

```
[ ] gb_model = GradientBoostingClassifier(random_state=24)
    gb_model.fit(X_train, y_train)
    preds = gb_model.predict(X_test)
    print(f"Accuracy on train data by Graident Boosting Classifier\
    : {accuracy_score(y_train, gb_model.predict(X_train))*100}")

    print(f"Accuracy on test data by Gradient Boosting Classifier\
    : {accuracy_score(y_test, preds)*100}")

    cf_matrix = confusion_matrix(y_test, preds)
    plt.figure(figsize=(12,8))
    sns.heatmap(cf_matrix, annot=True)
    plt.title("Confusion Matrix for Gradient Bossting Classifier on Test Data")
    plt.show()
```

```
Accuracy on train data by Graident Boosting Classifier: 100.0
Accuracy on test data by Gradient Boosting Classifier: 99.782135076252721
```



The below code is for creating a json format output which will be connected to the frontend using the Flask API.



```
    "predictions_classes":encoder.classes_
}

def predictDisease(symptoms):
    symptoms = symptoms.split(",")

    # creating input data for the models
    input_data = [0] * len(data_dict["symptom_index"])
    for symptom in symptoms:
        index = data_dict["symptom_index"][symptom]
        input_data[index] = 1


    input_data = np.array(input_data).reshape(1,-1)

    # generating individual outputs
    rf_prediction = data_dict["predictions_classes"][complete_rf_model.predict(input_data)[0]]
    nb_prediction = data_dict["predictions_classes"][complete_nb_model.predict(input_data)[0]]
    svm_prediction = data_dict["predictions_classes"][complete_svm_model.predict(input_data)[0]]
    dt_prediction = data_dict["predictions_classes"][complete_dt_model.predict(input_data)[0]]
    ab_prediction = data_dict["predictions_classes"][complete_ab_model.predict(input_data)[0]]
    gb_prediction = data_dict["predictions_classes"][complete_gb_model.predict(input_data)[0]]

    # making final prediction by taking mode of all predictions
    final_prediction = mode([rf_prediction, nb_prediction, svm_prediction, dt_prediction, ab_prediction, gb_prediction])[0][0]
    predictions = {
        "rf_model_prediction": rf_prediction,
        "naive_bayes_prediction": nb_prediction,
        "svm_model_prediction": nb_prediction,
        "dt_model_prediction": dt_prediction,
        "ab_prediction": ab_prediction,
        "gb_prediction":gb_prediction,
        "final_prediction":final_prediction
    }
    return predictions

# Testing the function
print(predictDisease("Fatigue,Cough,High Fever,Breathlessness,Mucoid Sputum"))
```

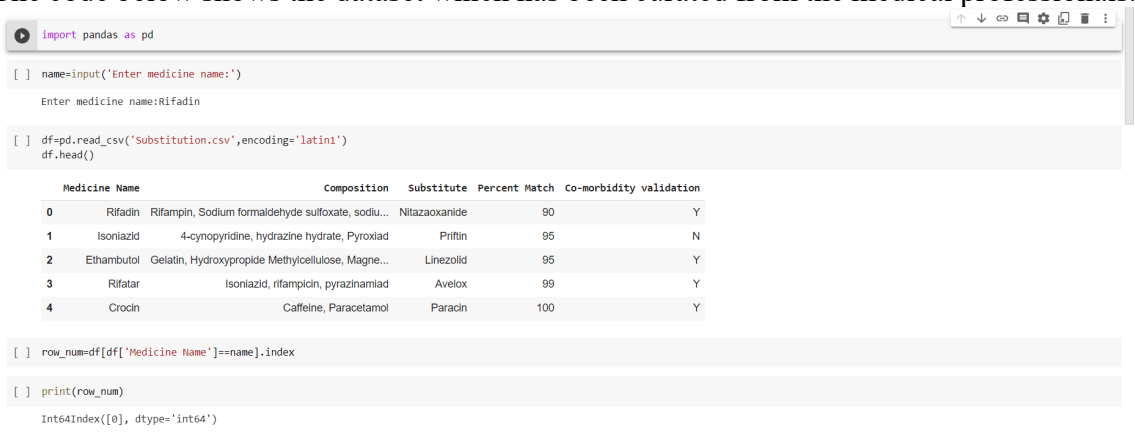
```
{'rf_model_prediction': 'Bronchial Asthma', 'naive_bayes_prediction': 'Bronchial Asthma', 'svm_model_prediction': 'Bronchial Asthma', 'dt_model_prediction': 'Bronchial Asthma', 'ab_prediction': 'Kappa', 'gb_prediction': 'Bronchial Ast
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but GaussianNB was fitted with feature names
```

Figure 31: Cumulation of ailment detection code

4.7 Medicine substitution code development

The substitute medicine feature is for finding the correct substitute to the medicines offered but also filtering them out according to the patient's physical well-being. This has been done by considering a number of factors such as the composition of the medicine, its percent match with its substitute and validations to co-morbidity conditions being applied or not.

The code below shows the dataset which has been curated from the medical professionals.



```
import pandas as pd
```

```
name=input('Enter medicine name:')
```

```
Enter medicine name:Rifadin
```

```
df=pd.read_csv('Substitution.csv',encoding='latin1')
df.head()
```

	Medicine Name	Composition	Substitute	Percent Match	Co-morbidity validation
0	Rifadin	Rifampin, Sodium formaldehyde sulfoxate, sodiu...	Nitazaoxanide	90	Y
1	Isoniazid	4-cynopyridine, hydrazine hydrate, Pyroxiad	Priftin	95	N
2	Ethambutol	Gelatin, Hydroxypropide Methylcellulose, Magne...	Linezolid	95	Y
3	Rifatar	Isoniazid, rifampicin, pyrazinamiad	Avelox	99	Y
4	Crocin	Caffeine, Paracetamol	Paracin	100	Y

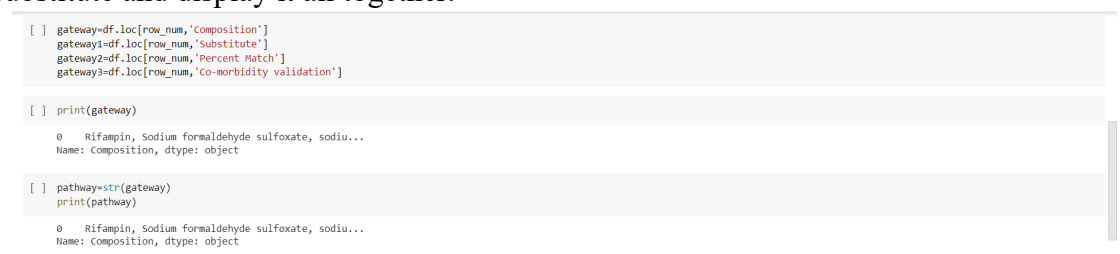
```
row_num=df[df['Medicine Name']==name].index
```

```
print(row_num)
```

```
Int64Index([0], dtype='int64')
```

Figure 32: Dataset curation

The snippet below is for obtaining the different particulars of the medicine and its substitute and display it all together.



```
gateway=df.loc[row_num,'Composition']
gateway1=df.loc[row_num,'Substitute']
gateway2=df.loc[row_num,'Percent Match']
gateway3=df.loc[row_num,'Co-morbidity validation']
```

```
print(gateway)
```

```
0    Rifampin, Sodium formaldehyde sulfoxate, sodiu...
Name: Composition, dtype: object
```

```
pathway=str(gateway)
print(pathway)
```

```
0    Rifampin, Sodium formaldehyde sulfoxate, sodiu...
Name: Composition, dtype: object
```

Figure 33: Obtaining data

The final output is displayed to the user in the format given below, where initially on entering the name of the medicine compound to be substituted, the output would be printed based on better odds of the recommended substitute on the parameters mentioned before.

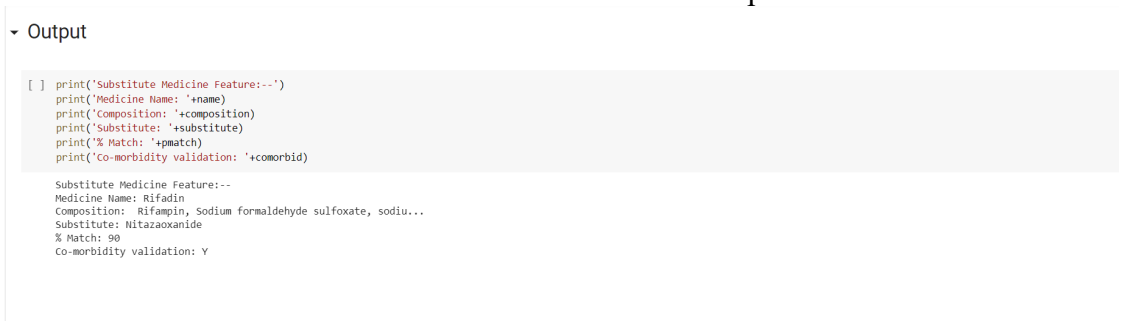
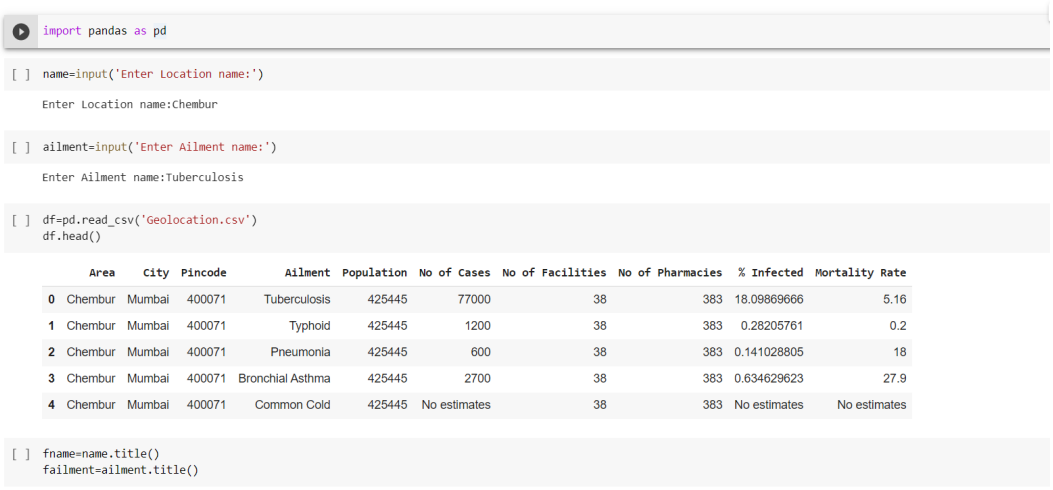


Figure 34: Final output

4.8 Geolocation Code Development

The Geolocation code is developed wherein the inputs of area and ailment are taken from the user using Geolocation function in Flutter. The coding is developed as follows where below is the snippet for taking inputs from the user, reading it and storing it into action variables

The snippet below is about taking inputs from the user and reading it and storing it into action variables



	Area	City	Pincode	Ailment	Population	No of Cases	No of Facilities	No of Pharmacies	% Infected	Mortality Rate
0	Chembur	Mumbai	400071	Tuberculosis	425445	77000	38	383	18.09869666	5.16
1	Chembur	Mumbai	400071	Typhoid	425445	1200	38	383	0.28205761	0.2
2	Chembur	Mumbai	400071	Pneumonia	425445	600	38	383	0.141028805	18
3	Chembur	Mumbai	400071	Bronchial Asthma	425445	2700	38	383	0.634629623	27.9
4	Chembur	Mumbai	400071	Common Cold	425445	No estimates	38	383	No estimates	No estimates

Figure 35: Geolocation input

The above code is about understanding the input and taking in respective functional values from the datastore.

```
[ ] fname=name.title()
    failment=ailment.title()

[ ] failment

    'Tuberculosis'

[ ] row_num=df[(df['Area']==fname) & (df['Ailment']==failment)].index

[ ] length=len(row_num)

[ ] if(length==0):
      print("Our database doesn't have a record of the credentials specified")
    else:
      gateway=df.loc[row_num,'Population']
      gateway1=df.loc[row_num,'No of Cases']
      gateway2=df.loc[row_num,'No of Facilities']
      gateway3=df.loc[row_num,'No of Pharmacies']
      gateway4=df.loc[row_num,'% Infected']
      gateway5=df.loc[row_num,'Mortality Rate']


      pathway=str(gateway)
      u1,population,u3,u4,u5,u6=pathway.split()

      pathway1=str(gateway1)
      countofwords=len(pathway1.split())
      words1,words2=pathway1.split('\n')
```

Figure 36: Taking of functional values

The output is provided below for the respective area and ailment provided

```
[ ]
    print('Geolocation Feature:--')
    print('Area: '+fname)
    print('Ailment: '+failment)
    print('No of cases: '+cases)
    print('No of facilities: '+facilities)
    print('No of pharmacies: '+pharmacies)
    print('% Infected: '+infected)
    print('Mortality Rate: '+mortality+'%')

Geolocation Feature:--
Area: Chembur
Ailment: Tuberculosis
No of cases: 77000
No of facilities: 38
No of pharmacies: 383
% Infected: 18.09869666
Mortality Rate: 5.16%
```

Figure 37: Module output

4.9 Medicine Availability Code Development

The code explained below is about finding the availability of medicines in the customer's area and provide the pharmacy address along with the specifics of the medicine available in pharmacy

The pharmacy data is loaded and here the input is taken in from the user in order to understand which area and what medicine he/she intends to search.

```
[ ] import pandas as pd

[ ] df=pd.read_csv('Pharmacy.csv')
    df.head()
```

	Area	City	Pharmacy Name	Pharmacy Latitude	Pharmacy Longitude	Pharmacy Address	Rifadin	Isoniazid	Rifatar	Crocin	Benadryl	Cetrizin	Sirturo	Zyvox	Tamiflu	Priftin	Tygacil
0	Chembur	Mumbai	Ashirwad Medical	19.05924	72.89684	Shop No.6 Runwal Shopping Centre Plot No.42 Ro...	2	0	16	25	10	0	7	0	1	2	2
1	Chembur	Mumbai	Tulsi Medical	19.05840	72.89837	2/204 Sukh Sagar Building D.k Sandu Marg Chemb...	4	0	15	30	12	2	2	0	1	3	2
2	Chembur	Mumbai	P & G Medico	19.05327	72.91572	Shop n.19/20,Gr floor, Neelkanth Garden CHS,Op...	4	2	23	23	8	2	2	3	0	4	4
3	Chembur	Mumbai	Balaji Medical	19.06260	72.90021	70/D, Uday Building, Shop No. 3 M D S Marg, Ch...	0	12	22	22	0	7	1	4	0	2	6
4	Chembur	Mumbai	Deonar Pharmacy	19.04925	72.91419	Shop No-1, Aastha- II, Beside Navadurga Apartm...	1	14	10	21	1	8	0	1	5	2	7

```
[ ] area=input('Enter the area name:')
    area

    Enter the area name:Vile Parle West
    'Vile Parle West'
```

Figure 38: Loading of pharmacy data

The output is provided in the form of a list which displays the medicine availability of different pharmacies in the location of the user.

```
[5] list=[]
    for ind in df.index:
      if(df['Area'][ind]==area):
        count=df.iloc[ind][med]
        if(count>0):
          thisdict={
              "Medical":df['Pharmacy Name'][ind],
              "Latitude":df['Pharmacy Latitude'][ind],
              "Longitude":df['Pharmacy Longitude'][ind],
              "Address":df['Pharmacy Address'][ind]
          }
          list.append(thisdict)
        else:
          continue
      else:
        continue


    print(list)

[{'Medical': 'Amar Medical Store', 'Latitude': 19.01591, 'Longitude': 72.8369, 'Address': 'Shop Nummer 1 R.N. Cooper Hospital Compound, 3, Gulmohar Rd, JVPD Scheme, Vile Parle West,
```

Figure 39: Medicine output

Chapter 5: Screenshots

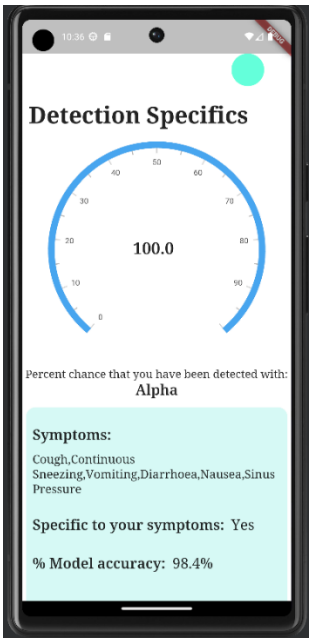
5.1 Front end development

The frontend behaves differently for every feature and the overall user journey on the app is explained in this section

Figure 40: Front end development
DETECTION FEATURE

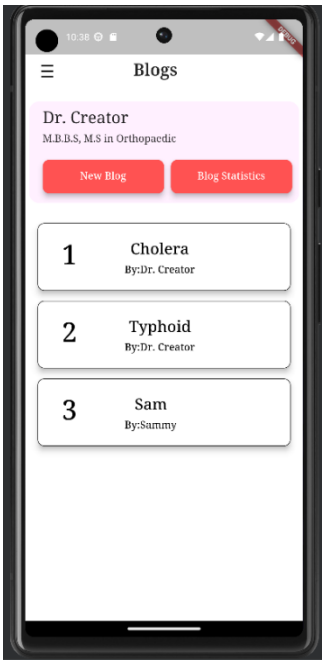
User inputs the symptoms

Loading screen for formulating output based on inputs given

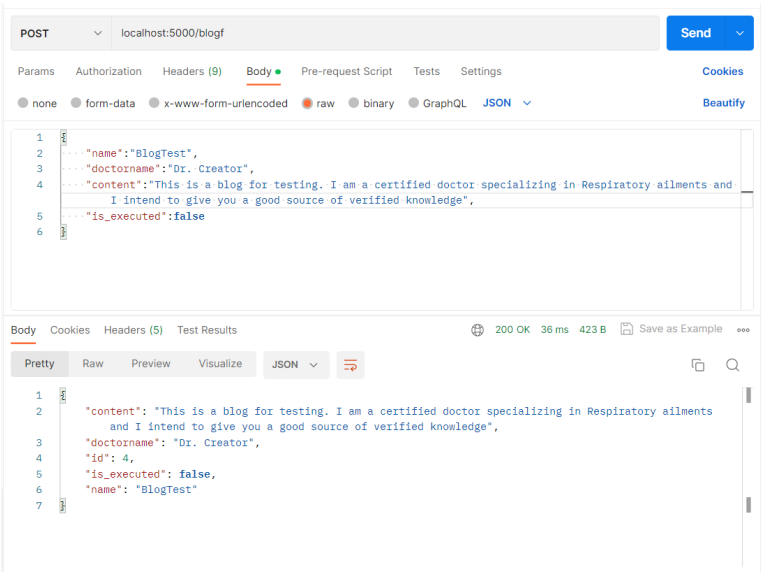


The output is provided with all the specifics

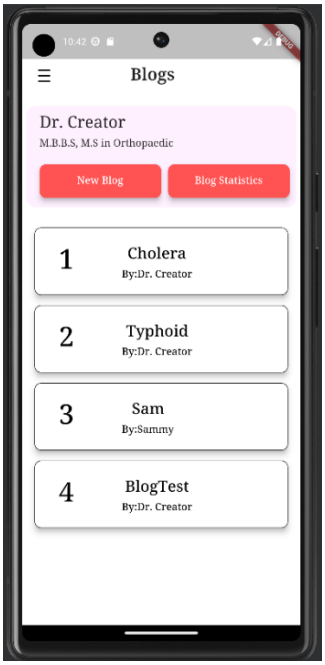
BLOG FEATURE



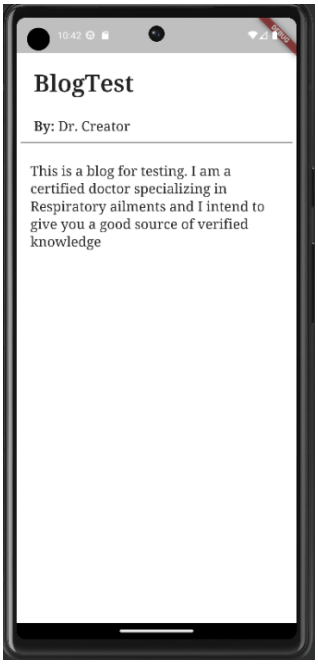
User can view the blogs.



The updation of the blog into the database

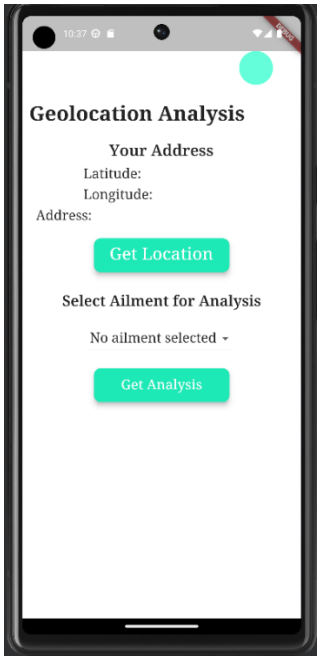


The updation of new blogs onto the application front end

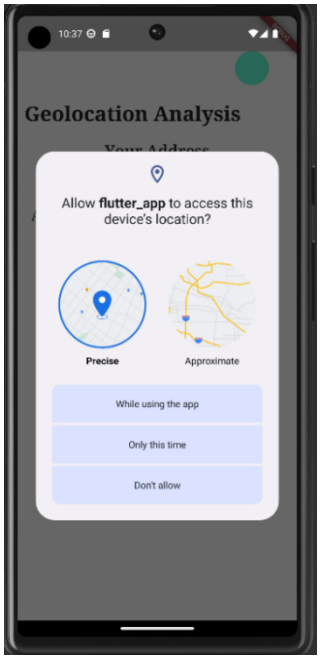


The content of the new blog is displayed on a separate page

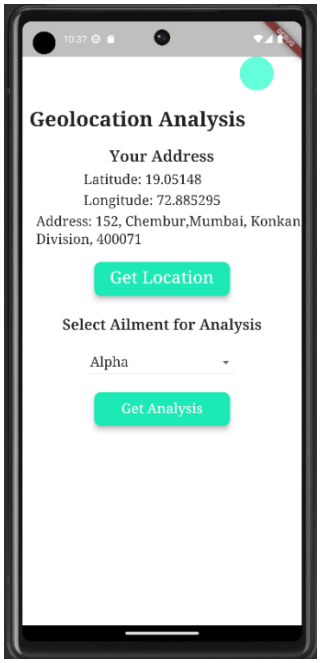
GEOLOCATION FEATURE



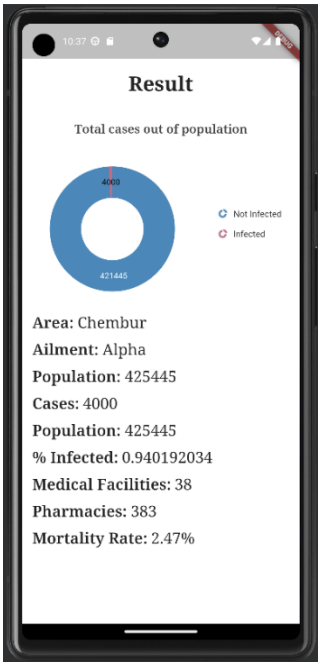
Intro Page to the Geolocation analysis consist of location and ailment sections



User permission is asked for the location

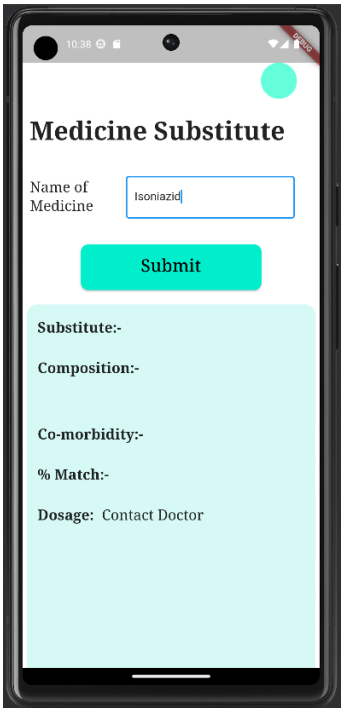


The location is displayed in the app.

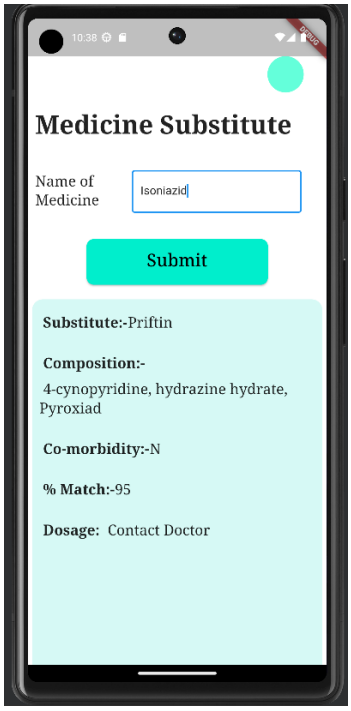


The final Geolocation analysis provides the information.

SUBSTITUTION FEATURE



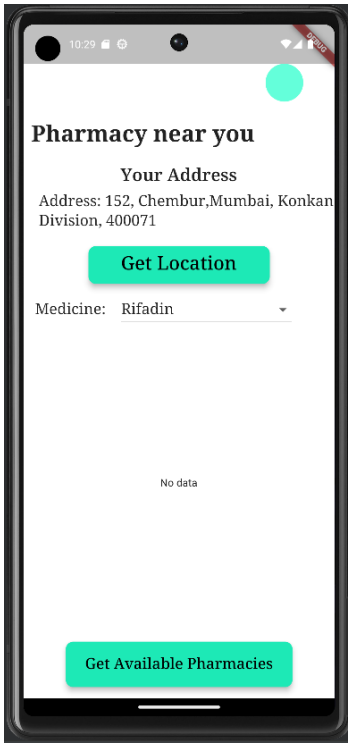
The intro screen is displayed with prompt for name



The output is provided as per the logic developed

PHARMACY DATA

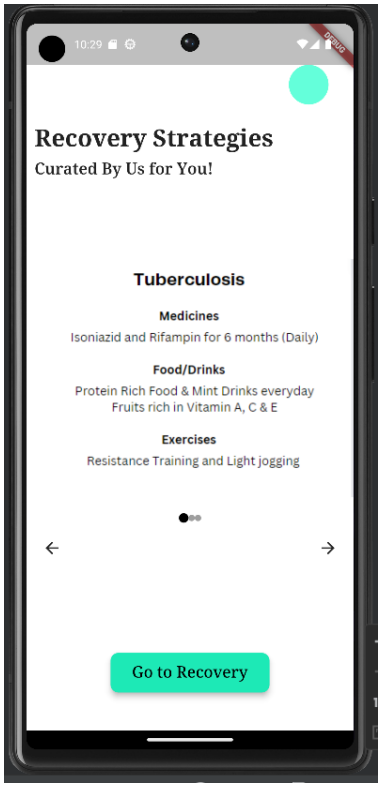
The Pharmacy Home Screen when loaded



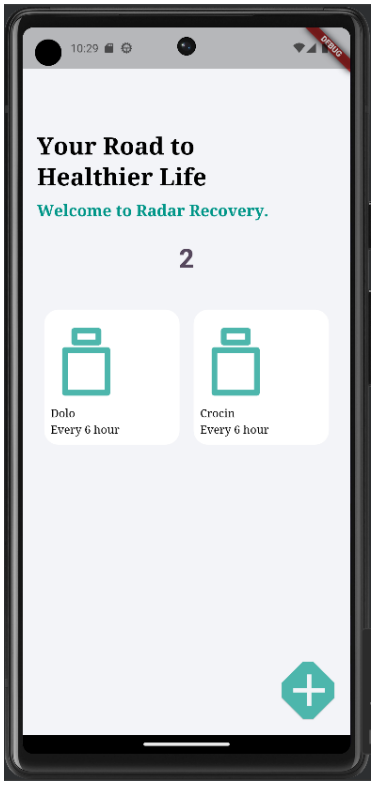
The User inputs the medicines and allows Location access

The Pharmacies with available medicines are displayed

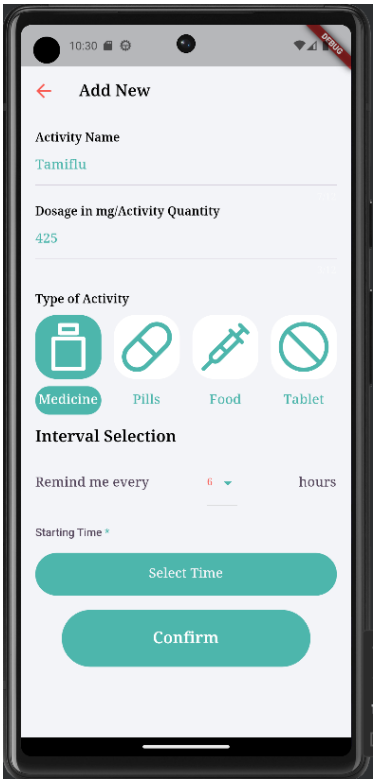
RECOVERY



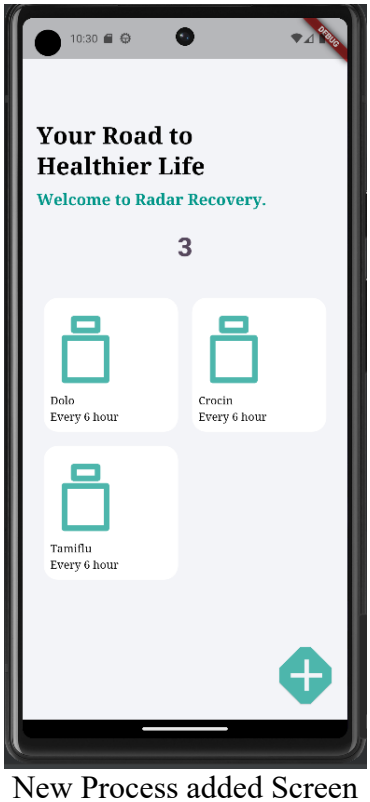
The Strategies Screen for the User



The Recovery Home Page



The New Process Page



New Process added Screen

Chapter 6: Project Test Report

6.1 Machine learning code

Sr No	Testing Case	Case Description	Expected output	Actual Output
1	No Input parameters	User doesn't enter any of the inputs if required	An error message is displayed stating empty inputs	A flutter toast message is displayed showing 'No inputs fed'
2	Respiratory Symptoms parameter fed	User selects and submits a combination of different symptoms	A Success message is displayed	A flutter toast message is displayed showing 'Success'

6.2 Substitution

Sr No	Testing Case	Case Description	Expected output	Actual Output
1	No Input Medicine	User doesn't enter any medicine name in the text field	An error message is displayed stating empty input	A flutter toast message is displayed showing 'No input fed'
2	Wrong medicine name entered	User enters a medicine name which doesn't exist	An error message is displayed stating that medicine doesn't exist	A flutter toast message is displayed showing 'Medicine doesn't exist'
3	Correct Medicine Name entered	User enters a medicine name which exists	A success message is displayed along with the output	A flutter toast message displayed 'Success' is displayed

6.3 Geolocation Analysis

Sr No	Testing Case	Case Description	Expected output	Actual Output
1	No Input Parameters fed	User doesn't enable GPS to find location or doesn't select the ailment	An error message is displayed stating empty input field	A flutter toast message is displayed showing 'Empty input'
2	Location and Ailment selected	User enters proper location and ailment information	A Success Message is displayed on the Screen	A flutter toast message is displayed showing 'Success'

6.4 Recovery

Sr No	Testing Case	Case Description	Expected output	Actual Output
1	No Steps fed as an input	User doesn't enter the descriptions of the tasks to be done	No action to be performed.	Flutter function doesn't move to the next screen
2	All Parameters fed and submitted	User enters all the information and clicks on submit.	A Success Message is displayed on the Screen	Flutter Splash Screen is displayed with a success icon

6.5 Pharmacy

Sr No	Testing Case	Case Description	Expected output	Actual Output
1	No Input Parameters fed	User doesn't enable GPS to find location or doesn't select the medicine name.	An error message is displayed stating empty input field	A flutter toast message is displayed showing 'Empty input'
2	Location and Medicine selected	User enters proper location and ailment information	A Success Message is displayed on the Screen	A flutter toast message is displayed showing 'Success' and Medical Store Names are displayed

6.6 Database

Sr No	Testing Case	Case Description	Expected output	Actual Output
1	No Input Credentials	User doesn't enter the username and Password in the text fields present	An error message is displayed stating empty fields	A flutter error message is displayed showing 'Required' below each field
2	Wrong Input Credentials	User enters wrong credentials in the text fields	An error message is displayed stating that User doesn't exist	A flutter toast message is displayed showing 'No user present'
3	Correct Credentials entered	User enters the correct username and password which exists	A success message is displayed on the screen	A flutter toast message displayed 'Success' is displayed and is directed to the Home Page

Chapter 7: Results and Discussion

The Flask backend server initializes as per the configurations. The flutter app builds without any errors and the APK files are generated accordingly. All the functionalities which are embedded in the application connect properly with the backend servers created and functions as per the initial stated requirement. The Machine learning models developed accept the symptoms and process them to generate a systematic ailment detection. Also, the SQL Alchemy and the MongoDB databases configured function properly for performing the required functionality of the application. The REST API calls function properly to establish the connection between the frontend and the backend, hence ensuring the smooth functioning of the entire application.

All the internal and external libraries used in the application configure properly in order to perform the actions as initiated by the user. Also, since the initiation of the application from the stakeholder's side, there have been no errors found along with a smooth walkthrough of the application from sign-in to the usage of the application. The build and stability of the application makes it possible to run on any Android device fulfilling all the before mentioned objectives to be carried out.

Chapter 8: Conclusion and Future Avenues

8.1 Conclusion

What started as an idea to bring doctors, patients and the whole medical infrastructure closer at a time when social distancing, fear and a lack of trust had broken down communications in the medical treatment process, has now transformed into a working application that uses the efficacy of a doctor's knowledge to streamline medical treatment processes.

Through extensive research, we have developed "RADAR" to help everyone live a healthier life and to assist doctors and medical professionals in delivering this. As has always been set out, the aim was never to replace the healthcare infrastructure but to optimize it and assist it with the application acting as the first line of defense.

8.2 Future Avenues

- To increase the accessibility of the application, in addition to the NLP, the application can also be converted into different vernacular languages and also an advanced level of speech to text and vice versa can be implemented for making the application close to hands free.
- To increase the scope of the detection feature of our application, a whole sum combination of all the ailments pertaining to the human body can be included which is what is desired ultimately by the doctors with whom we are tied up with
- To provide real time geolocation analysis, the application can be further integrated with the government portal for accessing and displaying real time data to the stakeholders.

Chapter 9: References

1. A. K. Singh, "A Comparative Study on Disease Classification using Machine Learning Algorithms," SSRN Electron. J., no. August, 2019, doi: 10.2139/ssrn.3350251
2. G. C. Hansen, K. H. Falkenbach, and I. Yaghmai, "Voice recognition system.," Radiology, vol. 169, no. 2, p. 580, 1988, doi: 10.1148/radiology.169.2.3175016
3. J. K. Kairi and A. K. Sharma, "Substitutes of prescription medicines - A review of concerns relevant to doctors and patients," J. Clin. Diagnostic Res., vol. 11, no. 8, pp. FE01–FE05, 2017, doi: 10.7860/JCDR/ 2017/28552.10439
4. B. Pandipati and R. P. Sam, "Speech to text Conversion using Deep Learning Neural Net Methods," vol. 12, no. 05, pp. 2037–2042, 2021
5. C. Huang et al., "Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China," Lancet, vol. 395, no. 10223, pp. 497–506, 2020, doi: 10.1016/S0140-6736(20)30183-5
6. M. Mosa, I. Yoo, and L. Sheets, "A systematic review of healthcare applications for smartphones. BMC Med.Inf. Decision Making," BMC Med. Inform. Decis. Mak., vol. 12, no. 67, pp. 1–31, 2012, [Online]. Available: http://www.biomedcentral.com/1472-6947/12/67