**Logo, company name

Description automatically generated**

**COMSATS University Islamabad (CUI)**

Software Design Description   
(SDS DOCUMENT)

for

**BreathEx**

Version 1.0

***By***

**Ahmed Kamran CIIT/SP20-BSE-004/ISB**

**Daniyal Hassan CIIT/SP20-BSE-025/ISB**

***Supervisor*Maam Sadia Maqbool**

***Bachelor of Science in Computer Science (2020-2024)***

**Table of Contents**

Revision History iii

Application Evaluation History iv

1. Introduction 1

2. Design Methodology and Software Process Model 1

3. System Overview 1

3.1 Architectural Design 1

4. Design Models 1

4.1. Activity Diagram……………………………………………………………………………………………………………………..1

4.2 Data Flow Diagram………………………………………………………………………………………………………............1

4.3 State Machine Diagram…………………………………………………………………………………………………………..1

4.4 Sequence Diagram………………………………………………………………………………………………………………….1

4.5 Class Diagram…………………………………………………………………………………………………………………………1

5. Data Design 2

5.1 Data Dictionary 2

6. Human Interface Design 2

6.1 Screen Images 2

6.2 Screen Objects and Actions 2

7. Implementation 3

7.1 Algorithm 3

7.2 External APIs/SDKs 5

7.3 User Interface 5

7.4 Deployment 8

8. Testing and Evaluation 9

8.1 Unit Testing 9

8.2 Functional Testing 9

8.3 Business Rules Testing 10

8.4 Integration Testing 10

Appendix A 11

Appendix B 13

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

Application Evaluation History

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |
|  |  |
|  |  |

Supervised by

Supervisor’s Name

**Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# **Introduction**

The product that we are going to built is BreathEx, which will help the patient to detect diseases like tuberculosis,covid, pneumonia by scanning x-rays of the patient using ML. There are various other functionalities in our application such as chatbot,infobox, medicine reminder, application feedback, ,prescription,report generation and management. This application will be helpful for both patient and doctor in order to manage patients record and help the doctor in curing the disease(tuberculosis, pneumonia, covid). Image processing and x-rays detection will be the prime focus of our work.

The application would be designed to analyze chest X-rays to detect signs of TB, COVID-19, and pneumonia. The application would use machine learning algorithms, such as deep learning and computer vision, to accurately diagnose the diseases based on the features identified in the X-rays. The application would be trained on a large dataset of chest X-rays, annotated with ground truth labels, to increase its accuracy in detecting the diseases. The application could be integrated with existing healthcare systems to support healthcare professionals in making prompt and accurate diagnoses. The application would have an intuitive user interface for healthcare professionals to easily access and use the system. The application would generate reports that provide information on the diagnosis, including visualizations of the X-rays and the results of the analysis. The application would be validated using appropriate metrics, such as sensitivity, specificity, and accuracy, to ensure its accuracy in detecting TB, COVID-19, and pneumonia. First of all user(patient) enter the system by login into the system. The  system will display list of all the diseases available in the system. After that user(patient) can select a disease to detect. Then system will display result of detection and display a report of disease with which user is suffering from. These reports will assist the doctors in proceeding the treatment of the patient. Then system will recommend medical suggestion to user for cure of disease for time being. User(patient) can also set alarms for taking medicine at specific time and days. The system can detect three major diseases which include covid 19 ,pneumonia, tuberculosis . Tuberculosis is detected with the analysis on x-rays which can achieved through ML by gathering datasets of different patients. The system is also capable of detecting COVID by analyzing X-rays of patients through ML. The system will give optimal result.

# **Design Methadology and Software Process Model**

Procedural methodology and Object-Oriented Programming both will be used in our project.AI Machine learning model will be trained by using OOP approach and the rest of the project will use procedural methodology.

**Rationale:**

The project is mainly using Object-Orietned programming. The reason of choosing this approach is that our app frontend is in flutter which uses dart. Dart is an object-oriented programming language. There are several benefits of using object oriented approach which includes encapsulation, inheritance, polymorphism, reusability and modularity. Using Object-Oriented approach in flutter helps us create well-organized, efficient, and flexible application that is easier to maintain ,develop and test

The incremental model will be our software process model for our project.

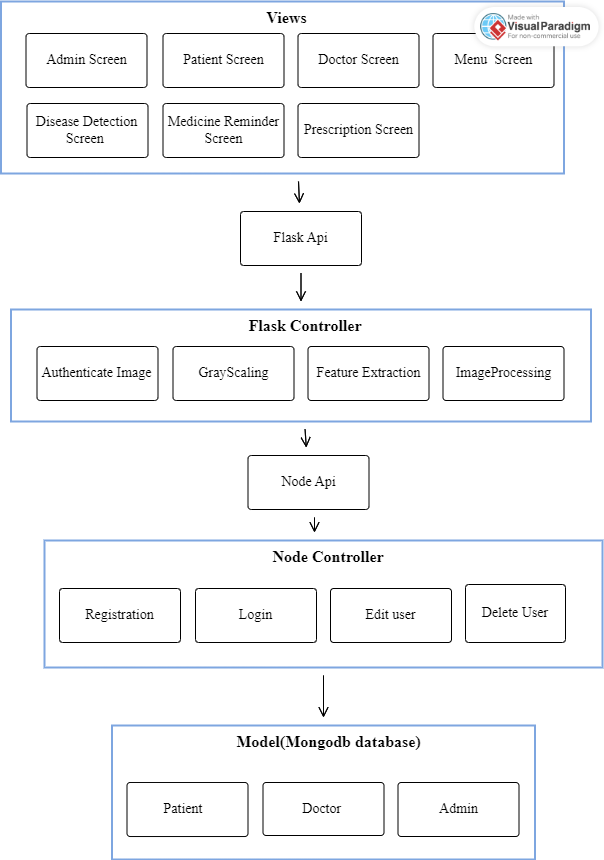
**Rationale:**

The rationale for choosing this approach is that we want our system to be created in small increments, with small by-products delivered. This methodology is chosen because it is easier to validate and test the short iterations. Another rationale for using the incremental model is that we want to start small with smaller by-products, then develop up to a high level, and finally to complete it.

# **System Overview**

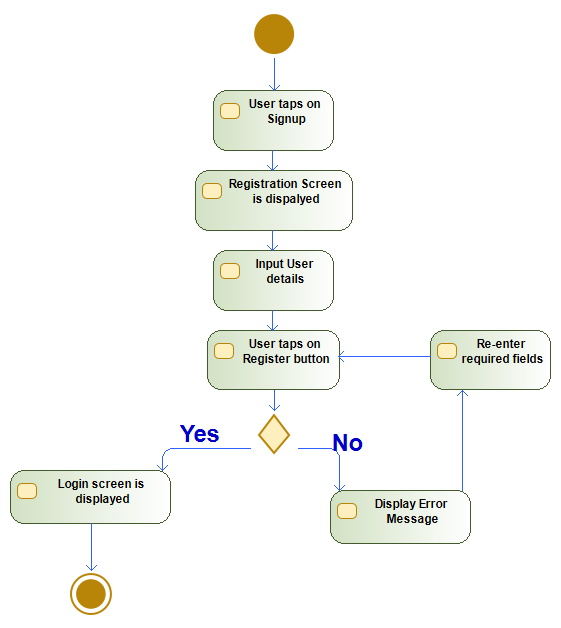
The purpose of this document is to provide a comprehensive system design and functionality. A architectural representation and design of the user's and the system's perspectives, respectively, are provided in the architectural diagram and all other design models diagrams for the project overview. In addition, an data flow diagram, detailed class diagram provide the comprehensive system details and data flow. Implementation details are included in pseudo code, user interface sections. Testing details are also show in this document

## **Architectural Design:**

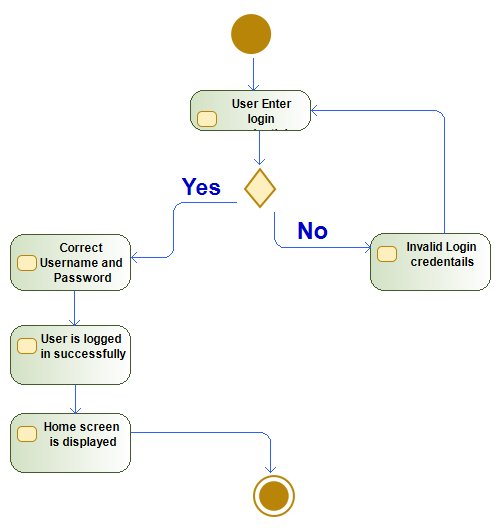


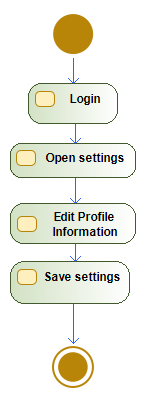
# **4. Design Models**

# **4.1 Activity Diagrams:**

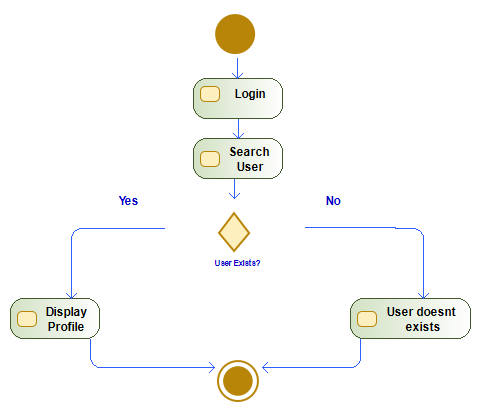
****

**Figure 1: Register**

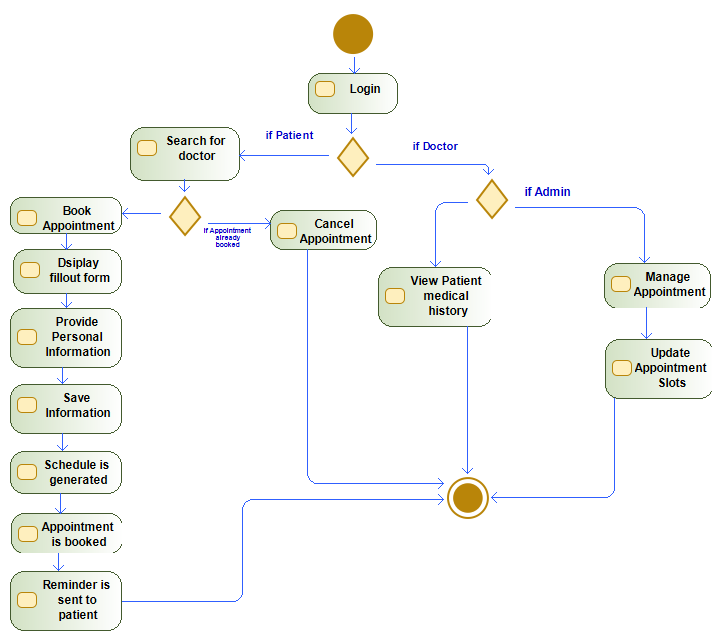
**  
Figure 2: Login**

****

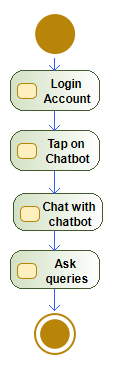
**Figure 3: Edit Profile**

****

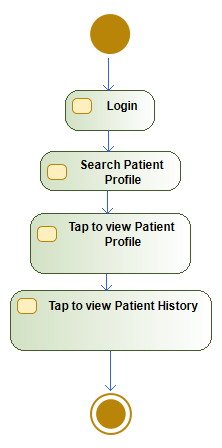
**Figure 4: Search Profile**

****

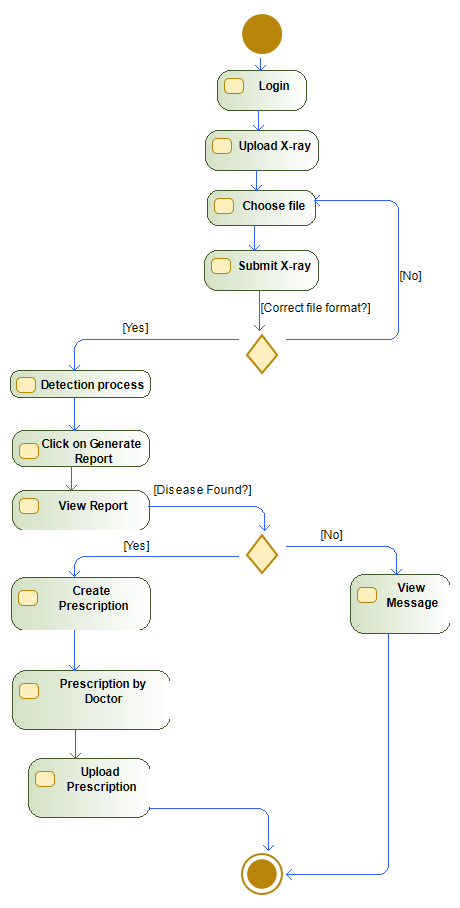
**Figure 5: Book Appointment**

****

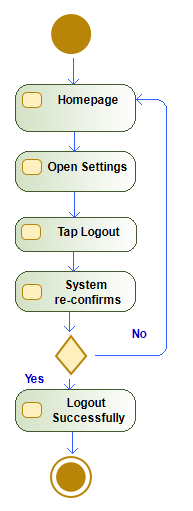
**Figure 6: Chatbot**

****

**Figure 7: View Patient History**

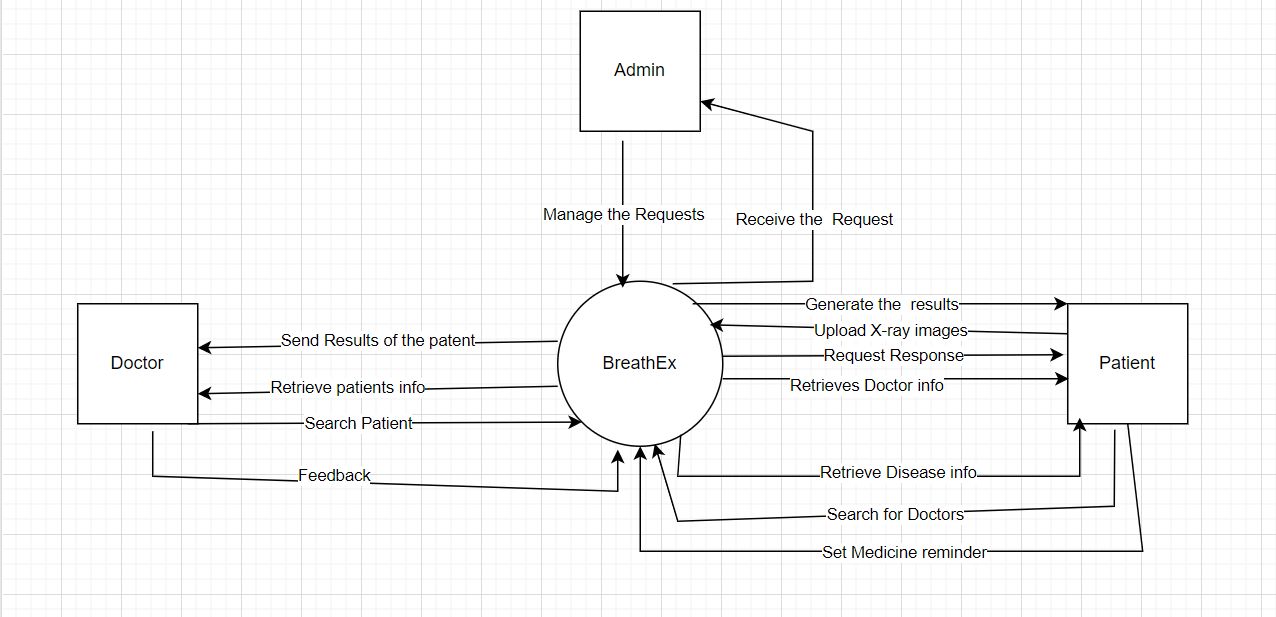
****

**Figure 8: Disease Detection**

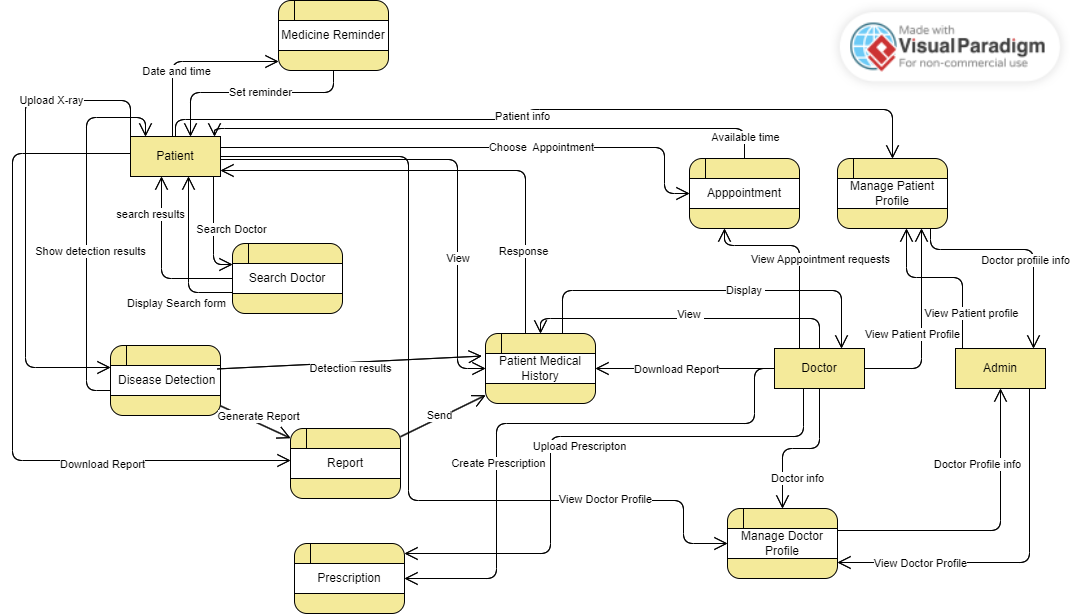
****

**Figure 9: Logout**

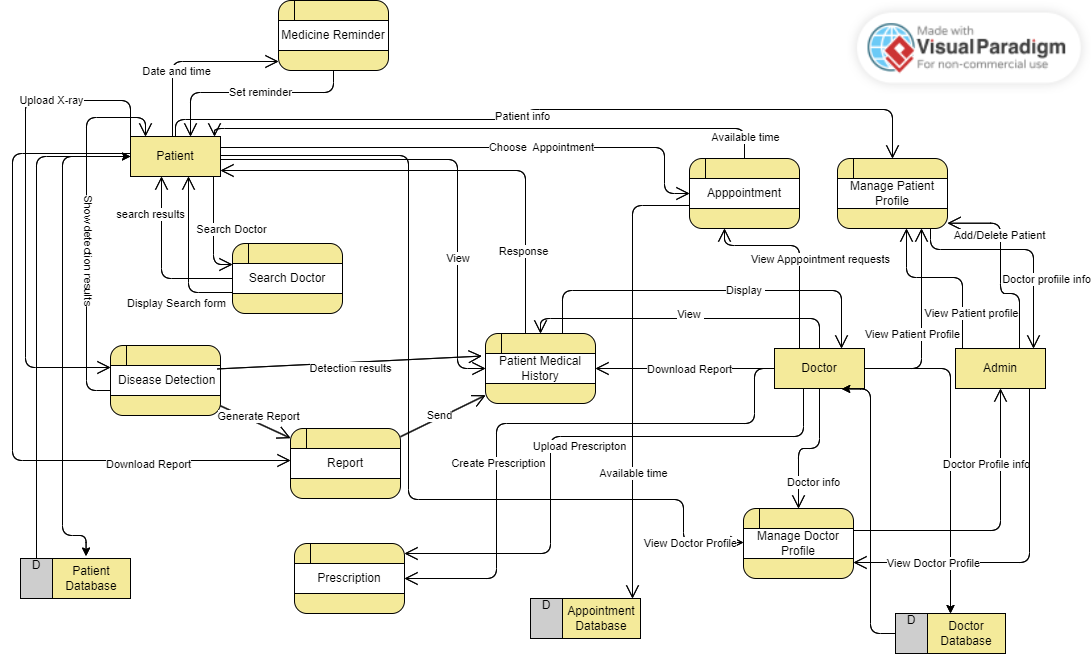
## **Data Flow Diagrams:**

****

**Figure 10: DFD Level 0**

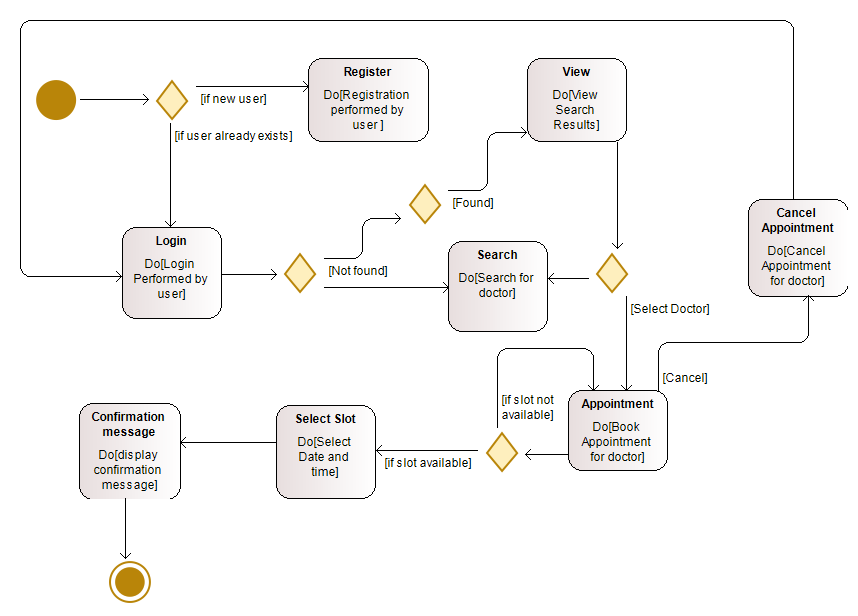
****

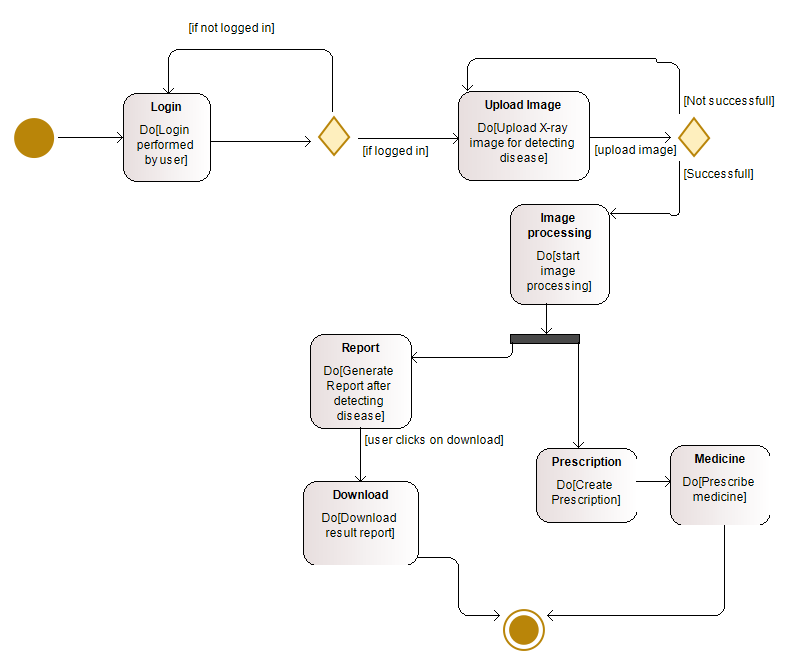
**Figure 11: DFD Level 1**

****

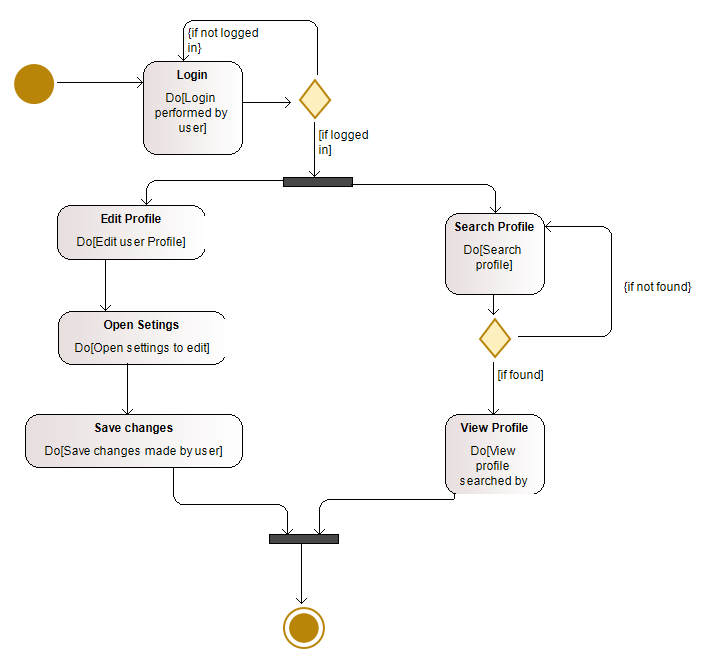
**Figure 12: DFD Level 2**

**4.2 State Transition Diagrams:**

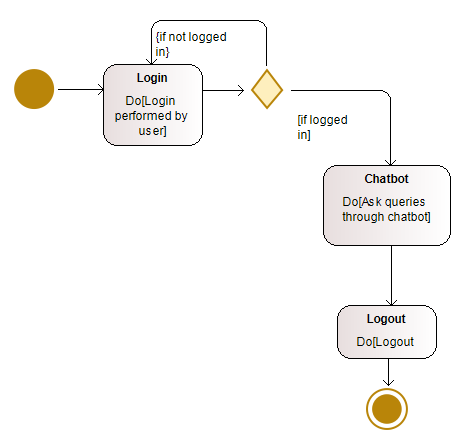
**Figure 13: State Transition diagram 1**

****

**Figure 14: State Transition diagram 2**

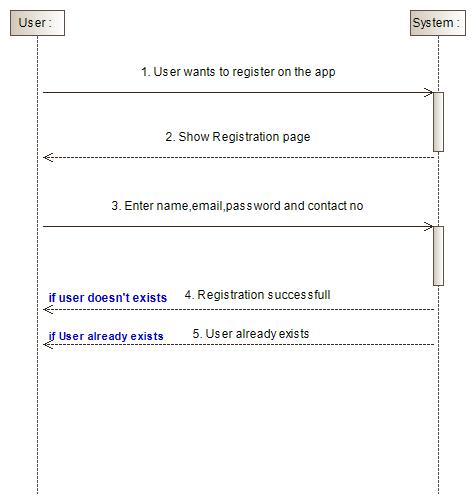
****

**Figure 15: State Transition diagram 3**

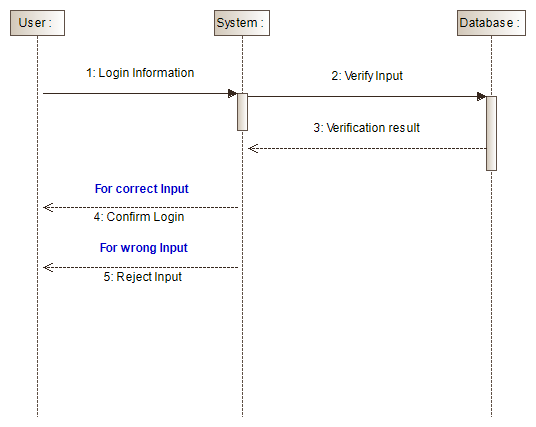
****

**Figure 16: State Transition diagram 4**

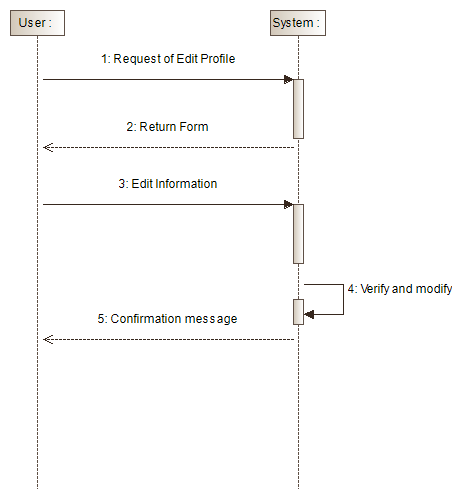
## **Sequence Diagrams**

****

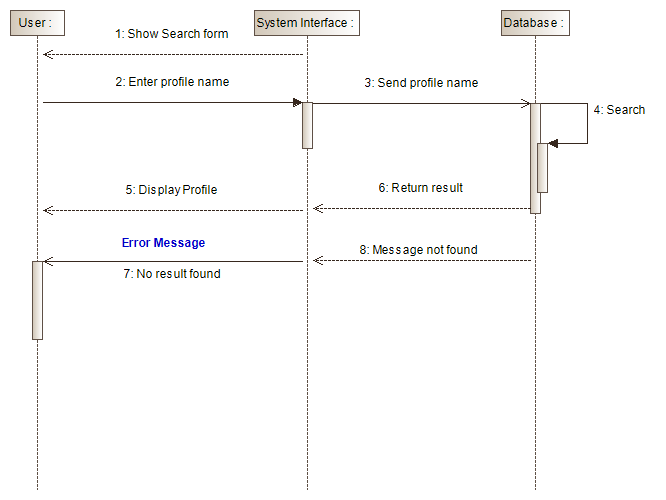
**Figure 17: Registration**

****

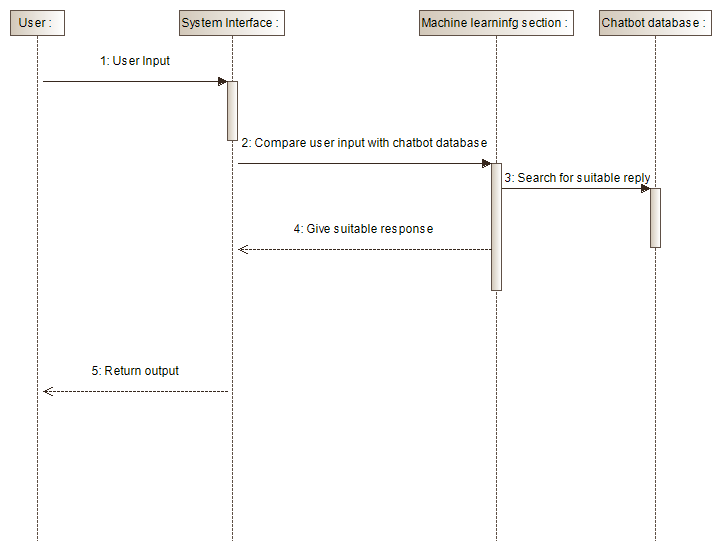
**Figure 18: Login**

****

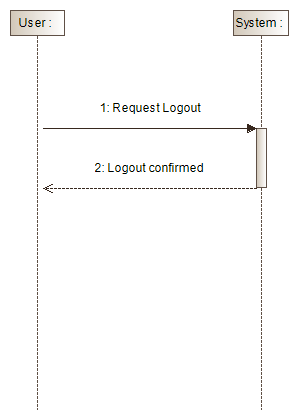
**Figure 19: Edit Profile**

****

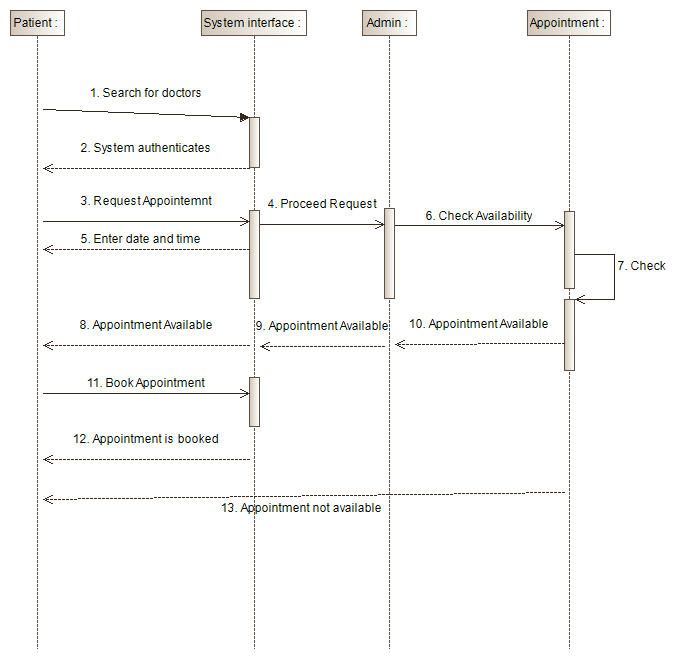
**Figure 20: Search Profile**

****

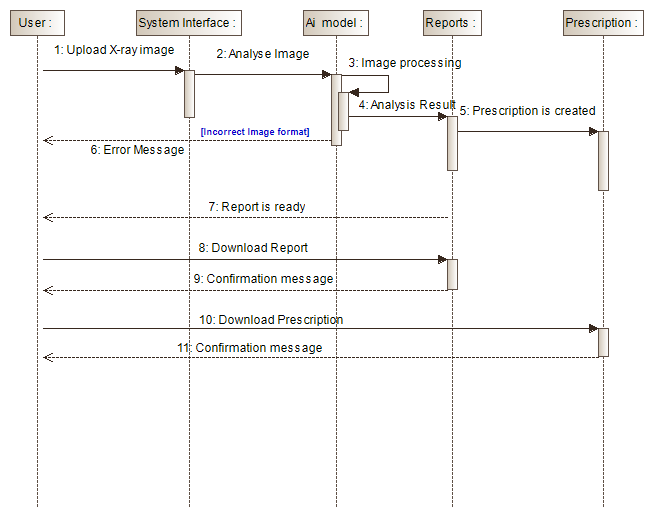
**Figure 21: Chatbot**



**Figure 22: Logout**

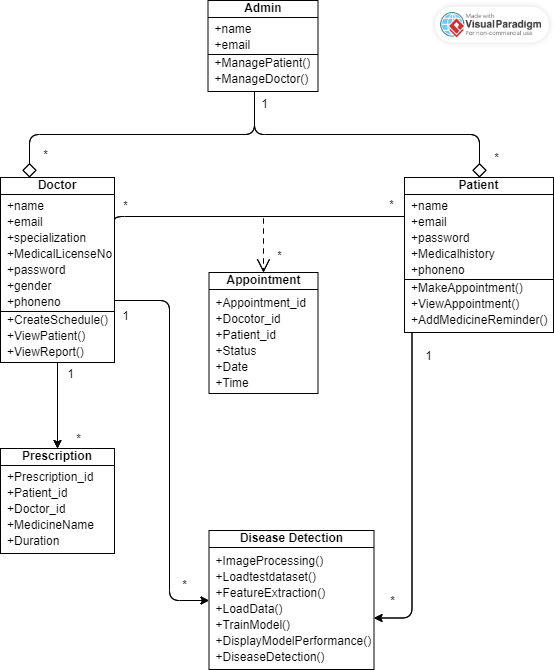


**Figure 23: Book Appointment**



**Figure 23: Disease Detection**

**4.5 Class Diagram:**

****

**Figure 24: Class Diagram**

# **Data Design**

The database that will be used for our project is Mongodb, The data of the users will be stored in the form of Schemas

## **5.1. Data Dictionary**

**Table 1: Data Dictionary of Doctor**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Range of Values** | **Description** |
| Name | String | 20 | Unique name of doctor |
| Email | String | 30 | Unique email of doctor |
| Specialization | String | 20 | Doctor specialization |
| MedicalLicenseNo | String | 15 | Unique Medical Lincense no |
| Password | String | - | Password for account security |
| Gender | String | 20 | gender of doctor |
| Phone\_no | Integer | 15 | Doctor personal contact no |
| Admin | Mongoose.Schema.Types.ObjectId | **-** | Admin details |
| patients | Mongoose.Schema.Types.ObjectId | **-** | Patients details |

**Table 2: Data Dictionary of Patient**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Range of Values** | **Description** |
| Name | String | 20 | Unique name of patient |
| Email | String | 30 | Unique email of doctor |
| Medical History | String | 50 | Patient’s medical history |
| Password | String | - | Password for account security |
| Phone\_no | String | 20 | Patient’s personal phone no |
| Doctors | Mongoose.Schema.Types.ObjectId | - | Doctor details |
| Admin | Mongoose.Schema.Types.ObjectId | - | Admin details |

**Table 3: Data Dictionary of Admin**

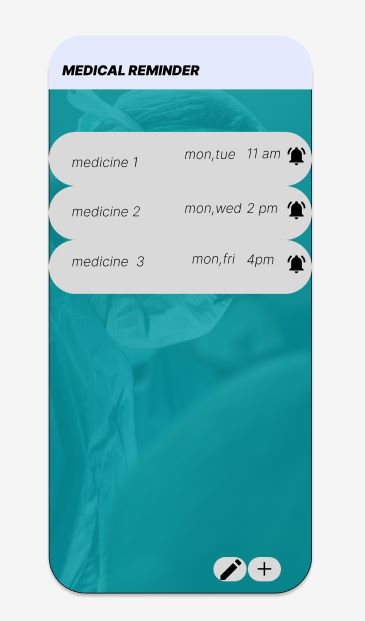
|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Range of Values** | **Description** |
| Doctors | Mongoose.Schema.Types.ObjectId | **-** | Doctor details |
| Patient | Mongoose.Schema.Types.ObjectId | **-** | Patient details |
| Email | String | 30 | Unique email of admin |
| Name | String | 20 | Unique name of admin |
| Password | String | **-** | Password for account security |

# **Human Interface Design**

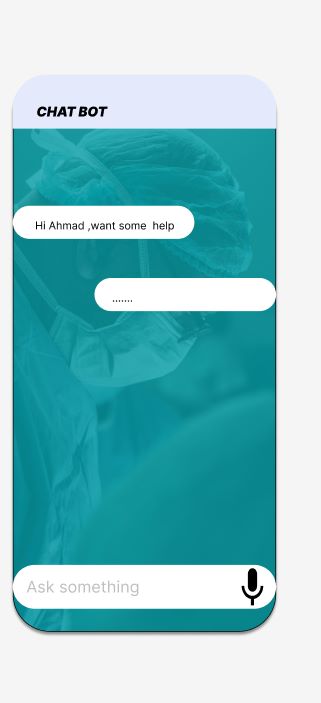
## **Screen Images**



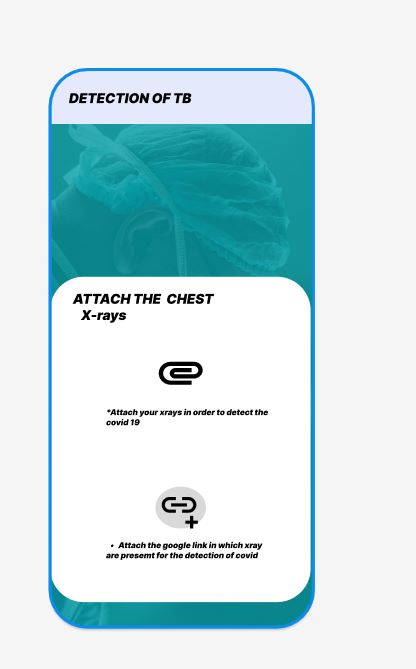
**Figure 25: Menu Screen**



**Figure 26: Medicine Reminder Screen**



**Figure 27: Chatbot Screen**

****

**Figure 28: Disease Detection Screen**

## **Screen Objects and Actions**

Figure 25: It is the menu screen of the application after login user is navigated to the menu screen

Figure 26: It is Medicine reminder which shows the all reminders with data and time

Figure 27: It is the Chatbot screen which shows user message and system response in the application

Figure 28: It is disease detection screen which ask the user to input x-ray image and attach report for disease detection,.

# **Implementation**

## **Algorithms**

We will use CNN Algorithm, which is mainly used in image recognition applications. As it focuses less on pre-processing, less human effort is required to implement its functionalities. The algorithm is simple to comprehend and quick to put into practice. Among all image classification algorithm it has highest accuracy.

### **CNN Algorithm**

The convolutional Neural Network CNN works by gathering an image, giving it a weight based on the many features in the image, and then separating them from one another

### **CNN Pseudo code**

## **External APIs/SDKs**

**Table 1 Details of APIs used in the project**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of API and version** | **Description of API** | **Purpose of usage** | **List down the API endpoint/function/class in which it is used** |
| GoogleCollab | For Image processing, Feature  extraction, Model training, Disease detection and Prediction of disease | This API is used to train the model in order to detect Tb, Pneumonia and COVID |  |

## **User Interface**

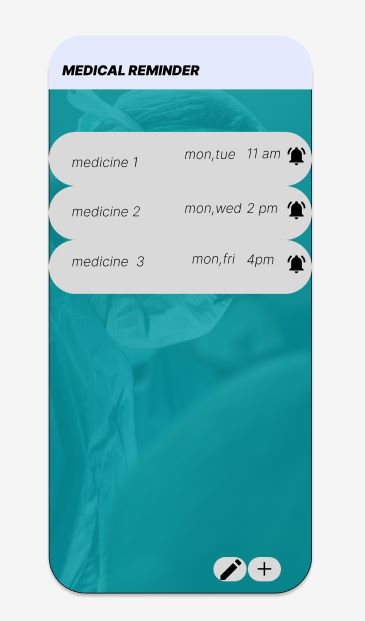
### **Menu Screen**

This is the Menu Screen which shows all the features of our application where user has to select any feature to use the particular feature



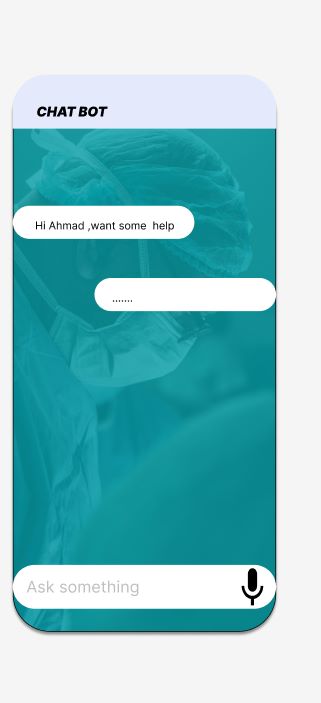
### **Medicine Reminder Screen**

This is the Medicine Reminder Screen where user will select date and time to set medicine reminder



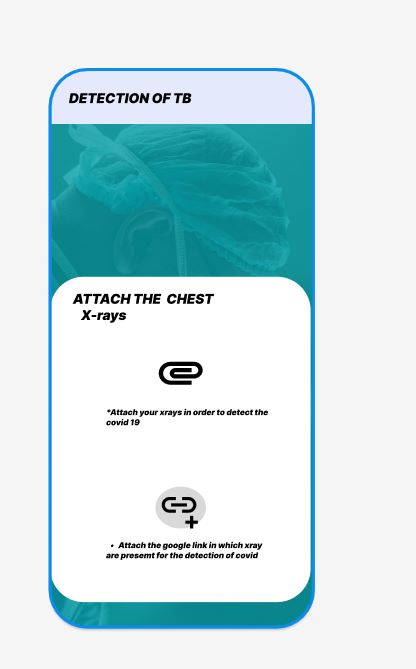
### **Chatbot**

This is the chatbot screen in which user has to write message and tap on send button to send message to and get medical assistance from system.



### **Disease Detection**

This is the disease detection screen in which user needs to upload his/her x-ray report and image to see result of detection

****

## **Deployment**

We will deploy our app by using flutter emulator and simulator. During development testing team can use emulators provided by flutter to test app on different platforms like IOS and Android.

**Physical Device:**

For testing our app in the physical device we can run our app directly on IOS or Android.

**Node Js Backend:**

Local development environment during development we can use node js backend locally on machine using node package manager and local package server

|  |  |
| --- | --- |
| **Softwares** | **Versions** |
| Flutter | 3.10.0 |
| Node JS | 18.15.0 |

# **Testing and Evaluation**