

MIST CAFETERIA APP

**Software Requirements Specification**

Table of Contents

1. [Preface 2](#_TOC_250023)
2. [Introduction 3](#_TOC_250022)
   1. [Purpose 3](#_TOC_250021)
   2. [Intended Audience 3](#_TOC_250020)
   3. [Scope 3](#_TOC_250019)
3. [Glossary 3](#_TOC_250018)
4. [Requirement’s discovery 4](#_TOC_250017)
   1. [Literature Review 4](#_TOC_250016)
   2. [Interview 6](#_TOC_250015)
   3. [Survey 6](#_TOC_250014)
5. [User Requirements 7](#_TOC_250013)
6. [System Architecture 7](#_TOC_250012)
7. [System Requirement Specification 8](#_TOC_250011)
   1. [System Requirements 8](#_TOC_250010)
   2. [Requirements Classification 9](#_TOC_250009)
8. [System Model 10](#_TOC_250008)
   1. [Context Diagram 10](#_TOC_250007)
   2. [Activity Diagram 11](#_TOC_250006)
   3. [Use Case Diagram 14](#_TOC_250005)
   4. [Sequence Diagram 20](#_TOC_250004)
   5. [State Diagram 25](#_TOC_250003)
9. [Appendix 28](#_TOC_250002)
   1. [Questionnaires 28](#_TOC_250001)
   2. [Some Responses 29](#_TOC_250000)
10. References 31

# Preface

Among the university cafeterias, the MIST Cafeteria is one of the largest. However, it is crucial to recognize and deal with the different operational issues that at the moment prevent its seamless service. These difficulties primarily take the shape of slow service and long lines for placing orders.

Customers must wait in line to get their meal tokens at the token counter, which creates the initial bottleneck. After receiving tokens, a sizable crowd gathers at the food counter, prolonging the service time even more. The inspection of the tokens to identify the precise foods and quantities that are linked with is the next step, after which each meal is packed individually and served. These processes add up to create significant delays, which is disruptive for students who can't eat on time and results in late for their subsequent class schedules.

To address these issues and enhance the overall dining experience at the Mist Cafeteria, we propose the development of a mobile application. This application aims to significantly reduce wait times by allowing patrons to place their orders in advance, streamlining the process, and providing an efficient solution. Furthermore, the application will incorporate convenient payment options, such as the use of bKash, thereby further improving the overall customer experience.

# Introduction

## Purpose

The "MIST Cafeteria App" is a mobile application system designed to streamline the food ordering process. Users can select food items from a menu, make payments using "bKash," and receive a digital token for cafeteria pickup. An admin page will display all incoming orders for cafeteria staff to prepare and organize. This automation aims to significantly reduce waiting times and enhance the overall dining experience at Mist Cafeteria. The app's core purposes include order collection, payment facilitation, menu presentation, and order management, all geared towards improving efficiency and convenience for users and cafeteria staff alike.

## Intended Audience

This application is primarily designed for the MIST community, including students, faculty members, and staff. It will be especially beneficial for students who need to quickly get a meal and make it to their classes on time. Additionally, this app is open to anyone who wishes to purchase food from Mist Cafeteria, extending its convenience beyond the MIST community.

## Scope

The "MIST Cafeteria App" is mainly a mobile application with a database. An Internet connection is required for this app. The app will facilitate secure payments through the "bKash" system, ensuring a convenient and cashless transaction process The Mist Cafeteria App focuses on streamlining the food ordering process within the MIST community and beyond.

# Glossary

**Database:** A database is a structured set of data held in a computer.

**bKash**: A secure mobile payment system used for processing financial transactions within the application.

**Token**: A digital representation of a user's order, generated by the app for presentation at the cafeteria counter for meal pickup.

**Admin Page**: The section of the application accessible to cafeteria staff for order management and meal preparation.

# Requirement’s discovery

## Literature Review:

A good number of articles, journals, conferences, and research papers were recited, and some important viewpoints were observed. Based on health, there is a lot of research, projects, documents & applications. But there are only a few such documents or applications which can act like a personal doctor or a healthcare assistant. The existing applications have a lot of limitations too. One of these apps is “Health Mug” which is based on AI analysis of a patient’s symptoms which suggests doctors. But this app cannot suggest any physical activities to the patient. Another app is “Life Plus Bangladesh” which is an online doctor appointment booking app and a medicine shop giving telemedicine & ambulance services. There is no app on the market that can talk with the patient to analyze the symptoms and give predictions on the patient’s health problems along with giving valuable suggestions. There is no existing system that takes real-time diagnostic values for analysis purposes. So, it is found from the reviews that users need a fully functional medical one-stop solution that can do all these jobs easily.

Some important articles:

|  |  |  |  |
| --- | --- | --- | --- |
| **Reference** | **Objectives** | **Outcomes** | **Methodology** |
| [10.1] | To make a system that can preserve patients’ previous medical records with a view to helping doctors to get an overview of the patient. | An android  application that can store the medical record history of a user so that it can be analyzed in the future. | * Patient and doctor both will register with personal details. * A unique id will be generated for a patient. * This id will be used in the future by doctors to study the medical history of the patient within the app. |
| [10.2] | To describe an AI-based  chat-bot  application that delivers | A chat-bot system that can interact with pregnant women and mothers to | * User message is processed through two modules. * Intent classification module checks the user input message and identifies the purpose. |

|  |  |  |  |
| --- | --- | --- | --- |
|  | instructions and support to  women and mothers. | provide help and support. | * Entity recognition module recognizes the user message structure and extracts main keywords. * Response generator module provides a useful response. |
| [10.3] | To make a healthcare  chat-bot using AI that will be able to detect illness and provide information on that basis. | A chat-bot that  can detect illness and provide information  regarding the detected illness. | * Chat-bot processes input data using algorithms. * It understands the input using the previous database of symptoms and algorithms. * Suggests necessary measures and doctors’ details near the user. |
| [10.4] | To build a chatbot that  communicates  with the user and can give treatments for various common diseases based on the symptoms using the  appropriate  algorithm and Recurrent Neural Network  algorithm. | Brings healthcare available to  everyone and endure early detection of diseases at home. | * The voice from the user is recorded using a microphone and the voice is converted to text using Google API. * The unstructured output of Google API will be taken as input and info is extracted using the Bag of Words method. * Two types of response systems are developed. * General Response is created by training the chatbot using the Seq2Seq model. * Medical Assistant: Apriori algorithm is used to predict the disease. * Google text to Speech API is used to convert the response text to speech. |
| [10.5] | To create an alternative to traditional medical  checkups in hospitals with a medical chat-bot | A platform  where the users can interact and track their health status and get  encouraged to remain healthy. | * At first while registering, the user must provide some personal details which will be kept confidential. * The admin can view all the user details and monitor users. * Users can communicate and the |

|  |  |  |  |
| --- | --- | --- | --- |
|  | companion |  | words are identified to detect symptoms using NLP.   * The chat-bot is trained with the symptoms dataset and can identify the disease using the KNN algorithm. * Finally, the chat-bot recommends the required treatment for the identified disease. |

## Interview:

Two interviews regarding healthcare availability were conducted. From conductions of these interviews, it is obvious that healthcare is still not easily accessible for all people. It's hard to find relevant doctor’s contact info for a particular illness most of the time.

Moreover, it is even harder to figure out which disease a person is suffering from without going for a formal checkup. Many people don’t want to go for a formal checkup as they see it as a waste of energy. Instead, they ask their friends/family for suggestions which is quite risky as healthcare is a sensitive issue.

## Survey:

Google form was used to interact directly with the targeted users to know their requirements. The results of this survey helped to detect and discover requirements. The questions can be found from the appendix section.

From the results of the survey, it is noticed that some healthcare-based problems of the participants can be solved by our system. These important points are given below

* + - About half of the participants think healthcare services are not always available to them and about 25% think that healthcare services are never available to them any time they want. So, an AI-based healthcare and consultancy app is helpful for our participants.
    - About 75% of the participants search on the internet for finding the solution to their medical problems. 60% of people think the results from the internet are somehow (50%) acceptable. So, an ai based system can be a solution for them to find their queries.
    - Only 1% of the participants check their temperature, heart rate, or oxygen level regularly whereas above 90% of participants agree that it is important to track these records continuously. This proposed system will be handy for keeping track of these records easily anywhere.
    - In case of emergency, about 40% of people cannot remain calm, and they do not have any doctor’s or ambulance’s contact number saved. This requirement can be fulfilled by the proposed system’s SOS feature.
    - About 40% of our participants are familiar with fitness or medical apps. So, it will not be harder for people to get used to our system. Most of the available systems don’t fulfill all the requirements of participants. This proposed app can solve this problem for them by being a one-stop solution.

# User Requirements

* 1. The system shall understand human language and various health metrics.
  2. The system shall detect illness/disease.
  3. The system shall suggest relevant general instructions for the cure.
  4. The system shall suggest relevant doctors.
  5. The system shall have an emergency ambulance/contact system.

# System Architecture

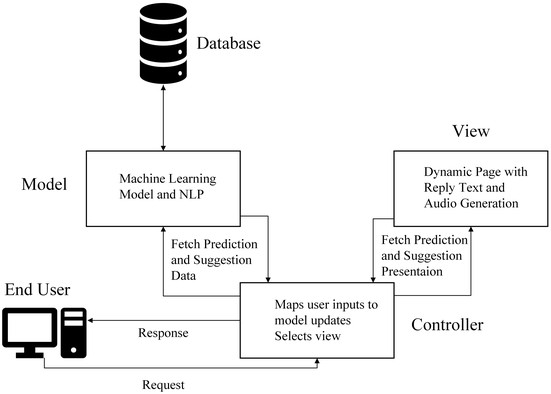


Figure 1: System Architecture

**Description:**

An Android application will be used as a user interface and an integrated handheld device will be used to take various health metrics.

There will be an identity management system to identify a particular user. Also, a hardware and an application management system will be used to coordinate the handheld device and the Android application.

The main application functionalities will be illness prediction, doctor suggestion, cure suggestion, and emergency message system.

The app will run on a cloud service and a database will be maintained.

# System Requirement Specification

## System Requirements:

* + 1. **Understanding human language and health metrics**
       1. The system shall take human text as input via mobile phone to interpret the meaning and understand the symptoms.
       2. The system shall use a finger oximeter heart rate module to measure oxygen level along with heart rate and interpret the significance of the received value.
       3. The system shall use a temperature sensor module for Arduino to measure temperature and interpret the significance of the received value.
    2. **Detecting sickness/illness**
       1. The system shall process all the provided information using machine learning and the database.
       2. The system shall give a prediction about the illness/disease type with 99% accuracy.
       3. The system shall give predictions via text..
    3. **Suggesting Cure**
       1. The system shall suggest various general instructions for the cure based on the predicted disease/illness by processing the database and machine learning model.
       2. Instructions shall be given via text.
    4. **Suggesting Doctors**
       1. The system shall have a database of available doctors’ contact information.
       2. Based on doctors’ expertise, the system shall suggest various doctors with contact information.
    5. **Emergency Contact**
       1. The system shall have an emergency button.
       2. If the button is pressed, an emergency message is sent to the nearest ambulance.
       3. Also, an emergency help message will be sent to predetermined close contacts.

## Requirements Classification

|  |  |  |
| --- | --- | --- |
| **System Requirements** | **Functional** | **Non-Functional** |
| Text input | ✔ | **X** |
| Real-time data collection | ✔ | **X** |
| Accuracy of interpreting the  significance of the received data | **X** | ✔ |
| Processing data using ML | ✔ | **X** |
| Prediction Accuracy | **X** | ✔ |
| Text Output | ✔ | **X** |
| Suggesting Cure | ✔ | **X** |
| Suggesting Doctors | ✔ | **X** |
| Emergency message | ✔ | **X** |
| Emergency ambulance service | ✔ | **X** |

# System Model

## Context Diagram

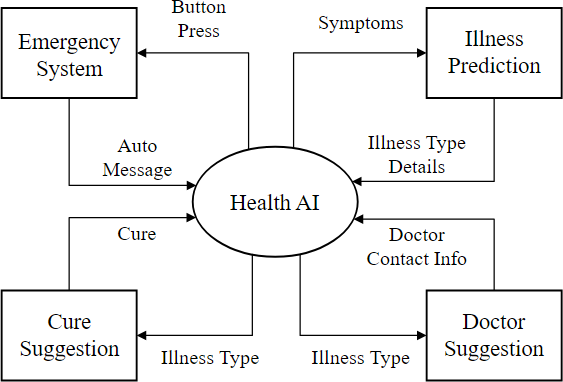


Figure 2: Context Diagram

**Description:**

There are 4 different subsystems in the context diagram.

1. Illness prediction module will take symptoms as input and provide information about illness type.
2. Cure suggestion module will take illness type as input and provide information about the cure.
3. Doctor suggestion module will take illness type as input and provide contact information of relevant doctors.
4. Emergency system module will be triggered by an SOS button press and it will send automated messages/signals to predetermined contacts and the nearest ambulance service.

## Activity Diagram

* + 1. **Activity Diagram for Disease Detection**

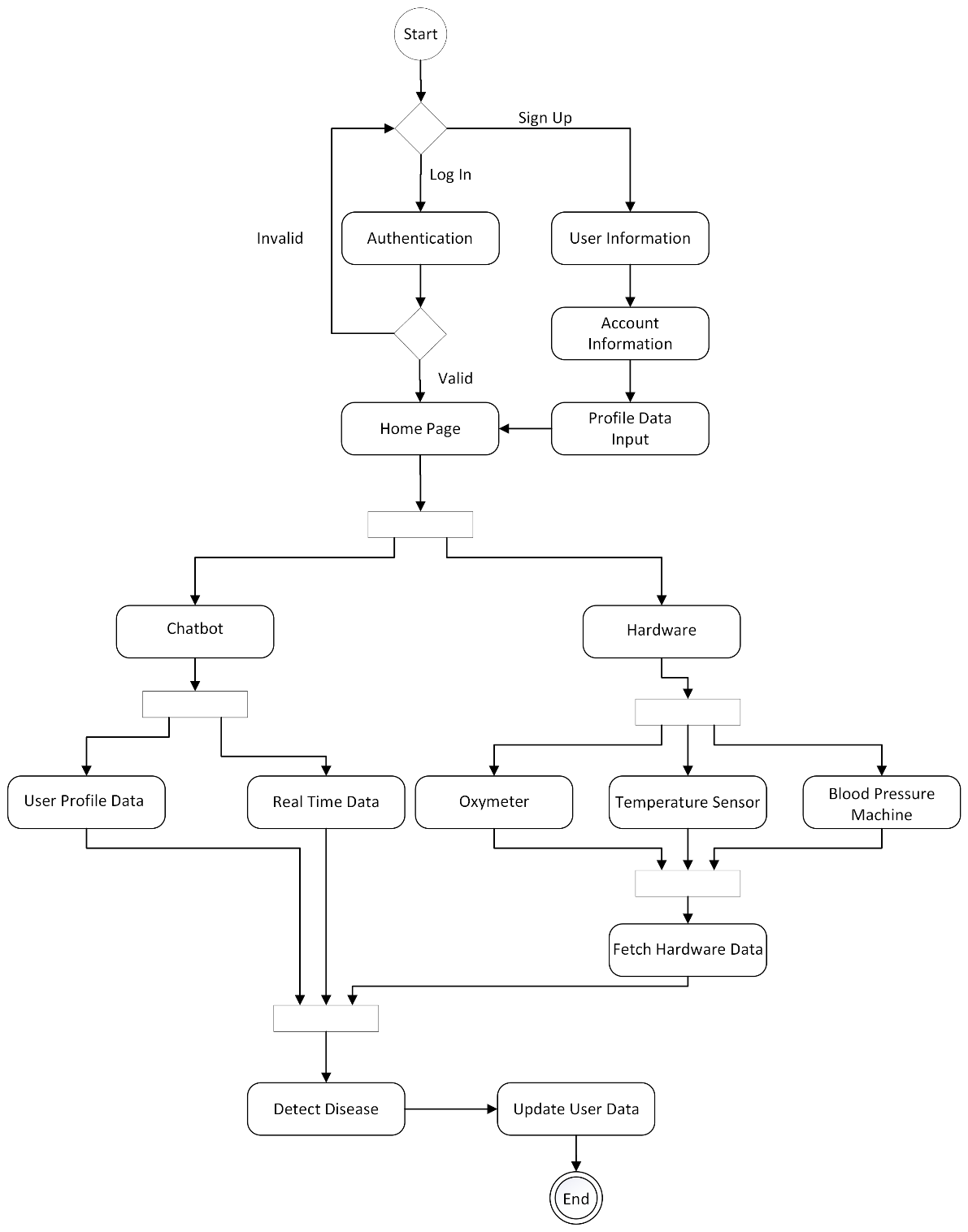


Figure 3: Activity Diagram for Disease Detection

Temperature Sensor

**Description:**

In the disease detection module real-time data is collected from user chat combined with user profile and sensor data to identify the disease. User profile is also updated with the latest data to keep the user profile up to date.

* + 1. **Activity Diagram for Taken Actions**

Temperature Sensor

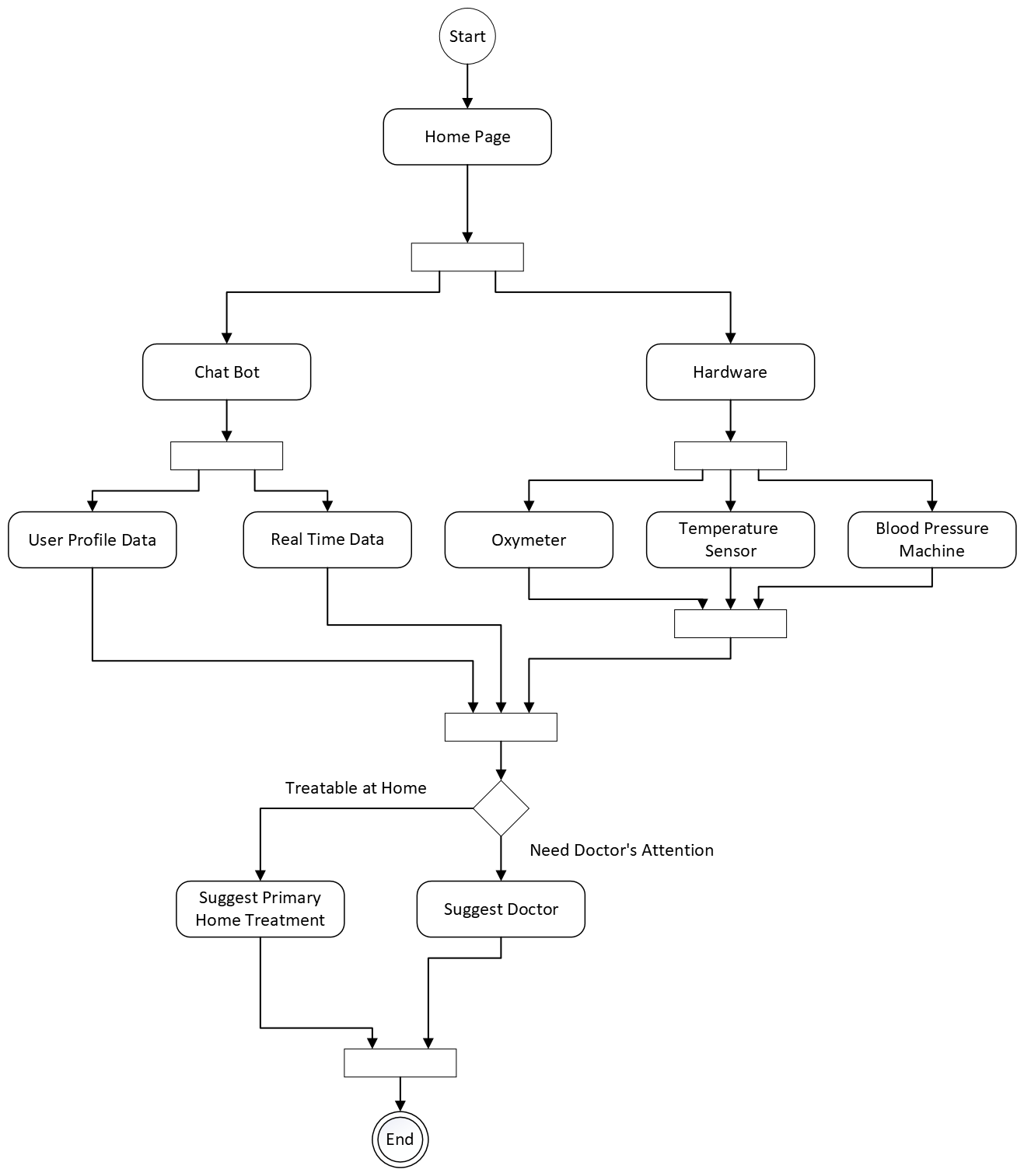


Figure 4: Activity Diagram for Taken Action

**Description:**

From user profile data, user chat and hardware data the machine learning algorithm will judge the user's condition. If the condition is treatable at home then primary home treatment will be suggested. Otherwise, the system will suggest nearby doctors.

* + 1. **Activity Diagram for Emergency System**

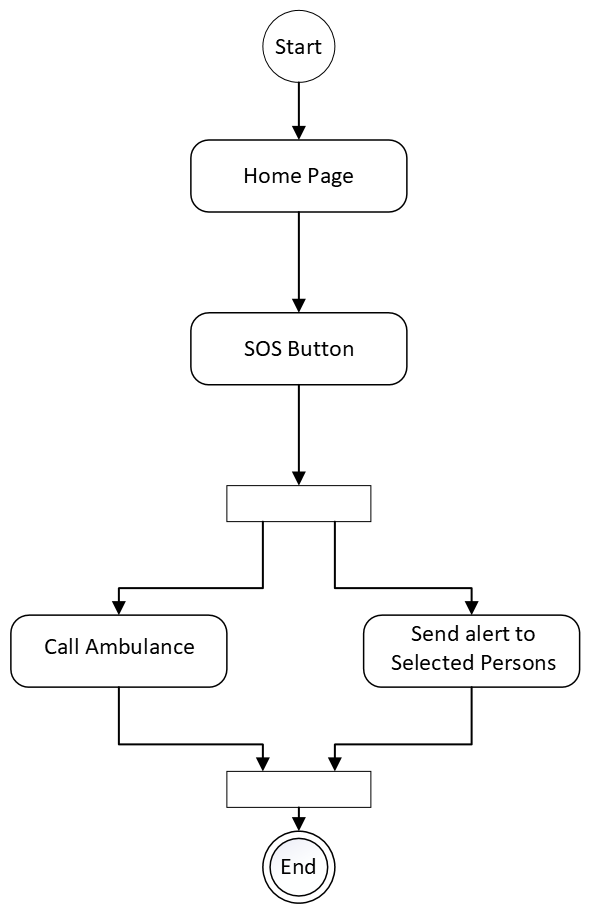


Figure 5: Activity Diagram for Emergency System

**Description:**

This activity diagram shows the work of the SOS button. Users will use this in case of

emergency. If the button is pressed, previously selected persons by the users will be notified and an ambulance will be called if wanted.

## Use Case Diagram

* + 1. **Illness prediction**

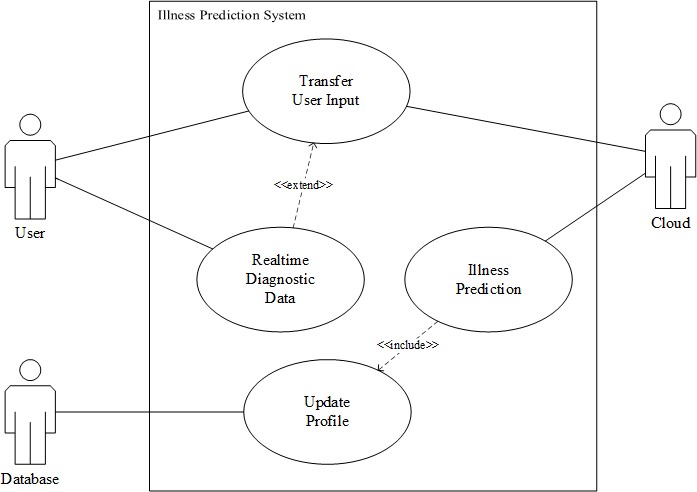


Figure 6: Use case diagram for Illness Prediction System

Tabular Description:

|  |  |
| --- | --- |
| Transfer user data | |
| Actors | User, Cloud. |
| Description | A registered user gives his or her problems as input which may be in text format. On the other hand, a user can measure his or her heart rate, oxygen level and give them as input also. This real time diagnostic data input is optional. Then all these user inputs are transferred to the cloud.  Predicated illness is then transferred to the user. |

|  |  |
| --- | --- |
| Data | Users’ problems, users’ real time diagnostic data. |
| Stimulus | Chatbot and diagnostic hardware enabled by user |
| Response | Uploads users’ problems and diagnostic data to cloud for analysis and  returns the predicted illness to the user. |
| Comments | Users must be registered. Real time data is not compulsory. |
|  |  |
| Real time diagnostic data | |
| Actors | User. |
| Description | User wears a handheld device and measures his or her heart rate,  temperature, oxygen level which is transferred to the app then to the cloud. |
| Data | Users’ real time diagnostic data which are heart rate, and oxygen level. |
| Stimulus | User activates a handheld diagnostic device. |
| Response | Sends the diagnostic data to the mobile application. |
| Comments | Users must wear the handheld device and fully finish the measurements. |

|  |  |
| --- | --- |
| Illness Prediction | |
| Actors | Cloud |
| Description | With the data which are uploaded into the cloud by the user all the  analysis is done, and the predicted disease or problem of the user is then figured out. Also, the current states and illness of the user is saved in the database |
| Data | Users inputted data, predicted illness. |
| Stimulus | User uploads his or her problems and diagnostic data |
| Response | Predicts the users’ problem |
| Comments | User must upload valid problems |

|  |  |
| --- | --- |
| Update Profile | |
| Actors | Database |
| Description | User’s problems, diagnostic data and predicted illness is stored in the  database for future usage. |
| Data | User’s problems, diagnostic data, and predicted illness. |
| Stimulus | Illness prediction predicts the problem successfully. |
| Response | All the data are saved in the users’ profile database |
| Comments | User must give permission to store his or her personal data. |

* + 1. **Cure Suggestions**

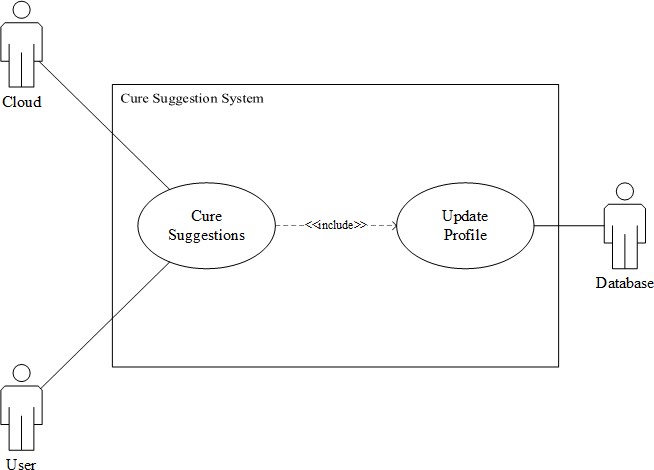


Figure 7: Use case diagram for Cure Suggestions System

Tabular Description:

|  |  |
| --- | --- |
| Cure Suggestions | |
| Actors | User, Cloud. |
| Description | From the cloud the predicted illness is taken, and cure of the  corresponding problem is given if the problem can be solved at home. |
| Data | Predicted Illness. |
| Stimulus | Illness prediction system gives predicated illness. |
| Response | Home remedies for the illness is given to the user |
| Comments | Users may follow the cure suggestion if it has no side effects for him or  her. |

|  |  |
| --- | --- |
| Update profile | |
| Actors | Database. |

|  |  |
| --- | --- |
| Description | Cures suggestions are saved in the database for future usage. |
| Data | Cure suggestions. |
| Stimulus | Cure suggestions gives any kind of suggestions. |
| Response | Saves data into the user’s profile database. |
| Comments | Users must give permission to store his or her personal data. |

* + 1. **Doctor Suggestions**

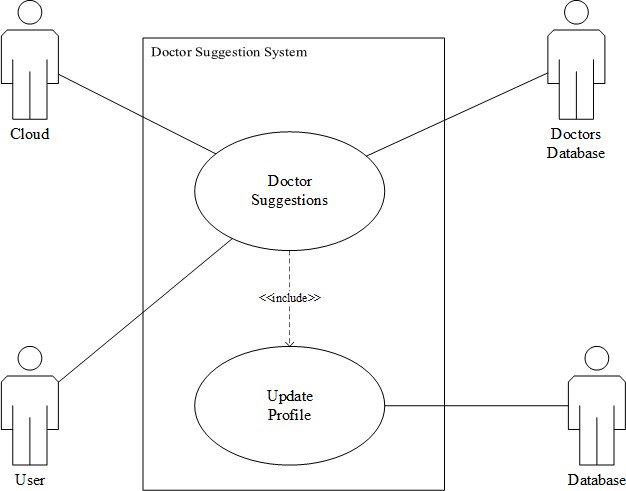


Figure 8: Use case diagram for Doctor Suggestions System

Tabular Description:

|  |  |
| --- | --- |
| Doctor Suggestions | |
| Actors | User, Cloud, Doctors database |
| Description | From the cloud predicted illness is taken and the specialist doctor on  that problem is searched from the doctor’s database. Nearest specialist doctor is then suggested to the user. |

|  |  |
| --- | --- |
| Data | Predicted illness, doctor’s database. |
| Stimulus | Illness is predicted which needs specialist doctors to resolve. |
| Response | Gives a list of nearest specialist doctors on the predicted illness. |
| Comments | User must give permission to access his or her location. |

|  |  |
| --- | --- |
| Update profile | |
| Actors | Database |
| Description | Doctor suggestion data is saved in the database for future usage. |
| Data | Doctor suggestion list. |
| Stimulus | Doctor suggestions give any kind of suggestions. |
| Response | Saves data into the user’s profile database. |
| Comments | Users must give permission to store his or her personal data. |

* + 1. **SOS/Emergency System**

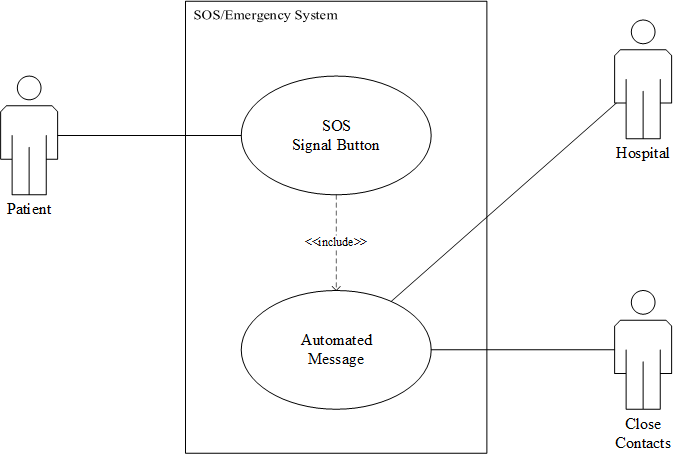


Figure 9: Use case diagram for SOS/Emergency System

Tabular Description:

|  |  |
| --- | --- |
| SOS Signal Button | |
| Actors | Patients. |
| Description | Patients may press the SOS button in case of emergencies which will  trigger an SOS automated message. |
| Data | User’s location and help wanted message text. |
| Stimulus | Button-press by user. |
| Response | Help wanted message. |
| Comments | Users must manually press the SOS button and give permission to use  his or her location. |

|  |  |
| --- | --- |
| Automated Messages | |
| Actors | Hospital and close contacts. |
| Description | Patients may press the SOS button which will lead to send help wanted  messages to hospitals and pre saved close contacts. |
| Data | User’s location and help wanted message text. |
| Stimulus | SOS button trigger. |
| Response | SMS to contacts and hospitals. |
| Comments | If no contacts are saved, then the message will only be sent to the  closest hospital. |

## Sequence Diagram

* + 1. **Login**

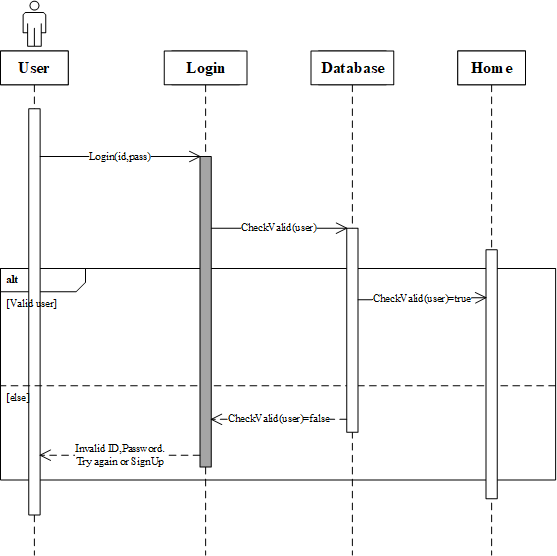


Figure 10: Sequence diagram for Login

* + 1. **Illness prediction**

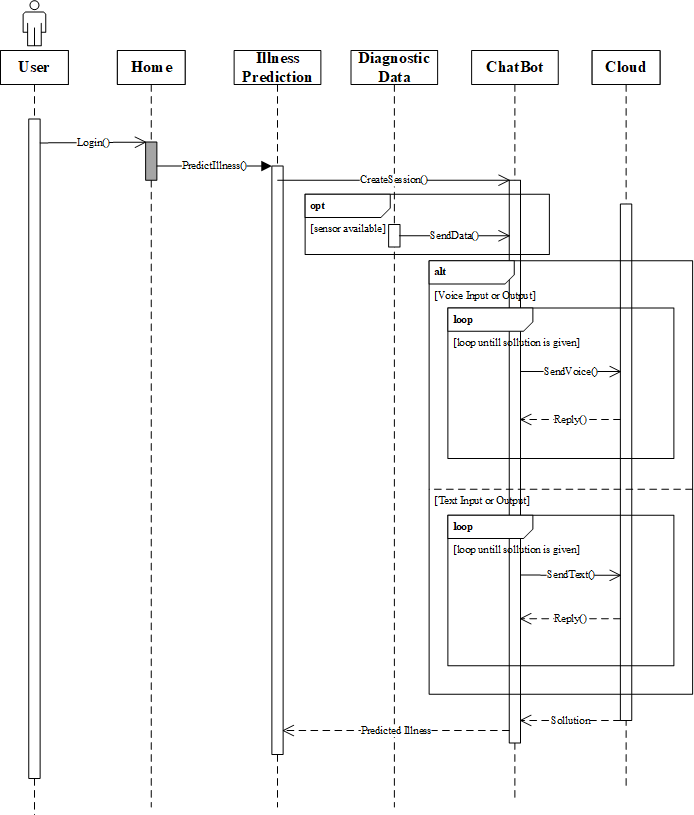


Figure 11: Sequence diagram for Illness Prediction System

* + 1. **Cure Suggestions**

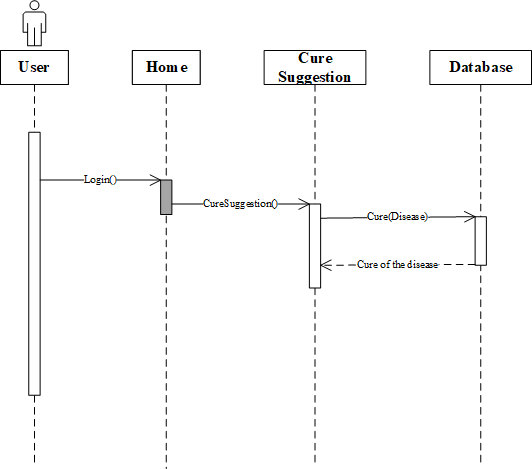


Figure 12: Sequence diagram for Cure Suggestions System

* + 1. **Doctor Suggestions**

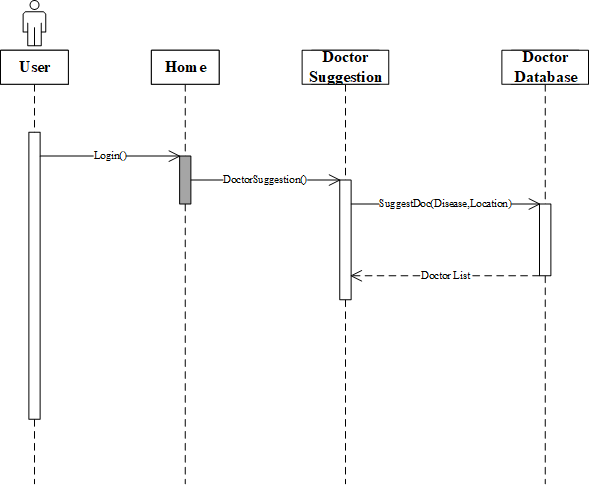


Figure 13: Sequence diagram for Doctor Suggestions System

* + 1. **SOS/Emergency System**

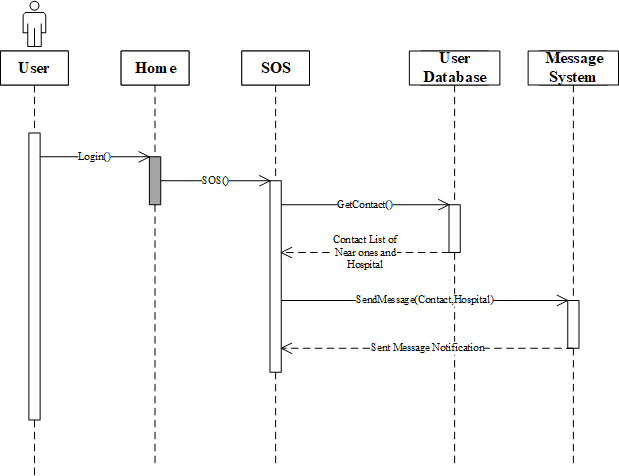


Figure 14: Sequence diagram for SOS/Emergency System

## State Diagram

* + 1. **Doctor Suggestions**

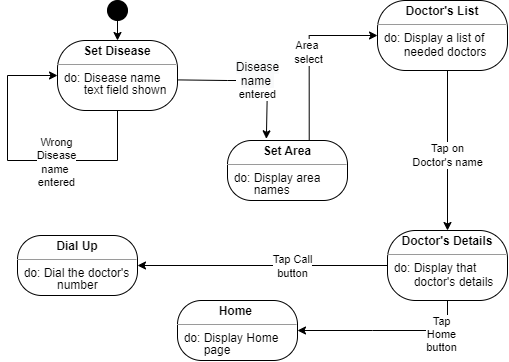


Figure 15: State diagram for Doctor Suggestions System

|  |  |
| --- | --- |
| **States** | **Description** |
| Set Disease | A text field where to enter the disease name. |
| Set area | A drop-down field to take the area name. |
| Doctor’s list | Display a list of eligible Doctors of that customer. |
| Dial Up | Display dial-up screen with the doctor’s contact number. |
| Home | Display system’s home page. |

|  |  |
| --- | --- |
| **Stimulus** | **Description** |
| Disease name entered | Entering disease names into the text field. |

|  |  |
| --- | --- |
| Area select | Select an area from the dropdown bar. |
| Tap on Doctor's name | Tap any doctor’s name. |
| Tap call button | Tap on the call/phone sign button. |
| Tap Home button | Tap on the Home button. |

* + 1. **Illness Prediction**

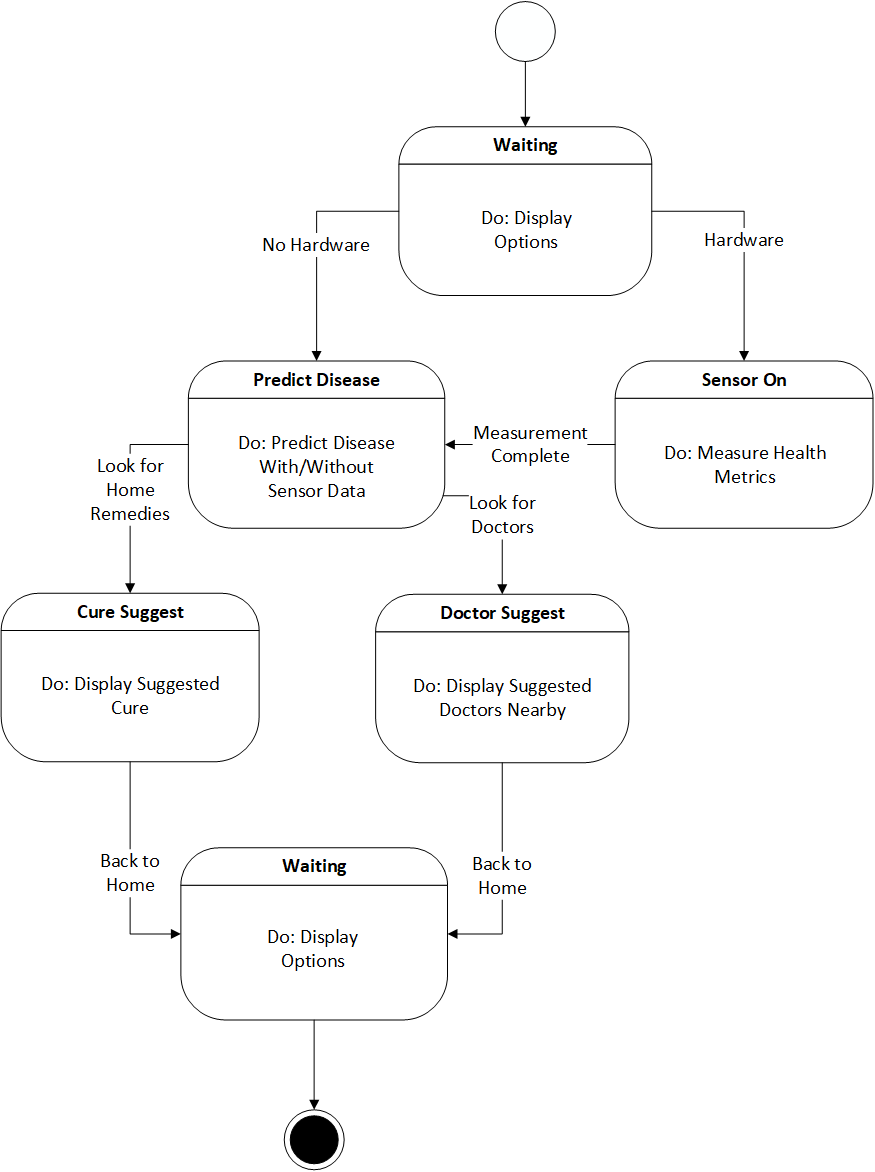


Figure 16: State diagram for Illness Prediction System

|  |  |
| --- | --- |
| **State** | **Description** |
| Waiting | The system is waiting for the user to select an option from the modules. |

|  |  |
| --- | --- |
| Sensor On | Sensors are turned on and health metrics are collected. |
| Predict Disease | Disease is predicted using user profile data and sensor data if there is any. |
| Cure Suggest | Home remedies are suggested based on the detected disease. |
| Doctor Suggest | User location is taken, and nearby specialist doctors are suggested for the detected disease. |

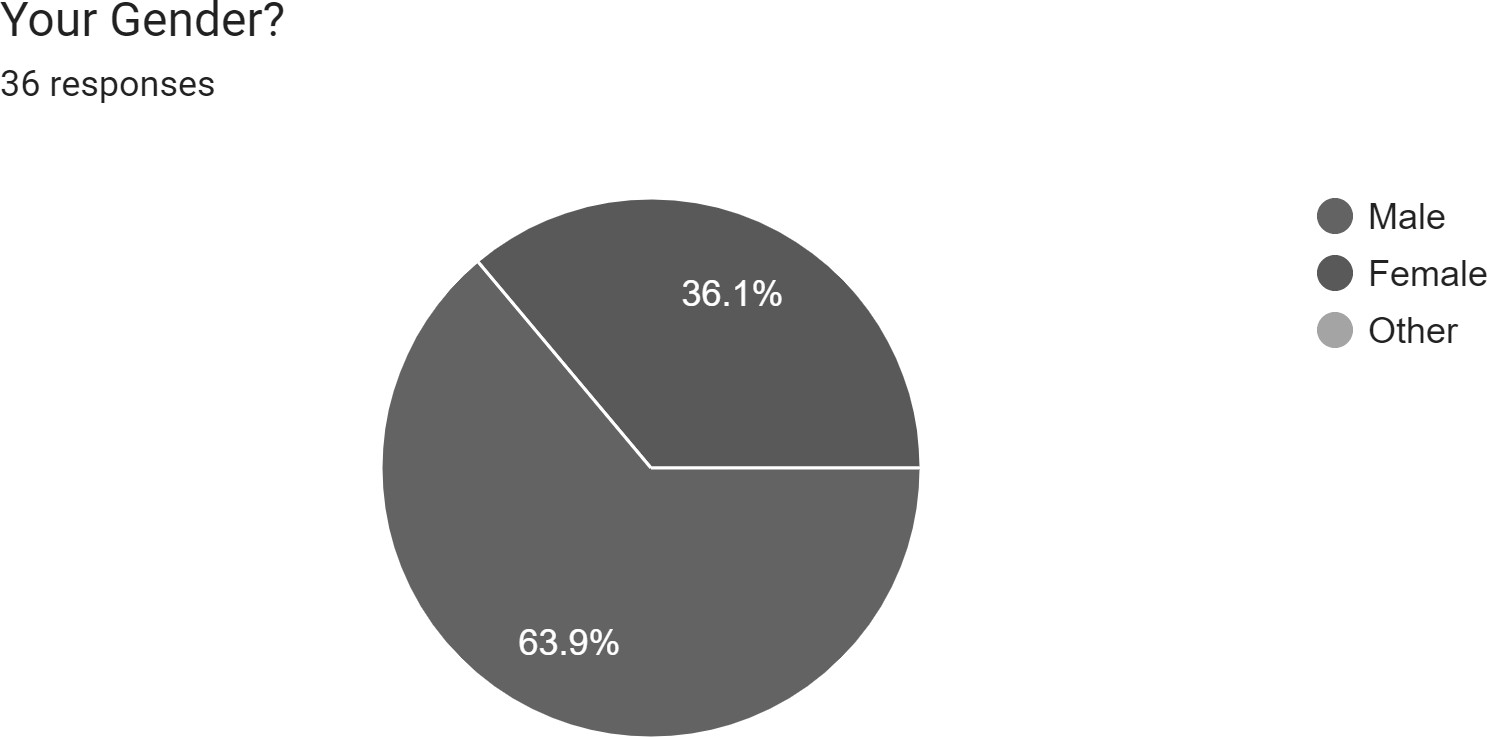
|  |  |
| --- | --- |
| **Stimulus** | **Description** |
| Hardware | User has connected Hardware |
| No Hardware | User doesn’t have connected hardware |
| Measurement  Complete | Measurement of health metrics complete |
| Look For Home  Remedies | User has selected the cure suggestion module |
| Look for Doctors | User has selected the doctor suggestion module |
| Back to Home | User has selected the Home option |

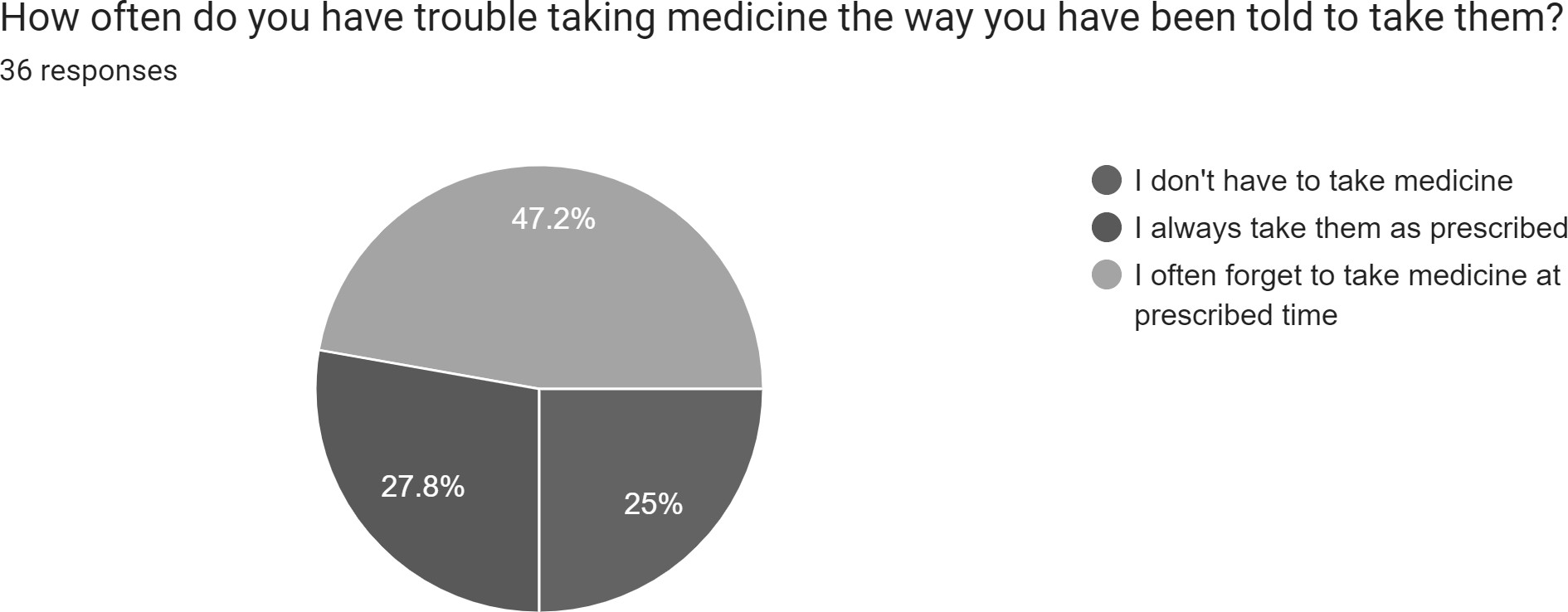
# Appendix

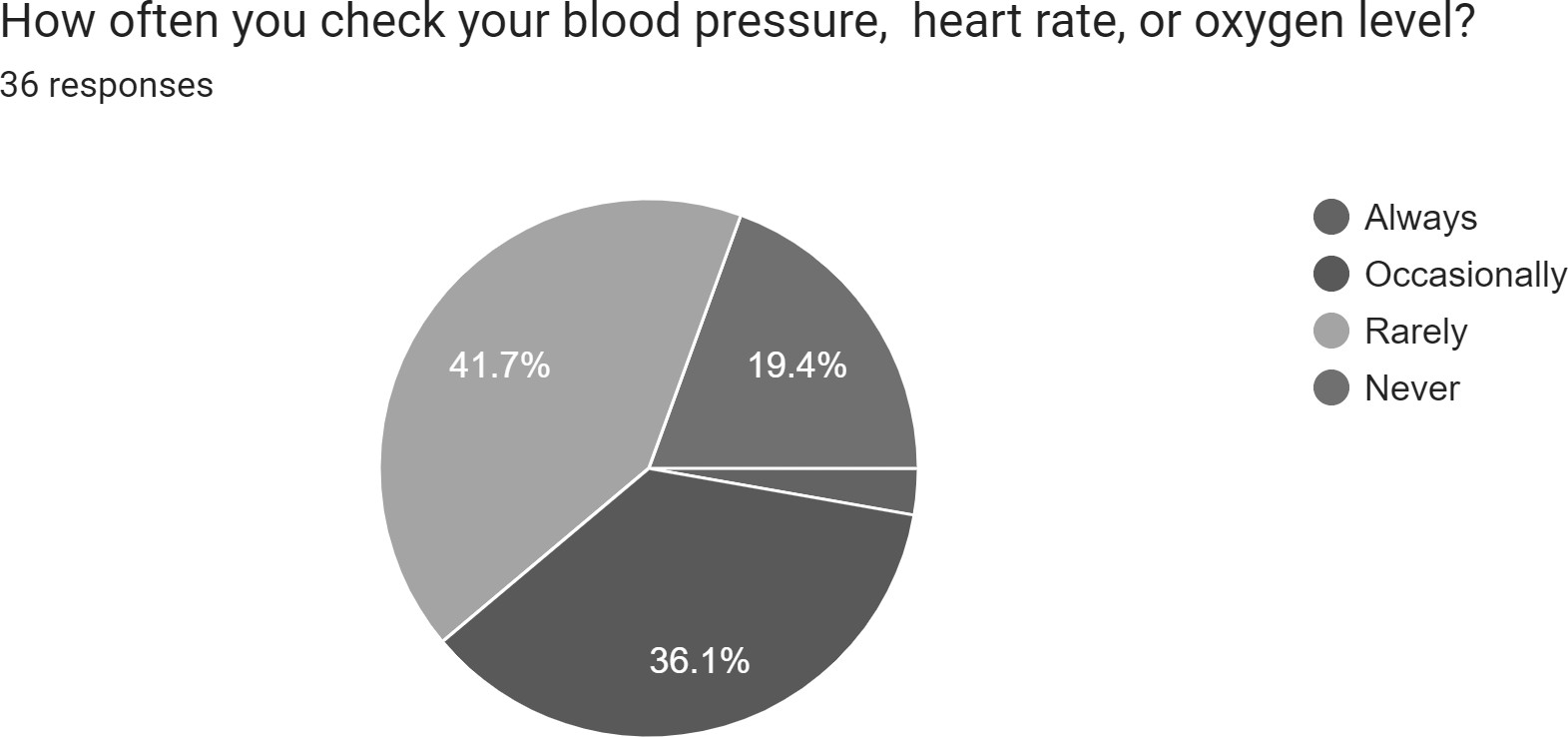
## Questionnaires

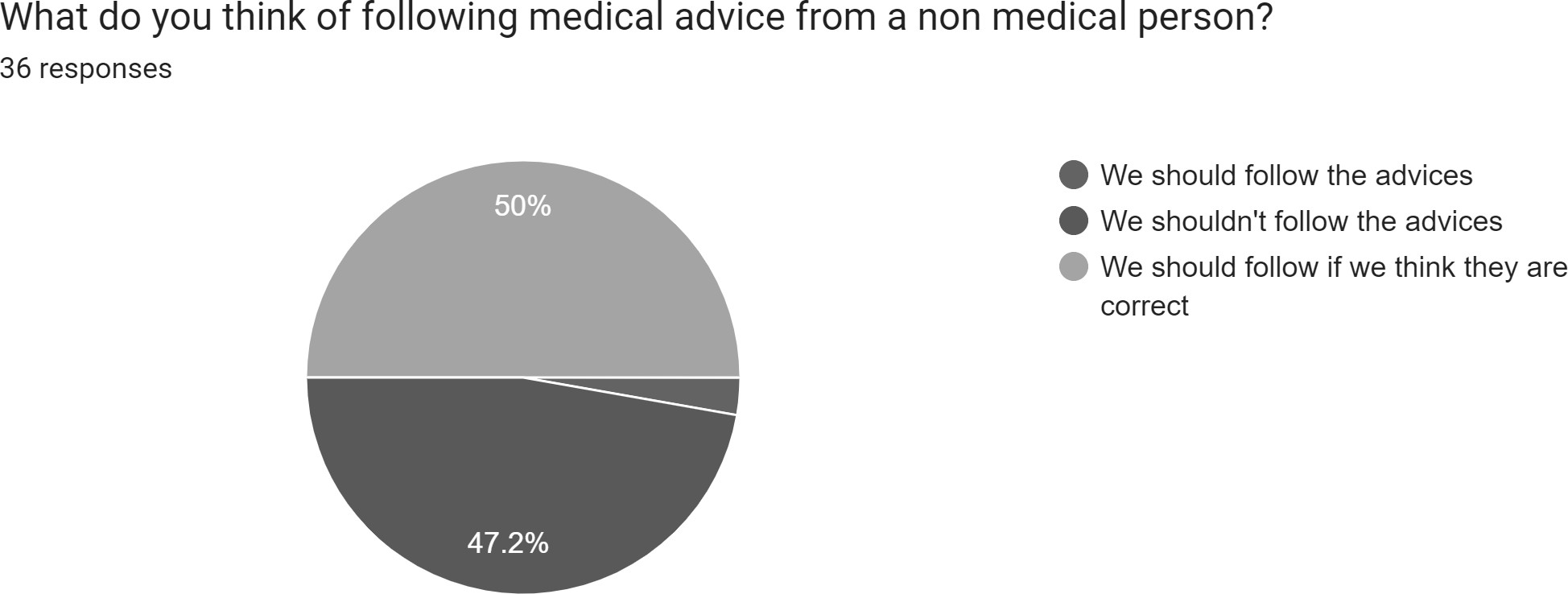
* + - Your Gender?
    - Your Age?
    - Do you have any chronic disease?
    - How often do you visit a doctor?
    - How clearly can you express your medical problems to your doctors or to your family?
* Do you have problems talking fluently with your doctor in person?
* Do you think healthcare services are always available to you any time you want?
* How often do you take health advice from your relatives?
* How often do you search on the internet about medical problems of yours?
* On a scale of 5 how much do you think that the results on the internet about your medical query are accurate?
* How often do you check your temperature, heart rate, or oxygen level?
* Do you measure these medical records (bp, heart rate, oxygen level) at home or go to the nearest pharmacy (drug shop)?
* Do you think it is necessary to keep a track of these (bp, heart rate, oxygen level) medical records?
* What do you think of following medical advice from a non-medical person?
* Do you think buying medicine without prescription should be allowed?
* Do you take medicine without a doctor's suggestion?
* How often do you have trouble taking medicine the way you have been told to take them?
* Are you comfortable discussing your medical problems with your doctor on text, voice, or video call?
* In case of an emergency do you prefer texting or talking to someone?
* Do you keep your doctor's contact number on your phone?
* On a scale of 5 how accurately can you identify which specialist doctor you need?
* How do you judge your knowledge about the doctors and services of the nearest hospitals around you?
* Have you ever used Google assistant, Siri or Alexa for searching medical queries?
* Are you comfortable to share your location with your close ones in case of any medical emergency?
* Do you have any medical apps on your phone? (If yes then can your write name of any)
* Do you use any healthcare or fitness tracking device or app? (If yes write the names of the device/app)
* In case of a medical emergency, how do you judge your ability to remain calm?
* What is the main problem you face in case of a medical emergency?

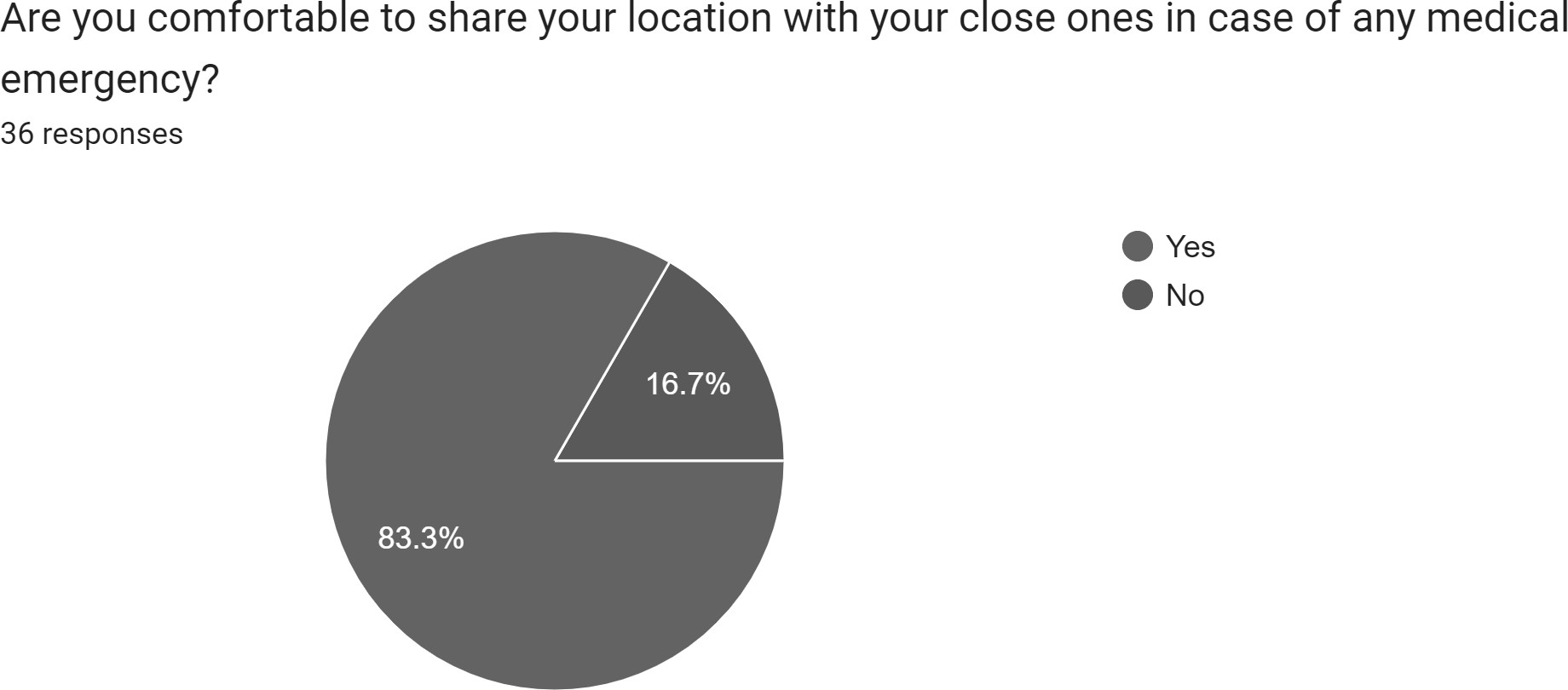
## Some Responses











# References

**[10.1]** Shilani, S., Singh, S. and Gupta, T., 2019. A Review Paper on Android based Personal Healthcare Companion. System, 6(04).

**[10.2]** Vaira, L., Bochicchio, M.A., Conte, M., Casaluci, F.M. and Melpignano, A., 2018, June. MamaBot: a System based on ML and NLP for supporting Women and Families during Pregnancy. In Proceedings of the 22nd International Database Engineering & Applications Symposium (pp. 273-277).

**[10.3]** Kurup, G. and Shetty, S.D., 2022. AI Conversational Chatbot for Primary Healthcare Diagnosis Using Natural Language Processing and Deep Learning. In Computational Intelligence in Pattern Recognition (pp. 259-272). Springer, Singapore.

**[10.4]** KC, G.P., Ranjan, S., Ankit, T. and Kumar, V., 2019. A personalized medical assistant chatbot: Medibot. Int. J. Sci. Technol. Eng, 5(7).

**[10.5]** Mathew, R.B., Varghese, S., Joy, S.E. and Alex, S.S., 2019, April. Chatbot for disease prediction and treatment recommendation using machine learning. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 851-856). IEEE.