Scalable Human-Machine Interaction System for Real-Time Care in the Internet of Health Things

by

Mahmoud Nasr Mohamed

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Masters of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2020

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The rise in numbers of individuals with weak immunity around the world and the aging of populations put an ever-growing pressure on healthcare and inevitably increases its cost. This phenomenon leads to larger portions of the population to which quality healthcare is not provided. To fight this trend, technological advancements in the Internet of Health Things aim to integrate smart sensors and devices to continuously monitor and assess the status of patients and older adults from the comfort of their own home at a fraction of the cost. Although solving specific problems each at a time advances the field and takes us a step closer to autonomous home care systems, the solution to these issues needs to consider the much larger picture to unify the approaches and cultivate benefits of many intelligent, but stand-alone, systems. The current work aims to explore the field of Internet of Health Things and its application to remote health monitoring and ambient assisted living for older adults. Picking up from where previous literature left off, this thesis proposes a multi-layered framework that provides a comprehensive solution to continuous healthcare. In particular, the framework was created with modularity, scalability, and expandability as the main priorities; to offer an all-purpose remedy to the problems in hand. To this end, the internal mechanisms of the framework are described in detail and the system is applied to remote health monitoring and ambient assisted living environments by interchanging its components. The implementations presented in this thesis expose the capability of the framework to harvest power of existing intelligent devices. Moreover, the two systems implemented consider multi-modal and natural human-machine interaction techniques that provide the user with the choice of their preferred interaction method. The main advantage of the proposed framework is that it offers an all-in-one solution to providing continuous healthcare without sacrificing the quality of care provided. On the contrary, the solution in this work allows deeper understanding of user's health, personalization, real-time analytics and recommendations, as well as aid for activities of daily living with state of the art technologies.

Acknowledgements

I would like to thank all the people who made this journey possible and successful. First, I would like to thank my supervisor, Prof. Fakhri Karray, for his continuous support and guidance throughout the duration of my studies. Also, I would like to thank Prof. Shady Shehata and Prof. Kumaraswamy Ponnambalam for taking the time to read my thesis and provide insightful comments to its improvement.

I would like to thank my roommates for their help, support, the great times we spent together, and for always making it feel like home. Special thanks go to my parents who supported me, though thousands of kilometers away, and are the main reason I was capable of reaching this point in my life and all my previous (and future) successful endeavours. Last but not least, I would like to thank my wife for her unconditional love and support every day. It was not easy, but was definitely worth it.

Thank you.

Table of Contents

List of Figures **viii**

List of Tables **x**

Nomenclature **xi**

1 Introduction **1**

1.1 Background 1

1.2 Problem Statement 2

1.3 Thesis Contribution 3

1.4 Thesis Outline 4

2 Literature Review **5**

2.1 Remote Health Monitoring 5

2.2 IoHT for Older Adults 9

2.2.1 Ambient Assisted Living 10

2.2.2 Social Robotics 13

3 Proposed Framework **20**

3.1 System Architecture 20

3.1.1 Multi-Layer Architecture 20

3.1.2 Communication Protocols 22

3.2 Fog Layer Modules . . . 24
3.2.1 Request Handler . . . 24
3.2.2 Data Handler . . . 25
3.2.3 Database Handler . . . 25
3.2.4 Agent Link . . . 25
3.2.5 Analytics Engine . . . 26
3.3 Summary . . . 27

4 WeCare System for Health Monitoring and Disease Prediction 28
4.1 System Architecture . . . 28
4.1.1 User Interface . . . 29
4.1.2 Health Assistant Bot . . . 29
4.1.3 Cloud Database . . . 31
4.2 Implementation and Results . . . 33
4.2.1 Smartphone Application . . . 37
4.3 Disease Prediction . . . 40
4.3.1 Rule-based Dataset Generation . . . 43
4.3.2 Machine Learning for COVID-19 Prediction . . . 46
4.4 Summary . . . 49

5 Ambient Assisted Living for Older Adults 51
5.1 System Architecture . . . 52
5.1.1 Central Brain . . . 52
5.1.2 Speech Interaction . . . 53
5.1.3 Autonomous Robotics for Task Achievement . . . 54
5.1.4 Smart Wearable . . . 55
5.2 Implementation and Results . . . 56
5.2.1 Static Commands . . . 56
5.2.2 Dynamic Commands . . . 59
5.3 Summary . . . 60

6 Conclusions and Future Work **61**
6.1 Conclusion 61
6.2 Future Work 62

References **64**

List of Figures

2.1 Commercial Smart Wearables for Remote Health Monitoring 7
2.2 Social Robots for Older Adults 13
2.3 Social Robot Implementations (1) 15
2.4 Social Robot Implementations (2) 17

3.1 System Design with 3 IoHT Layers 22

4.1 WeCare System Software Architecture 29
4.2 Rasa Pipeline 30
4.3 Structure of Data in DynamoDB from a Sample Patient Record (ID: p0307) 32
4.4 WeCare Log In Screen 33
4.5 WeCare System Tabs with Static Header 34
4.6 WeCare Web Interface Pages and Functionalities 34
4.7 WeCare System Dashboard Page 35
4.8 WeCare System Reports Page 36
4.9 WeCare System Categorical Parameters Visualization 37
4.10 Frequency Filters for Visualization in WeCare System 38
4.11 Category Filter of Health Vital Visualization in WeCare System 39
4.12 WeCare Insert Data Form 40
4.13 WeCare System Chat bot Assistant 41
4.14 WeCare Smartphone Application Functionalities 42

4.15 Different Access for Doctors and Patients in WeCare Smartphone Application 42
4.16 Additional WeCare Smartphone Application Functionalities 43
4.17 Temperature, Dry Cough, Fatigue, and Covid-19 Membership Functions . . 44
4.18 Histogram of created dataset for COVID-19 46
4.19 Instant Analytics for COVID-19 on WeCare Web Interface 49
4.20 Three-day COVID-19 Risk Analytics on WeCare Dashboard 49

5.1 Elderly Care System 52
5.2 Internal Modules in Brain of the Elderly Care System 53
5.3 Requesting Missing Entities in User Request 59
5.4 Speech Interaction for Robot Navigation 59

List of Tables

4.1 Physiological Parameters Monitored in WeCare 31
4.2 Confusion Matrix for kNN and Decision Tree Algorithms 48

5.1 Sample Training Data for Conversational Chat Bot 57
5.2 Complicated Interaction with Multiple Action Requests 58
5.3 Flexible Interaction Without Preset Commands 60

Nomenclature

SpO_2	Blood oxygen saturation
AAL	Ambient assisted living
API	Application programming interface
ASR	Automatic speech recognition
BP	Blood pressure
CGM	Continuous glucose monitoring
COTS	Commercial off-the-shelf
COVID	Coronavirus disease
DT	Decision Trees
ECG	Electrocardiogram
FN	False Negatives
FNR	False Negative Rate
FP	False Positives
FPR	False Positive Rate
GA	Google Assistant
GSR	Galvanic skin response
HMI	Human-machine interaction

HR	Heart rate
HRV	Heart rate variability
HTTP	Hypertext transfer protocol
IoHT	Internet of Health Things
IoT	Internet of Things
JSON	JavaScript Object Notation
kNN	k-Nearest Neighbors
M2M	Machine to machine
MQTT	Message Query Telemetry Transport
NLU	Natural language understanding
NoSQL	Non-sequential database
REDP	Recurrent EmbeddingDialogue Policy
REST	Representational State Transfer
RHM	Remote health monitoring
RHR	Resting heart rate
ROS	Robotic Operating System
RR	Respiration rate
SVM	Support Vector Machines
TN	True Negatives
TNR	True Negative Rate
TP	True Positives
TPR	True Positive Rate
UI	User interface
WHO	World Health Organization

Chapter 1

Introduction

1.1 Background

The continuous increase in world population has contributed to a rise in the portion of older adults throughout the world. According to the World Health Organization (WHO), the percentage of older adults is expected to increase from 12% in 2015 to 22% by 2050 [1]. Since aging is directly linked to suppressed immunity and older adults are known to be the most vulnerable to chronic illnesses, the number of people in need for healthcare is increasing drastically. Ultimately, the high demand for appropriate healthcare imply a rise in the cost of healthcare around the globe, a growing burden on healthcare facilities and healthcare professionals, and a pressing need for innovative solutions to mitigate these issues. With the advancement and miniaturization of sensors, communication technologies, and cloud-based storage and computing, the Internet of Things (IoT) has been gaining a lot of interest as a cost-efficient solution capable of revolutionizing the healthcare industry.

The Internet of Things is an ecosystem combining sensors, smart devices, computational units, and computer software within a network to collect, exchange, and analyze data. IoT sensors continuously collect and transmit real-time data about the environment, which is then stored in local and cloud databases for analysis and decision-making. For instance, a smart home environment with sensors to monitor room temperature and a heating, ventilation, and air conditioning (HVAC) system to regulate the temperature constitutes a simple IoT ecosystem. Similarly, sensors that read physiological data about the user such as heart rate, temperature, and blood pressure are combined in an ecosystem to monitor individuals' health. This ecosystem belongs to a branch of IoT called the Internet of Health Things (IoHT). These sensors that are able to obtain real-time vitals about users were first observed in wearables for fitness tracking such as the Fitbit[1], Apple Watch[2], and Garmin[3]. Such devices have enabled many users to track their activity and live a healthier lifestyle. Inspired by the idea of health tracking, many companies are opting to advance the current health industry by developing high-end sensors that are capable of tracking instantaneous physiological parameters. Accordingly, the health care industry progression engendered various applications that enable IoT in healthcare (for example VivaQuant[4] RX-1 and Dexcom G6[5]) and telemedicine, with a market size of $49.8 Billion in 2018, projected to reach $266.8 Billion by 2026 [2].

1.2 Problem Statement

Continuous health monitoring is required to lighten the load on the healthcare system without sacrificing care quality. This problem is most tangible with chronically ill patients and the community of older adults. Since the number of healthcare workers is decreasing and the costs of healthcare keep rising everyday, a solution that is independent of human presence is crucial. State of the art solutions explore the use of smart devices in an IoHT environment to provide continuous health monitoring. However, most of the proposed solutions are incomplete due to one of the following reasons:

- The systems are stand-alone and meant to track specific parameters and diseases, missing out on the great potential that lies in utilizing big data analytics on data from heterogeneous sources.
- The proposed architectures do not offer modularity to add and remove various components to personalize the solution to users' needs.
- Most studies do not account for technological advancements or well-established sensors and create custom sensors for their specific application instead.
- Cloud-based solutions suffer from an increase bandwidth requirements and higher latency. Therefore, implementations that depend solely on cloud storage and computing are not suitable for real-time response to emergency situations.

[1] https://www.fitbit.com/
[2] https://www.apple.com/watch/
[3] https://www.garmin.com/
[4] https://www.vivaquant.com/
[5] https://www.dexcom.com/en-CA

- Most proposed systems are not scalable which limits their future potential.
- Many studies focus on the core functionalities and do not offer intuitive methods for human-machine interaction. However, user studies stress that natural HMI is a cornerstone in the acceptability and usability of such solutions.

1.3 Thesis Contribution

The Internet of Health Things combines a variety of medical sensors, smart devices, and the cloud. These components allows an IoHT system to obtain data about the patient, analyze it to extract useful information, and permanently store the data to create a large database. Smart sensors are used to measure different body vitals such as the heart rate, blood pressure, glucose level, oxygen saturation, and the electrical activity of the heart (ECG). Combining several sensors can provide full history of the patient's health; thus allowing medical professionals to provide more personalized care and facilitating the transition from clinic-centric to patient-centric healthcare. Moreover, a continuous stream of physiological data will create a large amount, opening doors to various applications which benefit from machine learning and big data analytics to predict different diseases and complications. Such technology will allow alarming users of early onset of diseases which could potentially save numerous lives.

This thesis proposes a scalable multi-layered framework with a modular nature that offers an all-in-one solution and harvests all the benefits of the IoHT for real-time care. The proposed framework isolates different system components to allow plug-and-play of heteregeneous smart agents and functionalities. It tackles the gaps in previous literature by using commercial solutions as system agents. The framework also provides multi-modal natural user interaction and solidifies the concept by creating a graphical user interface, an intelligent chat bot, and a conversational speech interactive agents that are all able to access full system capabilities. Furthermore, the framework combines the low latency of fog (edge) computing and the scalability of the cloud for data storage and analytics. The flexibility and modularity of the proposed framework is showcased in its implementation for two major applications in the field of IoHT; remote healthcare monitoring and ambient assisted living.

1.4 Thesis Outline

The thesis is organized as follows. Chapter 2 provides literature review on the IoHT for remote health monitoring and discusses the studies done in care of older adults in ambient assisted living environments and with social robots. Chapter 3 describes the proposed framework and shows how different layers communicate and how the modules are utilized for a complete solution for human-machine interaction in an IoHT environment. In Chapters 4 and 5, the proposed framework is implemented (as a seed prototype) in a healthcare system for remote health monitoring and a multi-agent system for care of older adults, respectively. Chapter 6 concludes the thesis by highlighting the main contributions and future work directions to extend the current work.

Chapter 2

Literature Review

Previous studies in the domain of IoHT focused on different applications to offer a solution for remote health monitoring (RHM) as well as solutions that are specific to the aging population to offer an ambient assisted living (AAL) environment. Both fields are closely related with slight differences in the type of care provided and smart objects/devices utilized. This chapter provides an overview of recent work done in the the field with main goals of creating an IoHT environment for remote healthcare monitoring and ambient assisted living. It is important to mention that AAL and RHM involve utilizing smart sensors for health monitoring and disease prediction. Therefore, any discussion pertaining to inclusion of smart sensors in either RHM or AAL is also applicable to the other.

2.1 Remote Health Monitoring

One of the early works in the area of RHM was proposed by Maia et. al [3] where they introduced EcoHealth, a middleware platform capable of using wearable body sensors to measure physiological parameters and allow both patients and doctors to visualize and store data about the patient. The platform interacts with wearable sensors and smart devices using REST (Representational State Transfer - refer to Section 3.1.2) API through Bluetooth Low Energy (BLE). EcoHealth is also capable of sending notifications to its users with real-time data. The authors also explained how different physiological parameters could be used to predict and assess the patient's health status. For example, a high body temperature indicates that the patient is likely ill while sweating (using galvanic skin response) is a sign of stress or dehydration. In addition, fainting could be predicted by a sudden acceleration with a drop in heart rate.

The authors of [4] provided a review of various body sensors that can read physiological parameters and reviewed different communication technologies for data transmission from the sensors in an IoHT environment. The review covered how sensors can be used to monitor cardiac activity through electrocardiograms (ECG) and heart rate (HR) sensors which can be used to find various aspects of heart activity: the electric activity of the heart, heart rate , heart rate variability (HRV), and blood pressure (BP). The authors also explore and explain the way sensors measure body temperature, galvanic skin response (GSR), and blood oxygen saturation (SPO_2). In addition, they extended the study to show how multi-sensor systems can be used to monitor user activity and provide better insights about patients' health. Moreover, they showed the great benefit in textile-based sensors which are embedded in clothing. The study also highlighted how high body temperature is linked and can be used to predict insomnia, deterioration of cognitive functions, and severity of strokes and heart attacks. In addition, the study mentioned how SpO_2 measurements need to be maintained over 94% for functioning of cells and tissues in the body. Finally, the review concluded that wearable sensors can provide individuals with the ability to live independently while ensuring physical well-being with continuous and non-invasive health monitoring.

Both [5] and [6] proposed an architecture for RHM systems with three main layers:

- *Device Layer* which includes smart devices such as sensors, smartphones, and actuators.
- The *Fog Layer* contains devices within the local network that are capable of pre-processing the data from sensors, performing computations, and storing data locally. It also manages the varying requirements to connect to different devices.
- *Cloud Layer* capable of keeping all devices updated with most recent data, storing large amount of data, and performing big data analytics.

In addition, [5] suggested utilizing cheap and readily available commercial off-the-shelf (COTS) devices but offer no implementation of their proposition. On the other hand, [6] developed smart eyeglasses to measure HR and send warnings of abnormal readings to the user. The authors concluded the study with some recommendations to create a healthcare system that is centered about the patient rather than the clinic. Above all, they emphasize that the infrastructure needs to be scalable and the UI should be intuitive and easy to learn.

Particularly, several studies explored the potential benefits in monitoring patients with chronic illnesses and developing systems that are capable of detecting deteriorating conditions and providing early warning for dangerous medical situations. It is notable that

(a) Empatica E4 wristband (b) Dexcom G6 CGM (c) Apple Watch Series 4

Figure 2.1: Commercial Smart Wearables for Remote Health Monitoring

chronic diseases are expected to increase by 17% in the next 10 years [7], calling for a way to monitor patients and provide a better care methodology to prevent the adverse effects of late response to emergency situations. An early study involved developing a sensor that provides continuous ECG monitoring which can be used to predict cardiovascular diseases [8]. In [9], acceleration, GSR, BP, HR, HRV, and body temperature are measured during sleep using an Empatica E4 wearable (Figure 2.1a). The readings were then used to develop two types of models to predict early onset of migraines. A user-independent model was trained using all patients' data. This model was found to perform poorly. The authors argued the bad performance is because people experience migraines differently. Personal models were also created using data from each patient which gave an accuracy of 95.2% accuracy in predicting migraines attack. Therefore, the study showed how personalizing models using data from each patient separately can lead to better predictive models to detect pre-symptotic signs of an illness. Another study focused on continuous glucose monitoring (CGM) with diabetes patients and showed how it could improve their quality of life [7]. The authors described how storage and data access, visualization, and analysis are core features needed to provide an accurate diagnosis of the patient's conditions.

In addition, several commercial solutions exist that target monitoring of different chronic illnesses. For example, the Dexcom G6 [10], shown in Figure 2.1b, offers a CGM without fingersticks (which is common with glucose monitoring devices) and send alerts through their smartphone application (or to a smartwatch) about high or low blood glucose levels. Apple also introduced an electrocardiogram [11] in their Series 4 smart watch (shown in Figure 2.1c) which is capable of detecting irregular heart beat patterns as well as high and low heart rates.

The novel coronavirus 2019 (COVID-19) was first detected in Wuhan, China in December 2019. The virus hit the world with a global pandemic which overloaded medical facilities around the world and uncovered the gap between current technological advancements and the healthcare industry. COVID-19 is capable of spreading from an infected person even before symptoms start showing, which caused it to infect more than fourteen million (as per today's date of July 18, 2020) people and caused the death of more than 550,000 individuals [12]. In addition, polymerase chain reaction (PCR)-based tests that give the highest sensitivity in diagnosis of COVID-19 is costly and is limited in availability [13]. This emphasized the need for developing a monitoring system capable of utilizing physiological parameters to detect infection before symptoms appear. The authors of [13] described how early detection using wearables can prevent spread by stopping carriers of the virus from spreading it before symptoms appear. The study summarizes the relevance of different vital signs and give a summary of how each parameter is detected. The study covers how cardiovascular parameters such as HR, BP, HRV, and resting heart rate (RHR) can indicate if the individual is infected with the virus. In addition, the authors discuss the prevalence of low SpO_2, respiration rate, and body temperature with positive cases of COVID-19. Building on that study, the authors of [14] utilize a smartwatch to track HR, steps, and sleep of 31 participants. The study shows an association between the measured parameters and COVID-19 in 80% of the diagnosed cases using an offline model (after data has been collected). They also developed an online detection model with 67% accuracy in predicting if the person is infected. Although the accuracy is not optimum, the study shows how wearable technology can be utilized to find asymptotic cases (reported to be as much as 50% of cases) as well as provide an early warning system to reduce transmission of the virus.

Previous studies discussed the numerous benefits of adopting IoHT for remote health monitoring. They also provide insightful recommendations to design an all-inclusive solution for RHM and uncover the gaps in the implemented systems which are tackled in the current work. The benefits include healthcare facilities, medical professionals, and patients as shareholders and can be summarized as [6, 15, 16]

1. Health Care Facilities:
   - Reduce load on healthcare facilities as more people will be monitored remotely
   - Decrease hospitalization leading to a decrease in health costs
   - Fight shortage of healthcare professionals and increasing demand of healthcare as it decreases the need for direct doctor-patient interactions
2. Medical Professionals:

- Allow them to be more involved by having access to real time data
- Provide access to full patient history with real-time data
- Visualize patient data which allows better assessment
- Time saving due to the lower number of people who will need in-person checks
- Allow collaboration between different medical professionals as by storing the history of recommendations from doctors specialized in different domains
- Warn when patients need professional care by analyzing trends in patient's physiological parameters and detect abnormalities
- Provide a portable medical record to give any new health professionals a full picture of patient's history

3. Patient:

   - Allow real-time monitoring and health assessment therefore providing constant care at a high quality
   - Reduce hospitalization time and allow patients to lead a normal life
   - Save time and costs due to shorter stays at hospitals
   - Personalized care which allows the patient to feel safer and more in control of their health condition
   - Monitor patients with chronic illness and therefore provide early warning on possible deterioration in health
   - Analyze data in real-time and detect dangerous situations; thus resulting in faster and better care and an improved quality of life
   - Allow patients to contact their doctors therefore reducing hospital visits
   - Reduce risk on patients with chronic illnesses

2.2 IoHT for Older Adults

With the numbers of older adults on the rise around the world, older adults are requesting a solution that allows them to live independently and comfortably in their private residences. Currently, a caregiver is needed to achieve that by ensuring that the older adult is in safe and healthy and that all their needs are met. Caregivers can be divided into formal and informal caregivers [17]. Formal caregivers are medical professionals and nurses while informal caregivers are family members and friends who try to help the older adult in their daily lives. With the increasing costs of professional (formal) caregivers, and since family members can not afford to put in all the time needed to look after their elderly, a more innovative solution is required that is capable of providing continuous monitoring and services to this ever increasing portion of the population. IoHT solutions were offered as mobile applications which connect older adults to their families and caregivers. For example, Infosage, a mobile application developed at Harvard Medical School [18], offers a way to connect older adults as the primary users to their caregivers like their family members or nurses. It offers a medication reminder and tracker as well as an easy way to remind users of their appointments.

Ambient intelligence is a technology involving sensors that are adaptive and can respond to changes in their environment and user's requests; thus empowering people with a more comfortable living in their own homes. Ambient Assistive Living (AAL) for older adults combines ambient technologies with physically assistive technologies in a home environment to promote healthy living. The description of AAL environments is the same as that mentioned in wider scopes of creating systems for remote health monitoring in the IoHT with the exception that they are tailored to the needs of older adults. An additional unit that is present in many studies on assisting older adults is a social robotic agent capable of human-machine interaction (HMI) as well as task accomplishment. The current section first discusses IoHT solutions in AAL independent of robotics and then discusses works that focus on the utilization of social robotic agents. This division of literature is done to give a full background of the ideas utilized in the current work which combines recommendations from both aspects in the implementation found in Chapter 5.

2.2.1 Ambient Assisted Living

A review of AAL technologies and techniques is presented in [19]. The review explained how an appropriate AAL system should be able to:

- monitor and track the health of its users
- detect dangerous situations such as falls and provide an emergency alerting system
- assist with daily activities such as moving, eating, and communicating with family members, friends, and caregivers
- provide reminders for medications as well as daily activities that promote healthier living

The authors of [6], reviewed in Section 2.1, also discussed the various applications of IoHT in AAL. These applications including using IoHT to remind users to take their medications, highlight the user's health status, detect abnormal physiological parameters, and provide early warnings by predicting potential ailment.

Other studies focused on the design of the AAL system. For instance, the authors of [20] studied different architectures and models for AAL and analyzed them to come up with some recommendations to improve future designs of AAL frameworks. The authors recommended that AAL platform need to be standardized, maintainable, and exhibit independence in its layers/modules to allow changes during further development phases of the product. They highlight that creating a framework that takes into account how data is collected and combined from heterogeneous sources is one of the important aspects that require more focus from the research community.

Other studies focused on developing systems for AAL with specific functionalities. One of the first studies developed the mHealth framework [21]. MHealth was described as an open source Android based implementation that was presented to help the research community rapidly develop biomedical applications and cater for faster analysis, decision making, and more robust recommendations. They divided the framework into 4 layers, namely:

- Communication Manager: receives biomedical data collected from by sensors
- Storage Manager: stores raw data in a large database
- Data Processing Manager: Processes the raw data for better understanding and deeper analytics
- Visualization Manager: Presents a summary of the stored data in graphical form

The framework also includes functionalities such as abstraction of communication for receiving biomedical data from different sensors, permanent data storage, data processing to extract useful information, as well as a visualization system to provide summaries. The authors highlighted that future systems need to be expandable as more technologies emerge. In addition, they mention the need for more services to be added such as alerts and personalized user recommendations.

Another study presented an integrated platform for AAL with a multi-layer architecture [22]. The perception layer contains different sensors that are able to read health vitals about the patient. In the study, a heartbeat sensor, an accelerometer, and a body temperature sensor were used as wearable sensors and indoor humidity, temperature, light, and passive infrared sensors represented the non-wearable sensors in the environment. The second layer transfers data from the perception layer to the network and acts as the gateway to the devices. The integrated application layer contains all the stored data and can be thought of as the database for permanent storage. The authors built a prototype and presented preliminary results that the architecture proposed would provide a positive impact on the lives of older adults in an AAL environment by providing the following advantages:

- Accurate remote health monitoring with low errors
- Lower medical costs
- Feeling of independence while maintaining good health care for older adults
- More accurate disease prediction and management due to the availability of more data about the user

The authors of [23] combined the approaches in [21] and [22] in one framework that monitors older adults and utilizes machine learning algorithms to detect different physical activities. The authors made use of the multi-layer architecture presented in [22] and included the mHealth framework [21] in the perception layer. The study recorded ten participants while they perform 12 physical activities using the mHealth application to create a classifier for the various activities. The mHealth application transmitted the collected data to the cloud and data analytics layer which saved the data after applying the appropriate pre-processing to it. Finally, the application layer was used to provide a friendly UI. After the data was collected, it was used to train a multi-nomial naive bayes classifier, achieving an accuracy of 97.1% in detecting the 12 physical activities.

Another system was developed in [24] to detect abnormal actions such as falls using an ultra-wide radar sensor capable of reading heart rate, respiration rate (RR), and detecting movement without being physically attached to the user. The system exhibited accuracy up to 95% and 91% in measuring HR and RR respectively. In addition, the authors trained a support vector machine (SVM) model with radial basis function kernel to detect falls and achieved a specifity (True Negative Rate) of 90% and sensitivity (True Positive Rate) of 97%.

Similar to the works explored in Section 2.1, most studies in the field of AAL utilize cloud-based storage and computing. However, the delay that occurs during transporting data to and from the cloud for analysis leads to sub-optimal real-time performance. It also does not account for internet disconnections which would halt all functionalities the

(a) RIBA

(b) Paro Robot

Figure 2.2: Social Robots for Older Adults

system. Since such problems should be avoided in real-time monitoring, recent studies explored the idea of fog computing which utilizes a node at the edge of the local network of the user to store data temporarily as well as perform real-time analytics (similar to the work done in RHM [5, 6]). Fog computing-based patient monitoring system for ambient assisted living (FAAL) was proposed by [25] with a prototype that utilizes radar sensors to detect daily activities. In their conclusion, the authors highlighted that fog computing results in lower bandwidth requirements and latency; thus providing more efficient real-time response. Moreover, the study also found that utilizing fog computing for AAL and remote monitoring reduces the energy consumption of the system.

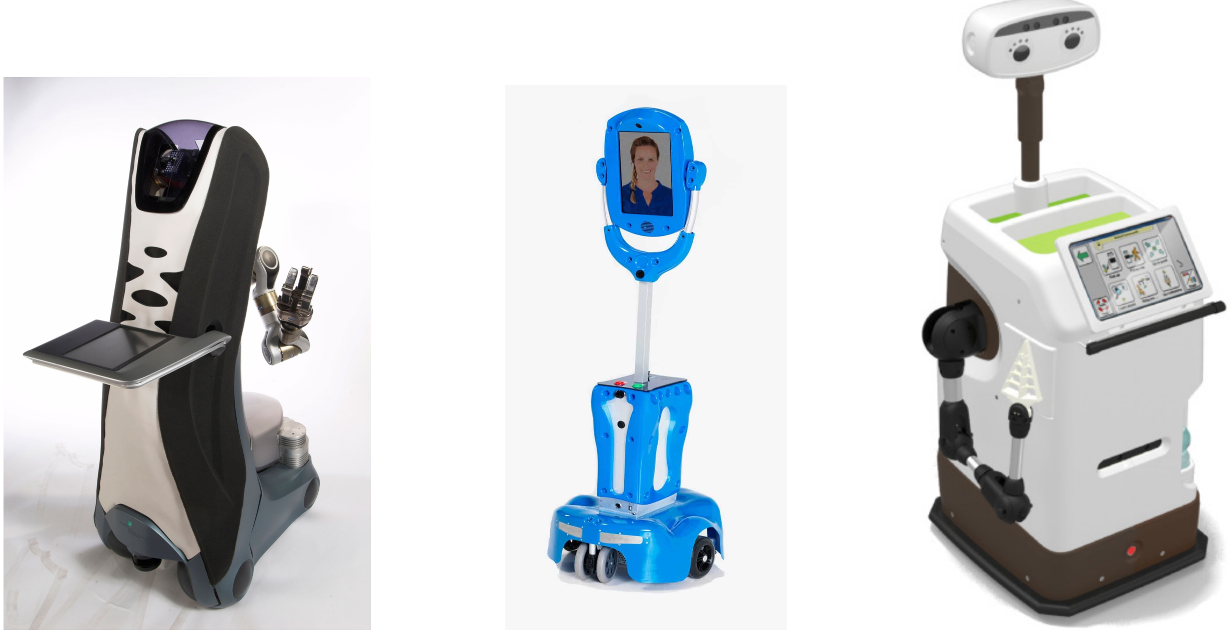
2.2.2 Social Robotics

A review of existing robot implementations for the care of older adults is presented in [26]. The review covered robotics tools such as an automated feeding spoon (Japanese Secom), a robot for carrying and transporting heavy objects called the Robot for Interactive Body Assistance (RIBA) (Figure 2.2a), and the Sanyo electric bathtub robot which aids older adults in bathing. Moreover, the Paro robot (Figure 2.2b), a robot which looks like a baby seal (popular in home care centers in Japan) was also mentioned to provide comfort, social interaction, and companionship for older adults. In addition, the review also described how robotic platforms were used to provide medication reminders, assistance with physical activities, engagement in activities to promote cognitive abilities, and human-like companionship to fight loneliness.

The ACCOMPANY project (Acceptable robotiCs COMPanions for AgeiNg Years) was a collaborative work between several researchers in the European Union (EU) that studied social human-robot interaction, robot learning, and memory visualization [17]. The project conducted user studies to understand the needs of different shareholders. The shareholders were first divided into three groups; formal caregivers, informal caregivers, and the older adults. Focus groups were conducted in which the problems facing the elderly were discussed to come up with the main areas to tackle in creating the social robot. In [27], the participants were first requested to write down the problems they see individually. Then, each group noted all the problems they see and grouped them into different problem dimensions and discussed their relative importance and how they are interconnected. Several problems were found from each user group and the similarities and differences were discussed. The authors concluded that the main problems facing older adults were:

1. Deteriorating physical health which calls for continuous monitoring by caregivers
2. Declining of the economic situation of the elderly and hence their inability to afford formal caregivers or a long-term care facility
3. Losing connection with family and friends
4. Difficulties in mobility

This was the first study in the long-term project which was shortly followed by a more elaborate discussion of future plans of the project as well as more user studies across the UK, Netherlands, and France in [17]. With the recommendations collected through the study groups, a prototype of a social robot was implemented based on a Care-O-Bot3 (Figure 2.3a) platform connected in an IoT environment. Primary user studies were conducted with 40 formal caregivers, 32 informal caregivers, and 41 older adults as participants across the 3 EU countries. The first user study identified 3 domains where problems led to loss of independence for the elderly; mobility, self-care, and social activities. Afterwards, a basic fetch-and-carry scenario where a robot finds, picks up, and carries a specific object was created and tested with participants. The study resulted in 68 user requirements, which were summarized into two scenarios. This was implemented in a later study presented in [28]. The first scenario involved retrieving a parcel from the front door, which is an extension of the simple fetch-and-carry scenario tested in [27]. The second scenario was to remind the user to drink water and bring it to them when they have spent 3 hours



(a) Care-O-Bot 3 (b) GiraffPlus (c) Hobbit

Figure 2.3: Social Robot Implementations (1)

without drinking. All interactions with the robot were made through a touch screen which suggested activities that are the user is most likely to request. To allow for such personalized recommendations, the robot was connected to multiple cameras and other sensors that detected the user's living patterns. The study concluded with some suggestions by the elderly including requesting the robot to open doors, greet visitors, and autonomously detect an emergency such as a fall and contacting a caregiver. In addition, all 3 user groups stated the need for personalized robots that can adapt to the user needs (personalization).

MOBISERV [29] provided a review of social robots in care of older adults and proposed a robotic platform integrated in an IoT environment with multiple agents. The paper suggests an architecture similar to those implemented in AAL technologies with a connection to smart objects such as home automation devices and smart wearables to read the user's health vitals, and a fog device that takes care of all computations in the system. The authors suggested adding a robotic agent that is capable of interacting with the older adults using speech to promote easier interaction and human-robot collaboration. Although the study did not mention an actual implementation of the system proposed, it provides insights on the implementation of a generic architecture that attends to all needs of the older adults through personalized agents.

Another work presented in [30] created a telehealth system based on a robotic platform called the GiraffPlus, shown in Figure 2.3b. The GiraffPlus robot was equipped with a touch screen and video conferencing to allow older adults to contact their caregivers and loved ones. After the implementation and testing the system with users, the study concluded that a robotic platform for care of older adults needs to be reliable, easy to use, consists of a few devices with which the user has to interact, and be personalized according to the needs of the older adult. Moreover, the Hobbit project [31] targeted developing an affordable social robot for elderly care capable of multi-modal interaction through a touch interface, voice interaction, and gesture control. The second prototype was tested with 49 users in three EU countries and the usability, acceptability, and affordability were assessed. The study concluded that the robot is found useful, easy to use, intuitive, and generally acceptable by the elderly. The authored emphasized that 49% of the participants preferred speech interaction over the other modes. The feedback from the study participants led the authors to create Hobbit 3 (the third prototype) shown in Figure 2.3c.

Another multi-stage project for developing a socially interactive robot for HMI and task achievement was given in [32] and [33]. SocialRobot (shown in Figure 2.4a) is a robotic platform created to mitigate the high costs of available platforms and offer a modular architecture for development of a social robot for elderly care. The robot created was based on ROS (Robotic Operating System - described in Chapter 5) and supported a MySQL database which keeps all required user data stored for personalized interactions called SoCoNet. The authors noted how the robotic system was capable of facial recognition and emotion classification which assisted in the personalization of the system to the user's mood. The SocialRobot robot architectural layers were first introduced in [32] where it consists of independent layers which isolate the hardware, the services provided by the system, and the database that stores data. A more recent work by the authors explored SocialRobot's application in a real-world environment of an elderly care center in the Netherlands [33]. The robot was placed in the care facility for a week and left to interact with users including caregivers, the center's staff, and the elderly. This study served as a pilot which gave the researchers insights on the usability and acceptance of their platform as well as some functionalities to add/change. For example, the users mentioned they wanted the robot to be able to play music, movies, and games. The caregivers also mentioned that it would be useful to allow the robot to carry objects and conduct memory training activities with the older adults. In addition, the older adults requested more advanced speech capabilities for a more natural interaction.

HomeMate, shown in Figure 2.4b, is a robotic agent for elderly care presented in [34]. It

(a) SocialRobot (b) HomeMate 3 (c) SYMPARTNER

Figure 2.4: Social Robot Implementations (2)

was aimed at older adults living at long-term care centers, where they are considered to have limited capabilities of Activities for Daily Living (ADL). The design of HomeMate went over 3 iterations, taking user feedback at each stage and adopting the altering the design accordingly. The robot had five main scenarios of service; it can play movies or music, allow for video chat with family members or friends, engage the elderly in games through voice and touch, and schedule reminders for medications or community events. The authors mentioned that the design was based on the principles of affordance in appearance (users can infer the robot capabilities from its appearance), a balance between autonomy and user control, and emphasizing dependability. Older adults were then recruited to test the system. All participants requested the 5 services through voice interaction, gesture control, a smartphone app, as well as the touch screen. The study did not mention how the scenario was structured and it ended with questionnaires which took the users' evaluation of the usefulness, ease of use and learning, and satisfaction. From the questionnaires, the authors determined that the participants found the scenarios useful, the system user friendly, and the gesture interaction easy and quick to learn. The study also noted that the elderly participants gave more importance to requiring simple and intuitive means of interaction that require minimum learning; referring to speech and natural language understanding as the main candidate. In addition, the authors also described the need for a method to communicate with the system when it is not in close proximity.

A long term study of 20 weeks where a robot was placed in a domestic environment of the elderly user was conducted in [35]. Participants got to use the robotic companion for one week each in their own home. The designed robot, called SymPartner (shown in Figure 2.4c) operated autonomously and focused on providing daily morning and evening routines, greetings at the apartment door, and reminders as well as health updates. In addition, the robot could find objects and recognize its users. Interaction was carried out through a graphical interface, but the robot could also respond to the user using speech. The companion acted completely autonomously with only remote assistance to fix minor issues. The 20 elderly participants were recruited from two service residential complexes in Germany where each of them lived alone in their own households. Evaluation of the usability, acceptability, and robustness of the robot companion was done following a pre-post design where participants' opinions were compared before and after the one-week duration of the experiment. The users participated in daily structured telephone interviews when the robot was in the house. The outcome of the study showed general satisfaction with the robot. The users developed an emotional bond with the companion which was partly attributed to its ability to adapt to the user's preferences, offering personalization. The authors noted that the users needed some time to get acquainted with the robot and learn how to request services. However, novelty effect was mentioned to have little impact since they spent a relatively long period with the robot. The participants found the reminders from the robot helpful, specially when it required little input from their side. They also mentioned that voice interaction made it easier to interact and request more complex behavior and activities without being physically close to the robot yet a more natural speech interaction would extend these benefits.

Besides that, [36, 26] and [37] recommended the inclusion of social robots in an environment with other smart agents designed for different tasks to result in a more beneficial system for elderly care. This is supported by the findings in [38] which state that technologies created for older adults are expected to be most successful when they require little effort to integrate into the environment and daily routines. These works shed a light on how social robots can be combined with other smart devices in an IoHT infrastructure to augment their intelligence and result in a more versatile and useful system. From the reviewed studies on social robotic platforms as well as those in AAL, the requirements for an IoHT system that incorporates multiple intelligent agents such as robots and sensors for the care of older adults can be summarized as follows:

1. Be extensible and modular to allow personalization of the system according to user needs

2. Provide a natural means of interaction that requires minimum learning and memo-

rization from the users through utilizing speech interaction with the ability to participate in a conversation fixed commands

3. The robotic system should be independent from the remaining system and allow different functionalities to be added or modified separately

4. Provide reminders of medications and appointments

5. Include a method of communication with the friends, family, and doctors (caregivers)

6. Monitor physiological parameters in real-time to provide a summary of the user's health and detect dangerous situations (falls, fainting, abnormal vital signs, heart attacks, etc ...) for faster response

7. Provide entertainment through music, movies, and cognitive games

In this chapter, a thorough review of previous work in the areas of remote health monitoring and ambient assisted living with IoHT environments is presented. Recommendations and requirements for RHM and AAL were summarized at the end of each section. Building on the recommendations and after some gaps were uncovered in previous implementations of both fields, the next chapter proposes a generic multi-layered framework for real-time care that can be utilized for RHM and AAL systems.

Chapter 3

Proposed Framework

In the previous chapters, the importance of integrating IoT and human-machine interaction solutions into healthcare were highlighted. In particular, emphasis was put on how IoHT could offer many advantages to both individuals and healthcare workers; thus, providing better healthcare, reducing costs, and decreasing the load on healthcare facilities and medical staff. While several solutions were proposed to tackle the emerging issues in healthcare, an all-round implementation that is flexible, expandable (involving old and new technologies) and scalable. Considering the requirements drawn from the current state of the art discussed in Chapter 2, the proposed framework enables linage of loosely coupled software components, in which each of its components makes use of little or no knowledge of other separate components to result in an all-in-one solution. The current chapter focuses on the proposed framework of which offers an all-in-one solution that is modular, versatile, and readily scalable.

3.1 System Architecture

3.1.1 Multi-Layer Architecture

The system in the current work combines the multi-layer architecture and recommendations in [6] and [5] and incorporates it with additional tools for real-time health monitoring and inclusion of smart autonomous agents. The system is composed of 3 layers as illustrated in Figure 3.1:

1. **Device Layer**: The device layer contains physical smart devices equipped with sensors such wearables that are able to monitor real-time physiological parameters such as heart rate, body temperature, and blood glucose level continuously. This also includes the user interface which allows the user to harvest the capabilities of the system. Devices in this layer provide the means for HMI with the users. It allows users to input data both manually and autonomously (through sensors) as well as access and visualize the data through multi-modal user interfaces.

2. **IoT Fog layer**: Fog computing (also called edge computing) refers to a decentralized computing structure which uses devices at the edge of the local network for communication, data storage, and computation. A fog device is defined as a device with the ability to receive and send data through wireless network connectivity as well as perform computations and store data locally. The name 'fog' is meant to portray that computations are done closer to the ground (compared to 'cloud', which is further in the sky). The fog layer was introduced to IoT systems to facilitate real-time computations to analyze and respond to data with minimal or no delay. The close proximity of the fog layer devices to the device layer allows faster response to the input data stream. In addition, fog devices act as a filter to store recent data, run sanity checks, analytics, and computations. Afterwards, only processed data that needs to be stored permanently is sent to the cloud, without affecting the response to the raw data stream. The fog layer connects to all smart devices in the network of the user and allows integration of heterogeneous devices. In addition, it contains most of the software modules utilized in the system and runs on a machine within the user's local network.

3. **Cloud Layer**: A cloud-based database and inference engine that is able to store permanent data and utilize machine learning algorithms and big data analytics for predictions and recommendations. The cloud layer extends the fog layer's capabilities by:

    - Synchronizing the data and making it available to all users (healthcare professionals and patients)
    - Keeping a secure data storage which is duplicated across several servers, thus increasing reliability
    - Provides expandable storage size and computational power allowing the system to be easily scalable

The device layer includes wearable sensors that read and transmit physiological as well as agents for manipulation of the physical world such as robotics. It also includes various

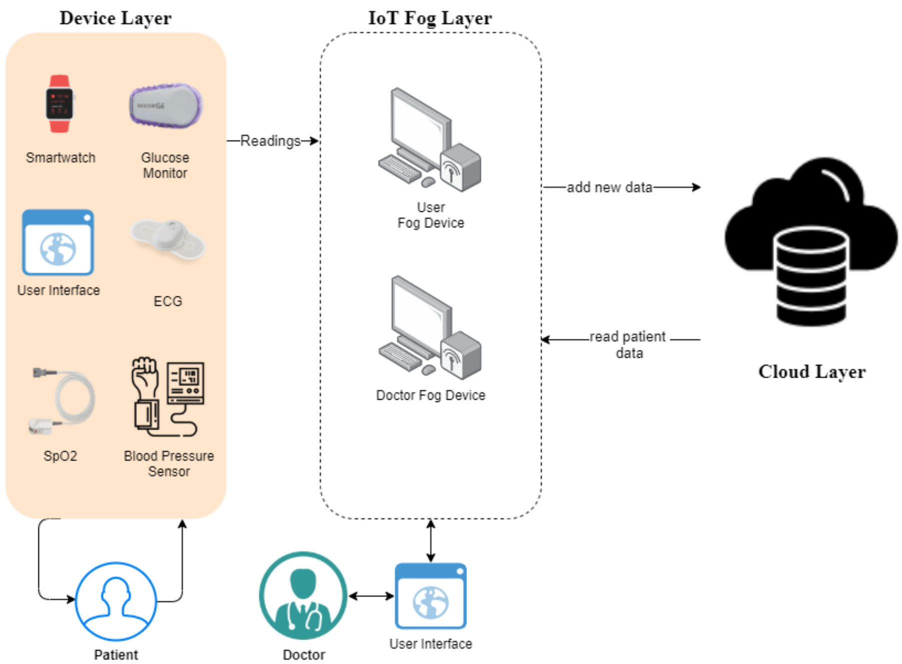


Figure 3.1: System Design with 3 IoHT Layers

user interfaces for HMI. All of the acquired data is transmitted to the IoT fog layer. In the current work, the fog device is a computer that resides within the user's local network. The fog device accepts the user's request, understands the required task, and performs it accordingly. After data processing is performed in the fog layer and preliminary analytical algorithms are applied to the data, the fog device is able to send notifications to the users and send data to the cloud layer. The cloud layer has a permanent copy of the user information. It also allows patients, medical professionals, and other authorized users to be able to view and alter the data remotely from any location.

The next sections discuss how different layers send and receive data and how the fog layer is divided further into different modules to allow easy expansion of the system and flexibility in its use cases.

3.1.2 Communication Protocols

The current section focuses more on the protocols used in transmitting the data rather than the wireless technologies on which the data is transmitted. The readers are referred to [4] for an in-depth study of various communication technologies used in remote healthcare including GSM (Global System for Mobile Communications), LTE (Long-Term Evolution), Bluetooth, Bluetooth Low Energy (BLE), Zigbee, Radio Frequency Identification (RFID), Near Field Communication (NFC), and WiFi. In the current work, BLE is used to connect the smartphone with wearable devices while WiFi is used for data transmission between the smartphone and the fog device and between the fog device and cloud. for all other forms of communications between the fog device and the cloud.

To allow different layers of the system to exchange data, and with the wide variety of devices that are expected to be utilized in the system, REST and MQTT communication protocols, being the most common in IoT applications, were explored and implemented in communications of different parts of the system. The use of both protocols creates a generic system that connects to a larger variety of smart devices. REST and MQTT protocols can be summarized as:

- **Representational State Transfer**: REST is an architectural style for communication between web services using hypertext transfer protocols (HTTP) requests. Webhooks use REST API to transfer data between modules on the same machine as well as remote agents. HTTP operations (GET, POST, PUT, and DELETE) cater for devices created by different manufacturers and based on varying protocols and data formats. Being a well-established protocol for data transfer, it allows accessing the functionalities of many different devices while avoiding compatibility problems [3].

- **Message Query Telemetry Transport**: MQTT is a light-weight machine to machine (M2M) communication protocol [39]. MQTT uses a server to which messages are sent, called a broker. A client connected to the broker can act as a publisher and/or a subscriber to different topics. Using the publish/subscribe protocol, a publisher client can send a message on a topic and all subscribing clients would receive it. MQTT allows for multi-threading where a single client can connect to multiple topics to publish and subscribe simultaneously [22]. MQTT, therefore, allows a large number of clients on different devices to connect and permits real-time communication due to its lightweight and speed. It is one of the most recently utilized communication protocols in IoT due to its speed of communication and flexibility. MQTT has well-established implementations in nearly all programming languages and is expected to increase in use for IoT purposes over the coming years.

3.2 Fog Layer Modules

The fog layer is divided into modules with specific roles. Modules can be changed, added, or removed according to the main requirements of implementation. In this section, a description of each module and its role is provided. Chapters 4 and 5 show how the application changes and how various modules are utilized to illustrate the modularity and flexibility of the proposed system.

3.2.1 Request Handler

The request handler is the first point of contact for any external request from the user. The request handler receives the request from the user through one of the possible user interfaces and directs it to the appropriate module for execution. Possible system requests are divided into three main categories:

- **Add Data**: New data from the user can be either biometrics (physiological data) or new reminders. Biometrics could be provided by the user through self testing, a medical professional, or a smart device that streams real-time data.
- **Request Data**: With a lot of data stored about the history of the patients health, the users are provided with the capability to explore the data in a meaningful format. Such data visualization and summaries would help patients keep track of their health status and offer medical professionals better overview of the patient's health. As a result, the request handler allows users to access the data and communicates with the report creator to generate said summaries and visualizations.
- **Request Action**: In presence of agents that can perform actions, the request handler is capable of formatting the request correctly for the agent link (Section 3.2.4). Such actions could be to send a notification to the user, move a robotic agent, or control smart devices in a smart home environment.

The request handler is able to understand and achieve user commands by passing it to the appropriate module. It accomplishes different tasks through its links to the other modules; it connects to the data handler to add or receive data, the agent link to perform actions, and the report creator to obtain summaries of user data.

3.2.2 Data Handler

When data about the user needs to be added, changed, or queried, the data handler takes care of the task. The data handler is set as the link between the request handler and the database handler. The data handler main roles can be summarized as:

- Ensures the user has the appropriate authorization to perform the action
- Checks the data for validity
- Restructures the data from the request handler into a format the database handler expects and vice versa
- Acts as a buffer between the request handler and database handler to allow changing the database (and consequently the database handler) without affecting the rest of the system

3.2.3 Database Handler

The database handler is used as a direct link to the cloud layer. It allows the system to communicate with the cloud using the appropriate API. This module translates a request for data into a query for the database (SQL or No-SQL) and returns the received data to the data handler. It also uploads new data points to the database after formatting it appropriately. Therefore, the database handler isolates the fog layer from the cloud and allows making changes to other modules without affecting the cloud link. Consequently, changes can be made to the database structure, type, or API without affecting the remaining modules of the system in hand. The database handler includes all options to add or remove records, users, or query items of interest.

3.2.4 Agent Link

The agent link is connected to the request handler and is in charge or sending commands to physical and virtual agents as well as receiving data from smart sensors. UIs are also classified as agents in this framework. In other words, this module links between the external physical world and the system. The physical world includes the users (through UIs), sensors, smart wearables, smartphones, and robots. The agent link is in charge of translating human commands to a format the request handler expects and translating system feedback and action commands into output to the user. System agents are divided into two categories; virtual agents and physical agents.

Virtual Agents

Interfaces for HMI constitute the virtual agents in the proposed framework. Different methods for HMI were utilized in this work, showcasing the flexibility of the systems developed and how they are able to provide the same functionalities across various platforms. In particular, the two developed systems utilize a web interface, an intelligent conversational agent, and speech interaction through virtual digital assistants. The exact implementation of these HMI agents is discussed in Chapter 4 and Chapter 5.

Physical Agents

Physical agents are those that have a presence in the physical world and can either sense environmental parameters or make alterations through actuation. For example, smart wearables read patient biometrics and communicate with the agent link to send the data to the system. In addition other sensors include cameras for event detection, GPS for localization of the user, as well as other smart home devices. The system architecture described here is capable of adding such agents seamlessly.

Another physical agent that is utilized in the current work is an autonomous indoor robot. The robot is an external module that only receives commands from the system and responds with feedback on the success of the operation. All robotic processing and task achievement implementations are separate from the core of the system, once again assuring modularity and personalizing the robotics according to user's needs.

3.2.5 Analytics Engine

All data sent to the system, whether through smart wearable devices, self-declared by the patients, or entered by physicians contains a lot of information and knowledge which can be unveiled with robust and accurate predictive models. Several studies have indicated how monitoring physiological health parameters can help track and predict dangerous scenarios for patients. This has been applied to cardiovascular diseases [8, 40], diabetes patients [7], and with older adults [22]. Evidently, the stored information about users could play a vital role in decreasing the risk of potential harm. The current proposed framework opens the doors to accessing all kinds of data in one integrated system. This has a lot of potential. For starters, a high body temperature indicates that the patient may be ill and in immediate need for medical attention. Moreover, ECG and blood pressure data can be used to give early signs of cardiac irregularity or even arrest. Likewise, other physiological symptoms can also be used to predict other illnesses and assure patients with chronic diseases receive quality care.

3.3 Summary

In this chapter, the multi-layered architecture of the generic system created for HMI in an IoHT environment is described. Because of its modular structure, the design possesses high flexibility and potential for scalability. To emphasize the strength of the proposed system architecture, the upcoming two chapters utilize the architecture presented for two real world scenarios. The next chapter will discuss the design and implementation of the WeCare system for remote health monitoring and disease prediction.

Chapter 4

WeCare System for Health Monitoring and Disease Prediction

Based on the reviewed literature, it can be concluded that the most important functionalities of any RHM system include data storage, access, analysis, and visualization. These capabilities allow doctors to monitor their patients' health status, give them more control, reduce hospitalization, and allow patients to feel safer. The WeCare system offers an adaptive, flexible, and scalable middleware solution for remote health monitoring. WeCare also allows users to add data manually through a custom UI and visualize summaries of the patient data for better assessment. Moreover, it is also equipped with an intelligent analytics engine to predict various illnesses and provide warnings to users. In this chapter, the WeCare system design is described in greater detail. A prototype version of the system with all components is created and the results are shown in Section 4.2. The impact of the system on healthcare professionals and patients is a positive leap in this sector. Accordingly, WeCare system is created as a seed prototype built using the framework described in Chapter 3.

4.1 System Architecture

The WeCare system is designed based on the generic framework described previously, with most of its software components residing in the IoT Fog Layer. The system is developed for continuous tracking of biological parameters and symptoms. It allows both patients and health workers to input data about the patient with different access levels. Patients can

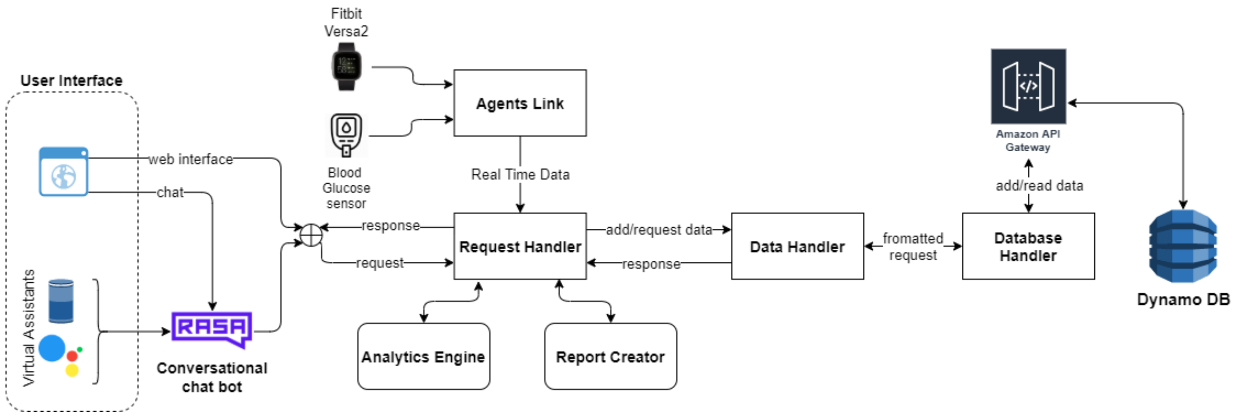


Figure 4.1: WeCare System Software Architecture

also self-test their health, providing more data for the prediction of various diseases and continuous monitoring. The software architecture of the WeCare is depicted in Figure 4.1. The system is divided into several modules as described in Section 3.2. Further description of the user interface, the intelligent chat bot, and the cloud database structure is provided next.

4.1.1 User Interface

A web-based user interface was designed which enables users to log in using their username and password. After logging in, the user is able to add new health vitals about the patient such as temperature, systolic and diastolic pressure, and heart rate. In addition, the user can add other symptoms such as dry cough, fatigue, sore throat, or headache. The web interface also allows patients and physicians to see the most recent updates about the patient as well as visualize the patient data over different time periods. These summaries allow the physicians to make a better assessment about the patient's health and prescribe treatment based on the history of their condition. In addition, it provides patients with a way to track their health status, providing with an enhanced feeling of safety and control.

4.1.2 Health Assistant Bot

From previous studies, it can be concluded that users prefer easier and more natural means of communication. Chat bots are currently integrated into various applications to answer frequently asked questions, provide an easier way to add or change user data, and even complete user orders (in e-commerce). From a healthcare perspective, a chat bot would provide an easier way to add physiological data, a summary of patients' condition, as well as answers to common medical questions from a credible source. An intelligent conversational chat bot system is implemented in the current work, capable of understanding natural human language; providing means for easier interaction with the system. *Rasa*[1] [41], an open source machine learning based framework for contextual assistants, is used as the natural language understanding (NLU) and dialogue management module in the current system. The open source nature of Rasa allows independent upgrade of its capabilities to provide new system functionalities as well as a lot of community support, making it the ideal choice as the system progresses through development stages.

Rasa Contextual Assistant

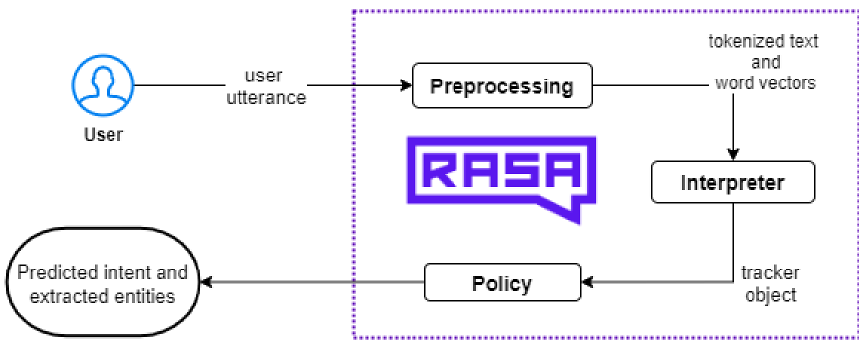


Figure 4.2: Rasa Pipeline

Rasa understands and responds to the users' requests by running what they said (utterance) through three stages: pre-processing, interpreting, and running the policy, as demonstrated in Figure 4.2. Rasa is divided into two components. The first component is the (NLU) model which performs all required pre-processing steps such as tokenizing the text and removing invalid characters and white spaces. Tokenization segments the given text into smaller units of words, numbers, and symbols called tokens. This process is done by defining a separator between words. The characters between the white spaces define the word boundaries and allow the tokenization to take place. After the tokens are created, they are mapped to vectors of real numbers through a process called word embedding. The output is then passed to an interpreter that is able to extract intents and entities and add them to a dictionary [41]. The user's intent is what the chat bot understands

[1] https://rasa.com/

Table 4.1: Physiological Parameters Monitored in WeCare

Name	Data Type	Values (units)
Temperature	Float	25-50 (C)
Systolic Pressure	Float	0-200 (mmHg)
Diastolic Pressure	Float	0-200 (mmHg)
HeartRate	Float	0-250 (bpm)
Blood Glucose Level	Float	4-9 (mmol/L)
Fatigue Muscle Pain Dry Cough Sore Throat Nasal Congestion Runny Nose Shortness of Breath Loss of Taste or Smell	Categorical	None, Mild, Moderate, Severe

as their main request while entities represents auxiliary information provided in the user's input, required for completing a certain action. The extracted information is stored to keep record of the state of the conversation in a tracker object. The tracker is sent to Rasa Core (the second component) which decides the next action to perform by running it through the policy. The current policy utilized in the Rasa framework is a Recurrent Embedding Dialogue Policy (REDP) [42] which is inspired by the neural embedding model proposed in the Starspace algorithm [43]. REDP makes use of supervised word vectors which is extracted from the provided training data. Unlike the Starspace algorithm, REDP takes into account the history of the conversation stored by the tracker. Hence, REDP is able to learn the effect of previous actions and user utterances on the current state of conversation and adapt its predictions accordingly. It is also able to train on a small amount of data and is language agnostic [42]. The chosen action then updates the tracker for future reference.

4.1.3 Cloud Database

Data received by the fog layer is processed and run through the analytics module. The system then sends notification to users for any detected abnormalities. Afterwards, the database handler formats the data to adhere to the cloud database API and sends it to the

Figure 4.3: Structure of Data in DynamoDB from a Sample Patient Record (ID: p0307)

cloud layer for permanent storage. DynamoDB[2], by Amazon Web Services (AWS), is the main cloud database in use in the current system. DynamoDB is a NoSQL (non-sequential) database which allows more control over the structure of each item separately. A NoSQL database is non-tabular and usually works with key-value pairs. NoSQL databases provide a flexible schema that allows different parts of the data to be added separately as well as scaling of the database to include more keys without changing the existing data. It also creates relations between data nested in the same data structure without using separate tables. As a result, it is possible to declare a subset of the physiological parameters with DynamoDB without using extra storage space. In other words, the users can enter only the parameters they can read at that moment, making user interactions more natural. The database stores items in nested key-value pairs which can be represented as a JSON[3] (JavaScript Object Notation) object [44]. JSON is a lightweight data-interchange format which is easy for both humans and machines to read and create. JSON is based on JavaScript Programming Language and it provides a format that is independent of the programming language used and is widely adopted for data exchange purposes.

[2] https://aws.amazon.com/dynamodb/

[3] https://www.json.org/json-en.html

4.2 Implementation and Results

In the implementation of the WeCare system, special attention was given to COVID-19 symptoms extracted from [45, 46] and those reported in [13]. These symptoms are chosen to be the physiological parameters monitored by the WeCare system. The general division of the records in DynamoDB is shown in Figure 4.3. Each patient record contains a list of doctors authorized to access their data, some personal information including their full name, date of birth, and nationality, as well as physiological data. The database can be accessed only through the anonymous patient ID, therefore avoiding any access to personal information and adding privacy to user information. The patient health vitals (physiological data) each contains separate entries with the date and time as the key and the reading as the value of the parameter inserted.

The utilized physiological parameters are shown in Table 4.1 along with the data types and values expected for each. The prototype presented here is created for doctors to monitor their patients' health and be able to enter data manually. This is to be extended later to involve patients and other healthcare professionals with varying levels of access. For example, patients will be able to self-declare general symptoms like those implemented currently while medical staff will have access to add/view more specialized data. For instance, doctors should be allowed access to data such as enzyme activity, lab test results, and other diagnosis that should only be available to healthcare professionals. Currently, the system provides doctors with the ability to insert data in two ways through the created web interface; either by filling a form which contains all the data types, or by chatting with

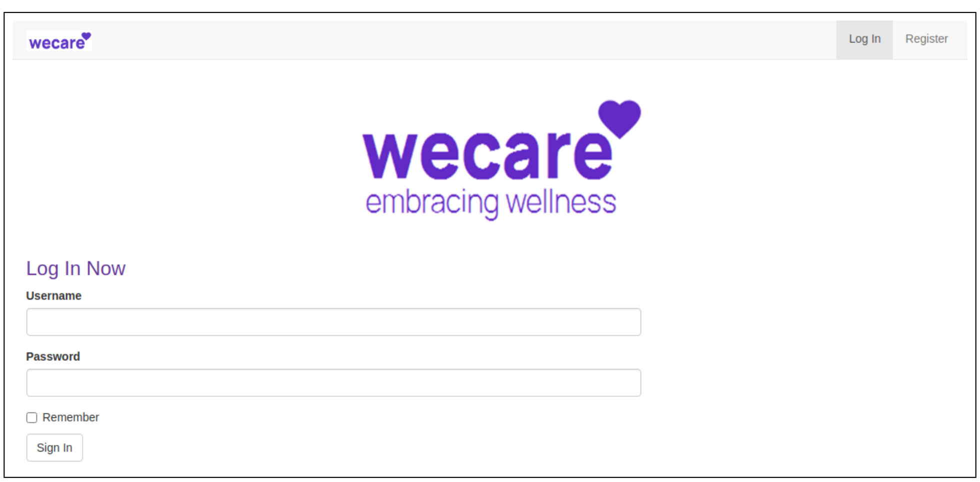


Figure 4.4: WeCare Log In Screen

a conversational chat bot.

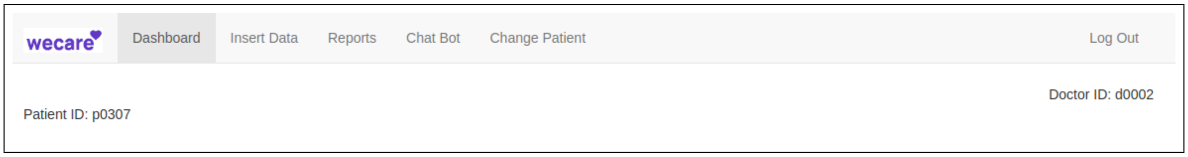


Figure 4.5: WeCare System Tabs with Static Header

The WeCare system described previously was implemented using Python in the back-end. The web interface was created using HTML, CSS (Cascading Style Sheets), and JavaScript. The main interface is composed of 5 main web pages as well as those to register and log in (shown in Figure 4.4). To demonstrate, Figure 4.5 shows navigation toolbar with the tabs to the five main pages and Figure 4.6 briefly depicts their respective functionalities. Once the user logs in, they are prompted to enter the ID of the patient they would like to access. The system then checks if the doctor is authorized to access the patient and retrieves the patient data from the cloud. If access is granted to the doctor, summary of the patient's most recent data (over the past week) is compiled and visualized in the *Dashboard* as shown in Figure 4.7. The dashboard contains all recent data about the patient and the time it was taken. At the top of the page, the system shows the ID of the doctor using the system as well as that of the patient they have accessed as shown in Figure 4.5. This information is always shown at the top of all the web pages. From the dashboard, the doctor can click on any of the "Details" buttons to view the history of a specific parameter for the past week. This also redirects the user to the *Reports* page.

The *Reports* page, shown in Figure 4.8, lets the user pick the window (start and end date and time) they want to visualize the data between. The user also chooses one of four values for frequency: (1) hour (2) day (3) month (4) year. Finally, they choose the type of

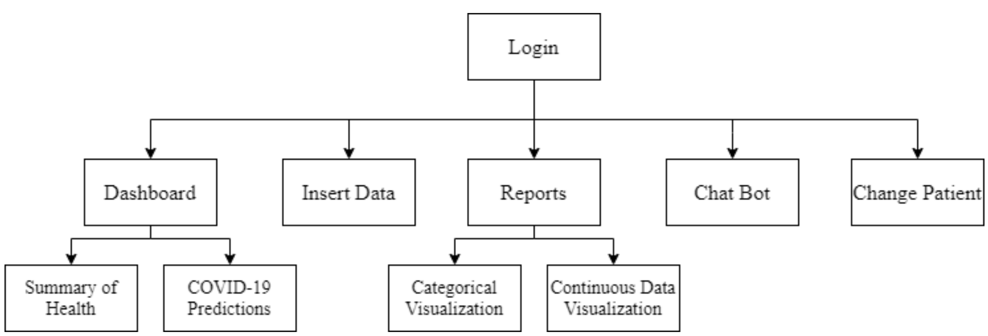


Figure 4.6: WeCare Web Interface Pages and Functionalities

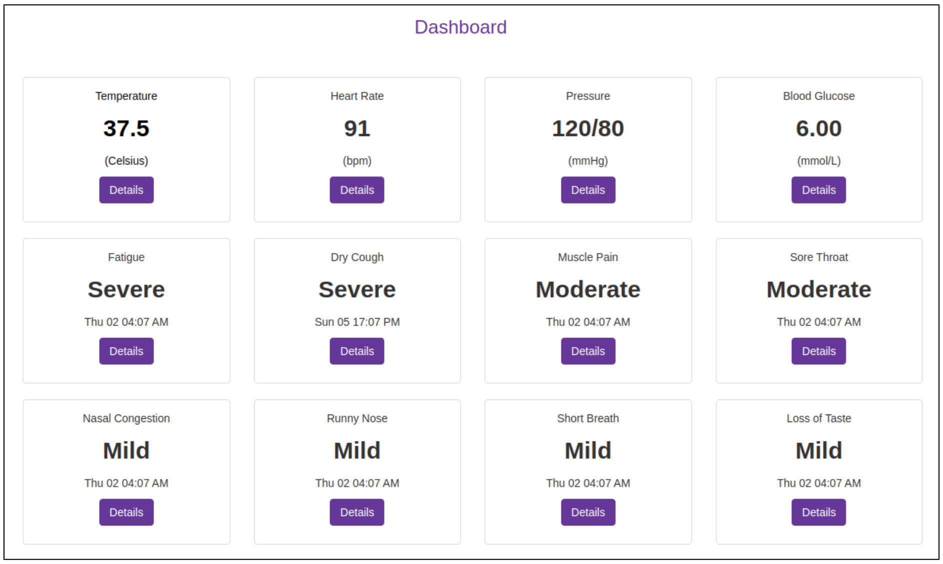


Figure 4.7: WeCare System Dashboard Page

vital they would like to plot (Figure 4.9). These custom plots provide doctors with more detailed data to help them better understand how different physiological parameters change over the short and long term (days, weeks, months, or years). Figure 4.10 shows the heart rate plot filtered by hour and day to illustrate how the frequency filter affects the plotted data. The user can also choose specific values to visualize for categorical parameters as shown in Figure 4.11. As a result, they can see how many times the patient had a more prominent level in one of the parameters over the same time range and frequency filter.

The *Insert Data* tab directs the user to the page where they can enter new data about the patient, shown in Figure 4.12. It is set as a form with empty cells for float variables and a drop-down menu for categorical variables. The float variables have checks to make sure they fall within their expected ranges. If not, they are disregarded and a warning is issued to the user. Moreover, the web interface takes into account that the user is not obliged to enter all data points at once as mentioned previously in the database structure. Furthermore, since each physician is usually looking after several patients, the WeCare web interface provides a page to easily change the patient in use from the *Change Patient* tab.

The chat bot is located in another screen and it provides a more natural method of interaction with the users to enter data. The chat bot also allows to easily switch between patients and enter data right away. This will also allow for easy integration of a voice assistant in later stages. Using this assistant, the physician will be able to enter data

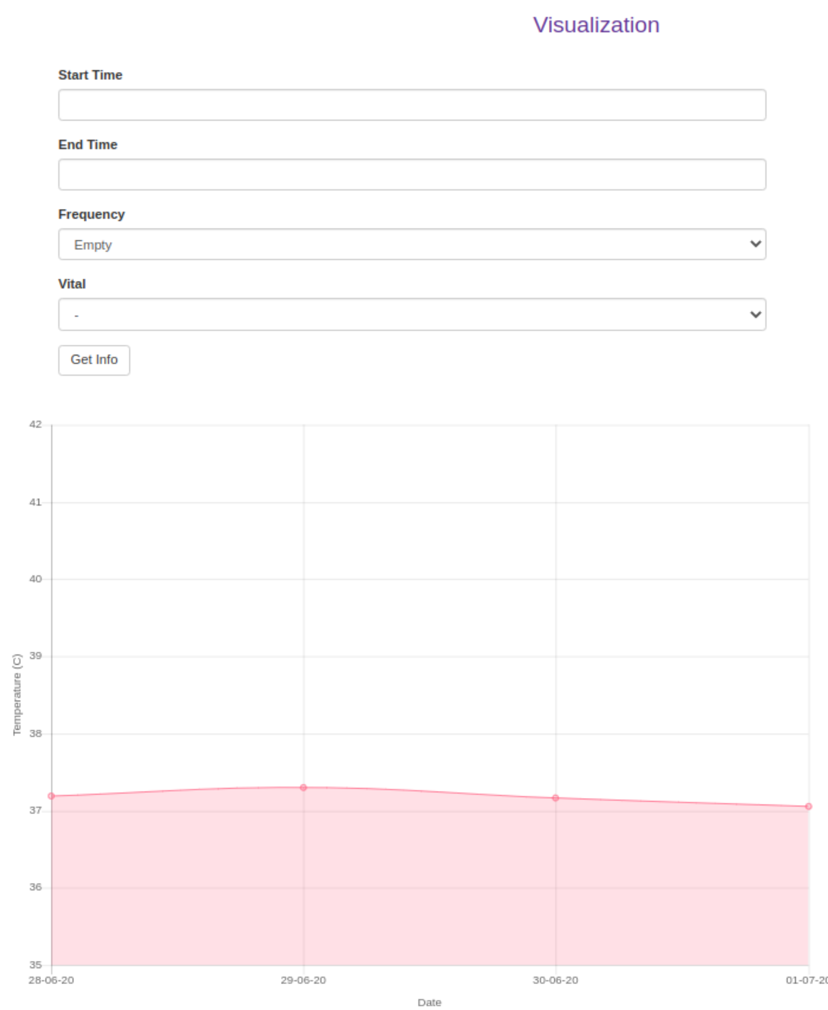


Figure 4.8: WeCare System Reports Page

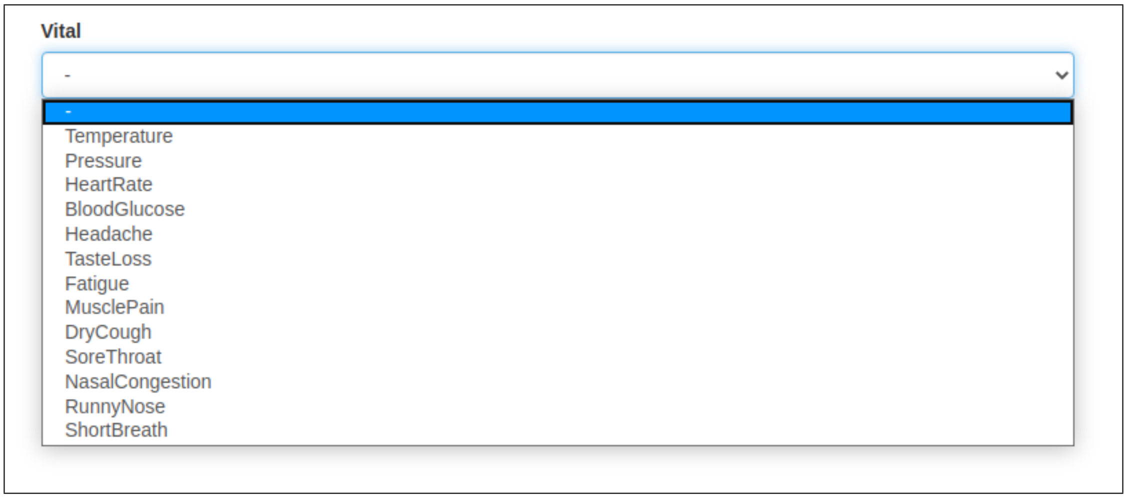


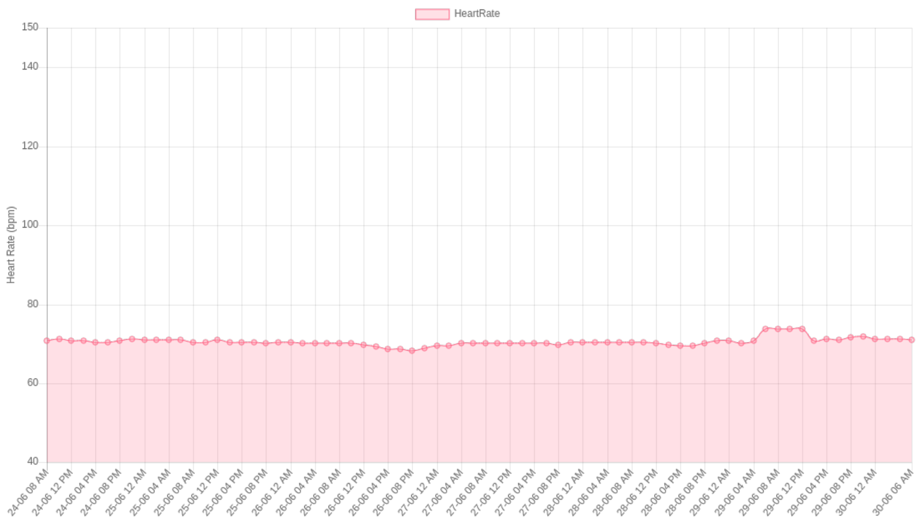
Figure 4.9: WeCare System Categorical Parameters Visualization

about different patients quickly. A sample interaction with the chat bot is shown in Figure 4.13 where the doctor is granted access to a patient and then enters the temperature, dry cough, and heart rate.

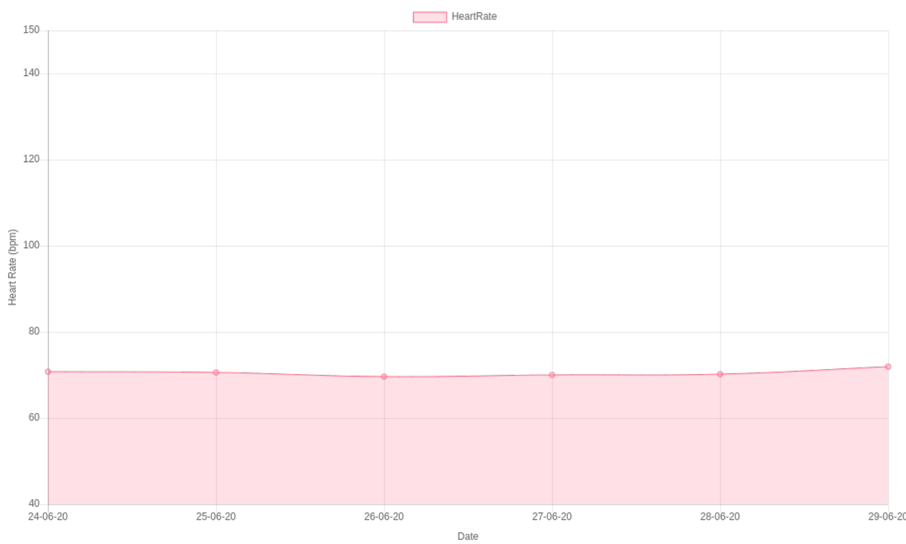
4.2.1 Smartphone Application

A prototype for a smartphone application is also designed with the main graphical interface adopting the same underlying system in the WeCare web Interface. The design of the smartphone application shows the next step of development for the WeCare. For example, the doctor is able to view a summary of a patient's health through the dashboard, add new data through the form, view summary reports, and interact with the chat bot to add new data as shown in Figures 4.14a, 4.14b, 4.14c, and 4.14d respectively. The application design also shows how patients and doctors will have slightly different interfaces and functionalities. In addition to the functionalities the WeCare system presented possesses, the mobile application showcases the next step of development for the system with additional features such as:

- Separate login options for doctors and patients (Figure 4.15a). Each user level allows for different functionalities (Figures 4.15b and 4.15c) and level of access. As mentioned earlier, the patient will be able to self declare general symptoms they are experiencing while the doctor will have access to adding more specialized data.



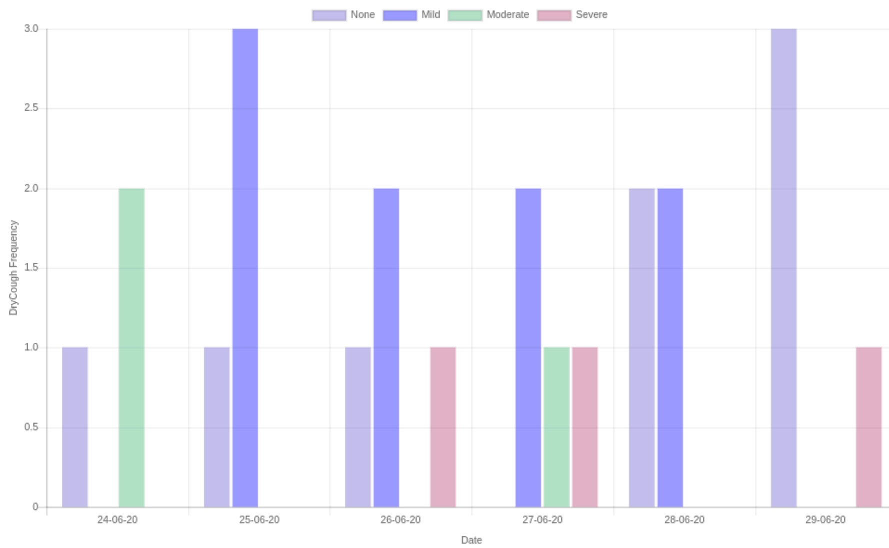
(a) Heart Rate per Hour



(b) Heart Rate per Day

Figure 4.10: Frequency Filters for Visualization in WeCare System

- Allows each doctor to access all their patients directly through a list (Figure 4.15d). The doctor can also see a summary of the patient's health and chat with them if needed as shown in Figure 4.16a. Likewise, each patient will find a list of their doctors and will be able to chat with them.



(a) Dry Cough (All Levels)

(b) Moderate and Severe Dry Cough Levels

Figure 4.11: Category Filter of Health Vital Visualization in WeCare System

- The application provides reminders for appointments and medications and provides patients with a calendar with their appointments on their home page (refer to Figure 4.16b. They are also able to see a summary of their health similar to that in Figure 4.14a.

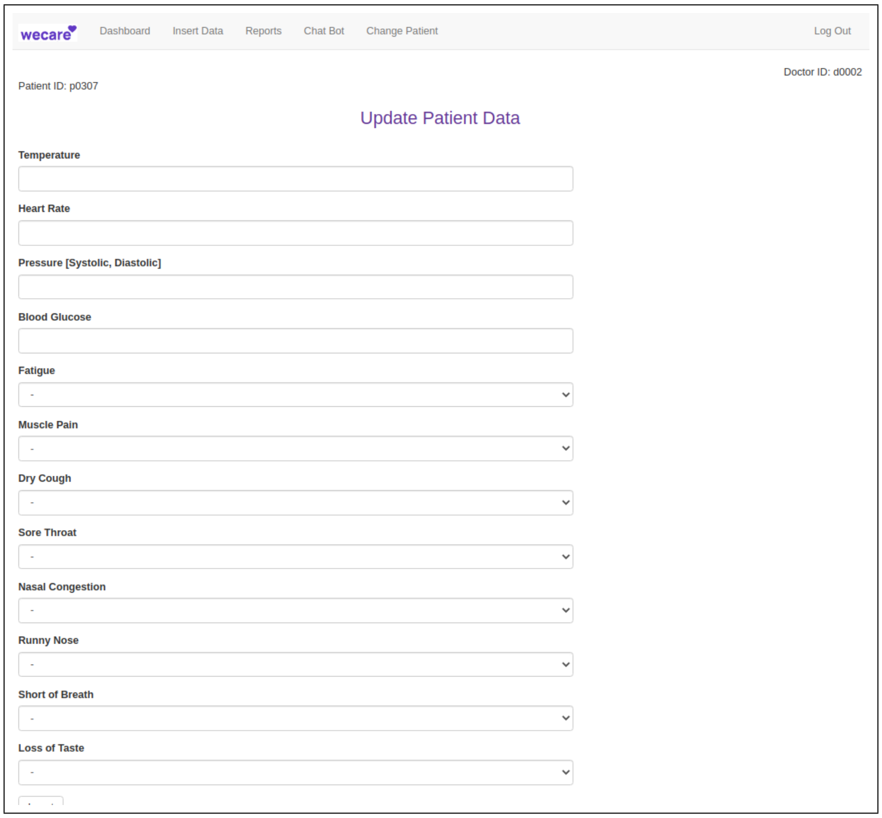


Figure 4.12: WeCare Insert Data Form

- The analytics screen of the mobile application will provide the predicted risk level for different illnesses based on machine learning models as shown in Figure 4.16c.

4.3 Disease Prediction

Disease prediction is a vital part of a RHM system. Consequently, the WeCare system explores this idea and integrates a machine learning model to predict the probability (risk level) of a patient being infected with COVID-19 using 3 of its most prevalent symptoms as part of its analytics engine. The emergence of the novel coronavirus in recent days has rendered the society helpless and many businesses had to stop suddenly. Many people may be carriers of the disease and therefore spread it and not know about it until it is too late. This calls for a way to pre-screen patients and provide warning of early onset

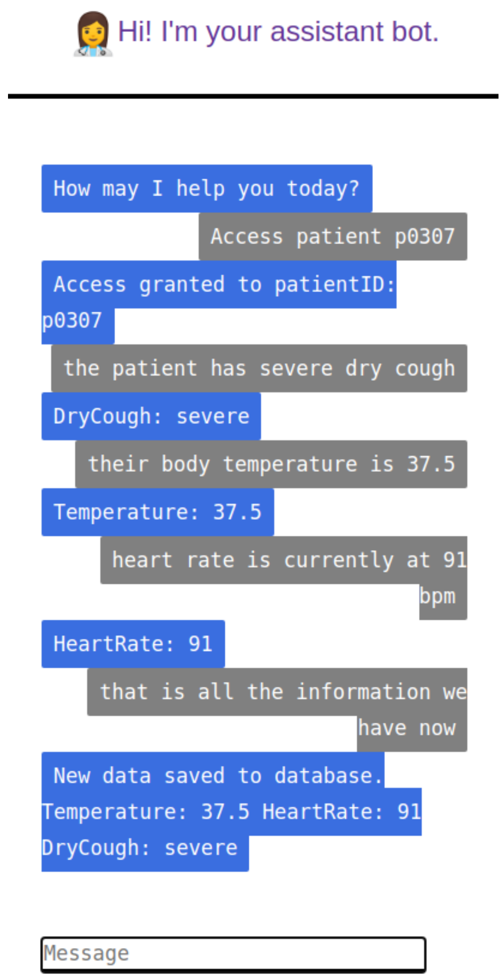
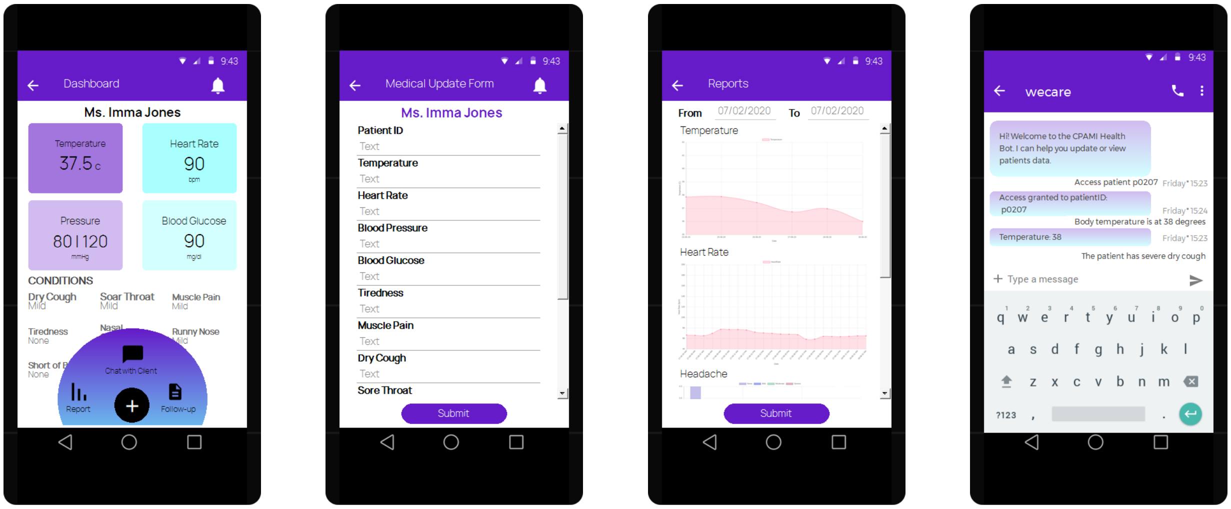


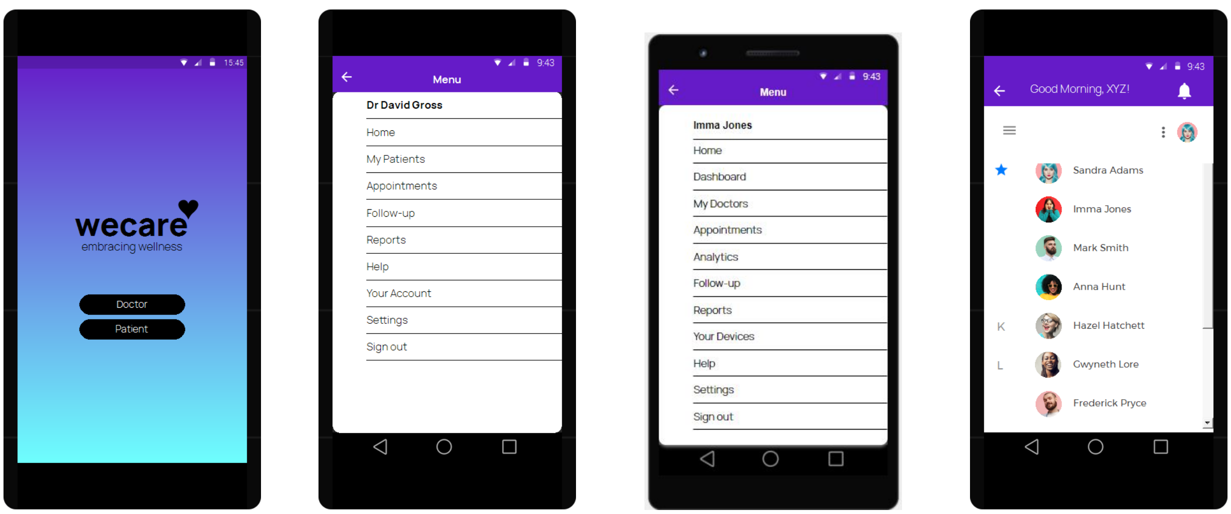
Figure 4.13: WeCare System Chat bot Assistant

of the viral infection without overloading medical facilities. The current work provides a preliminary system that is able to utilize self-declared patient symptoms and biological data to determine the risk of the patient being infected with COVID-19. A thorough search was made for a dataset that relates different symptoms people show with COVID-19 positive and negative cases as the class labels, however such a dataset is not publically available. This can be attributed to the fact that the COVID-19 is still a new disease and this data is yet to be collected for it. As a result, an artificial dataset was created using a fuzzy inferencing with a set of rules based on different symptoms as described next.



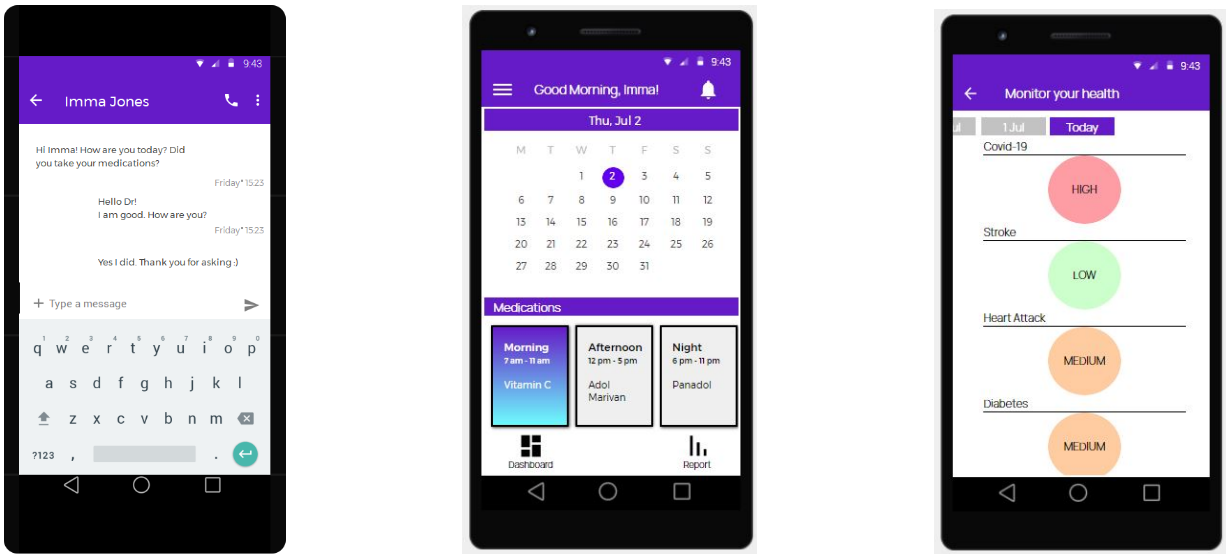
(a) Dashboard (b) Insert Data (c) Reports (d) Chat Bot

Figure 4.14: WeCare Smartphone Application Functionalities



(a) Login Screen (b) Doctor's Menu (c) Patient's Menu (d) List of Patients

Figure 4.15: Different Access for Doctors and Patients in WeCare Smartphone Application



(a) Chat with Patient (b) Patient Home Page (c) Analytics

Figure 4.16: Additional WeCare Smartphone Application Functionalities

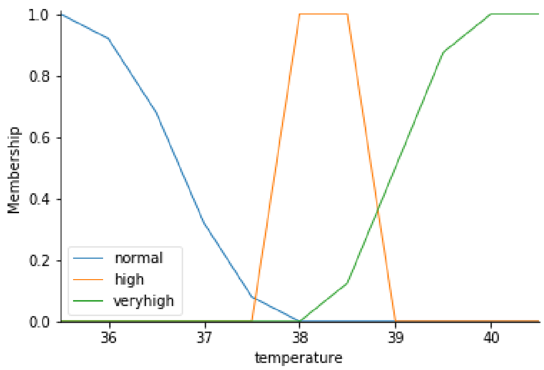
4.3.1 Rule-based Dataset Generation

All data points involved in the WeCare prototype are symptoms of COVID-19. However, according to [45] and [46], fatigue, dry cough, and high fever are the most prevalent symptoms in patients diagnosed with COVID-19 infection. These 3 biological parameters were used as the base of the system and fuzzy membership functions were defined for each.

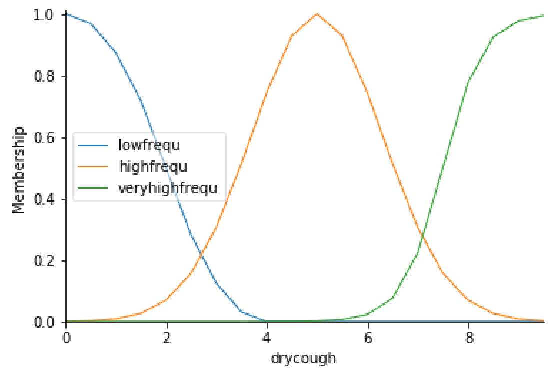
All membership functions created are shown in Figure 4.17 and are defined as follows:

- *Temperature* has three fuzzy sets: normal (N), high (H), and very high (VH). The support set is between 35.5 and 41 degrees Celsius with increments of 0.5. N is defined using a Z-membership function that reaches 0 at 38 degree. H is defined using a trapezoidal membership function between with its vertices at 37.5, 38.0, 38.5, and 39.0. VHT is defined using an S membership function and starts increasing from around 38 degrees, achieving a membership value of 1 at 39.5.

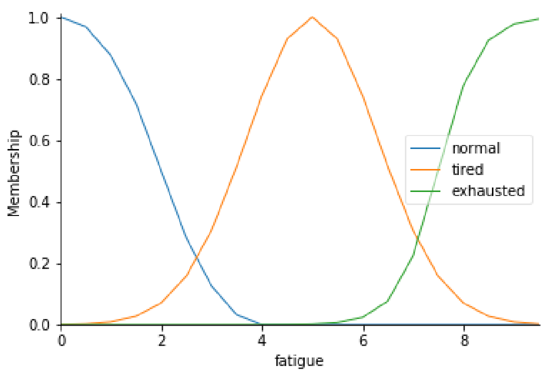
- *Dry cough* is a categorical variable, it was created to have a support set between 0 and 10 (with an increment of 0.5), expressing the severity and frequency at at which the patient is dry coughing. Three membership states are created: low frequency (LF), high frequency (HF), and very high frequency (VHF). LF is defined by a



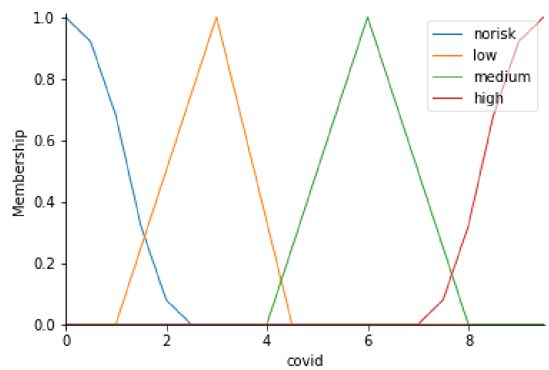
(a) Temperature Membership Function



(b) Dry Cough Membership Function



(c) Fatigue Membership Function



(d) COVID-19 Membership Function

Figure 4.17: Temperature, Dry Cough, Fatigue, and Covid-19 Membership Functions

Z membership function which reaches 0 at 4.0. HF uses a Gaussian membership function with a mean of 5.0 and a spread (σ) of 1.3. VHF is defined using a S membership function which starts rising from 7.0 and peaks at 9.5.

- *Fatigue* expresses how tired the patient is and is created to be very similar to dry cough. Both fuzzy variables have the same support set and use the same membership functions but with different names. Three fuzzy states were set at normal (N), tired (T), and exhausted (E) which are defined just like LF, HF, and VHF respectively in dry cough.

- *Covid* is the consequent of inferencing on the three membership variables of temperature, dry cough, and fatigue. It is expressed using 4 levels which give the risk level of a patient being infected with COVID-19. A no risk (NR) level uses a Z membership function which reaches 0 at 2.5. Low risk (LR) and Medium (MR) risk use triangular membership functions which start at 1 and 4, peak at 3.0 and 6.0 and reach 0 at 4.5 and 8.0 respectively. High risk (HR) uses a S membership function which starts at 7 and reaches its maximum value (1.0) at 9.5.

It is important to note that the fuzzification of the categorical variables was created to be able to use fuzzy inferencing. During inferencing, the following values were used for the different levels the user chooses from:

- None: 0.0
- Mild: 3.0
- Moderate: 6.0
- Severe: 9.0

A rule base was created using some set rules from combinations of the levels of the fuzzy membership variables. Some of the rules used in inferencing are:

1. if Temperature is VH AND Dry Cough is VHF AND Fatigue is E, THEN Covid is HR
2. if Temperature is N AND Dry Cough is LF AND Fatigue is N, THEN Covid is NR
3. if Temperature is N AND Dry Cough is LF AND Fatigue is T, THEN Covid is LR
4. if Temperature is N AND Dry Cough is HF AND Fatigue is T, THEN Covid is MR

Using these rules for inferencing with Mamdani, a sweep was made over all possible combinations of values for the three fuzzy variables and the consequent was obtained. All the input and output was then saved to be used to train and test machine learning algorithms for the prediction of COVID-19. The levels of the fuzzy variable of covid are then to be considered as the classes that the machine learning module should classify. The dataset created was then analyzed to get some insights of the data distribution. A histogram of the dataset is shown in Figure 4.18. The class distributions are then:

- No Risk: 448 samples
- Low Risk: 5552 samples

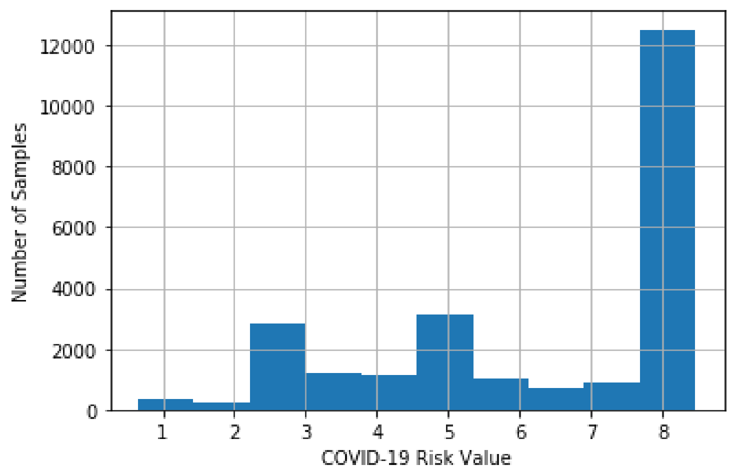


Figure 4.18: Histogram of created dataset for COVID-19

- Medium Risk: 6144 samples
- High Risk: 11856 samples
- Total Samples: 24000

In conclusion, fuzzy logic inferencing was used only to create an artificial data set for the risk level of COVID 19 with the three symptoms that are most prevalent in patients who tested positive; fatigue, dry cough, and fever.

4.3.2 Machine Learning for COVID-19 Prediction

Evaluation Metrics

Before discussing the use of machine learning (ML) for predicting the risk level of COVID-19, a quick overview of performance metrics is necessary. First, the confusion matrix is one of the methods used to evaluate the classification performance of a model. With a two class classification problem (0 and 1), the confusion matrix (shown in Figure **??** contains important four terms that are used to derive many evaluation metrics:

- *True Positives (TP)*: A count of the number of samples predicted to belong to class 1 where the true class is also 1.

- *True Negatives (TN)*: A count of the number of samples predicted to belong to class 0 where the true class is also 0.
- *False Positives (FP)*: A count of the number of samples predicted to belong to class 1 while their true class is 0.
- *False Negatives (FN)*: A count of the number of samples predicted to belong to class 0 while their true class is 1.

Using these terms, the following metrics can be computed:

- *Accuracy*: A ratio of the number of correct predictions made over all predictions made.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- *Precision*: Ratio of correctly predicted positive samples to the total number of predicted positive samples.

$$Precision = \frac{TP}{TP + FP}$$

- *Recall*: Ratio of correctly predictive positive samples to the total number of positive samples in the dataset. It is also sometimes called the True Positive Rate (TPR).

$$Recall = \frac{TP}{TP + FN}$$

- *F1 Score*: Gives a ratio between the precision and recall by calculating their harmonic mean. F1 score accounts for imbalanced datasets better than the accuracy metric.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

While there are more evaluation metrics that are used in comparing performance of different ML models, the ones mentioned here are enough for the upcoming discussion of results. The binary (two-class) metrics can be extended to multiple classes by considering each class separately. For each class, the samples classified as that class are assigned to the positive class (1), while all other classifications are assigned to the negative class (0).

From there, the evaluation metrics are calculated for each class separately. These metrics provide better insights on the performance of the model on each of the classes than the overall accuracy.

Implementation of Machine Learning Methods for COVID-19 Prediction

After the dataset was created. It was divided into 80% for training and 20% for testing. Several machine learning algorithms were trained and tested including k-Nearest Neighbor (kNN), linear Support Vector Machines (SVM), SVM with a gaussian kernel, and decision trees (DT). After analysing the results based on several metrics including accuracy, precision, recall, and F1-score, decision trees and kNN were found to outperform the rest of the methods. The overall accuracy of DT is 98.02% while that of kNN is 97.31%. Since the dataset is imbalanced, the classifier could be biased towards the class with large number of samples. Therefore, other evaluation metrics are considered to better assess the performance of the models created. The results for the kNN and DT models are summarized in Table 4.2 which shows how DT perform equally as good or slightly better than kNN in all classes on all metrics. Although the difference in accuracy between kNN and DT is small, the precision, recall, and f1-score of kNN on 3 of the classes is lower, leading to more frequent mistakes when presented with data from these classes. Recall that the artificial dataset was created using fuzzy inferencing; which, in essence, is a set of if-then rules applied to fuzzy membership values. Likewise, decision trees work using a set of nested if-then rules, which can explain their superior performance over the remaining methods.

It can be seen that the results are superior and may be unrealistic in a real dataset. Therefore, a few steps can be taken to provide a more robust COVID-19 predictor. First, some noise should be injected in the dataset to allow the algorithm to generalize better. Moreover, cross validation should be applied to accommodate for different train-test splits and give a better view of the overall performance of the algorithms.

Table 4.2: Confusion Matrix for kNN and Decision Tree Algorithms

	kNN			**Decision Trees**		
Class	Precision	Recall	F1-score	Precision	Recall	F1-score
High Risk	0.99	0.99	0.99	0.99	0.99	0.99
Medium Risk	0.95	0.95	0.95	0.96	0.96	0.96
Low Risk	0.96	0.96	0.96	0.97	0.97	0.97
No Risk	0.93	0.93	0.93	1.00	0.95	0.98

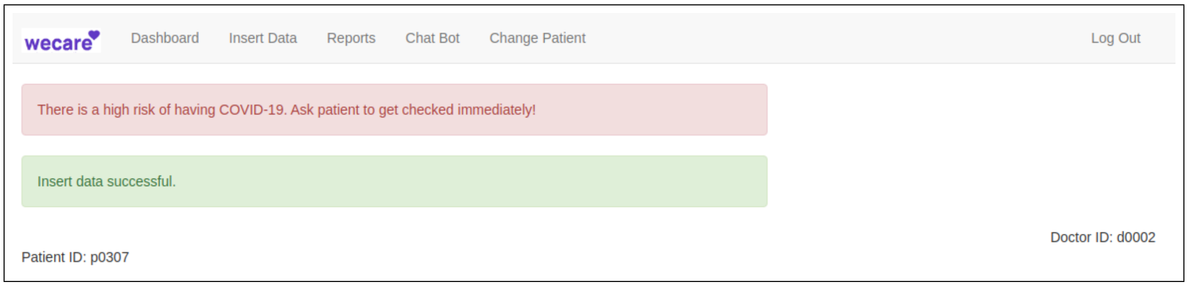
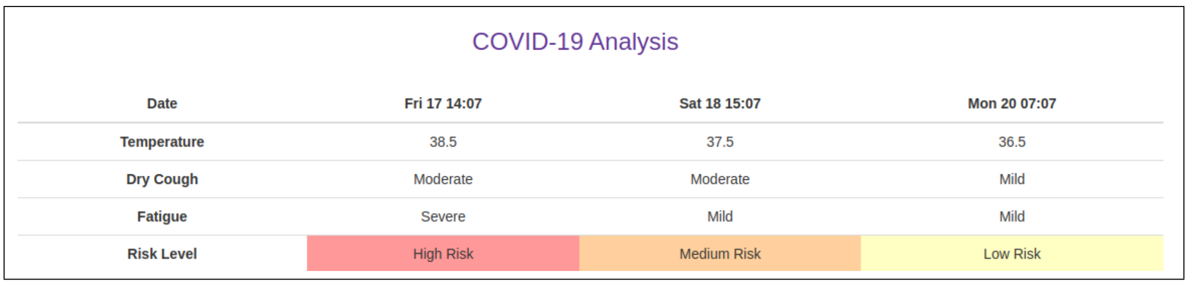


Figure 4.19: Instant Analytics for COVID-19 on WeCare Web Interface



COVID-19 Analysis

Date	Fri 17 14:07	Sat 18 15:07	Mon 20 07:07
Temperature	38.5	37.5	36.5
Dry Cough	Moderate	Moderate	Mild
Fatigue	Severe	Mild	Mild
Risk Level	High Risk	Medium Risk	Low Risk

Figure 4.20: Three-day COVID-19 Risk Analytics on WeCare Dashboard

A decision tree was trained on the data and saved to be used in the Analytics module within the fog layer of the WeCare system. The user gets direct feedback from the analytics module whenever they enter all three required physiological parameters as shown in Figure 4.19. Moreover, the system also provides a summary of the risk level on the patient over the past 3 days in the *Dashboard* page as shown in Figure 4.20

4.4 Summary

This chapter proposes the WeCare system for remote health monitoring and describes its main capabilities and building blocks. A prototype is built following the framework described in Chapter 3 thus emphasizing the practical mode of operation of different modules. A web interface was created to allow healthcare professionals to interact with the system to add new patient data through either form filling or by chatting with a conversational agent. Doctors can also make use of the powerful visualization tools presented in the system to understand the patient's status better and provide more personalized treatment. Adding to that, the system connects to a NoSQL database in the cloud for scalable data storage.

WeCare tracks four continuous and 8 categorical physiological parameters, which constitute the symptoms of COVID-19. From these symptoms, a dataset was created through rule-based fuzzy inferencing on the three most prevalent symptoms. Using the artificial dataset, different machine learning models were trained and the best was used as the seed for an expandable analytics engine. All in all, WeCare illustrated a sample system which builds on the flexible framework discussed in Chapter 3 to provide a versatile solution for RHM. Chapter 5 uses the same framework to design a system for ambient assisted living of older adults with advanced human-machine interaction and social robots.

Chapter 5

Ambient Assisted Living for Older Adults

Ambient Assisted Living (AAL) involves combining smart objects, such as sensors, smartphones, actuators, and software, all connected in the IoHT to provide healthcare services and assisted living to older adults. As mentioned, AAL for older adults has been tackled previously in literature where various agents were combined to facilitate better and more independent living for older adults. Other works considered the use of social robots to provide older adults with a task-achievement agent, a social partner, as well as maintain social contact with their family and friends. Moreover, the great benefit behind AAL, social robots, and the combination of both for the care of older adults was also discussed in detail. In particular, several studies concluded that the more natural the HMI method is, the more usable and acceptable users find the system.

In this chapter, a system designed for older adults is showcased which combines recommendations from previous studies in one solution; cultivating all their benefits. It utilizes the same general structure described in Chapter 3 with a wearable sensor to obtain heart rate measurements, a robotic system for task achievement, as well as a NLU module coupled with virtual assistants to provide speech interaction capability. In particular, the current work has four main functionalities:

- Provide a way of natural interaction through speech
- Remind older adults of medications and appointments
- Provide a flexible architecture that allows users to easily add other smart and robotic agents in an IoHT environment

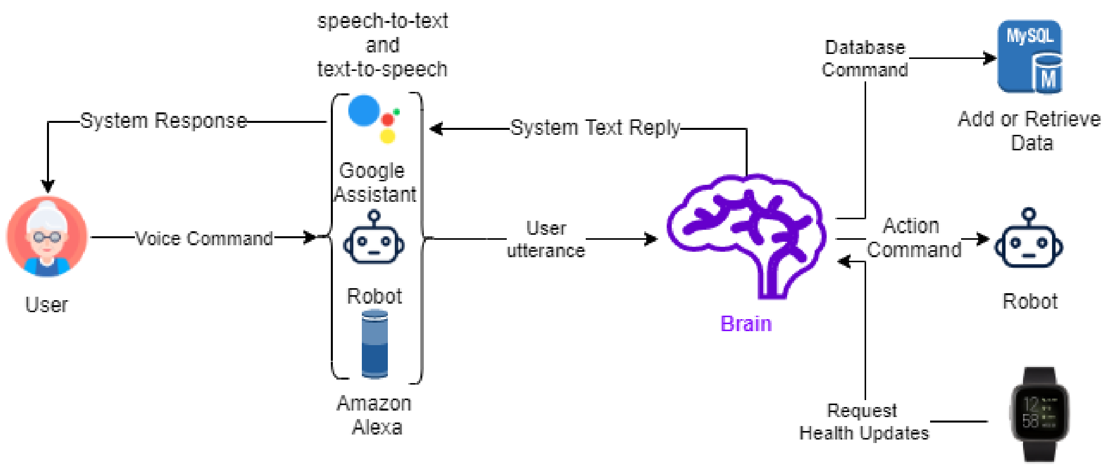


Figure 5.1: Elderly Care System

- Control robots to move, autonomously navigate, and retrieve objects

A deeper look into the system is presented in the current chapter while highlighting its various capabilities.

5.1 System Architecture

The elderly care system is created to provide older adults with a way to track their health and medications and be able to request several tasks from robotic systems. The system described here is an all-in-one solution that yields the power of pre-existing smart devices and brings them together to provide the elderly with a flexible, seamless means of HMI to serve their needs in the most natural way to humans; speech [47]. The system also uses commercial off-the-shelf wearables, virtual assistants, and robots. These COTS devices eliminate the need to create the devices from scratch, thus harvesting the power of well-established implementations while augmenting the intelligence of the system by integrating various agents.

5.1.1 Central Brain

The core of the system in this implementation is called the *Brain*. The Brain contains all the fog layer modules for request understanding and task achievement and is linked to

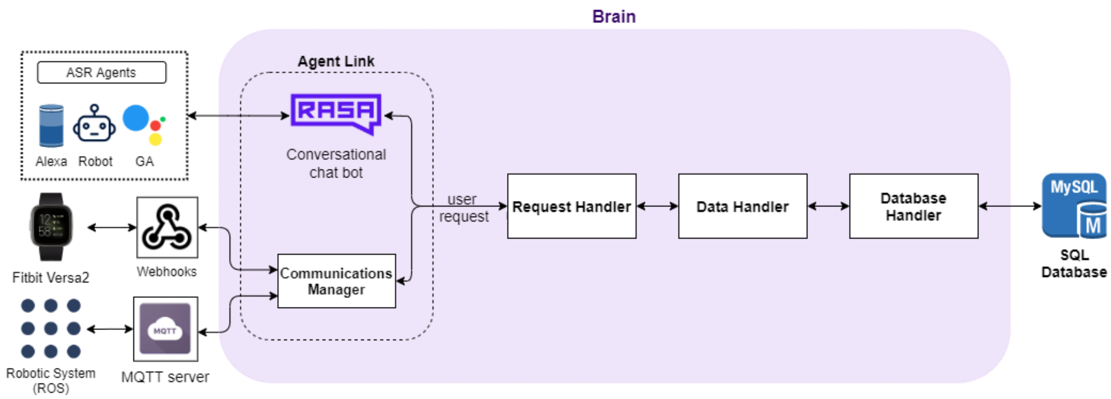


Figure 5.2: Internal Modules in Brain of the Elderly Care System

different agents to receive or issue commands as shown in Figure 5.1. The figure depicts how the system is linked to different agents and communicates with the older adults and the various system agents.

A more detailed view of the internal modules of the Brain is illustrated in Figure 5.2. Great resemblance is noticeable with the architecture used in the WeCare system for RHM as the system being discussed utilizes the same versatile framework, which is the main contribution of the current work. The agent link handles all external interactions with the IoHT environment. For example, it connects with wearables for real-time health monitoring and the robotic system to send navigation commands. In addition, the agent link contains the Rasa agent which handles text utterances to extract the user's intent and send the user's request to the request handler. The communication manager sub-module allows the agent link to communicate using MQTT and REST protocols. The agent link is capable of distributing the requested action to the appropriate agent by storing the capabilities of each agent locally; thus the remaining system modules do not interfere in the details of task achievement. Needless to say, the provided architecture allows adding or modifying system capabilities separately.

5.1.2 Speech Interaction

A concern that arises when high tech solutions are offered to older adults is the technological barrier that exists. As stated in [48], a fraction of the aging population are uncomfortable interacting with smartphones and touch screens. Consequently, it would be of great benefit if there is a more natural interaction method that requires minimum user training. Speech has been suggested by several studies as the most promising way for older adults as it is the most natural way for interaction [49, 34, 35, 31]. In light of the great advancements in virtual assistants like Alexa and Google Assistant (GA) and their powerful automatic speech recognition (ASR) capabilities, making use of such commercial solutions becomes a promising idea. In fact, previous work tackled the idea of integrating Alexa in a multi-robot system [50] and showed it could facilitate natural control by humans in a robotic environment. The current work harvests the advanced capabilities of speech-to-text engines in GA and Alexa and uses a central chat bot system for language understanding and dialogue management. Centralizing NLU and dialogue management allows the system to provide the same functionality across platforms without the need to develop and maintain the system separately for each of virtual assistant to accommodate for their API (application programming interface). Consequently, the solution proposed becomes generic and applicable to the largest fraction of possible users.

The flow of data once the user interacts with the system through speech is illustrated in Figure 5.1. The elderly care system receives the speech command from the user through a virtual assistant (GA or Alexa) which uses ASR to translate the signal to text and passes it to the brain of the system. The message is received at the agent link which processes the text utterance using the conversational chat bot based on Rasa framework. The reader is referred back to section 4.1.2 where Rasa was explained in more detail. The conversational chat bot is able to understand the user intent and extract useful information needed to achieve the user request. According to the user's intent, a specific action is then requested from the request handler.

5.1.3 Autonomous Robotics for Task Achievement

Social robots are being utilized for care of older adults where they can help connect them with their families or care givers, move objects, and help those with low mobility to move around and interact with their environment. In the current implementation, the agent link communicates through MQTT with the robotic system. Due to its speed and efficiency, MQTT was found to be a better communication protocol in linking robotics to IoT environments [51].

The robotic system was implemented using a collection of open source software libraries called the Robotic Operating System *ROS*[1] [52] and an open source robotics simulator offered by the Open Source Robotics foundation; Gazebo[2]. ROS is the most utilized software for robotics and it contains a large number of packages for all kinds of applications

[1] https://www.ros.org/
[2] http://gazebosim.org/

and a huge support by both the academic and industrial communities. ROS utilizes a publish-subscribe methodology to transfer messages. The ROS network is a group of nodes which communicate through topics. Each topic is similar to a billboard on which publishers send messages and subscribers can see and read all posted messages A single node can subscribe and publish to several topics at the same time. Similarly, several robots can communicate by registering their topics and nodes on the same ROS network (large similarity to MQTT prototocol). A ROS master node connects to the MQTT broker and listens from the agents link in system's Brain. According to the nature of the task, the master is able to direct the request to the appropriate robotic agent. This allows the system to exhibit independence between the different agents. In other words, the central Brain of the system do not need to know how the action is actually performed by the robotic system, and the robot does not get involved in understanding the user's request.

5.1.4 Smart Wearable

Smart wearable sensors allow non-invasive monitoring of real-time health parameters. The use of these wearables allow for continuous monitoring of health status and detection of sudden changes that may indicate a dangerous situation to the older adult. For example, acceleration can be used to detect falls, a sudden drop in heart rate may mean indicate fainting, and ECG data can be used to detect early signs of a heart attack.

Wearables first entered the scene with health trackers for fitness purposes. Recently, more interest has gone into wearables being utilized for remote health monitoring. For example, the blood glucose monitor Dexcom [10], the Ora Ring[3] that measures body temperature, and the Apple Watch [11] which has recently received approval for their ECG to be used for clinical purposes. Most of these products however work with proprietary software that works only with their smartphone application. The current system is developed to be be able to receive data from various wearables, providing a full picture of the user's health.

In this implementation, a Fitbit Versa 2[4] is used to transmit real-time HR data. To integrate the Fitbit, a widget was created using JavaScript that receives the data from the Fitbit Versa 2 and transmits it using REST API to elderly care system. The agents link receives the data and sends it as a request to add data to the request handler in a similar method to that described previously in the WeCare system.

[3] https://ouraring.com/

[4] https://www.fitbit.com/us/products/smartwatches/versa

5.2 Implementation and Results

The system was implemented and integrated with Google Assistant (GA) as the virtual assistant, a robotic system and a Fitbit Versa 2 as a smart wearable for health monitoring. To send the parsed text from GA to the system, a google action was created. The google action gets triggered by the user saying (or typing) "*I want to speak to my care system.*" to GA. From that point until the user exits the system, the user messages are all processed by the system as described earlier. The conversational agent is able to recognize the intents shown in Table 5.1. The second column of the table shows a small sample of the training data that the REDP model uses. The Rasa system is able to generalize to understand different ways the user may request different actions. In the current section, a few examples of how the system understands and responds to user intents are shown. The types of dialogues the user can engage with the system in are classified into two categories: static and dynamic commands. Static commands are mainly concerned with health updates and requests to set up appointments and reminders. On the other hand, dynamic commands include requests to the robotic system to move around or autonomously navigate to a location.

5.2.1 Static Commands

As mentioned, the interaction through GA starts with the user triggering the google action. Once the system is triggered, all user messages are processed using the described conversational chat bot which extracts the intent of the user and entities in the message (as well ad the confidence of each classification). Table 5.2 showcases an interaction between a user and the system with static commands. It can be seen from the provided example how the system is capable of handling a natural conversation that goes beyond just achieving the task and adds a social nature to the interaction. In addition, the system is capable of understanding natural language without scripted commands, removing the need for training or memorization to utilize it.

In the interaction shown in Table 5.2, the user asks the system to add a reminder for a new medication that is to be taken on Sunday. The system responded by requesting the time of the medication. This shows how the system deals with requests that need several entities to be found before execution. The system detects which parameters are missing and asks the user to provide them in a form filling manner while maintaining a natural conversation with the user. Figure 5.3 depicts how the system is capable of looping through the parameters required and asking for them until they are all provided before executing the action.

Table 5.1: Sample Training Data for Conversational Chat Bot

Intent name	**Sample Utterance(s)**
greet	Hello/Hi/Heyy bot
goodbye	goodbye/talk to you later
affirm	yes/that sounds good/indeed
ask_med	What medications do I have on Monday/When do I have to take Tylenol
ask_state	How are you today/How have you been
deny	No/Wrong/I don't think so
inform_med	the medication is called Tylenol/that would be at 9 am/on both Sunday and Friday
health_enquiry	how is my health today/what is my current heart rate?
mood_great	I am great/great/Feeling well
mood_unhappy	not feeling well/I'm feeling terrible/not so good
move_cmd	move forward 1 meter/go back 2 meters/make a right turn
navigate	go to the kitchen/meet me in the living room
stop	exit now/stop what you are doing/cancel
thanks	thank you/thanks/you are great help
ask_weather	what is the weather like today?/what is the temperature now?
store_location	save this place as the living room/store your current position as the kitchen/please remember this place as the dining table

Table 5.2: Complicated Interaction with Multiple Action Requests

User Utterance	**Extracted Intent: Confidence**	**Entities**	**Response**
Hello	*greet*:1.0		Hey!
Remind me to take alpha on Sunday	*request_medication*: 1.0	med_name: alpha, Date: Sunday	Can you tell me what time, including a.m. or p.m?
Set that at 11 am	*request_medication*: 1.0	med_name: alpha, Date: Sunday, Time: 11 am	Ok I will remind you of alpha on Sunday at 11 am. Do you need me to do something else?
What is my health status today?	*health_enquiry*: 0.94		Your heart rate is currently 70.
Ok, what pills do I have to take on Sunday?	*ask_med*: 0.97	Date: Sunday	You have to take alpha at 11 am and beta at 3 pm.
No, thanks	*thanks*: 0.97		Glad I could help!

Once all entities required to add the medication reminder are extracted, the agent link sends the request to the request handler to add the reminder to the database. The request handler forwards the request to the data handler and from there to the database handler which puts the data in the appropriate format for the database and adds the data. In this system, a local *MySQL*[5] database is utilized to save the user information and medications. Although the request handler can directly link to the database, the mentioned route is used to ensure generality and flexibility. It also shows how the same framework proposed in Chapter 3 can be tailored to various applications.

The same route in module functionalities and communication is followed when a user asks for an update about their health status or enquires about their medications; The conversational agent understands the user intent, sends the request to the request handler, which asks for the formatted data from the data handler. The database handler communicates with the database to retrieve the required data and convey it to the data handler. The system responds with the latest HR reading available when the user asks for an update of their health. The user can also enquire about the medications they need to take on a

[5] https://www.mysql.com/

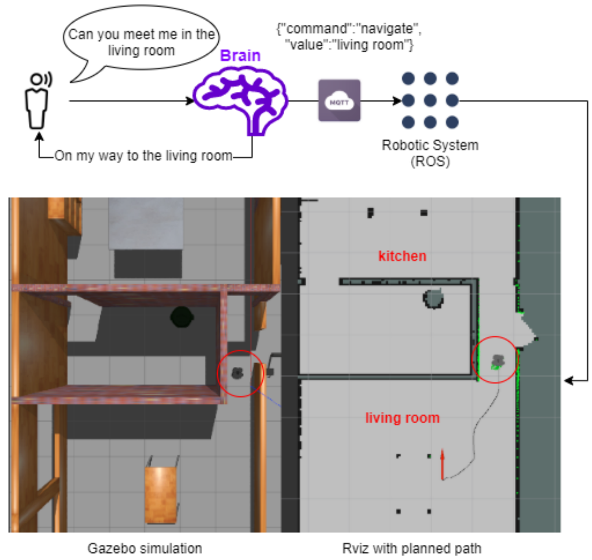


Figure 5.3: Requesting Missing Entities in User Request

Figure 5.4: Speech Interaction for Robot Navigation

specific day or when they have to take a particular pill. The system uses the data stored in the MySQL database to convey this information back to the user. Both requests and their classified intent, extracted entities, and the system response are shown in Table 5.2.

5.2.2 Dynamic Commands

The capabilities of the system extend beyond maintaining a natural conversation and saving/retrieving information to a robotic system capable of performing dynamic commands and making changes in the older adult's environment. Most robotic systems available that are powered by a speech system are able to understand only scripted commands that require memorization by its user [34, 50]. This calls for familiarity with the system as well as some training. As mentioned in previous studies, a system would be more acceptable and usable by older adults if these requirements are relaxed. This is tackled in the current system through the natural speech interaction module explained previously. Table 5.3 shows a sample of the utterances used to train the REDP model on different user intents for the robotic system and the extracted (structured) message the system transmits forward to the request handler. Afterwards, the request handler sends the action request to the agent link and the ROS master node.

Table 5.3: Flexible Interaction Without Preset Commands

User utterance	Extracted Command
move forward 1 meter/ go straight/ go forward/ move to the front	{command: "go", value: "1"}
turn 90 degrees/ rotate 90/ turn to the right/ turn anticlockwise	{command: "turn", value: "90"}
go backward 3 meters/ take 3 steps back/ move back 3 meters	{command: "go", value: "-3"}
go to the kitchen/ come meet me in the kitchen/ navigate to kitchen/ can you please go to the kitchen	{command: "navigate", value: "kitchen"}

The system was thoroughly was tested where the robot is requested to move around and navigate to different stored locations in the map. The user can also move a robot to a specific location and then request the system to store the location with a specific name (using the store_location intent), thus enabling the older adult to save locations of interest such as the front/back door or a place they usually keep their personal belongings for faster task-achievement and a more effective human-robot collaboration. Figure 5.4 shows a simulated robot (turtlebot3) navigating to the living room after the user says "*Can you meet me in the living room*" and the conversational agent successfully extracts the user's intent and sends the command to the robotic system.

5.3 Summary

In the current chapter, a system was created for care of older adults in an AAL environment based on the IoHT. Besides smart devices and sensors, the system created also integrates robotics to provide means of task accomplishment. The prototype presented builds on the multi-layer framework proposed in Chapter 3. It provides the user with continuous HR monitoring, natural speech understanding for HMI using virtual assistants, as well as a ROS-based autonomous robot. Two different types of commands were tested with the system to elaborate on its functionalities and the system was shown to naturally understand user's utterances and perform the required tasks. This implementation of the Brain's architecture incorporatesvheterogeneous agents and actions which illustrates the flexibility of the system. The next chapter concludes the work presented in this thesis and highlights the main contributions. It also provides some future areas of research to explore.

Chapter 6

Conclusions and Future Work

This chapter provides a brief summary of the thesis proposition and highlights its key results and contributions in Section 6.1 Future research directions are then outlined in Section 6.2.

6.1 Conclusion

The current work investigates the application of IoHT in two applications of huge interest to researchers: remote health monitoring (RHM) and ambient assisted living (AAL). After extracting recommendations from literature and identifying gaps in previous studies, the thesis proposed a framework with three main layers: the device layer comprised of smart devices, the fog layer which offers local data processing, analytics, and storage, and the cloud layer for unlimited storage size and big data analytics. Detailed description of the software modules is presented with the role of each module and how it helps achieve the ultimate goal of creating a flexible system applicable to different domains.

To support the contributions of the proposed framework, it is implemented in two major applications of IoHT. First, the framework is applied to RHM in the prototype WeCare system. WeCare illustrates how physiological parameters can be collected through a web interface as well as a chat bot. It uses the fog layer for real-time analytics and a cloud database for storage. WeCare provides real-time predictions of the risk level of COVID-19. The three main symptoms of COVID-19 (fatigue, dry cough, and body temperature) are used to create an artificial dataset based on a fuzzy logic system with four risk levels as the classes; No Risk, Low Risk, Medium Risk, and High Risk. Different classical machine

learning algorithms are then applied and decision trees where found to outperform the rest with an overall accuracy of 98.02%. The WeCare web interface allows healthcare professionals to enter data about the patient and view summaries of the patient's health status, and it provides warnings based on the COVID-19 predictive model.
Moreover, the framework is also applied to AAL in a platform for care of older adults which integrates a natural speech interaction interface, a wearable sensor, and robotic agents. The elderly care platform builds on recommendations from previous literature whilst using the same framework proposed in this thesis. The implementation illustrates how the system provides natural HMI and integrates various agents. It also provides the older adults with the ability to control a robotic agent in their environment using natural speech. These two systems shed light on how the loosely coupled software modules can be combined to accommodate for varying system requirements.

Conclusively, this thesis provided an all-in-one solution for real-time health monitoring. Therefore, the main thesis contribution is the proposition of a framework that:

1. integrates heterogeneous smart agents for health monitoring, data analysis, and task accomplishment.

2. provides real-time response to emergencies (fog layer) as well as big data analytics (cloud layer).

3. offers a modular and flexible architecture to accommodate for various applications as well as adding/modifying functionalities easily.

4. is easily expandable with new technologies and advanced smart devices as they emerge. In other words, the framework allows new agents and system modules to be added seamlessly without affecting its functionalities.

5. isolates different system functionalities to allow personalizing the system to user needs.

6. allows users to access different system functionalities through multiple modes of communication including speech interaction for intuitive and natural HMI.

6.2 Future Work

The work done in this thesis offers a generic framework for IoHT environments and implements two systems for RHM and AAL. Both systems implemented here are seed prototypes, and therefore some future work directions can be suggested.

The capabilities of both systems can be improved in several ways. For starters, the RHM system should be augmented with wearable sensors for real-time data collection of different physiological parameters. Using the real-time data streams, models can be created to detect abnormalities in the readings and send warnings to the user and their doctor. In addition, different authorization levels need to be implemented and the system should be able to provide the appropriate interface and capabilities. Similarly, the elderly care system should incorporate robotic systems capable of more complex behavior such as fetch-and-carry. Cameras and other environmental sensors should also be used to identify the older adult's activities and detect dangerous situations.

Using the fog layer and the cloud layer could be restructured to follow a lambda architecture which provides the same functionalities as the proposed fog-cloud combination. The lambda architecture offers scalable and generic data processing which accommodates faults in data transmission. The speed layer can be used for real-time analytics, the serving layer can present reports and summaries to users, and the batch layer can be used for big data analytics. The lambda architecture will also allow the use of an event bus where all agents can push their requests and get their tasks done in order without overloading the system when a huge number of agents are present.

Besides adding extra functionalities, the main framework does not tackle two aspects that are crucial to providing it a solution with its implementation with users:

1. *Security and Privacy*: The framework needs to ensure secure transfer of the patients' personal information as well as their physiological data and history of illness. Extra measures need to be put in place as well to ensure data is not transmitted from the cloud unless the user has the appropriate credentials to access the data. Privacy concerns are of utmost importance in healthcare applications due to the sensitivity of the data being held, therefore encryption methods, levels of security, and other measures should be taken to ensure the user's data cannot be hacked.

2. *Communication Technologies*: Communication between different parts of the system is important as it affects the bandwidth requirements and speed of data transfer. In the proposed implementations, suggestions from previous literature as well as constraints in the available agents guided the choice of BLE and WiFi for communication. However, elaborate studies need to be done on the various communication technologies to compare their transfer rate, power consumption, and applicability to different smart devices. This will highly affect the efficiency and usability of the system.

References

[1] World Health Organization, *World report on ageing and health.* World Health Organization, 2015.

[2] Fortune Business Insights, "Telehealth market size, share, and industry analysis," 2020.

[3] P. Maia, T. Batista, E. Cavalcante, A. Baffa, F. C. Delicato, P. F. Pires, and A. Y. Zomaya, "A web platform for interconnecting body sensors and improving health care.," in *MoWNet*, pp. 135–142, 2014.

[4] S. Majumder, T. Mondal, and M. J. Deen, "Wearable sensors for remote health monitoring," *Sensors*, vol. 17, no. 1, p. 130, 2017.

[5] G. Petrović, V. Dimitrieski, and H. Fujita, "Cloud-based health monitoring system based on commercial off-the-shelf hardware," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 003713–003718, IEEE, 2016.

[6] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven iot ehealth: Promises and challenges of iot in medicine and healthcare," *Future Generation Computer Systems*, vol. 78, pp. 659–676, 2018.

[7] A. M. Longva and M. Haddara, "How can iot improve the life-quality of diabetes patients?," in *MATEC Web of Conferences*, vol. 292, p. 03016, EDP Sciences, 2019.

[8] K. J. Kappiarukudil and M. V. Ramesh, "Real-time monitoring and detection of" heart attack" using wireless sensor networks," in *2010 fourth international conference on sensor technologies and applications*, pp. 632–636, IEEE, 2010.

[9] P. Siirtola, H. Koskimäki, H. Mönttinen, and J. Röning, "Using sleep time data from wearable sensors for early detection of migraine attacks," *Sensors*, vol. 18, no. 5, p. 1374, 2018.

[10] Dexcom Continuous Glucose Monitoring, “Benefits of CGM,” 2020.

[11] Apple Newsroom, “ECG app and irregular heart rhythm notification available today on Apple watch,” 2018.

[12] W. H. Organization”, “Who coronavirus disease (covid-19) dashboard,” 2020.

[13] D. R. Seshadri, E. V. Davies, E. R. Harlow, J. J. Hsu, S. C. Knighton, T. A. Walker, J. E. Voos, and C. K. Drummond, “Wearable sensors for covid-19: A call to action to harness our digital infrastructure for remote patient monitoring and virtual assessments,” *Frontiers in Digital Health*, vol. 2, p. 8, 2020.

[14] T. Mishra, M. Wang, A. A. Metwally, G. K. Bogu, A. W. Brooks, A. Bahmani, A. Alavi, A. Celli, E. Higgs, O. Dagan-Rosenfeld, *et al.*, “Early detection of covid-19 using a smartwatch,” *medRxiv*, 2020.

[15] J. J. Rodrigues, D. B. D. R. Segundo, H. A. Junqueira, M. H. Sabino, R. M. Prince, J. Al-Muhtadi, and V. H. C. De Albuquerque, “Enabling technologies for the internet of health things,” *Ieee Access*, vol. 6, pp. 13129–13141, 2018.

[16] P. K. D. Pramanik, A. Nayyar, and G. Pareek, “Wban: Driving e-healthcare beyond telemedicine to remote health monitoring: Architecture and protocols,” in *Telemedicine Technologies*, pp. 89–119, Elsevier, 2019.

[17] F. Amirabdollahian, R. op den Akker, S. Bedaf, R. Bormann, H. Draper, V. Evers, G. J. Gelderblom, C. G. Ruiz, D. Hewson, N. Hu, *et al.*, “Accompany: Acceptable robotics companions for ageing years—multidimensional aspects of human-system interactions,” in *2013 6th International Conference on Human System Interactions (HSI)*, pp. 570–577, IEEE, 2013.

[18] Y. Quintana, F. Darren, B. Crotty, J. Ruchira, E. Kaldany, M. Gorenberg, L. Lipsitz, D. Engorn, J. Rodriguez, A. Orfanos, *et al.*, “Infosage: Supporting elders and families through online family networks,” in *AMIA Annual Symposium Proceedings*, vol. 2018, p. 932, American Medical Informatics Association, 2018.

[19] P. Rashidi and A. Mihailidis, “A survey on ambient-assisted living tools for older adults,” *IEEE journal of biomedical and health informatics*, vol. 17, no. 3, pp. 579–590, 2012.

[20] A. Abtoy, A. Touhafi, A. Tahiri, *et al.*, “Ambient assisted living system’s models and architectures: A survey of the state of the art,” *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 1, pp. 1–10, 2020.

[21] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, "mhealthdroid: a novel framework for agile development of mobile health applications," in *International workshop on ambient assisted living*, pp. 91–98, Springer, 2014.

[22] A. Rashed, A. Ibrahim, A. Adel, B. Mourad, A. Hatem, M. Magdy, N. Elgaml, and A. Khattab, "Integrated iot medical platform for remote healthcare and assisted living," in *2017 Japan-Africa Conference on Electronics, Communications and Computers (JAC-ECC)*, pp. 160–163, IEEE, 2017.

[23] L. Syed, S. Jabeen, S. Manimala, and A. Alsaeedi, "Smart healthcare framework for ambient assisted living using iomt and big data analytics techniques," *Future Generation Computer Systems*, vol. 101, pp. 136–151, 2019.

[24] G. Diraco, A. Leone, and P. Siciliano, "A radar-based smart sensor for unobtrusive elderly monitoring in ambient assisted living applications," *Biosensors*, vol. 7, no. 4, p. 55, 2017.

[25] J. Vora, S. Tanwar, S. Tyagi, N. Kumar, and J. J. Rodrigues, "Faal: Fog computing-based patient monitoring system for ambient assisted living," in *2017 IEEE 19th international conference on e-health networking, applications and services (Healthcom)*, pp. 1–6, IEEE, 2017.

[26] A. Vercelli, I. Rainero, L. Ciferri, M. Boido, and F. Pirri, "Robots in elderly care," *DigitCult-Scientific Journal on Digital Cultures*, vol. 2, no. 2, pp. 37–50, 2018.

[27] H. Lehmann, D. S. Syrdal, K. Dautenhahn, G. Gelderblom, S. Bedaf, and F. Amirabdollahian, "What should a robot do for you?-evaluating the needs of the elderly in the uk," in *ACHI 2013, The Sixth International Conference on Advances in Computer-Human Interactions*, pp. 83–88, 2013.

[28] S. Bedaf, P. Marti, F. Amirabdollahian, and L. de Witte, "A multi-perspective evaluation of a service robot for seniors: the voice of different stakeholders," *Disability and Rehabilitation: Assistive Technology*, vol. 13, no. 6, pp. 592–599, 2018.

[29] M. Nani, P. Caleb-Solly, S. Dogramadzi, T. Fear, and H. van den Heuvel, "Mobiserv: an integrated intelligent home environment for the provision of health, nutrition and mobility services to the elderly," 2010.

[30] S. Coradeschi, A. Cesta, G. Cortellessa, L. Coraci, C. Galindo, J. Gonzalez, L. Karlsson, A. Forsberg, S. Frennert, F. Furfari, *et al.*, "Giraffplus: A system for monitoring

activities and physiological parameters and promoting social interaction for elderly," in *Human-Computer Systems Interaction: Backgrounds and Applications 3*, pp. 261–271, Springer, 2014.

[31] D. Fischinger, P. Einramhof, K. Papoutsakis, W. Wohlkinger, P. Mayer, P. Panek, S. Hofmann, T. Koertner, A. Weiss, A. Argyros, *et al.*, "Hobbit, a care robot supporting independent living at home: First prototype and lessons learned," *Robotics and Autonomous Systems*, vol. 75, pp. 60–78, 2016.

[32] D. Portugal, L. Santos, P. Alvito, J. Dias, G. Samaras, and E. Christodoulou, "Socialrobot: An interactive mobile robot for elderly home care," in *2015 IEEE/SICE International Symposium on System Integration (SII)*, pp. 811–816, IEEE, 2015.

[33] D. Portugal, P. Alvito, E. Christodoulou, G. Samaras, and J. Dias, "A study on the deployment of a service robot in an elderly care center," *International Journal of Social Robotics*, vol. 11, no. 2, pp. 317–341, 2019.

[34] S. Lee and A. M. Naguib, "Toward a sociable and dependable elderly care robot: Design, implementation and user study," *Journal of Intelligent & Robotic Systems*, pp. 1–13, 2019.

[35] H.-M. Gross, A. Scheidig, S. Müller, B. Schütz, C. Fricke, and S. Meyer, "Living with a mobile companion robot in your own apartment-final implementation and results of a 20-weeks field study with 20 seniors," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2253–2259, IEEE, 2019.

[36] H. M. Do, M. Pham, W. Sheng, D. Yang, and M. Liu, "Rish: A robot-integrated smart home for elderly care," *Robotics and Autonomous Systems*, vol. 101, pp. 74–92, 2018.

[37] T. Sorell and H. Draper, "Robot carers, ethics, and older people," *Ethics and Information Technology*, vol. 16, no. 3, pp. 183–195, 2014.

[38] G. Toms, F. Verity, and A. Orrell, "Social care technologies for older people: Evidence for instigating a broader and more inclusive dialogue," *Technology in Society*, 2019.

[39] A. Banks, E. Briggs, K. Borgendale, and R. Gupta, "Mqtt version 5.0," *OASIS Standard*, 2019.

[40] P. Chatterjee, L. J. Cymberknop, and R. L. Armentano, "Iot-based decision support system for intelligent healthcare—applied to cardiovascular diseases," in *2017*

7th International Conference on Communication Systems and Network Technologies (CSNT), pp. 362–366, IEEE, 2017.

[41] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open source language understanding and dialogue management," *arXiv preprint arXiv:1712.05181*, 2017.

[42] V. Vlasov, A. Drissner-Schmid, and A. Nichol, "Few-shot generalization across dialogue tasks," *arXiv preprint arXiv:1811.11707*, 2018.

[43] L. Y. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston, "Starspace: Embed all the things!," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[44] I. Ecma, "The json data interchange format," 2013.

[45] Centers for Disease Control and Prevention, "Symptoms of coronavirus," 2020.

[46] A. Smith, "What are the early symptoms of coronavirus (covid-19)?," 2020.

[47] J. G. Wilpon, D. B. Roe, *et al.*, *Voice communication between humans and machines*. National Academies Press, 1994.

[48] E. Vaportzis, M. Giatsi Clausen, and A. J. Gow, "Older adults perceptions of technology and barriers to interacting with tablet computers: a focus group study," *Frontiers in psychology*, vol. 8, p. 1687, 2017.

[49] N. Dethlefs, M. Milders, H. Cuayáhuitl, T. Al-Salkini, and L. Douglas, "A natural language-based presentation of cognitive stimulation to people with dementia in assistive technology: a pilot study," *Informatics for Health and Social Care*, vol. 42, no. 4, pp. 349–360, 2017.

[50] C. Ouali, M. M. Nasr, M. A. AbdelGalil, and F. Karray, "Voice controlled multi-robot system for collaborative task achievement," in *International Conference on Robot Intelligence Technology and Applications*, pp. 345–360, Springer, 2017.

[51] M. H. Amaran, N. A. M. Noh, M. S. Rohmad, and H. Hashim, "A comparison of lightweight communication protocols in robotic applications," *Procedia Computer Science*, vol. 76, pp. 400–405, 2015.

[52] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.