mVitals - An Intelligent Edge Computing Based Wireless Mobile Healthcare System

Final Year Project

Session 2017-2020

4th Year Student

A project submitted in partial fulfilment

Of

BS in Computer Science (CUI)

Department of Computer Science

COMSATS University Islamabad, Lahore Campus

28 December 2020

Project Detail

Type (Nature of project)	[ ] **D**evelopment [ ] **R**esearch [*] **R**&**D**
Area of specialization	

Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	SP17-BCS-029	Noor Jaffri	sp17-bcs-029@cuilahore.edu.pk	
(ii)	SP17-BCS-101	Muhammad Irtiza	sp17-bcs-101@cuilahore.edu.pk	
(iii)	SP17-BCS-080	Hafsa Khan	sp17-bcs-080@cuilahore.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I am Noor Jaffri S/D/o Syed Hamid Hussain Jaffri group leader of FYP under registration no CIIT/SP17-BCS-029/LHR at Computer Science Department, COMSATS Institute of Information Technology, Lahore. I declare that my FYP proposal is checked by my supervisor and the similarity index is ________% that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix A.

Date: _____________ Name of Group Leader: Noor Jaffri Signature: ______________

Name of Supervisor: Imran Raza
Designation: Assistant Professor
Signature: ________________

Co-Supervisor (if any): _______________________
Designation: ________________________
Signature: ________________________

HoD: ________________________
Signature: ________________________

Abstract

Internet of Things (IoT) made it possible to interconnect millions or billions of different "Things" to create huge networks. IoT based applications include but not limited to smart homes, automated parking systems, fitness bands and health monitoring systems, etc. IoT made it possible to build autonomous and effective real-time health care monitoring systems. It allows effective and flexible real-time health care monitoring systems, equipped with sensors that collect the patient's data regarding vital signs and reduce human error. The existing systems use cloud computing technology where the collected data is stored, processed and analysed on cloud servers. As the human population has been increasing exponentially, the public healthcare sector is focusing on advanced wearable devices rather than traditional monitoring systems. Due to a lack of continuous monitoring, many patients have been suffering in hospitals as well as in homes. To overcome this problem, technical experts are discovering new and viable approaches based on various technologies e.g., cloud, edge and fog computing. The proposed intelligent system i.e., mVitals, targets such patients who require real-time healthcare monitoring. It will provide an advanced wearable system, based on edge computing. The system will be highly portable and easy to use. The key objective of mVitals is to monitor the patients' vital signs such as heart rate, electrocardiogram (ECG), body temperature, etc. in real-time. The communication of the system will be based on wireless technology i.e. Bluetooth, so the patient wearing it will not feel tangled with wires. mVitals will be designed using Arduino Nano and various sensors such as ECG Module, heart rate, body temperature, blood pressure and breathing rate sensor. The patient's data will be acquired via sensors and transferred to an edge server to reduce latency and bandwidth utilization. Edge server i.e. mobile phone, will transfer the real-time data to the AI server. AI server will apply machine learning algorithms on acquired data for noise reduction, analytics, predictions and alert generation. Medical history and reports of the patients will be stored in a real-time database which will further help doctors to assess the patient's condition and suggest treatment plans based on machine learning. Healthcare is not only the most promising application of IoT technology, but also these devices and systems have the potential to enhance the quality of patients' monitoring.

Acknowledgement

By the grace of Allah and the prayers of our beloved parents, we successfully completed mVitals project. We would like to acknowledge and thank the IGNITE National Technology Fund, for funding our project. This project is a result of hard work of several creative and intelligent minds, to whom we express our gratitude and regards. Furthermore, we would like to extend our appreciation and gratitude to our Supervisor **Mr. Imran Raza,** for his dedicated supervision and encouragement, throughout the entire project.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Project Overview

The Internet of Things (IoT) has made a great evolution over the years [1]. It is a system of devices, some with built-in sensors, connected through the internet that collects data and apply analytics to extract valuable information. This information can be used to detect patterns, provide recommendations, and suggest future actions, etc. Today, applications of IoT systems can be observed in every domain especially in healthcare for remote monitoring of patients, even in real-time. The focus is not only to monitor vital signs and electrocardiogram (ECG), but the healthcare industry is working on telehealth, smart hospitals, wearables, etc. Most IoT systems are based on cloud-computing technology because it provides data storage, privacy, easily manageable data, and unified access. Furthermore, because of IoT contributions in healthcare, the Internet of Medical Things (IoMT) has become a branch of IoT because of IoT contributions in healthcare [2].

The Internet of Things is bringing the healthcare industry a new life. One of the best ways is where the doctors can use the appropriate patient's digital information to take suitable decisions. It increases the quality of patient information in the medical field. mVitals will utilize edge computing in combination with intelligent IoT devices. This will allow the system to collect patient data, send it to an edge server for noise reduction and provide real-time information. This system will use different sensors like ECG, heart rate, body temperature, blood pressure and breathing rate sensor to monitor patient's health. These sensors will be attached to a microcontroller like Arduino. To monitor a patient's health, the caretaker will use an end-user device to screen the vitals, which would be connected to an edge server wirelessly.

1.2 Cloud Computing Based IoT Devices

IoT systems use several architectural approaches. Most IoT systems are based on cloud-computing technology, which provides data storage, privacy, easily manageable data and unified access. The Cloud computing method is dependent on the Internet that allows computers and other equipment to share software and hardware information[3].

1.3 Edge Computing Based IoT Devices

IoT devices are adapting edge computing because of the high costs and complexity in centralized systems. IoMT is adapting edge computing [4] over the current centralized cloud infrastructure, because of the increased demand of data-driven care in the health industry. Edge computing is a decentralized architecture deployed in the proximity of the user to reduce latency and bandwidth utilization. It is more secure, efficient, inexpensive, and scalable; it also provides local manipulation of data [5]. Furthermore, in the healthcare monitoring system, the functionality can be very crucial for the patient in case of a low bandwidth rate. Edge computing solves this problem and is beneficial for applications that require intensive computations and low latency. Besides low latency, edge computing is preferred over cloud in remote locations as well, where there is limited or no connection to a central server location.

1.4 Objectives

- Design and implement an intelligent portable system for real-time healthcare monitoring.
- Develop a smart medical monitoring system that will utilize in hospitals, ambulances, etc. and collaborate with the smart home idea to make it part of daily life activities.
- Deploy deep learning techniques for training and testing of models, to provide reliable disease diagnosis.
- Design a system incorporating a Clinical Decision Support System (CDSS), which will enhance the patient's diagnosis with better analytics.
- Assist the doctors/caretakers to monitor the patient and provide a reliable notification mechanism in case of any critically.
- Make use of wireless technology, to get rid of jumbled up wired hardware systems and allowing patients to roam freely.
- Develop an edge computing-based system backed up with cloud services, to provide storage and analysis of data in real-time.
- Store and manage the health record of the patients for different purposes.
- Provide an adequate visual representation of the patient's vital signs and ECG, so even a non-medical person can monitor the patient.
- Reduce the costs associated with employing expensive monitoring equipment.

1.5 Problem Statement

WHO (World Health Organization) country profile shows that in Pakistan 25.3% of individuals had high BP, 19% had CVD (Cardiovascular disease) diseases, 6% had chronic respiratory diseases [6]. Pakistan being a developing country does not possess the resources for medical treatment with the high population growth. A survey was conducted in 2017-2018 that showed there is only one bed for 1580 patients [7]. Doctors can't treat most of the patients with limited resources. A remote healthcare system is needed which may free up more beds in hospitals that can benefit patients who need urgent help. The recent pandemic situation has limited people only to their homes. This situation called for a remote healthcare system that could maintain safety between doctor and patient.

1.6 Assumptions and Constraints

1.6.1 Assumptions

- The system will develop for the real-time vital signs monitoring of patients.
- The system will assist doctors towards reliable disease diagnosis using edge computing technology.
- A platform for doctors to examine the detailed continuous data of a patient`s vital signs with accuracy and visual representation.
- An online database-oriented system will keep the detailed data of each patient organized.

- An artificially intelligent system that will identify the disease and learn from the data acquired each time a patient uses the system and give useful suggestion towards the best treatment plans.

1.6.2 Constraints

- To use this system, the user must have mVitals device.

- The project must be completed in the giving period.

- Most of part of data will be stored on a cloud database, not on local devices.

- The user using this system must have basic knowledge of information technology devices i.e. mobile phones.

1.7 Project Scope

- Most of the current healthcare monitoring systems have jumbled up hardware. To tackle this problem, most of the functionalities performed by our system use wireless technology, such as Bluetooth.

- According to [8], due to insufficient equipment in hospitals for emergency conditions, patients suffer, and lives are at stake. To overcome this problem, an intelligent healthcare monitoring system is introduced, which continuously monitors the vital signs and ECG of patients through wirelessly connected devices.
- Transfer delay of trauma patients to hospitals leads to complications and suffering [9]. Even seconds can make a difference. This was the motivation to introduce such a system that even a non-medical person could use to monitor vital signs and alert hospitals on time in case of any criticality.
- As the Electronic Health Records (EHRs) [10] are replacing paper records, the proposed system will store the patients' data in an online-secured database. This allows the authorized personnel to access the records anytime and anywhere.
- This system monitors the vital signs and ECG of the patient remotely. Hence, reducing the effort of doctors and patients.
- This smart medical monitoring and diagnostic system will not only be utilized in hospitals, ambulances, etc. but also collaborate with the smart home idea to make it part of daily life activities.

- The system will use deep learning algorithms, which learn from past data of patients. It provides predictions about the critical condition of patients.

2 Requirement Analysis

2.1 Existing Systems Study

As the health industry begins to incorporate IoT devices and systems, many wired or unwired gadgets and armbands are developed to monitor patients' health in real-time. However, they are either not very versatile or efficient. A fitness tracking smartwatch is introduced in [11] that collects and displays users' heart rate, steps in a day, and calories burnt. However, it has only limited sensors and does not monitor all the vital signs such as oxygen level and respiratory rate. Thus, the scope of its health care monitoring functionality is limited.

The Phillips Guardian Solution[12] is designed to deploy in hospitals. It connects with EMR and ADT using the same route which is deployed in ICU monitors for seamless data exchange. The Validated EWS data goes straight to EMR to help meet the standards of care It Sends data only to the nursing staff. This system also uses cloud computing and is only built for hospital beds.

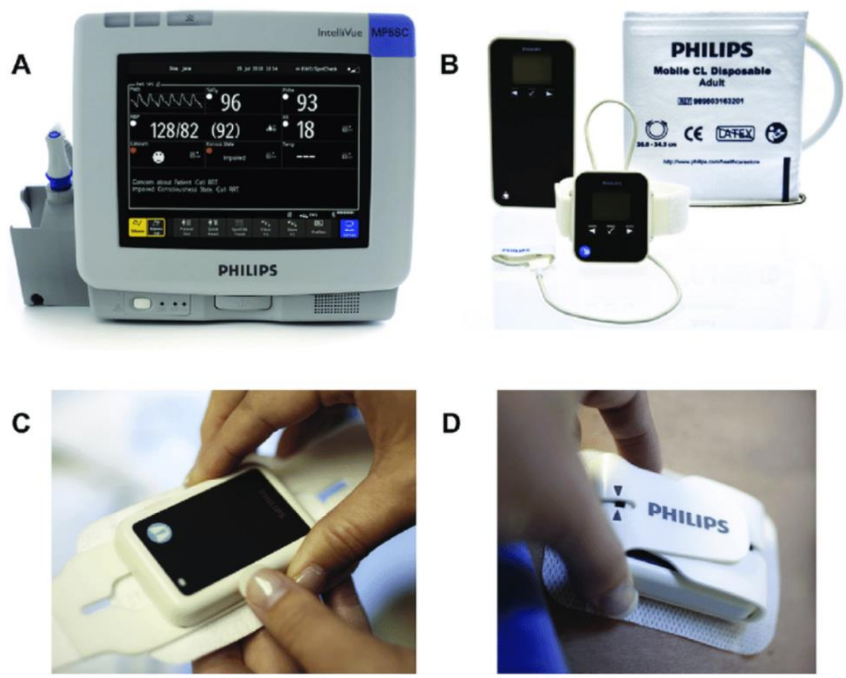

Figure 1- The Phillips Guardian [12]

The Life Scope G9[13] is a system that provides comprehensive parameter monitoring with data storage, comprehensive arrhythmia, and ST-segment analysis. It also includes a 12-lead ECG capability, hemodynamic and pulmonary calculations. The Life Scope G9 is a bedside monitor designed. It offers a synergetic display of patient information based on present and past medical history on a simple user interface for higher clinical intervention. Life Scope G9 is a high-precision monitor. It is expensive and it can only be deployed in hospitals. Its data cannot be viewed on any other device.

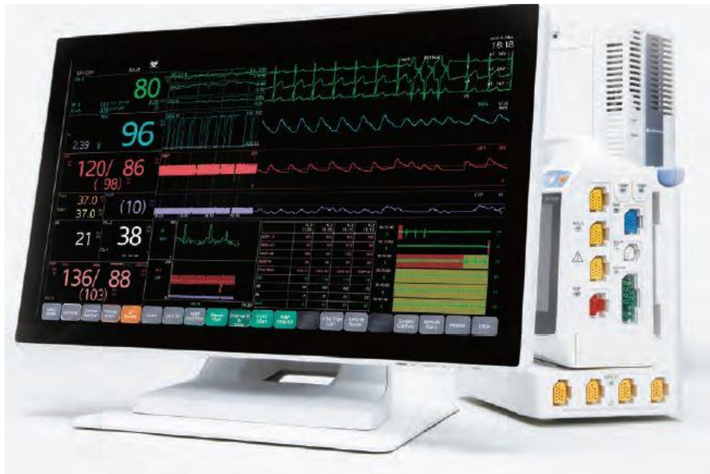

Figure 2- Life Scope G9 Bedside Monitor[13]

Researchers at Monash University in Australian has developed a gadget that can monitor blood pressure continuously in different situation such as while exercising, walking, or running and even sleeping. Its accuracy drops to 83% during exercise. Their solution of a continuous-wave radar (CWR) sensor placed on the sternum, and a photoplethysmogram (PPG) sensor placed on the left earlobe. It is wireless, comfortable, and cuff-less, but it serves a signal function[14].

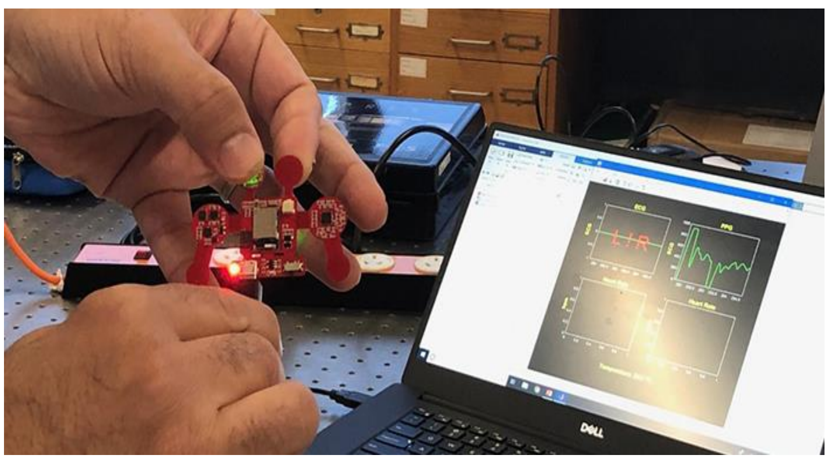

Figure 3- Wearable Monitor for Blood Pressure[14]

Sotera Wireless Inc. developed a small and portable multipurpose device called ViSi Mobile [15]. It is capable of monitoring patients' heart rate, blood pressure, pulse rate, respiration rate, skin temperature, etc. in Figure 4. The major drawbacks of this device are that it is very expensive and usually works on cellular networks, which can be unavailable in some regions. However, some of the newer and expensive versions can also be connected via Wi-Fi. Figure 4 shows the hardware connectivity of ViSi Mobile with a human body, which has plenty of wires that could be irritating for the patient.

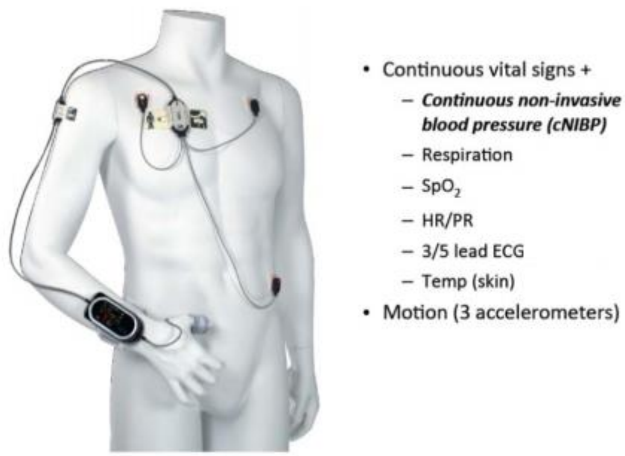

Figure 4- ViSi Mobile's Hardware [16]

The F8 is a fitness[17] tracking smartwatch. It is introduced to collect and display users' heart rate, steps in a day, and calories burnt. It is applicable on android 4.4 and IOS 8.4 or above. However, it has only limited sensors and does not monitor all the vital signs such as oxygen level and respiratory rate. The Accuracy of this smartwatch is questionable.

Figure 5- F8 Smartwatch[18]

A pulse oximeter [19] is developed to determine the arterial oxygen saturation in the blood of the user, and heart rate during sports at high altitudes, etc. It has an adjustable LED display and Low battery consumption. However, it also serves a single function and is a delicate device and it does not store any data.

Figure 6- Beurer PO 30[19]

The Nihon Kohden's Life Scope G3[20] is a wearable vital sign telemeter. Based on edge technology. The patient's vital signs and waveforms can be displayed using a colour LCD of 3.2 inches. It alarms the user in case of a sudden change in the patient's condition. It only works with 3 sensors to monitor ECG, respiration, and SpO_2 of the patients.

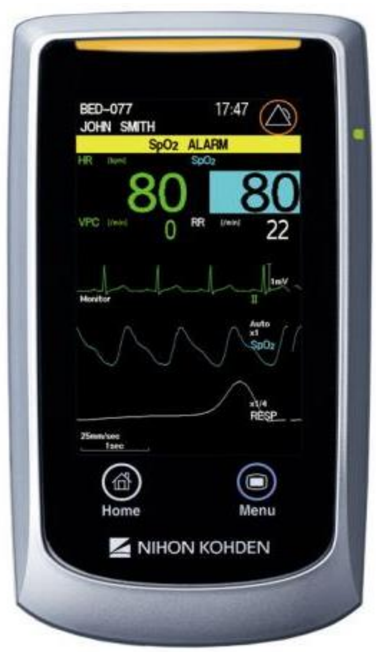

Figure 7- Life Scope G3[20]

2.2 Stakeholders List

- Doctor

- Paramedical Staff
- Healthcare Facilities
- Patient
- Caregiver
- AI Server

2.3 Requirements Elicitation

2.3.1 Functional Requirements

FR01: Signup

Table 1- FR01: Signup

Req. No.	**Functional Requirements**
FR01-01	A doctor will be able to create an account and provide his/her information such as name, specialization, contact number, email, and hospital/clinic name. This record will be automatically stored in the database.
FR01-02	A patient will be able to create an account and provide his/her information such as name, age, contact name and email. This record will be automatically stored in the database.
FR01-03	The doctor as well the patient registered with that doctor will be able to access that particular doctor's profile after authorization.

FR02: Login

Table 2- FR02: Login

Req. No.	**Functional Requirements**
FR02-01	The patient and doctor will be able to log into the system and avail the features provided by the system.
FR02-02	The system shall allow the user to recover their forgotten password by recovery methods.

FR03: Edit Doctor's Profile

Table 3- FR03: Edit Doctor's Profile

Req. No.	Functional Requirements
FR03-01	The system shall provide an interface to the doctor to edit his/her profile from the database.

FR04: Register New Patient

Table 4- FR04: Register New Patient

Req. No.	Functional Requirements
FR04-01	The doctor/caretaker will be able to register a patient and provide the patient's information such as name, gender, age, contact number, email, and date of birth, address and medical history.
FR04-02	Patient's information shall be automatically stored in the database and accessible by the doctor as well as the patient.

FR05: Edit Patient`s Profile

Table 5- FR05: Edit Patient`s Profile

Req. No.	Functional Requirements
FR05-01	The system shall allow the patient to edit the profile from the system database.

FR06: Remove Patient

Table 6- FR06: Remove Patient

Req. No.	Functional Requirements
FR06-01	A patient will be able to remove his/her profile from the doctor's profile by the doctor, with whom he is registered, as well as by the patient himself.

FR07: View Vital Signs

Table 7- FR07: View Vital Signs

Req. No.	Functional Requirements

FR07-01	The system shall allow the user to view an interface with all the vital sign information of the patient.

FR08: Search and access to Patient

Table 8- FR08: Search and access to Patient

Req. No.	Functional Requirements
FR08-01	The doctor will be able to view the list of all his/her registered patients.
FR08-02	The doctor will be able to view complete details of all his/her registered patients.
FR08-03	The doctor will be able to Search his/her registered patients.

FR09: Diagnosis and Treatment Plan

Table 9- FR09: Diagnosis and Treatment Plan

Req. No.	Functional Requirements
FR09-01	The system will provide automatic diagnosis and suggestions for treatment plans to the user. It will store meaningful data of the patients into the database and applies machine learning models to it for the system to learn and aid user.
FR09-02	The system will allow the doctor to edit the generated treatment plan of specific patient according to requirement.

FR10: Run Patient Diagnosis

Table 10- FR10: Run Patient Diagnosis

Req. No.	Functional Requirements
FR10-01	The doctor will be able to manually run diagnosis, along with automatic diagnosis, on his/ her patients and view the results.

FR11: Set Appointment for Patient

Table 11- FR11: Set Appointment for Patient

Req. No.	Functional Requirements

FR11-01	The doctor will be able to set up an appointment for the patient in his/her treatment plan, provided by the system.
FR11-02	The doctor will be able to provide appointment details such as location, date, time, doctor's name, etc.

FR12: Alert Generation

Table 12- FR12: Alert Generation

Req. No.	Functional Requirements
FR12-01	In the case of criticality in the patient's condition, the system will send an alert to the patient's doctor, nearby hospital based on GPS or an ambulance.

FR013: View History

Table 13- FR013: View History

Req. No.	Functional Requirements
FR013-01	The doctor and patient will be able to view the historical data of the particular patient, which will be already stored in the system's database. They will allow us to click on a vital sign and a screen will display a history of that particular vital sign information for each hour, day or month.

Table 14- FR014: View Report

FR014: View Report

Req. No.	Functional Requirements
FR014-01	The doctor and patient will allow us to view historical data of the particular patient, which will be already stored in the system's database.
FR014-02	Patients' real time as well as historic reports will be generated and viewed by doctors, patients and administrators.

FR015: Delete Account

Table 15- FR015: Delete Account

Req. No.	Functional Requirements

FR015-01	The system will allow the doctor to permanently delete his/her account.
FR015-02	The system will allow the patient to permanently delete his/her account.

2.3.2 Non-Functional Requirements

NFR01: Performance and Scalability

Table 16- NFR01: Performance and Scalability

NFR01-01	The starting page's average load time should not be more than 5 seconds.
NFR01-02	The average processing time taken by the system to complete the authentication process should not be greater than 5 seconds.
NFR01-03	System Mean Time to Failure (MTTF) should not be more than 1 minute.
NFR01-04	The average system response time should not be greater than 10 seconds.
NFR01-05	The system must successfully and effectively run on a client device with 2 GB RAM or above, 10 GB storage and a good internet connection.

NFR02: Security

Table 17- NFR02: Security

NFR02-01	Only authorized users should be able to access the system.
NFR02-02	Any unauthorized users should not have any access control for the database.
NFR02-03	Any user should not be allowed to view history, report, diagnosis or treatment plans of other users.
NFR02-04	All acquired data from the sensors are directly saved to the cloud database.

NFR03: Defects-Maintainability

Table 18- NFR03: Defects-Maintainability

NFR03-01	The average defect escaped ratio of the system should not be more than 5% per month.

NFR03-02	Post Release defects fixing should not take more than 8 hours.
NFR03-02	Escaped defects should be fixed on system updates.

NFR04: User Documentation

Table 19- NFR04: User Documentation

NFR04-01	Complete documentation of the system should be provided to the user.
NFR04-02	To avoid any inconvenience, help options must be provided throughout the system.
NFR04-03	Help must easily accessible using tooltips and graphical representation.

NFR05: Data Recovery

Table 20- NFR05: Data Recovery

NFR05-01	In case of system /server crashes all data should be recoverable within 30 minutes.

NFR06: Usability

Table 21- NFR06: Usability

NFR06-01	The system will be clear and responsive on each element.
NFR06-02	Each element should be represented with the help of an icon in the system.
NFR06-03	Sliders, tooltips and icons shall be used in the system.

NFR07: Data Integrity

Table 22- NFR07: Data Integrity

NFR07-01	Acquired data must be filtered before saving in cloud database.
NFR07-02	Each user’s data shall be stored separately.

2.4 Use Case Description

Table 23- UC01: Sign Up

2.4.1 UC01: Sign Up

Use Case Name	Signup	
Abstract	No	
Purpose	To register new users to use/access system.	
Actors	Doctor, Patient	
Importance	Primary	
Overview	This use case register new users and authorizes them to use system features according to user type.	
Requirements		
Status	Essential	
Uses		
Pre-conditions	None	
Post-conditions	If registration is successful, the user will be logged into the system.	
	Actor Actions	**System response**
	Typical Course of Actions	
	1. User starts the system.	
		1a. System requests to select user type.

	2. User selects user type and click on the text filled which are to be filled.	
	3. User clicks on 'Sign Up' button.	
		3a. System validates the user input, registers and logs user into the system.
	Alternative Course	
	1. User entered invalid text fields.	
		1a. System validates the user input and displays error message
	Exceptional Course	
		1a. Database connection error occurs.

2.4.2 UC02: Login

Table 24- UC02: Login

Use Case Name	Login
Abstract	No
Purpose	To authenticate users to use/access system.
Actors	Doctor, Patient
Importance	Primary

Overview	This use case authenticates users and authorizes them to use system features according to user type.
Requirements	
Status	Essential
Uses	Reset Password
Pre-conditions	Account should be created.
Post-conditions	If authentication is successful, the user will be logged into the system.

	Actor Actions	**System response**
	Typical Course of Actions	
	1. User starts the system.	
		1a. System requests to select user type and enter username and password.
	2. User selects user type and enters username and password.	
	3. User clicks on ‘Log In’ button.	
		3a. System validates the user input and logs user into the system.
	Alternative Course	
	4. User entered invalid username and/or password.	

		1a. System validates the user input and displays error message
	Exceptional Course	
		1a. Database connection error occurs.

2.4.3 UC03: Forget Password

Use Case Name	Forget Password
Abstract	No
Purpose	To authenticate users to recover their login password.
Actors	Doctor, Patient
Importance	Primary
Overview	This use case enables users to recover the forgotten password to their accounts.
Requirements	
Status	Essential
Uses	n/a
Pre-conditions	
Post-conditions	Password is set/changed.

	Actor Actions	**System response**
	Typical Course of Actions	

	1. User clicks on “Forget Password” button.	
		1a. System displays a form with fields; email, new password, confirm new password,
	2. User enters email, new password, confirm new password,	
	3. User clicks on “Reset” button.	
		3a. System validates the user input and updates the current user password.
	Alternative Course	
	1. User entered invalid password format.	
		1a. System validates the user input and displays error message.
	Exceptional Course	
		1a. Database connection error occurs.

2.4.4 UC04: Reset Password

Table 25- UC04: Reset Password

Use Case Name	Reset Password
Abstract	No

Purpose	To authenticate users to reset login password.	
Actors	Doctor, Patient	
Importance	Primary	
Overview	This use case enables users to reset the passwords to their accounts.	
Requirements		
Status	Essential	
Uses		
Pre-conditions	User is logged in using old password.	
Post-conditions	Password is set/changed.	
	Actor Actions	**System response**
	Typical Course of Actions	
	4. User clicks on "Change Password" button.	
		1a. System displays a form with fields; current password, new password, confirm new password,
	5. User enters current password, new password, confirm new password,	
	6. User clicks on "Reset" button.	

		3a. System validates the user input and updates the current user password.
	Alternative Course	
	2. User entered invalid password format.	
		1a. System validates the user input and displays error message.
	Exceptional Course	
		1a. Database connection error occurs.

Table 26- UC05: Edit Profile

2.4.5 UC05: Edit Profile

Use Case Name	Edit Profile
Abstract	No
Purpose	To enable patients and doctors to edit profile information.
Actors	Patient, doctor
Importance	Primary
Overview	This use case will enable patients and doctors to edit their profile information like contact number, address, email address.
Requirements	
Status	Essential
Uses	Authentication

Pre-conditions	Patient/doctor is logged in and authorized.	
Post-conditions	Patient/doctor has made changes in profile information.	
	Actor Actions	**System response**
	Typical Course of Actions	
	3. Patient/doctor enters ‘Edit Profile’ section.	
	4. Patient/doctor clicks on the text field that is to be edited.	
	5. Patient/doctor inputs new details and click on ‘Update Info’ button.	
		5.a. System validates the input data from Patient/doctor.
		5.b. System updates the new profile details in database and displays success message.
	Alternative Course	
		Data input format not correct – System displays error message.
	Exceptional Course	
		1a. Database connection error occurs.

2.4.6 UC6: Delete Account

Table 27- UC6: Delete Account

Use Case Name	Delete Account	
Abstract		
Purpose	To enable doctor and patient to permanently delete account.	
Actors	Doctor, Patient	
Importance	Primary	
Overview	This use case will enable the doctor and patient to permanently delete account.	
Requirements		
Status	Essential	
Uses		
Pre-conditions		
Post-conditions	Account deleted.	
	Actor Actions	**System response**
	Typical Course of Actions	
	1. Doctor/Patient clicks on the "Profile" button.	
		1a. System will display profile of doctor/patient.

	2. Doctor/Patient clicks the “Edit Profile” button.	
		1a. System allows the doctor/ patient to edit the profile.
	3. Doctor/Patient clicks the “Delete Account Permanently” button.	
		1a. System will delete the account permanently.
	Alternative Course	
		Error while deleting the account – System displays error message.
	Exceptional Course	
		1a. System error occurs.

Table 28- UC7: Patient Registration

2.4.7 UC7: Patient Registration

Use Case Name	Patient Registration
Abstract	
Purpose	To enable doctor to register a new authorized patient.
Actors	Doctor
Importance	Primary

Overview	This use case will enable the doctor to register a new authorized patient in his/her profile.	
Requirements	Patient’s account should be made.	
Status	Essential	
Uses		
Pre-conditions	Doctor must be logged in the system. Patient is present in the database	
Post-conditions	New patient will register in doctor’s profile.	
	Actor Actions	**System response**
	Typical Course of Actions	
	4. Doctor clicks on the “Add New Patient” button.	
		1a. System will display the text fields for adding patient’s information.
	5. Doctor fills the text fields for adding patient’s information and clicks “Add” button.	
		1a. System will add the patient.
	Alternative Course	

		Error while adding patient – System displays error message.
	Exceptional Course	
		1a. System error occurs.

Table 29- UC8: View History

2.4.8 UC8: View History

Use Case Name	View History
Abstract	
Purpose	To enable patient and doctor whom that particular patient is registered, to view history.
Actors	Patient, Doctor
Importance	Primary
Overview	This use case will enable the patient and their doctor to view the history.
Requirements	
Status	Essential
Uses	
Pre-conditions	The patient's vital sign data should be present in system database. Patient/Doctor is logged in and authorized.
Post-conditions	History of patient has shown.

	Actor Actions	**System response**
	Typical Course of Actions	
	Patient or doctor clicks on the “View History” button.	
		System will display the history records of patient.
	Alternative Course	
		Error while opening history – System displays error message.
	Exceptional Course	
		1a. mVitals device connection/configuration error occurs.
		2a. System error occurs.

Table 30- UC9: View Report

2.4.9 UC9: View Report

Use Case Name	View Report
Abstract	
Purpose	To enable patient and doctor whom that particular patient is registered, to view report of patient’s vital signs.
Actors	Patient, Doctor
Importance	Primary

Overview	This use case will enable the patient and their doctor to view the report of the patient’s vital signs.	
Requirements	Continuous real-time data gathering and storing.	
Status	Essential	
Uses		
Pre-conditions	Patient/Doctor is logged in and authorized.	
Post-conditions	Report of patient has shown.	

	Actor Actions	**System response**
	Typical Course of Actions	
	Patient or doctor clicks on the “Generate Report” button.	
		System will display the generated report recorded vital signs of patient.
	Alternative Course	
		Error while generating report – System displays error message.
	Exceptional Course	
		1a. mVitals device connection/configuration error occurs.
		2a. System error occurs.

2.4.10 UC10: View Diagnosis

Table 31- UC10: View Diagnosis

Use Case Name	View Diagnosis	
Abstract		
Purpose	The doctor will be provided a set of diagnosis by the system from the acquired data of the registered patient.	
Actors	Doctor	
Importance	Primary	
Overview	This use case will enable the doctor to view the diagnosis from the acquired data of the registered patient.	
Requirements	The patient should be registered by the doctor.	
Status	Essential	
Uses		
Pre-conditions	Doctor is logged in and authorized. Patient is added and present in the database.	
Post-conditions	A list of diagnosis will display.	
	Actor Actions	**System response**
	Typical Course of Actions	
	Doctor clicks on the “Diagnosis” button.	

		System will display the list of diagnosis from the patient's data.
	Alternative Course	
		Error while diagnosis process – System displays error message.
	Exceptional Course	
		1a. System error occurs.

2.4.11 UC11: View Treatment Plan

Table 32- UC11: View Treatment Plan

Use Case Name	View Treatment Plan
Abstract	
Purpose	To enable doctor to generate treatment plan suitable for patient.
Actors	Doctor
Importance	Primary
Overview	This use case will enable the doctor to generate a treatment plan for the patient.
Requirements	Patient registering by the doctor.
Status	Essential
Uses	
Pre-conditions	The data history of patient's vital signs should be present in system.

Post-conditions	Treatment plan generated by the system displays.	
	Actor Actions	**System response**
	Typical Course of Actions	
	Doctor clicks on the “Treatment Plan” button.	
		1a. System will display the treatment plan suitable for the patient.
	Alternative Course	
		Error while generating treatment plan – System displays error message.
	Exceptional Course	
		1a. System error occurs.

2.4.12 UC12: Edit Patient’s Treatment Plan

Table 33- UC12: Edit Patient’s Treatment Plan

Use Case Name	Edit Patient’s Treatment Plan
Abstract	
Purpose	To enable doctor to edit the generated treatment plan for patient.
Actors	Doctor
Importance	Primary

Overview	This use case will enable the doctor to edit the already generated treatment plan of the patient according to requirements.	
Requirements	Patient registering by the doctor.	
Status	Essential	
Uses		
Pre-conditions		
Post-conditions	Edited treatment plan displays.	
	Actor Actions	**System response**
	Typical Course of Actions	
	1. Doctor clicks on the “Treatment Plan” button.	
		1a. System will display the treatment plan suitable for the patient.
	2. Doctor clicks on the “Edit” button.	
		1a. System allows the doctor to edit the generated treatment plan of patient.
	Alternative Course	
		Error while editing the treatment plan – System displays error message.
	Exceptional Course	

		1a. System error occurs.

2.4.13 UC13: Remove Patient

Table 34- UC13: Remove Patient

Use Case Name	Remove Patient	
Abstract		
Purpose	To enable doctor to remove patient from his/her profile.	
Actors	Doctor	
Importance	Primary	
Overview	This use case will enable the doctor to remove a patient from his/her profile.	
Requirements	Patient registering by the doctor.	
Status	Essential	
Uses		
Pre-conditions		
Post-conditions	Patient removed.	
	Actor Actions	**System response**
	Typical Course of Actions	
	6. Doctor clicks on the “Registered Patients” button.	

		1a. System will display all the patients registered by doctor.
	7. Doctor select the patient to be deleted and clicks "Delete" button.	
		1a. System allows the doctor to delete the patient.
	Alternative Course	
		Error while deleting the patient – System displays error message.
	Exceptional Course	
		1a. System error occurs.

2.4.14 UC14: View Vital Signs

Table 35- UC14: View Vital Signs

Use Case Name	View Vital Signs
Abstract	
Purpose	To enable patient to view his/her vital signs.
Actors	Patient
Importance	Primary
Overview	This use case will enable the patient to view his/her vital signs.
Requirements	mVitals device should be connected/configured properly.

Status	Essential	
Uses		
Pre-conditions	Patient is logged in and authorized.	
Post-conditions	Vital signs shows.	
	Actor Actions	**System response**
	Typical Course of Actions	
	1. Patient logged in and connect device.	
		1a. System will display all vital signs of patient.
	Alternative Course	
		Error while displaying vital signs of the patient – System displays error message.
	Exceptional Course	
		1a. System error occurs. 2a. mVitals device connection/configuration error.

2.4.15 UC15: Search Patient

Table 36- UC15: Search Patient

Use Case Name	Search Patient
Abstract	No

Purpose	To enable doctor to search patient from the database and view his profile.	
Actors	Doctor	
Importance	Primary	
Overview	This use case will enable doctors to search patients from the database and view his profile information and perform any other action he is allowed to perform.	
Requirements		
Status	Essential	
Uses		
Pre-conditions	Doctor is logged in and authorized. Patient is added and present in the database	
Post-conditions	Patient is displayed.	
	Actor Actions	**System response**
	Typical Course of Actions	
	Patient enters 'Search Patients' section.	
		System displays list of all the Patients in database.
	Alternative Course	
		No Patient is found– System displays error message.

	Exceptional Course	
		Database connection error occurs.

2.4.16 UC16: Alert Messages

Table 37- UC16: Alert Messages

Use Case Name	Alert Messages	
Abstract	No	
Purpose	To send alert messages to doctor in case of emergency.	
Actors	Doctor	
Importance	Primary	
Overview	This use case will enable alert generation and send it to the patient's doctor.	
Requirements		
Status	Essential	
Uses		
Pre-conditions	Doctor is logged in and authorized. Patient is logged in and authorized.	
Post-conditions	Alert messages send.	
	Actor Actions	**System response**
	Typical Course of Actions	

	Abnormal readings coming from the mVitals device.	
		System sends alert notifications to the patient's doctor/nearby hospital.
	Alternative Course	
		No Doctor is found– System displays error message.
	Exceptional Course	
		Database connection error occurs.

2.4.17 UC17: Set Appointment

Table 38- UC17: Set Appointment

Use Case Name	Set Appointment
Abstract	No
Purpose	To enable doctor to set an appointment for the patient.
Actors	Doctor
Importance	Primary
Overview	This use case will enable doctors to set an appointment for the patient.
Requirements	
Status	Essential

Uses		
Pre-conditions	Doctor is logged in and authorized. Patient is added and present in the database	
Post-conditions	Patient is displayed.	
	Actor Actions	**System response**
	Typical Course of Actions	
	1. Doctors clicks “Set Appointment” button.	
		1a. System will display a form of text fields i.e. select patient, time and date.
	2. Doctor fills the text fields i.e. select patient, time and date and clicks “Set” button.	
		1a. System will set an appointment for the patient.
	Alternative Course	
		No Patient is found– System displays error message.
	Exceptional Course	
		Database connection error occurs.

2.4.18 UC18: Data Acquisition

Table 39- UC18: Data Acquisition

Use Case Name	Data Acquisition	
Abstract		
Purpose	To enable system to acquire data from the mVitals device for the vital signs monitoring.	
Actors	AI Server	
Importance	Primary	
Overview	This use case will enable AI server to get data from the bodysuit, process it, extract required features from it and identify disease or whether the person has disease or not.	
Requirements		
Status	Essential	
Uses		
Pre-conditions	mVitals device is connected and configured properly.	
Post-conditions	Data is acquired from the mVitals device.	
	Actor Actions	**System response**
	Typical Course of Actions	
	mVitals device is configured and patient wears it.	

		AI server gets all the data coming from the sensors of mVitals device.
	Alternative Course	
		Device not connected or configured properly– System displays error message.
	Exceptional Course	
		Connection error occurs. Configuration error occurs.

2.4.19 UC19: Data Processing

Table 40- UC19: Data Processing

Use Case Name	Data Processing
Abstract	
Purpose	To enable the system to process the acquired data.
Actors	AI Server
Importance	Primary
Overview	This use case will enable AI server to process the data acquired from the mVitals device. Processing includes noise removal from the data and data normalization to remove any unnecessary data.
Requirements	
Status	Essential
Uses	Data Acquisition

Pre-conditions	Data is acquired from the mVitals device.	
Post-conditions	The data is processed properly and is ready for further use.	
	Actor Actions	System response
	Typical Course of Actions	
	AI server gets the data from the mVitals device and applies some methods to process it.	
		Data is processed and ready for further use.
	Alternative Course	
		Data not found or unable to process data– System displays error message.
	Exceptional Course	
		Data acquisition error occurs. System error occurs.

2.4.20 UC20: Feature Extraction

Table 41- UC20: Feature Extraction

Use Case Name	Feature Extraction
Abstract	
Purpose	To enable AI server to extract required features from the processed data.
Actors	AI server

Importance	Primary	
Overview	This use case will enable AI server to extract required features from the processed data that will be further classified and on the basis of these features, diagnosis will be carried out.	
Requirements		
Status	Essential	
Uses		
Pre-conditions	Data is acquired and processed properly.	
Post-conditions	Required features from the data are extracted.	
	Actor Actions	System response
	Typical Course of Actions	
	AI server will extract required features for diagnosis purpose from the processed data.	
		Required features for diagnosis are extracted.
	Alternative Course	
		Unable to perform the required action– System displays error message.
	Exceptional Course	
		System error occurs.

2.4.21 UC21: Diagnosis

Table 42- UC21: Diagnosis

Use Case Name	Diagnosis	
Abstract		
Purpose	To enable system to identify the fluctuation in vital signs and diagnosis diseases.	
Actors	AI server	
Importance	Primary	
Overview	This use case will enable the system to diagnose the disease from the vital signs of patients and decide if the fluctuation of vital signs are at dangerous level.	
Requirements		
Status	Essential	
Uses		
Pre-conditions	mVitals device is connected/configured Properly	
Post-conditions	Fluctuation in vital signs is identified and details are displayed to the doctor.	
	Actor Actions	**System response**
	Typical Course of Actions	
	AI server identifies the fluctuation in vital signs using some machine learning algorithm.	

		Abnormal readings are identified and displayed to doctor.
	Alternative Course	
		System failed– System displays error message.
	Exceptional Course	
		System error occurs.

Table 43- UC22: Treatment Plan

2.4.22 UC22: Treatment Plan

Use Case Name	Treatment Plan
Abstract	
Purpose	To enable system to prepare a treatment plan from acquired data.
Actors	AI server
Importance	Primary
Overview	This use case will enable the system to create a treatment plan from the acquired data and decides which treatment should be better for the patient according to patient's vital signs readings.
Requirements	
Status	Essential
Uses	
Pre-conditions	mVitals device is connected/configured Properly

<table>
<tr><td colspan="2">Post-conditions</td><td>Treatment plan for the patient is created and displays to the doctor.</td></tr>
<tr><td></td><td>Actor Actions</td><td>System response</td></tr>
<tr><td></td><td>Typical Course of Actions</td><td></td></tr>
<tr><td></td><td>AI server create treatment plan using some machine learning algorithm.</td><td></td></tr>
<tr><td></td><td></td><td>Treatment plan is created and displayed to doctor.</td></tr>
<tr><td></td><td>Alternative Course</td><td></td></tr>
<tr><td></td><td></td><td>System failed– System displays error message.</td></tr>
<tr><td></td><td>Exceptional Course</td><td></td></tr>
<tr><td></td><td></td><td>System error occurs.</td></tr>
</table>

2.5 Use Case Design

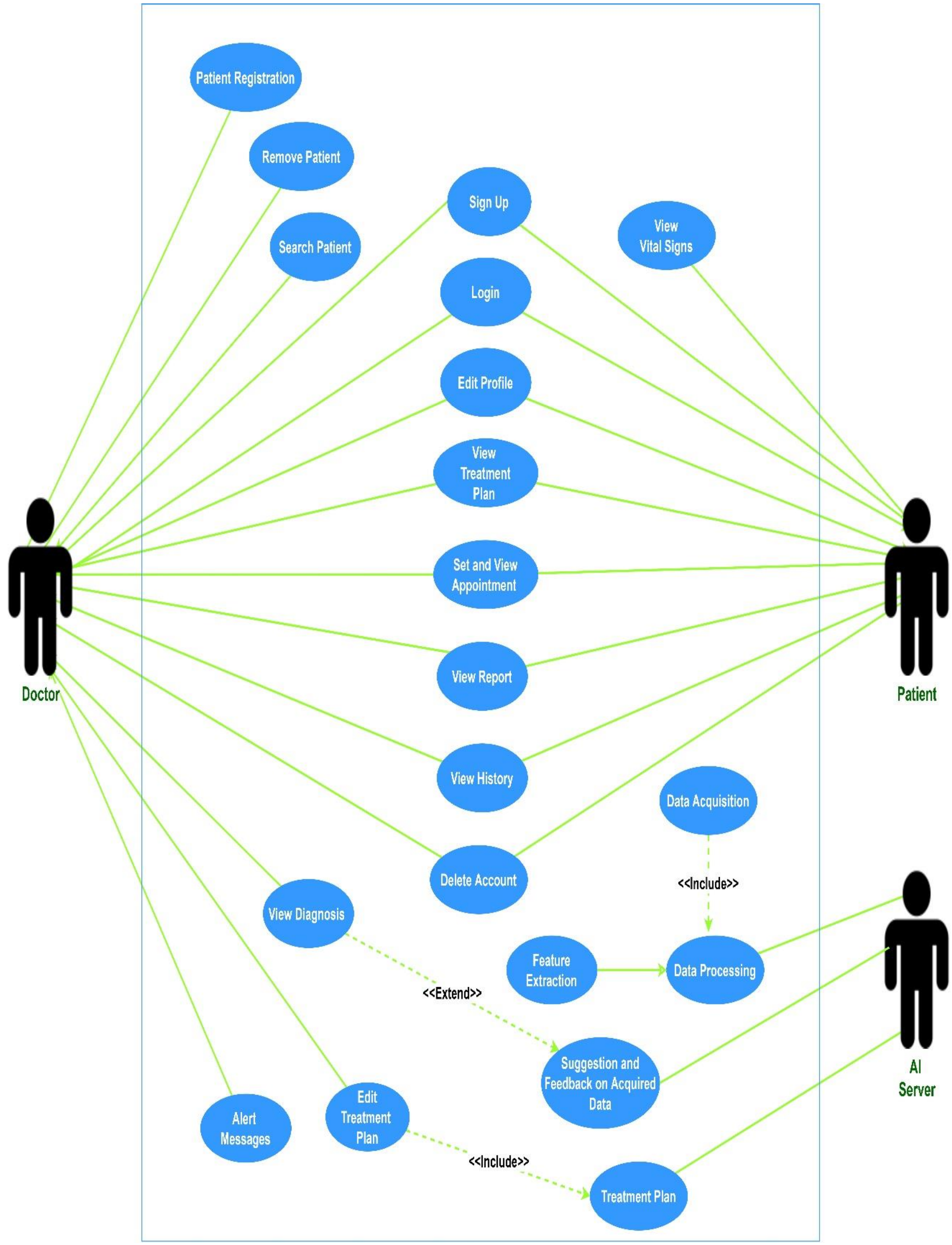

Figure 8- Use Case Diagram of System

2.6 Software Development Life Cycle Model

For the implementation of this project, Scrumban, an agile project development methodology that is a hybrid of Scrum and Kanban, will be deployed.

2.6.1 Why Use Scrumban Model?

Scrumban is widely used in the development and maintenance of projects. We are adapting this methodology because it embraces the features of both Scrum and Kanban where the former is used as a way of working and the latter is used to view, understand and improve performance. It also uses key metrics to estimate the average time for completion of a specific task, a highly versatile approach for workflow management, reduces the impact of errors, increases productivity and waste minimization efficiency.

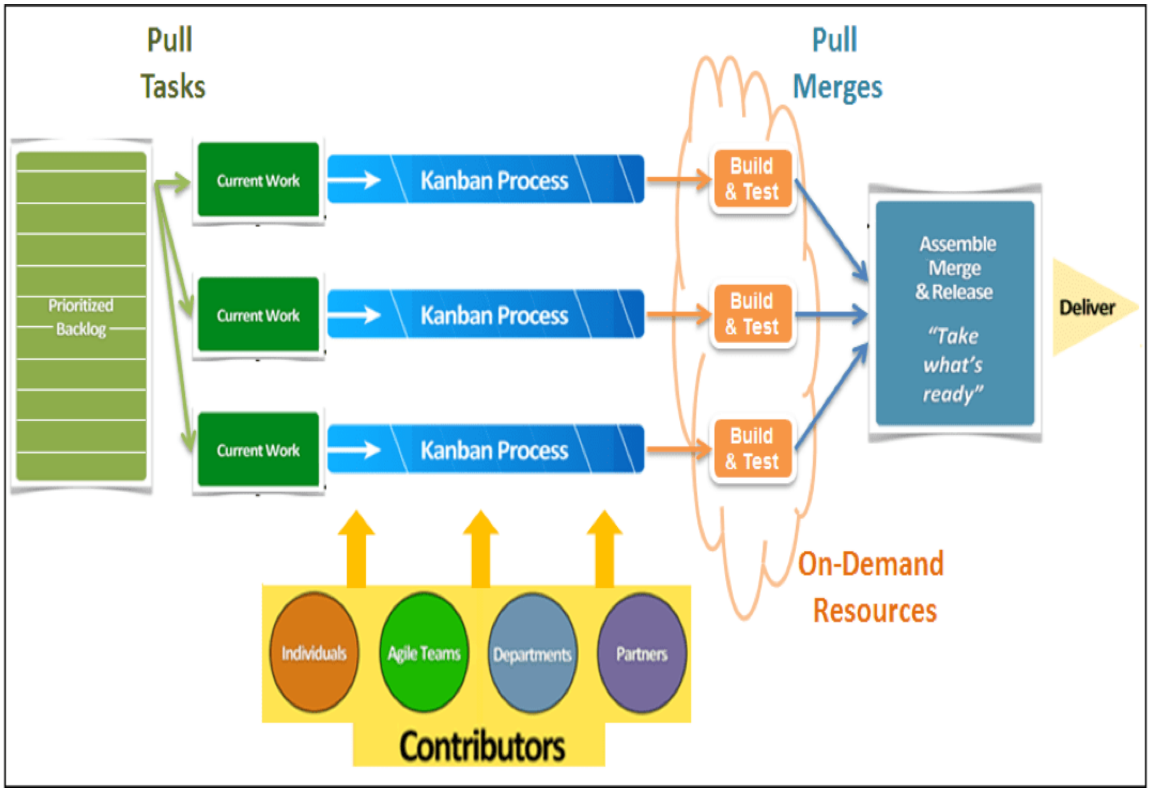

Figure 9- SDLC Model for the System [21]

3 System Design

3.1 Work Breakdown Structure (WBS)

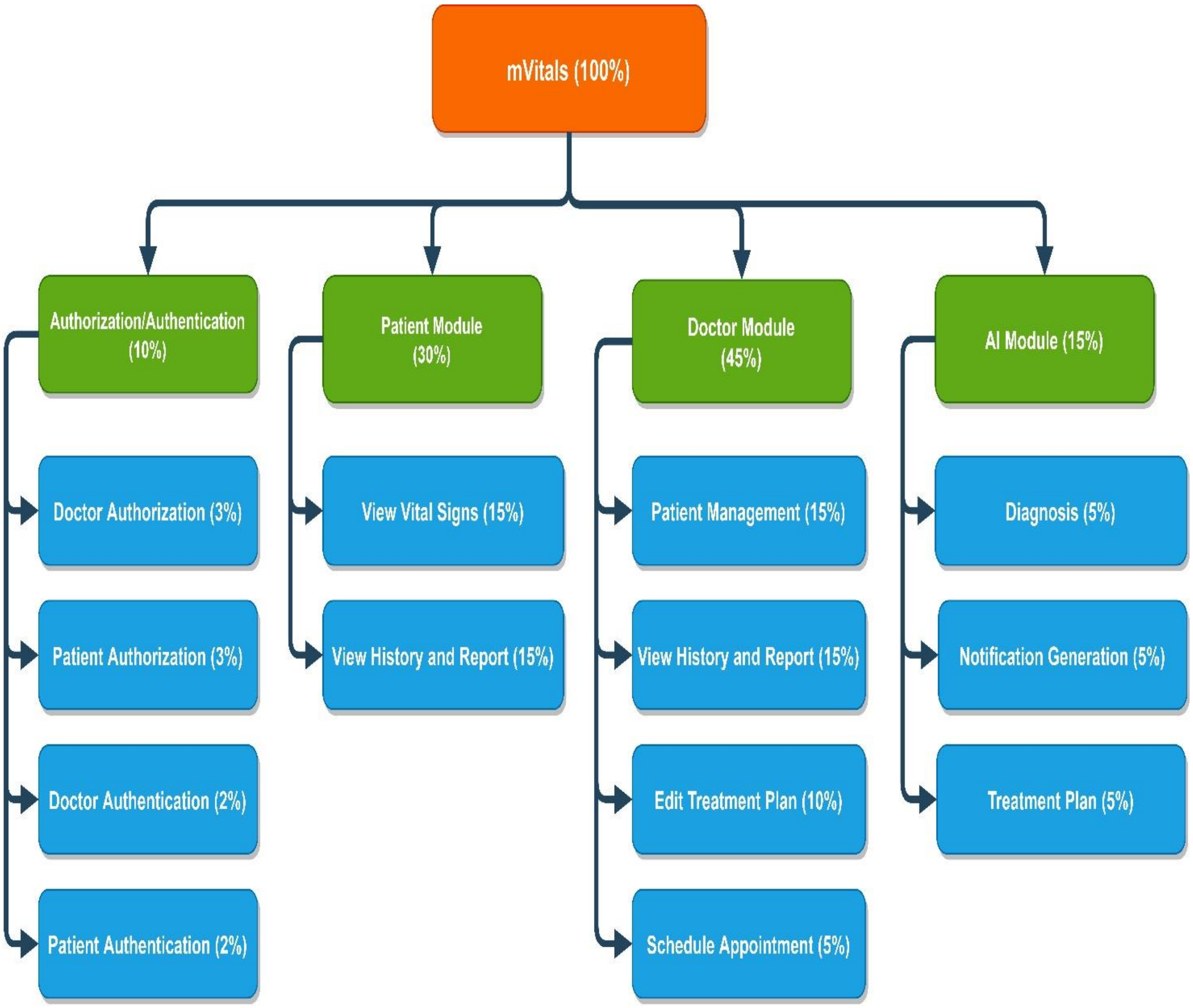

Figure 10- Work Breakdown Structure of System

3.2 Software Architecture

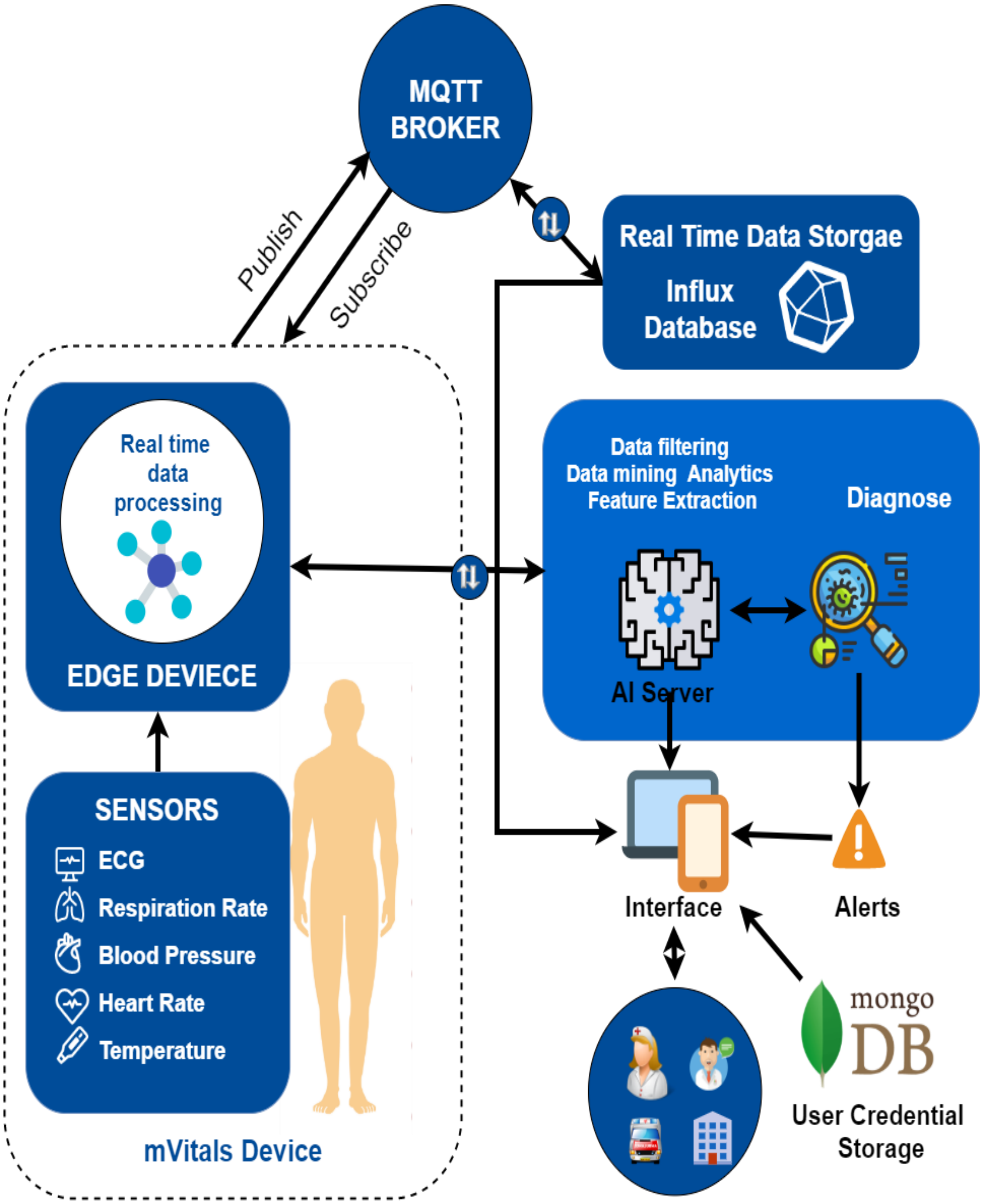

Figure 11- Architecture Diagram of mVitals System

Figure 5 shows an architecture of an IoT based health monitoring system that works on edge technology to get computation one step closer to their users for quicker response time, even with high latency. The system has a device called mVitals device which has further two modules. The first module is the sensors that the patient is going to be wearing to get the sensory data of five sensors used in this system. The second module is an edge device that is going to filter the data and apply analytics and filter extraction to generate a diagnose and send alerts to the interface where the user and the respective doctor can see. Data is stored temporarily in an edge device to reduce latency and bandwidth utilization. It is also accountable for data filtering, data analysis, and feature extraction for an AI module. In the AI module by using machine learning algorithms a diagnosis will be generated as well as a treatment plan. Diagnose and treatment plans will be accessible to both patients and doctors using an interface. Furthermore, it will aid in generating a report of the patient's condition. To achieve the edge technology mVitals uses MQTT protocol to sends the sensory data wirelessly to the server and save the real-time data in the Influx database and the personal data of patients and doctors are storing on Mongo DB database. After data being sent to the interface, the motioning can happen, meaning that the doctor and the patient or their caretaker can monitor the vital signs. A web-app of mVitals will be installed on patients' and doctors' phones or tablets through which both will be able to monitor the real-time physiologic parameters of a patient at the same time. The system will be able to generate alerts if a patient's condition gets critical. The system will allow the doctors to manually add diagnoses and change the course of treatment if needed.

3.2.1 Wireless Sensor Environment

All the sensors will be encapsulated inside a wireless device and this device will wirelessly connect with the system via a microcontroller. The patient will only need to wear the device and the sensors will automatically start collecting the data. As the device will be portable, patients will allow roaming freely in the network.

3.2.2 Hardware

The hardware used in this system is my signals HW kit. MySignals HW Kit includes multiple sensors. The system uses five sensors. Which includes Temperature, Spo2, blood pressure, body position and ECG sensor. Arduino UNO is used as a microcontroller and lastly Raspberry pi 3 as an Edge Device. Hardware is assembled something like this, sensors attached to their respective wires to my signals HW kit and Arduino is soldered by with my signals kit which sends sensory data to raspberry pi through USB 2.0 cable type a/b.

3.2.3 Real-time sensor Data Acquisition

The vital signs sensors will record the patient's physiological parameters using ECG Module, heart rate sensor, body temperature sensor, blood pressure sensor, Spo2 sensor, and a position sensor in real-time with the help of a microcontroller i.e., Arduino Nano. Filtration will be applied to this acquired data, to remove noise, and then it will be transferred to the edge network via an edge node i.e., Raspberry Pi.

3.2.4 Real-time Data Storage

Influx database is used to store the real time sensory data, that is sent through the edge device via MQTT protocol. Using MQTT protocol, different mVitals devices (sensor kits) can be used to publish data stream to the broker and doctors and patients can subscribe to these device's topic and view data. Storing these real time data streams will allow the user to view historic data of the patient, on their phones or computers.

3.2.5 Authentication and Authorization Module

Patients or Doctors will be able to create their accounts and then log in using the information they entered while creating their profile. Once logged in patients or doctors will be able to use services/features provided by the system.

3.2.6 Patient Module

In the Patient module, patients will be able to view their history, real-time monitor their vital signs. Patients will also be able to view the report generated by the system based on task progress, medical analysis, and doctor feedback. Patients will also be able to look at the treatment planned by their doctors. Patients will get notified in a critical situation as well as their doctors.

3.2.7 Doctor Module

In the Doctor module, the doctor register or delete a patient from their profile as well as the system database. The doctor will be able to view the list of patients registered to him. The doctor will also be able to view the real-time data of each patient. The doctor can also write a treatment plan, delete the plan, and modify the plan for each patient according to their needs. The doctor will be notified if any of the patient is in critical condition. Doctors will also be able to diagnose the patients and add their diagnostics report to the system. Doctors will also be able to view the report generated by the system based medical analysis.

3.2.8 AI Module

In the AI module, the AI agent will be able to acquire data from the wearable sensors, process the data, and extract required features from it using machine learning algorithms. AI agents will then generate a diagnosis and tell whether our patient is in critical condition or not. The AI agent will use machine-learning algorithms to analyse past data, to improve healthcare by predictions and better analytics.

3.2.9 Local Data Storage

The system is built on local edge network that stores data locally in InfluxDB time series database. Furthermore, patient's and doctor's credentials are stored in MongoDB, for user authentication and user profiles. To view the historic sensory data of a specific patient, a query can be requested, from user application, to InfluxDB.

3.2.10 Monitoring Module:

In the monitoring module, patients/doctors will be able to monitor the real-time data received from wearable sensors through a responsive application that can function on mobile phones and computers.

3.3 Database Diagram

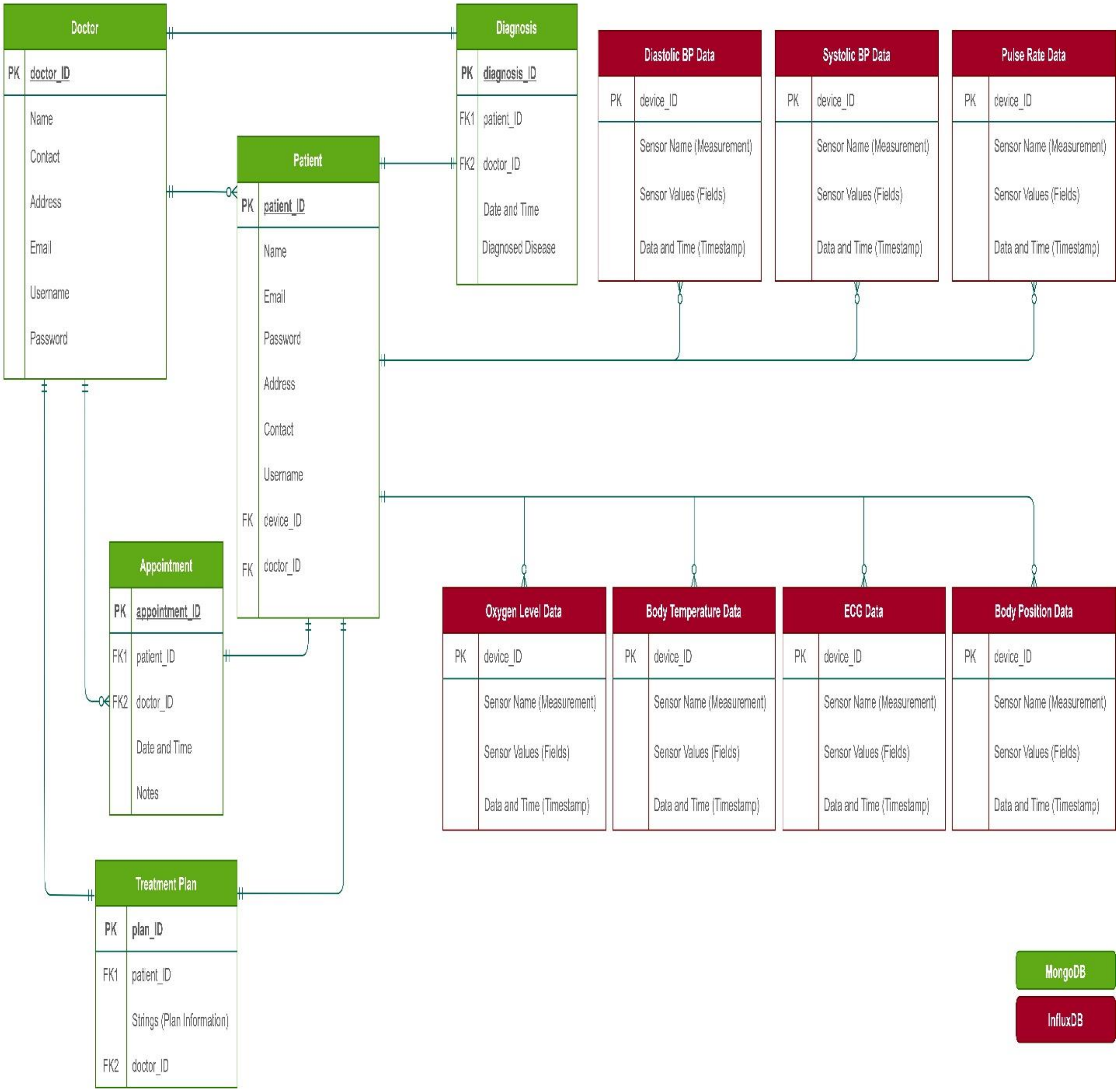

Figure 12- Database Diagram of mVitals

3.4 Network Diagram (Gantt chart)

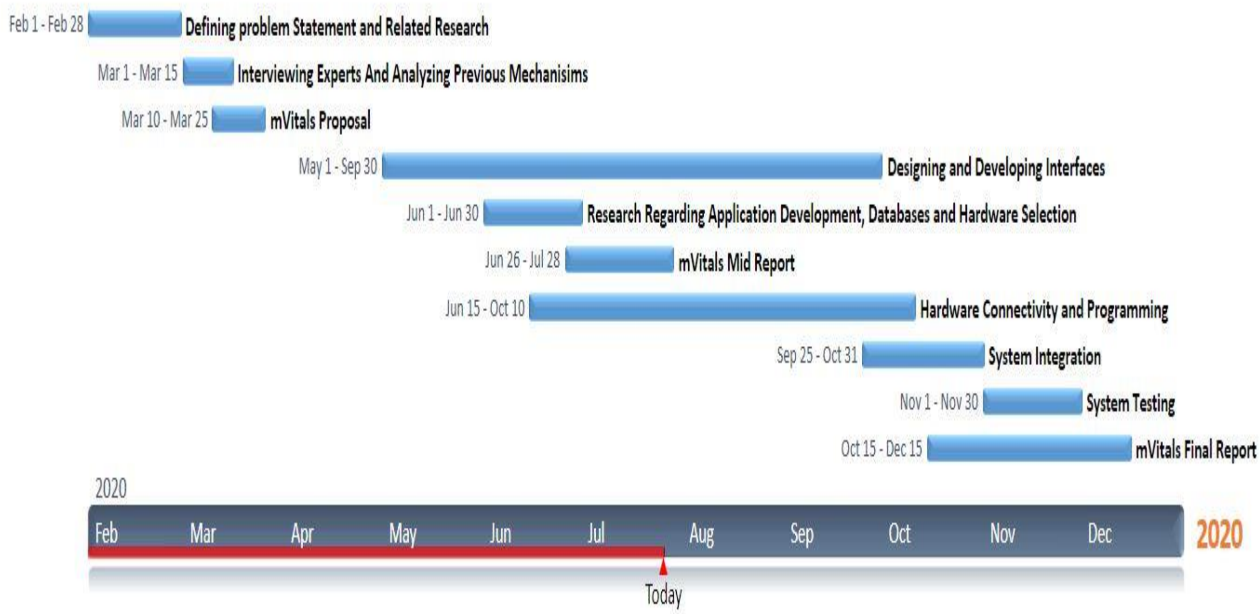

Figure 13- Gantt Chart of mVitals

4 System Implementation

First thing we realize in the pre-processing of our system is what will be the frontend of the system application and how can we implement it. So, we started our implementation by designing a web application for mVitals and then integrate it with the sensors and edge device.

4.1 Hardware Assembling

Hardware used in this system is my signals HW kit. MySignals HW Kit includes multiple sensors. The system uses five sensors, which includes Temperature, Spo2, blood pressure, body position and ECG sensor[22]. Arduino UNO is used as a microcontroller[23] and lastly Raspberry pi 3 [24] as an Edge Device. Figure 8 shows the hardware that is used to build mVitals device (sensor kit).

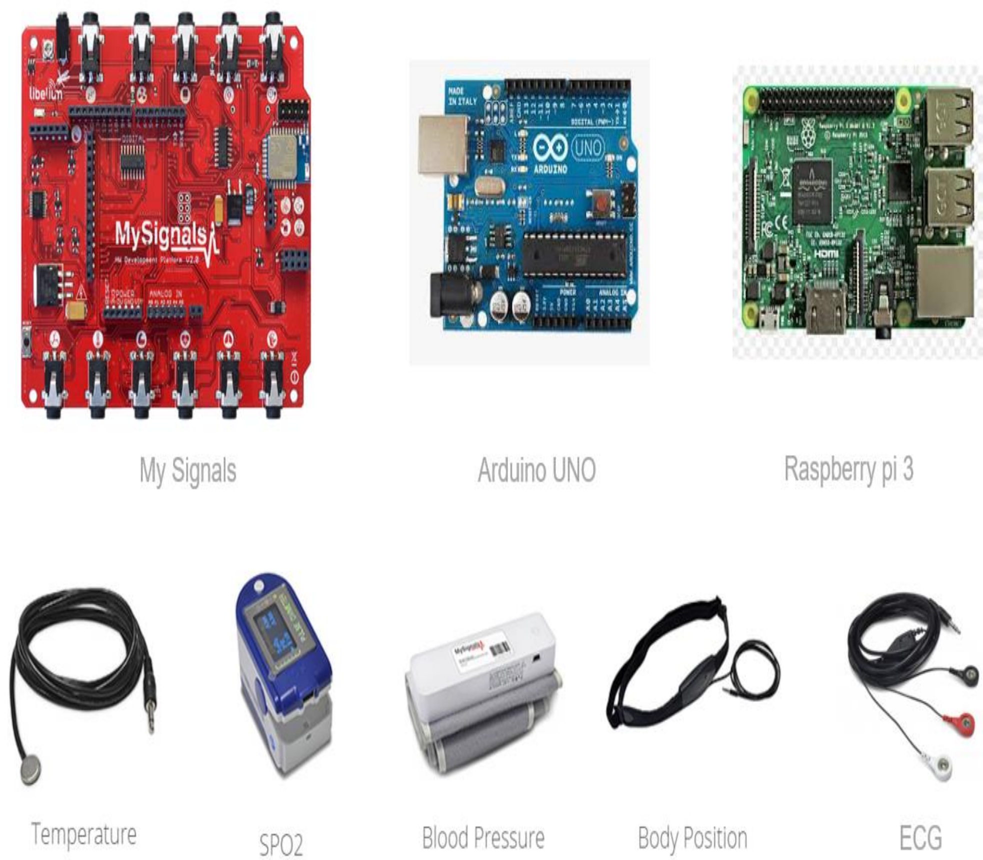

Figure 14- MySignals HW Kit with Sensors [22], Arduino Uno [23], Raspberry Pi 3 [24]

4.2 Circuit Diagram

Figure 15 explains the circuit diagram of how these devices are assembled. The sensors are attached with their respective wires to MySignals HW kit and Arduino is soldered by with MySignals kit, which sends sensory data to raspberry pi through USB 2.0 cable type a/b.

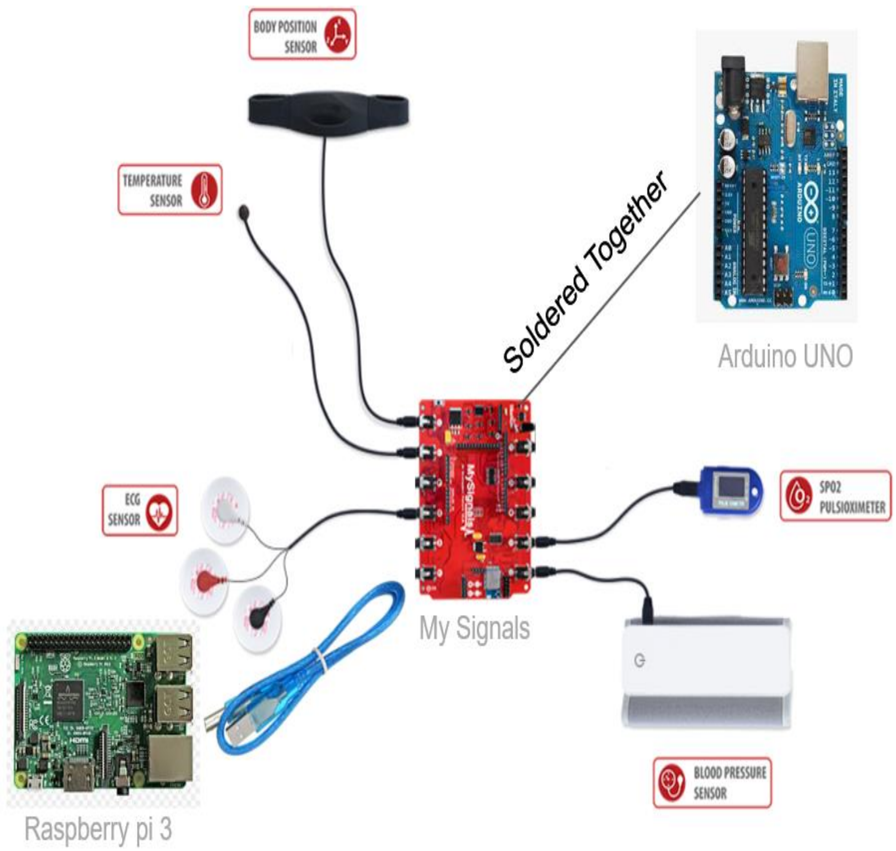

Figure 15- Circuit Diagram of mVitals

4.3 Real-time Sensory Data Acquisition

MySignals HW kit comes with a built-in MySignals.h library to work with the sensors attached to it. Arduino IDE was used to burn the code to the microcontroller. The code was designed to send five sensory data values to the raspberry pi 3.

Each of these sensors produces different output. Temperature sensor reads human body temperature and gives the value in Celsius, Spo2 gives pulse, breathing rate and oxygen level in blood. A blood pressure sensor gives systolic and diastolic pressure. Position Accelerometer monitors five different patient positions which includes standing/sitting, supine, prone, left and right. Lastly, for continuous heart rhythm monitoring, ECG sensor sends a continuous data stream[22]. This data stream is sent to raspberry pi in a string format in real-time. The steam is separated with a signal coma to identify each value. The first 3 values are for the blood pressure sensor that by default sends “a” if the sensor is off. When the button is turned on the blood pressure stops the real- time monitoring and sends “e” while the cuff band is filling with air. As soon as the function reads “i”, it gives the value of systolic and diabolic pressure, respectively. The second sensor is spo2 that gives “no data available’ when it is off. After turning on using the button it gives “@”, meaning the sensor is working but has not started to give the data yet. The message converts to “pulse per minute, oxygen in blood” from “@” when the sensor starts to receive the data. Thirdly the temperature sensor in Celsius and fourthly ECG output and lastly the position sensor’s value can be seen in the output stream from left to right.

4.4 Edge Networking

4.4.1 Data Acquisition

Sensor data from the mVitals kit is transferred to an edge node i.e., Raspberry Pi. The communication between these two devices is established using a serial transmission through a USB cable. Raw data is read from the cable at real time, and each sensor's value is separated. The separated values are stored in an array so that various checks can be applied in data filtering.

4.4.2 Data Filtering

Once the data is separated, various checks are applied depending on the length of the array. For example, if the length of the array is 5, then only 5 variables are initialized with respective values. If a value for some sensor is missing, then the program automatically initializes its variable with "n/a" for a uniform schema for the database and user application.

4.4.3 Data Publishing

The system uses JSON data format for versatility and uniform data pattern. Each variable, storing sensor's value, is converted into JSON format python's json.dump() function. After conversion, each value is published over the network with a specific topic. This topic includes the mVitals sensor kit ID and the sensor's name.

4.4.4 Network Broker

For handling all the data communication between edge nodes, a local MQTT broker is placed at the centre of the network. It is installed on an edge node and controls the publishers and subscribers. For ease of use, the system is using Mosquitto MQTT broker, as it only requires an application to be installed on a device and a simple command, "net start mosquito", to start the service.

4.5 Real-time Data Storage

For real time data storage, a time series database is used i.e., InfluxDB. To access the transferred data over the network, a subscriber is created on an edge node. This subscriber reads data through subscribing to each sensor's topic and stores it into a variable for further computation. The InfluxDB uses The Line Protocol for storing data. Once data is stored in a variable, it is converted to the Line Protocol. After the conversion, data is passed to write.points() function, which calls the POST function of the database.

4.6 Web-App Development

mVitals system web-app development was based on two modules, frontend, and backend. We have developed application's frontend in React.js, whereas for the backend we are working with Node.js.

4.6.1 Interface Frontend

We have built a web-app for the mVitals system. For the development of this web-application UI, the framework we have used is React. The main reason to select a web application over an Android or IOS native application is because it only requires two things i.e. a web-browser and an active internet connection, for efficient working. We have created adaptive layout of mVitals system's web-app with the help of responsive CSS for the development of react-components because it allows to render elements in the React DOM regardless of the size of the screens.

4.6.2 Frontend Functionalities

All the users can facilitate themselves by the efficient and real-time working of mVitals system by specifying their registered email and passwords. New users can register themselves by entering their credentials. On the dashboard panel, users can monitor the real-time sensory data coming from the sensors with the help MQTT broker. Users can edit profiles and new data is automatically modified in the database. All the doctors are allowed to set appointments for their patients. Historical data can be viewed on the historical data panel in the form of graphs to check and monitor all the fluctuations in the sensory data readings.

4.6.3 Interface Backend

For the implementation of mVitals system backend, we have used Node.js. The web-app frontend and backend are communicating together with help of restful APIS. With the help of HTTP request, these restful API's allow the system to access and store data. The database which is used to store all the users' credentials is Mongodb. When the user enters his registered email and password in the login page fields, a Get operation is called to check the validity and authenticity of the user by comparing the entered data and saved data in the database. In case of a wrong email or password, the system does not allow unauthentic users to enter the mVitals system.

4.7 Problems Faced During Implementation

4.7.1 Inconsistent Sensory Data Stream Issue

One of the challenges we faced while acquiring the sensor data is that the stream was inconsistent. The reason for that is because blood pressure only sends data when it is tuned one using the button same is the case with spo2 sensor. To overcome the problem, data filtration is applied at the edge device that is raspberry pi 3 in this scenario.

```
23.68,,Prone position
23.48,,Prone position
i,Diastolic: 78,Systolic: 96,Pulse/min: 91,,81,90%,,23.98,,1.77,,Prone position
a,,81,90%,,23.98,,1.95,,Prone position
```

Figure 16- Inconsistent Data Stream in String Format

4.7.2 Data Acquisition and Filtering Issue

We faced some problems while reading real-time data from the MySignal hardware kit on the edge node. We used wireless communication, but it required a separate wireless module for the Arduino board. This affected the overall size of the mVitals sensor kit, which would have made it difficult for the patient. So, we solved the communication with a serial USB cable, from Arduino to Raspberry Pie.

Furthermore, the data read from the Arduino was irregular due to compatibility issues of the sensors. Some of the sensors were digital and some of them were analog. We had to try a lot of checks and data filtering to tackle the irregular values. With time, we cleaned the codes for better performance and readability, for future improvements. We separated each sensor's value and stored them in an array. Then applied simple checks based on the length of the array. After data filtering, it was converted into JSON format for versatility and a uniform schema.

4.7.3 Problems Faced in Data Storage

Whenever a system has a real-time data communication, there is a synchronization problem. To no exception, we faced such problems in storing time-series data into the database. At times, mVitals sensor kit was producing data very quickly and the system was having latency, which led to data loss over the network. Sometimes the process of storing data was so quick, as compared to data generation, that junk values were being stored.

So, we had to create some specific events with which each process, i.e., data publishing and data storage, ran. For example, in database script, we set the calling of POST function (of database) to whenever the program received a message from a particular topic. This was the problem of storage of junk values ("N/A") was avoided.

4.8 Interface UI

4.8.1 Login Panel

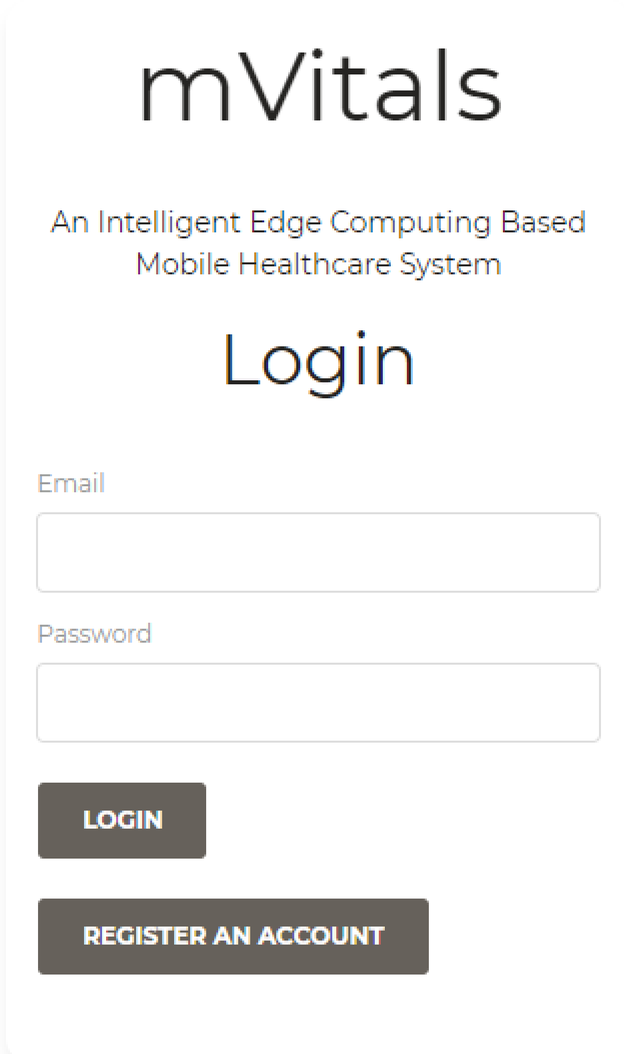

Figure 17- Login Page of mVitals

Figure 17 shows the login page of the mVitals web-app. Only the authentic users can log in to the system with the help of email and password.

4.8.2 Registration Page

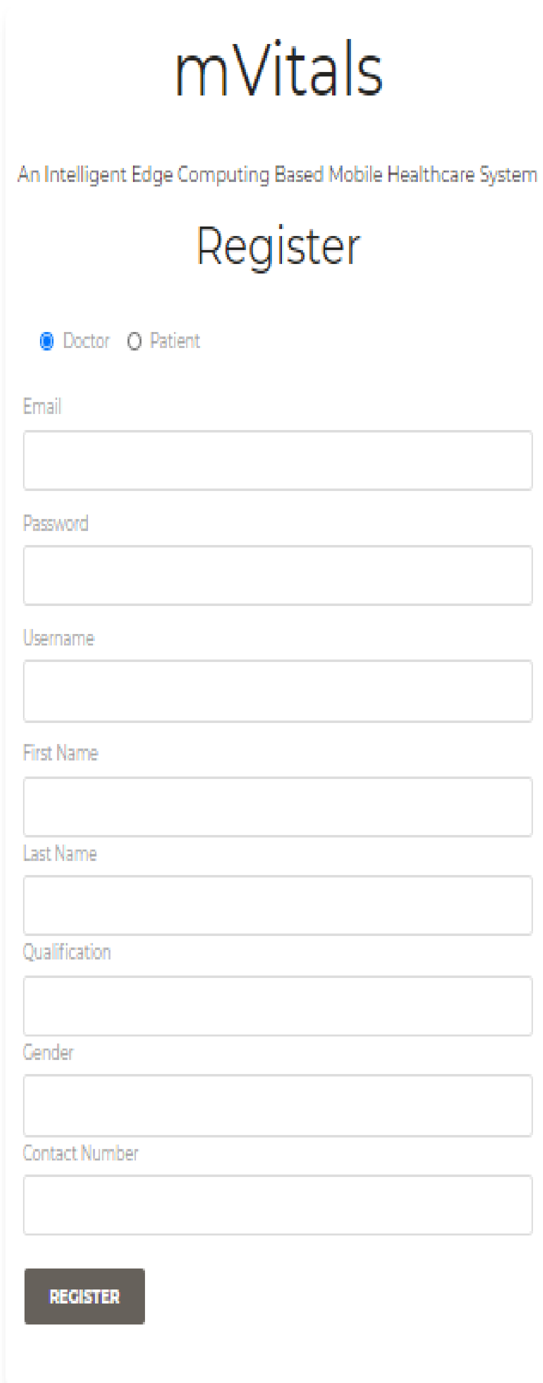

Figure 18- Registration Page for Doctor

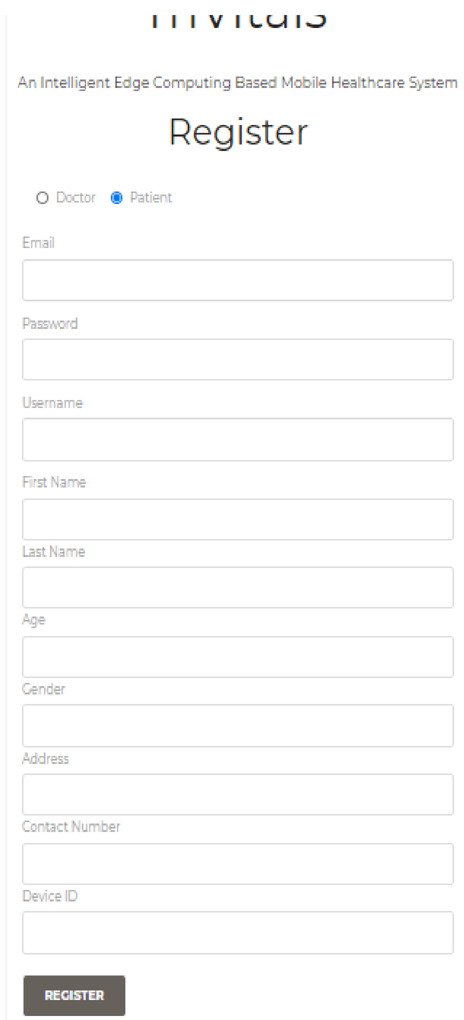

Figure 19- Registration Page for Patients

Figure 18 and figure 19 are showing the registration page for the doctors and patients, respectively. The new users must select the type of user and that specific form will be displayed on the registration page. To register himself with the system, the user needs to enter all the fields.

4.8.3 Dashboard Page

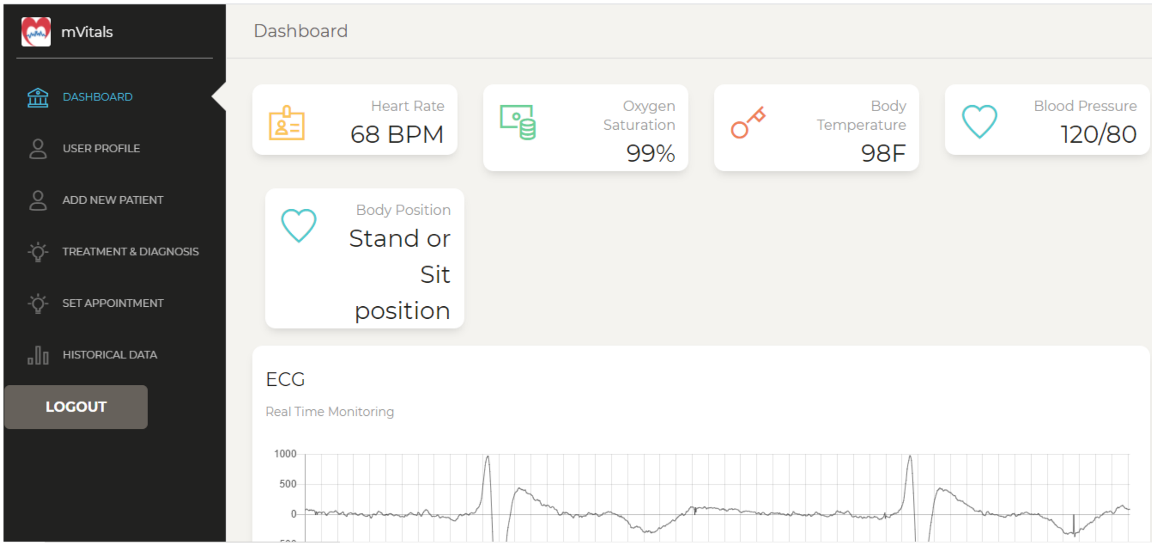

Figure 20- Dashboard Page of mVitals

Figure 20 shows the dashboard of the mVitals system. On this panel user can view the real-time data coming from the sensors with the help of MQTT broker. On the left side, a sidebar has been attached, with the help of which the users can easily jump to other panels.

4.8.4 User Profile Page

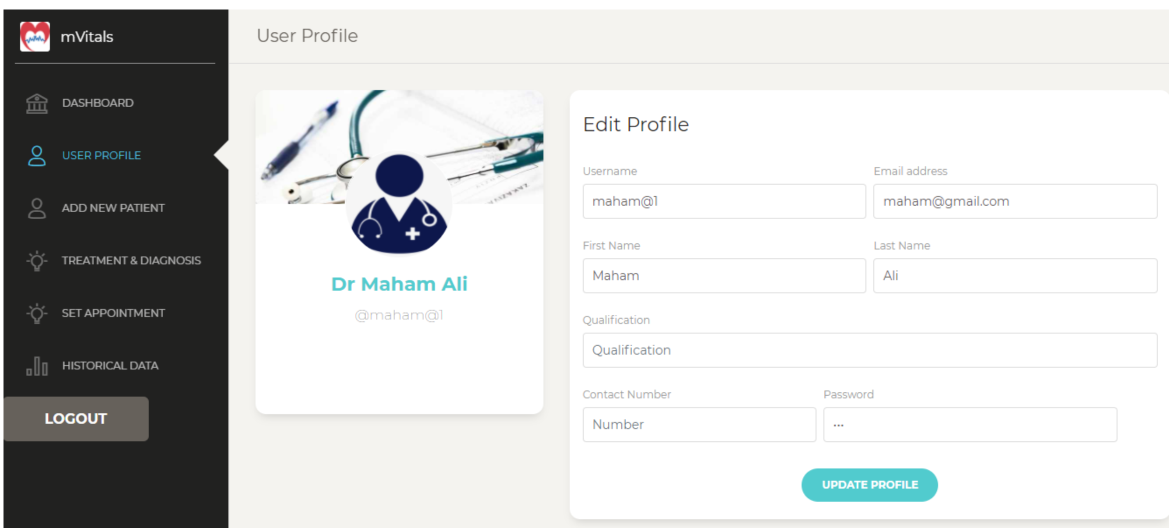

Figure 21- User Profile Page

On the user profile page, users cannot only view their profiles but also modify them. With the help of restful APIS, new data will be updated in the database.

4.8.5 Add New User

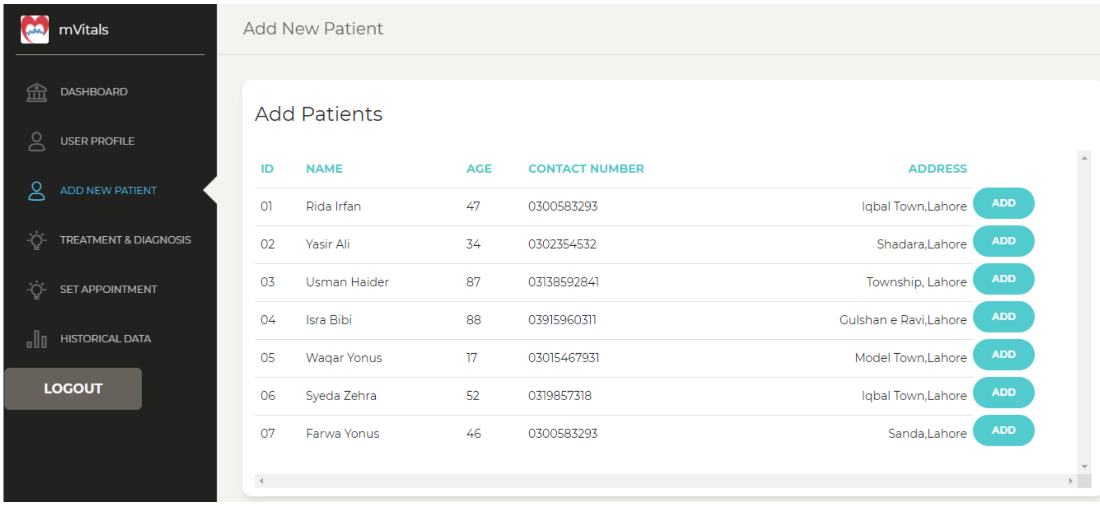

Figure 22- Add New User Page

On this panel, the list of all the patients will be displayed in front of the doctors and doctors add the required patient by clicking the add button.

4.8.6 Treatment and Diagnosis

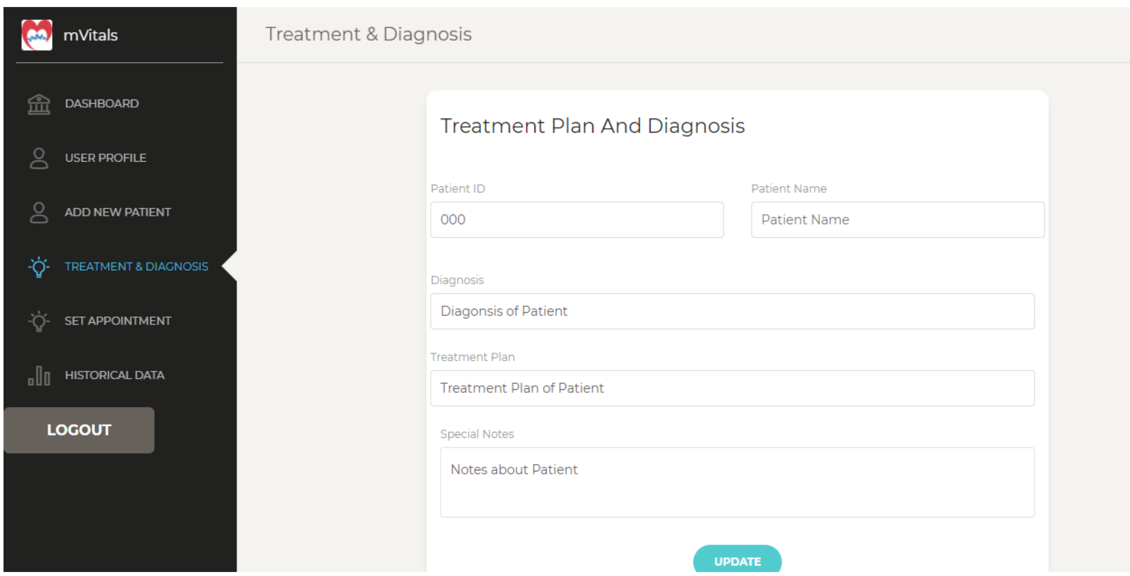

Figure 23- Treatment and Diagnosis Page

Figure 23 shows the treatment and diagnosis page of mVitals system. With the help of machine learning and artificial intelligence, the system diagnoses the problem and then suggests a suitable treatment plan for the patients and doctors are allowed to modify these treatment plans.

4.8.7 Set Appointment Page

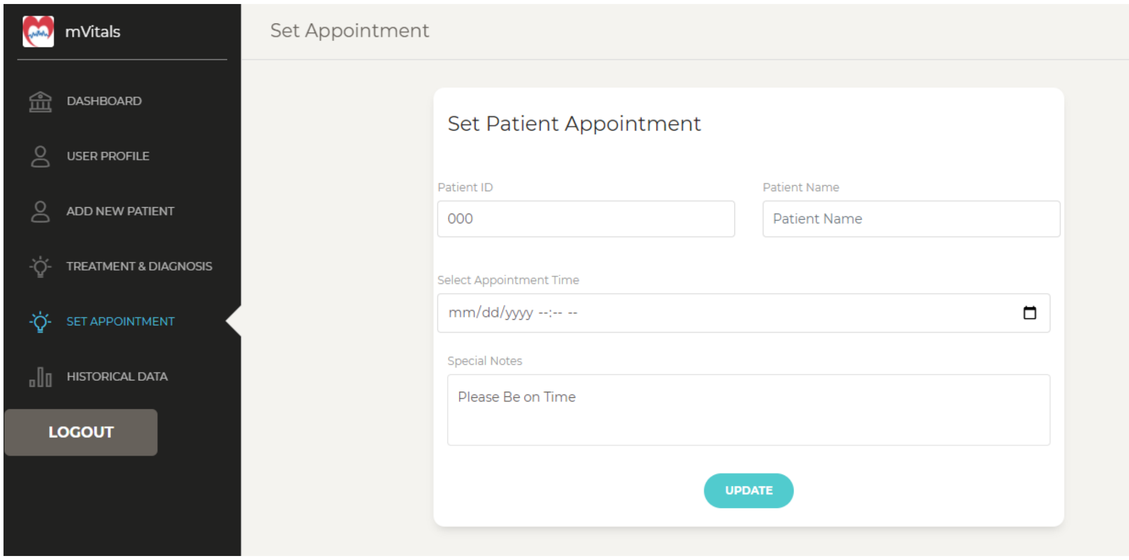

Figure 24- Set Appointment Page

As figure 24 shows the set appointment page where the doctors can easily set appointments for the required patients.

4.8.8 Historical Data Page

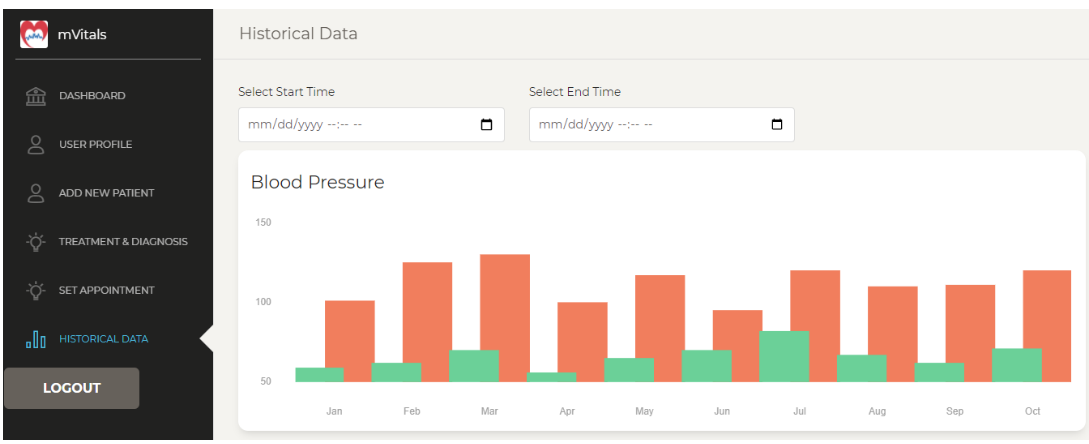

Figure 25-Historical Data Page 1

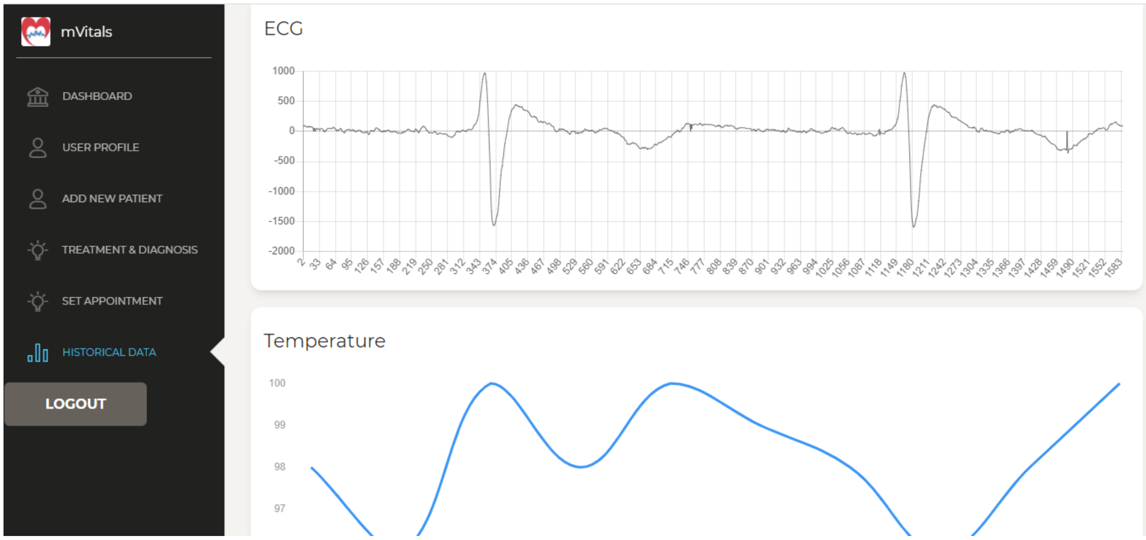

Figure 26- Historical Data Page 2

In the figure 25 and 26, the historical data page has been displayed. On this page, the historical time series data of the patient's sensors is displayed.

4.8.9 Logout

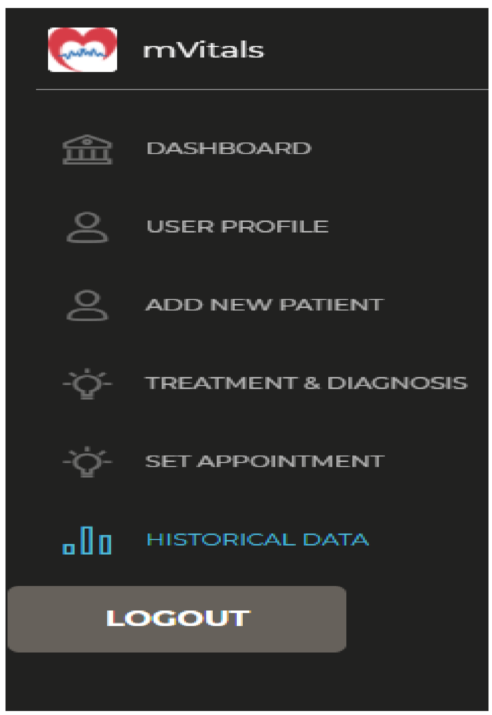

Figure 27- Logout

By clicking the logout button on the sidebar, which is shown in the figure 27, the user logouts from the system and jumps to the login page.

4.9 Tool and Technologies

Following are the list of all the tools, technologies and libraries which are using in the implementation of this project.

4.9.1 Hardware

- MySignals HW Kit
- Temperature Sensor
- Spo2 Sensor
- Blood Pressure Sensor
- Body Position Sensor
- ECG Sensor
- Arduino Uno
- Raspberry Pi 3

4.9.2 Languages

- React
  - React is used for the development of mVitals web-application frontend.
- Node.js
  - Node.js is used for the backend development of mVitals system.
- C++
  - For sensors configuration and data acquisition, we have used C++.
- Python
  - Python is a high-level language, used for application development, machine learning, data science, data filtering, etc. It is an object oriented and interpreted language. It can also be used for edge computing.
- JavaScript
  - A high-level language used for frontend and backend development of user application.

4.9.3 Libraries, Modules and Protocols

- MQTT Protocol
  - MQTT [25] is an easy to implement and a light weight publish-subscribe protocol that allows it clients to transfer data from one node to another. It usually uses TCP/IP to transmit data.
- mySignals.h Library
  - mysignals.h is a high-level library that easily manages 16 sensors integrated in the board. mVitals only uses five of them. [26]
- wire.h Library
  - The wire library allows you to communicate with I2C devices, often also called "2 wire" or "TWI" (Two Wire Interface). [27]
- spi.h Library
  - This library allows you to communicate with SPI devices, with the Arduino as the master device. [28]
- PAHO MQTT
  - This python library [29] provides a Client class which enables the user applications to connect with the MQTT broker. It allows the application to publish messages to a topic, and to subscribe to topics to access published messages.

- PySerial
    - This python module [30] Allows the python application to establish a serial communication with an external device. It allows thee application to read data from a serial port coming from, for example, a USB cable.
- RPi GPIO
    - It [31] is a python library written for Raspberry Pi. It allows applications on Raspberry Pi to control and query the GPIO (General Purpose Input Output) pins of the board.
- InfluxDB-Python
    - It [32] is a client, written in python, for applications to interact with InfluxDB.
- Datetime Module
    - Datetime [33] is a python module that allows applications to work with data and time. There are various functions to work with time, time intervals and dates.
- JSON Module
    - JSON [34] is a python module that allows applications to work with JSON (JavaScript Object Notation). It is used to convert python objects into JSON format and vice versa.

4.9.4 Databases

- Influxdb
    - This open-source time-series database has been optimized for fast and high availability storages, and retrieval of real time series data.
- Mongodb
    - The user's personal information has been stored in Mongodb.

5 System Testing

5.1 Test Cases

Following are the test cases for different modules of our system.

5.1.1 TC-01: Doctor Registration

Table 44- TC-01: Doctor Registration

TC-ID: TC-01	**Author of Test Case:** Noor Jaffri
Priority: High	**Created At:** 02/10/20
Name: Doctor Registration	**Executed By:** Noor Jaffri
Title: Add a new doctor to the system and verifying their given information	**Executed At:** 02/10/20

Description: Check the proper working of Registration page.

Pre-conditions: Doctor should be navigated to the Registration page.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/ Fail)	Notes
1	Go to the Registration page		An account for the user should be created	Account created successfull y	Pass	
2	Email	Dotor1@gmail.com			Pass	
3	Password	“123@567”			Pass	
4	Age	“25”			Pass	
5	Gender	“Male/Female”			Pass	
6	Qualification	“MBBS”			Pass	
7	Username	“doctor123”			Pass	
8	Click on the “Register” button				Pass	Successful account creation

Post Condition:

Doctor has successfully registered to the mVitals system.

5.1.2 TC-02: Patient Registration

Table 45- TC-02: Patient Registration

<table>
<tr><td colspan="3">TC-ID: TC-02</td><td colspan="4">Author of Test Case: Noor Jaffri</td></tr>
<tr><td colspan="3">Priority: High</td><td colspan="4">Created At: 02/10/20</td></tr>
<tr><td colspan="3">Name: Patient Registration</td><td colspan="4">Executed By: Noor Jaffri</td></tr>
<tr><td colspan="3">Title: Add a patient user to the system and verifying their given information</td><td colspan="4">Executed At: 02/10/20</td></tr>
<tr><td colspan="3">Description: Check the proper working of Registration page</td><td colspan="4"></td></tr>
<tr><td colspan="7">Pre-conditions: Patient should be navigated to the Registration page</td></tr>
<tr><th>Step</th><th>Test Steps</th><th>Test Data</th><th>Expected Result</th><th>Actual Result</th><th>Status (Pass/ Fail)</th><th>Notes</th></tr>
<tr><td>1</td><td>Go to the Registration page</td><td></td><td>An account for the patient should be created</td><td>Account created successfully</td><td>Pass</td><td></td></tr>
<tr><td>2</td><td>Email</td><td>Patient1@gmail.com</td><td></td><td></td><td>Pass</td><td></td></tr>
<tr><td>3</td><td>Password</td><td>“123@567”</td><td></td><td></td><td>Pass</td><td></td></tr>
<tr><td>4</td><td>Age</td><td>“17”</td><td></td><td></td><td>Pass</td><td></td></tr>
<tr><td>5</td><td>Gender</td><td>“Female/Male”</td><td></td><td></td><td>Pass</td><td></td></tr>
<tr><td>6</td><td>Device ID</td><td>“98usxd33”</td><td></td><td></td><td>Pass</td><td></td></tr>
</table>

7	Click on the “Register” button				Pass	Successful account creation

Post Condition: Patient has successfully registered to the mVitals system.

5.1.3 TC-03: Login

Table 46- TC-03: Login

TC-ID: TC-03	**Author of Test Case:** Noor Jaffri
Priority: High	**Created At:** 02/10/20
Name: Login	**Executed By:** Noor Jaffri
Title: To verify the details of the user and to open their account	**Executed At:** 02/10/20
Description: Check the proper working of the login page.	
Pre-conditions: User should provide a valid email and password.	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Go to the Login page		User should be able to log in	User is navigated to their dashboard after logging in	Pass	
2	Provide email	Doctor@gmail.com			Pass	

3	Provide password	“123@567”			Pass	
4	Click login button				Pass	

Post Condition:

Doctor is taken to their dashboard.

5.1.4 TC-04: Reset Password

Table 47- TC-04: Reset Password

TC-ID: TC-04				**Author of Test Case:** Noor Jaffri		
Priority: High				**Created At:** 02/10/20		
Name: Rest Password				**Executed By:** Noor Jaffri		
Title: To reset the login password of the user account				**Executed At:** 02/10/20		
Description: Check the proper working of the reset password page.						
Pre-conditions: User should provide a valid email and password.						
Step	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**	**Notes**
1	Go to the Edit Profile page		Edit Profile page should be opened	User is navigated to login page	Pass	
2	Click change password		Reset password page should be opened	User is navigated to Reset password page	Pass	
3	Provide email	Doctor1@gmail.com			Pass	
4	Click Edit button		Rest password link should be	Rest password link sent on user	Pass	

			sent on email	email address		

Post Condition: Doctor is taken to their dashboard.

5.1.5 TC-05 Edit Profile

Table 48- TC-05 Edit Profile

TC-ID: TC-05	**Author of Test Case:** Noor Jaffri
Priority: Low	**Created At:** 03/10/20
Name: Edit Profile	**Executed By:** Noor Jaffri
Title: Change user information	**Executed At:** 03/10/20
Description: Allow the user to change the information of their profile.	
Pre-conditions: User should be available in the database and logged in the system.	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Go to the user profile page		Users profile page should be opened	User is navigated to the profile page	Pass	
2	Change the desired information	Name: Doctor Qualification: MBBS			Pass	

3	Click on Edit Button				Pass	
Post Condition: User`s profile is edited and successfully updated.						

5.1.6 TC-06 Add Patient

Table 49- TC-06 Add Patient

TC-ID: TC-06	**Author of Test Case:** Noor Jaffri
Priority: High	**Created At:** 03/10/20
Name: Add Patient	**Executed By:** Noor Jaffri
Title: Add Patient	**Executed At:** 03/10/20
Description: Allow the doctor to add their patient in the system.	
Pre-conditions: Doctor should be available in the database and logged in the system.	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Go to the user Add Patient page		Patients list page should be opened	Doctor is navigated to the patients list page	Pass	

2	Click Add new patient button	List of all the patients should be displayed			Pass	
3	Click on Add Button		Patient is added successfully pop up window should appear		Pass	
Post Condition: Patient is now associated with the doctor.						

5.1.7 TC-07 Treatment Plan and Diagnosis

Table 50- TC-07 Treatment Plan and Diagnosis

TC-ID: TC-07	**Author of Test Case:** Noor Jaffri
Priority: High	**Created At:** 03/10/20
Name: Data Acquisition	**Executed By:** Noor Jaffri
Title: Collect patient`s sensors data for diagnosis	**Executed At:** 03/10/20
Description: Check the proper working of mVitals sensors device.	
Pre-conditions: mVitals device should be working and configured.	

Step	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**	**Notes**

1	Wear Smart mVitals Device		mVitals device should be worn	Device is worn by the patient	Pass	
2	Configure and connect mVitals device with the system with the help edge device MQTT broker		mVitals should be connected to the system	mVitals device is connected to system	Pass	
3	Fetch Real-time sensory data from the device		Real-time data should be coming to the system on the real time.	Real-time data is coming from the broker.	Pass	
4	Process Coming Sensory Data		Real-time processing should be made on the data for treatment plan	Real-time data processing is made for the treatment plan	Pass	
5	Display treatment plan and diagnosis		Real-time treatment plan and diagnosis should be displayed on the treatment plan and diagnosis page	Treatment plan and diagnosis is displaying in on the page	Pass	

Post Condition: Treatment plan and diagnosis is successfully displayed.

5.1.8 TC-08 Historical Data

Table 51- TC-08 Historical Data

TC-ID: TC-08	**Author of Test Case:** Noor Jaffri
Priority: High	**Created At:** 04/10/20
Name: Data acquisition	**Executed By:** Noor Jaffri
Title: Collect's patient data from the database.	**Executed At:** 04/10/20
Description: Display the historical sensory data of the user on the historical data page.	
Pre-conditions: User should be available in the database and logged in the system.	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Go to historical data page		Historical data page should be opened	User is navigated to the Historical data page	Pass	
2	Select specific time and date	Starting time: 7:33 PM Ending time: 8:00 PM Date: 11/10/20	Time and date should be selected	User easily selected starting and ending time with date	Pass	

4	Click Enter Button				Pass	

Post Condition: Historical data of the patient is displayed.

5.1.9 TC-09 Set Appointment

Table 52- TC-09 Set Appointment

TC-ID: TC-09	**Author of Test Case:** Noor Jaffri
Priority: Low	**Created At:** 03/10/20
Name: Set Appointment	**Executed By:** Noor Jaffri
Title: Doctor setting appointment for patient	**Executed At:** 03/10/20
Description: Allow the doctor to set an appointment for the patient.	
Pre-conditions: Doctor and patient should be available in the database and doctor logged in the system.	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Go to the user set appointment page		Set appointment page should be opened	User is navigated to the Set appointment page	Pass	

2	Select the specific patient	Name: Patient ID: 010212000			Pass	
3	Select specific time and	Time: 7:00 PM Date: 17/10/20			Pass	
4	Click Set button				Pass	
Post Condition: Appointments for the patient is successfully made.						

5.1.10 TC-10 View Vital Signs

Table 53- TC-10 View Vital Signs

TC-ID: TC-10	**Author of Test Case:** Noor Jaffri
Priority: High	**Created At:** 04/10/20
Name: View Vital Signs	**Executed By:** Noor Jaffri
Title: Real-time sensory data display	**Executed At:** 04/10/20
Description: Allow user to view real-time sensory data from the mVitals device	
Pre-conditions: User should be available in the database and logged in the system.	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes

1	Go to the user Dashboard page		Dashboard page should be opened	User is navigated to the Dashboard page	Pass	
2	Data fetching from the MQTT broker				Pass	
4	Displaying real-time values of sensors				Pass	
Post Condition: User's sensory data is displaying to the dashboard.						

5.1.11 TC-011 Edit Treatment Plan and Diagnosis

Table 54- TC-011 Edit Treatment Plan and Diagnosis

TC-ID: TC-011	**Author of Test Case:** Noor Jaffri
Priority: High	**Created At:** 03/10/20
Name: Data Acquisition	**Executed By:** Noor Jaffri
Title: Collect patient`s sensors data for diagnosis	**Executed At:** 03/10/20
Description: Check the proper working of mVitals sensors device.	
Pre-conditions: mVitals system should display the treatment plan and diagnosis for the patient.	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Configure and connect mVitals device with the system with the help edge device mqtt broker		mVitals should be connected to the system	mVitals device is connected to system	Pass	
2	Fetch Real-time sensory data from the device		Real-time data should be coming to the system on the real time.	Real-time data is coming from the broker.	Pass	
3	Process Coming Sensory Data		Real-time processing should be made on the data for treatment plan	Real-time data processing is made for the treatment plan	Pass	
4	Display treatment plan and diagnosis		Real-time treatment plan and diagnosis should be displayed on the treatment plan and diagnosis page	Treatment plan and diagnosis is displaying in on the page	Pass	

5	Doctor can edit the treatment plan and diagnosis				Pass	
Post Condition: Treatment plan and diagnosis is successfully modified.						

5.1.12 TC-12: Logout

Table 55- TC-12: Logout

TC-ID: TC-12	**Author of Test Case:** Noor Jaffri
Priority: High	**Created At:** 04/10/20
Name: Logout	**Executed By:** Noor Jaffri
Title: To terminate the activity session for the user	**Executed At:** 04/10/20
Description: Check the proper working of the logout page.	
Pre-conditions: User should click the logout button.	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Click to logout button		User should be able to log out	User is navigated to their login page after logged out	Pass	

2	Click the confirm button				Pass	
Post Condition: User is taken to the login page.						

6 Individual Contribution

6.1 Hafsa Khan

Table 56- Contribution By Hafsa Khan

#	Activity	Decryption	Duration
Hardware Assembling			
1	Research	Research regarding different hardware, to find the suitable hardware for this system.	2 weeks
2	Individual sensor monitoring	Run a code for each sensor and read the values	2 weeks
3	Circuit diagram	Designing a circuit diagram and combining the hardware modules	1 week
4	Assemble edge device	Sending data from sensors to edge device	1 day
Development			
5	Sensors congregate monitoring	Assembling a code together that gives reading of 5 sensors using mysignals.h library. In Adruino IDE.	1 month
Testing			
6	Sensory data monitoring	Testing real-time data of sensor and applied changes if it crashed.	1 week
Machine Learning			

7	Dataset	Finding patient's dataset for the diagnose of blood pressure.	1 week
8	Model training	Training different models on that dataset.	1 week
ML Testing			
9	Accuracy	Achieve high accuracy that best fit the ML model	1 week
10	Prediction	Generating prediction of the diagnose.	1 week
Documentation			
11	Documentation	Documenting the work done throughout the project	1 week

6.2 Muhammad Irtiza

Table 57- Contribution by Muhammad Irtiza

#	Activity	Decryption	Duration
Edge Networking			
1	Network Design and Implementation	Designed different edge networks and analysed their performance, ease of use and simplicity, for environments such as hospitals. Implemented these networks to test which was the best for our set of challenges.	3 weeks
2	Data Acquisition from Sensor Kit	Wrote scripts in python using Thonny integrated environment. Data is read via USB 2.0.	2 weeks
3	Data Filtering on Edge Node	Data filtration has been applied to the acquired data	2 weeks

		for a uniform schema for the database and user application.	
Server/Broker			
4	Creating MQTT Broker/ Server	For controlling the communication between the edge nodes, a local MQTT broker is placed at the centre of the network. Mosquitto MQTT broker. is running on an edge node that controls the publishers and subscribers.	3 weeks
5	Wireless Data Transmission (Publishing Sensor Values)	After data filtering, each sensor's value is published on a specific topic including device ID and sensor name.	2 weeks
Data Storage			
6	Real-time Data Storage	For real time data storage, a time series database is used i.e., InfluxDB. To access and store the transferred data over the network, a subscriber is created on an edge node. This subscriber reads data through subscribing to each sensor's topic and formats that data further and stores it into the database.	1.5 month
Testing			
7	Network Testing	Tested the edge network for latency, disconnection, load, synchronization between nodes and data integrity.	3 weeks
8	Querying Database	Tested the stored data with dummy querying, to check	1 week

		latency, complexity, and data integrity.	
Research and Documentation			
9	Research	Carried out research and comparisons of technologies and tools, to analyse which was the best for each scenario. Analysing various tools in different situations for their performance and ease of use.	2 weeks
10	Documentation	Documenting the work done and procedures followed, throughout the project.	3 weeks

6.3 Noor Jaffri

Table 58- Contribution by Noor Jaffri

#	Activity	Decryption	Duration
Development			
1	React web-App UI development	Creation of web-application with the help of responsive CSS.	1 month
2	React Web-App Backend implementation	Designing of restful-API for backend development.	1 month
3	Web-App and edge device connection	Use of MQTT broker for real-time data communication.	2 weeks
4	Users' Credentials Storage	All users’ credentials are stored in Mongodb.	2 weeks
5	Historical Data	Connection of mVitals web-app and InfluxDB with the help of HTTP request.	1 week

Testing			
6	Real-time Data Monitoring on Web-app	Write python script to send sensory data values to the MQTT broker and then display it on the web-app.	1 week
7	Functional Testing	Tested mVitals web-app functionalities on different devices.	1 week
8	Performance Testing	Monitored mVitals web app performance on different devices	1 week
Documentation			
9	Research	Carried out research and comparisons of technologies and tools that would work on different devices.	2 weeks
10	Documentation	Documenting the work done throughout the project	1 week

7 Conclusion

The proposed system will be mobile and wearable enabling wireless communication to continuously monitor vital signs and ECG of a patient. The system will utilize Bluetooth for communication between devices making it highly portable. Due to portability, the patient wearing a wearable sensor-device will be able to move freely while the caretaker/doctor can view vital signs information from his mobile device without being physically present with the patient. The system will store records of patients' history as well as real-time monitoring data for future reference. The purpose was to present a design of a middleware platform to support better data acquisition operations in mobile health care monitoring environments. The data will be stored in a secure and efficient database system and only authorized people will have the remote accessibility of a patient's medical records, regardless of whereabouts with high processing speed. Incorporation of Clinical Decision Support System (CDSS) will enhance the patient's diagnosis and provide better analytics using deep learning techniques. The system will aid in devising treatment plans for patients based on deep learning and past data. This cost effective and efficient system will represent data in an easy-to-understand manner. Alerts and notifications will be sent to the concerned person in case of any critical change in the health parameters of patients so that the patient never stays unobserved. The aim is to improve the quality of life-related to health care, with the help of continuous monitoring. The health caretakers can screen, analyse, and diagnose their patients constantly. The proposed system uses machine-learning algorithms to analyse past data, to improve healthcare by predictions and better analytics. It will not only be available in the hospitals and ambulances but will also be integrated with the smart home systems.

7.1 Problems Faced and Lessons Learned

After the submission of the proposal, different problems related to the domain, tools, and technologies for the system were faced. The first issue encountered was the selection of hardware which was final from our accounts in terms of budget and wear-ability, but because of COVID-19 and the lockdown, our team was unable to acquire what was planned. To start working, hardware was necessary, so we had to work with the hardware which was available in the lab. In this pandemic situation, we had to come into the lab to complete the project on time, and with the risk of getting infected with coronavirus. After working with the sensors, we faced the problem of having an inconsistent data stream which made it difficult to store the values of each sensor. The problem was solved by filtering the data on the edge device. Another issue encountered was that weather we use machine learning algorithms or deep learning algorithms. So, after consulting with our field related teacher we decided to use a machine learning algorithm. The lesson that we learned is that uncertain events will affect the project, but the team should work with what they have and find another way.

7.2 Future Work

- More sensors such as diabetes sensor, Electrocardiography (ECG) with 12 leads (limb and precordium leads), etc. will be installed on the system for more accuracy and detailed monitoring.
- mVitals with implementation of VR technology and adaptation for Microsoft HoloLens.
- The system can be installed in the hospitals of the entire city, which will be connected to a centralized server for more experienced monitoring.
- In the future, the system can be incorporate with the smart medication system for the right dose.

8 References

[1] M. Tropmann-Frick, “Internet of things: Trends, challenges and opportunities,” *Commun. Comput. Inf. Sci.*, vol. 909, pp. 254–261, 2018, doi: 10.1007/978-3-030-00063-9_24.

[2] A. Limaye and T. Adegbija, “A Workload Characterization for the Internet of Medical Things (IoMT),” in *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI*, Jul. 2017, vol. 2017-July, pp. 302–307, doi: 10.1109/ISVLSI.2017.60.

[3] “6 IoT in Healthcare Examples You Should Know | Built In,” *builtin.com*, Jul. 04, 2019. https://builtin.com/internet-things/iot-in-healthcare (accessed Jul. 31, 2020).

[4] Nick Ismail, “Moving from central to the edge: Is cloud decentralisation inevitable?,” *information-age*, Jun. 01, 2018. https://www.information-age.com/cloud-decentralisation-inevitable-123472539/ (accessed Jul. 31, 2020).

[5] C. Chang, S. N. Srirama, and R. Buyya, “Mobile Cloud Business Process Management System for the internet of things: A survey,” *ACM Comput. Surv.*, vol. 49, no. 4, pp. 1–42, Dec. 2016, doi: 10.1145/3012000.

[6] S. Naseem, U. K. Khattak, H. Ghazanfar, and A. Irfan, “Prevalence of non-communicable diseases and their risk factors at a semi-urban community, Pakistan,” *Pan Afr. Med. J.*, vol. 23, Mar. 2016, doi: 10.11604/pamj.2016.23.151.8974.

[7] Eminraja, “Pakistan’s Healthcare Sector: There Is Only 1 Bed To Accomodate 1580 Patients - Siasat.pk News Blog,” *Apr 28, 2018*. https://blog.siasat.pk/pakistans-healthcare-sector-1-bed-accomodate-1580-patients/ (accessed Jul. 31, 2020).

[8] M Waqar Bhatti, “‘Delays in reaching hospitals contribute to high trauma deaths,’” *The News*, Jan. 03, 2016. https://www.thenews.com.pk/print/85963-Delays-in-reaching-hospitals-contribute-to-high-trauma-deaths (accessed Jul. 31, 2020).

[9] “The Impact of Technology in Healthcare | AIMS Education,” *AIMS education*, Jun. 02, 2019. https://aimseducation.edu/blog/the-impact-of-technology-on-healthcare/ (accessed Jul. 31, 2020).

[10] “F8 Smartwatch Weather Forecast Fitness Tracker Watch Waterproof Ip67 Heart Rate Monitor F8 Smart Watch Smart Bracelet Wristband - Buy F8 Smart Watch Smart Bracelet Wristband,F8 Smart Wristband,Smart Bracelet Wristband Product on Alibaba.com.” https://www.alibaba.com/product-detail/F8-Smartwatch-Weather-Forecast-Fitness-Tracker_62235386764.html?spm=a2700.galleryofferlist.0.0.4ffe3fe9AHkc5Q (accessed Jul. 31, 2020).

[11] “PO 30 - Pulse oximeter | beurer,” *beurer*. https://www.beurer.com/web/gb/products/medical/ecg-and-pulse-oximeter/pulse-oximeter/po-30.php (accessed Jul. 31, 2020).

[12] “IntelliVue Guardian EWS | Philips Healthcare.”

https://www.usa.philips.com/healthcare/clinical-solutions/early-warning-scoring/intellivue-guardian-ews (accessed Dec. 28, 2020)..

[13] “Life scope G9 - Nihon Kohden Europe - PDF Catalogs | Technical Documentation.” https://pdf.medicalexpo.com/pdf/nihon-kohden-europe/life-scope-g9/69520-176947.html (accessed Dec. 28, 2020).

[14] “Wearable Monitor Provides Continuous Blood Pressure Data | Medgadget.” https://www.medgadget.com/2020/01/wearable-monitor-provides-continuous-blood-pressure-data.html (accessed Dec. 28, 2020).

[15] J. K. T. and P. R. A. and M. E. Johnson, “Sotera Wireless,” *Digital Stategies*. https://digitalstrategies.tuck.dartmouth.edu/wpcontent/uploads/2016/10/Sotera_Wireless_6-0035.pdf (accessed Sep. 04, 2012).

[16] “(43) Sotera Wireless - Visi Mobile - YouTube.” https://www.youtube.com/watch?v=WF8UGt3IipY (accessed Jul. 31, 2020).

[17] “F8 Smartwatch Weather Forecast Fitness Tracker Watch Waterproof Ip67 Heart Rate Monitor F8 Smart Watch Smart Bracelet Wristband - Buy F8 Smart Watch Smart Bracelet Wristband,F8 Smart Wristband,Smart Bracelet Wristband Product on Alibaba.com.” https://www.alibaba.com/product-detail/F8-Smartwatch-Weather-Forecast-Fitness-Tracker_62235386764.html?spm=a2700.galleryofferlist.0.0.4ffe3fe9AHkc5Q (accessed Jul. 31, 2020).

[18] “F8 Smartwatch Weather Forecast Fitness Tracker Watch Waterproof Ip67 Heart Rate Monitor F8 Smart Watch Smart Bracelet Wristband - Buy F8 Smart Watch Smart Bracelet Wristband,F8 Smart Wristband,Smart Bracelet Wristband Product on Alibaba.com.” https://www.alibaba.com/product-detail/F8-Smartwatch-Weather-Forecast-Fitness-Tracker_62235386764.html?spm=a2700.galleryofferlist.0.0.4ffe3fe9AHkc5Q (accessed Dec. 28, 2020).

[19] “PO 30 - Pulse oximeter | beurer,” *beurer*. https://www.beurer.com/web/gb/products/medical/ecg-and-pulse-oximeter/pulse-oximeter/po-30.php (accessed Jul. 31, 2020).

[20] “Life Scope G3.” https://eu.nihonkohden.com/en/products/patientmonitoring/life_scope_g3.html (accessed Dec. 28, 2020).

[21] . “SCRUMBAN | Pearls of Wisdom,” *Pearls of Wisdom*, May 14, 2014. https://agilepearls.wordpress.com/tag/scrumban/ (accessed Jul. 31, 2020)..

[22] “MySignals HW v2 - eHealth and Medical IoT Development Platform for Arduino.” https://www.cooking-hacks.com/mysignals-hw-ehealth-medical-biometric-iot-platform-arduino-tutorial.html (accessed Dec. 27, 2020).

[23] “Arduino Uno Rev3 | Arduino Official Store.” https://store.arduino.cc/usa/arduino-uno-rev3 (accessed Dec. 27, 2020).

[24] “Buy a Raspberry Pi 3 Model B – Raspberry Pi.” https://www.raspberrypi.org/products/raspberry-pi-3-model-b/ (accessed Dec. 27, 2020).

[25] Margaret Rouse, Alexander Gillis and Peter Waher, " What is MQTT and How Does it works?", IoT Agenda, https://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport (accessed Dec. 26, 2020)

[26] "MySignals HW v2 - eHealth and Medical IoT Development Platform for Arduino". https://www.cooking-hacks.com/mysignals-hw-ehealth-medical-biometric-iot-platform-arduino-tutorial.html (accessed Dec. 26, 2020).

[27] "Arduino - Wire", SM, December 24, 2019. https://www.arduino.cc/en/reference/wire (accessed Dec. 26, 2020).

[28] "Arduino - SPI", SM, December 24, 2019. https://www.arduino.cc/en/reference/SPI (accessed Dec. 26, 2020).

[29] "paho-mqtt . PyPI", September 22, 2020. https://pypi.org/project/paho-mqtt/ (accessed Dec. 26, 2020).

[30] Chris Liechti , "Welcome to pySeries's Documentation - pySeries 3.0 documentation", PySeries, 2001-2015. https://pythonhosted.org/pyserial/ (accessed Dec. 26, 2020).

[31] "RPi.GPIO . PyPI", July 21, 2019. https://pypi.org/project/RPi.GPIO/ (accessed Dec. 26, 2020)

[32] "InfluxDB-Python - InfluxDB 5.3.1 documentation", https://influxdb-python.readthedocs.io/en/latest/include-readme.html (accessed Dec. 26, 2020).

[33] "datetime - Basic date and time types - Python 3.9.1 documentation", https://docs.python.org/3/library/datetime.html (accessed Dec. 26, 2020).

[34] "json - JSON encoder and decoder - Python 3.9.1 documentation", December 27, 2020. https://docs.python.org/3/library/json.html (accessed Dec. 27, 2020).

Document Viewer

Turnitin Originality Report

Processed on: 28-Dec-2020 17:25 PKT
ID: 1481667478
Word Count: 14850
Submitted: 1

mVitals By Noor Jaffri

Similarity Index	Similarity by Source	
13%	Internet Sources:	7%
	Publications:	3%
	Student Papers:	8%

exclude quoted exclude bibliography exclude small matches mode: quickview (classic) report
Change mode print refresh download

1% match (student papers from 10-Jun-2015)
Submitted to Higher Education Commission Pakistan on 2015-06-10

1% match (Internet from 28-Dec-2020)
https://www.alibaba.com/product-detail/F8-Smartwatch-Weather-Forecast-Fitness-Tracker_62235386764.html?spm=a2700.galleryofferlist.0.0.4ffe3fe9AHkc5Q

1% match (Internet from 21-Jun-2020)
https://www.coursehero.com/file/45168061/Test-Case-LMSdocx/

<1% match (publications)
"Mobile Health", Springer Science and Business Media LLC, 2015

<1% match (Internet from 10-Nov-2020)
https://www.medgadget.com/2020/01/wearable-monitor-provides-continuous-blood-pressure-data.html

<1% match (student papers from 29-May-2017)
Submitted to Higher Education Commission Pakistan on 2017-05-29

<1% match (student papers from 27-May-2020)
Submitted to Université Saint-Esprit Kaslik on 2020-05-27

<1% match (student papers from 16-Dec-2019)
Submitted to Higher Education Commission Pakistan on 2019-12-16

<1% match (student papers from 05-Aug-2020)
Submitted to Higher Education Commission Pakistan on 2020-08-05

<1% match (student papers from 18-Jan-2019)
Submitted to UCSI University on 2019-01-18

<1% match (student papers from 30-Jul-2020)
Submitted to Higher Education Commission Pakistan on 2020-07-30

<1% match (Internet from 23-Jul-2020)
https://eu.nihonkohden.com/products/patientmonitoring/life_scope_g9.html

<1% match (student papers from 21-Jun-2013)
Submitted to Higher Education Commission Pakistan on 2013-06-21

<1% match (student papers from 15-May-2018)
Submitted to Higher Education Commission Pakistan on 2018-05-15

<1% match (publications)
Xiuli Tang. "Research on Smart Logistics Model Based on Internet of Things Technology", IEEE Access, 2020

<1% match (student papers from 12-May-2016)
Submitted to Higher Education Commission Pakistan on 2016-05-12

<1% match (student papers from 04-Dec-2019)
Submitted to Özyegin Üniversitesi on 2019-12-04

<1% match (student papers from 18-Dec-2014)
Submitted to Higher Education Commission Pakistan on 2014-12-18

<1% match (Internet from 21-Oct-2020)
https://blog.siasat.pk/pakistans-healthcare-sector-1-bed-accomodate-1580-patients/

<1% match (student papers from 08-Jan-2018)
Submitted to Universiti Teknologi Petronas on 2018-01-08

<1% match (Internet from 31-Jul-2020)
https://www.thenews.com.pk/print/85963-Delays-in-reaching-hospitals-contribute-to-high-trauma-deaths

<1% match (student papers from 08-Jan-2018)
Submitted to Higher Education Commission Pakistan on 2018-01-08

<1% match (student papers from 09-Jul-2014)
Submitted to City and Islington College, London on 2014-07-09