

TALLINN UNIVERSITY OF TECHNOLOGY  
School of Information Technologies

Md Taibur Rahman 194273IASM

# **Smart Device Application Based on Health Care System**

Master's thesis

Supervisor: Vladimir Viies, PhD  
Associate Professor

Tallinn 2021

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Md Taibur Rahman 194273IASM

# **"Tervishoiusüsteemil põhinev nutiseadme rakendus"**

[Lõputöö liik: magistritöö]

Juhendaja: Vladimir Viies, PhD  
Associate Professor

Tallinn 2021

## **Author's declaration of originality**

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Md Taibur Rahman

09.05.2021

## **Abstract**

The primary goal of this work is to provide a theoretical and practical idea about how to develop the health care system so that we can call it a “smart health care system”. The primary purpose is to create an innovative device application for the android environment. Every citizen, especially the older generation, children, and the disabled person, can get the treatment without any hassle. Besides, the Government can get helpful health information about every citizen. The paper will provide the idea of developing an innovative device application, and the application in the android environment will make the health care system more convenient. The work will also provide the basic idea of an innovative device application that shows how a doctor or family doctor and patients can connect in an emergency. The designated doctor can also check the information of his/her patient's status every week. The innovative device application will work by getting the information or data from various smart devices like Fitbit, Mintti vision health device, checkme pro, etc. Doctors and patients will be connected to the system, and poorly conditioned patients can get rapid solutions. The thesis consists of three main parts, each describing its objectives:

1. Review the consists of the android environment, its properties, main frameworks, and tools. Compares it to other mobile operating system and give examples of how applications work on it.
2. Discuss state of the art, the selection of smart device and its accuracy, and functionality implementation.
3. The design and implementation of the smart health care system based on an application (Smart Health Care) in the android environment and its security.

This thesis is written in English and is 79 pages long, including 5 chapters, 30 figures, and 7 tables.

## List of abbreviations and terms

OS	Operating System
GPS	Global Positioning System
SDK	Software Development Kit
IoT	Internet of Things
IDE	Integrated Development Environment
DVM	Dalvik Virtual Machine
UI	User Interface
OOP	Object-Oriented Programming
PHP	Hypertext Pre-processor
API	Application Programming Interface
LRU	Least Recently Used
LAMP	Linux, Apache, MySQL, PHP/Perl/Python
SQL	Structured Query Language
ECG	Electrocardiogram
QR	Quick Response
BP	Blood Pressure
MEWS	Modified Early Warning Score
SD	Standard Deviation
DFD	Data Flow Diagram
UML	Unified Modelling Language
WGS	World Geodetic System
RAM	Random Access Memory
UNIX	Uniplexed Information and Computing Service

## Table of contents

1 Introduction .....	11
2 Android Platform and Tools .....	15
2.1 Android Application Framework.....	15
2.1.1 Android System Architecture .....	17
2.1.2 Development Environment.....	21
2.1.3 Backend Architecture Possibilities .....	22
2.2 Android Memory Management .....	23
2.3 Android Device and Processor Management .....	24
2.4 Database.....	26
2.5 MVC Design Pattern .....	27
2.6 Mapkit.....	29
3 Smart Device Selection .....	30
3.1 State of the Art.....	30
3.2 Existing Technology and Device.....	33
3.2.1 Existing Smart devices .....	33
3.2.2 Selected Device and Working Method.....	33
3.2.3 Results and Accuracy .....	35
3.3 Data Flow Analysis of the System .....	36
3.4 Functional Block Diagram.....	41
3.5 System Validation and the Relationship.....	43
3.5.1 Block Definition Diagram .....	43
3.5.2 Use Case Diagram .....	44
3.5.3 Sequence Diagram.....	46
3.5.4 Analysis .....	47
4 Design, Implementation and Result .....	48
4.1 Specification of Smart Health Care .....	48
4.2 The management system of Smart Health Care .....	49
4.3 Implementation and Result of Smart Health Care.....	50
4.4 Design principles and methods of Smart Health Care.....	56

4.5 Market Value of Smart Health Care System .....	58
4.6 Security Mechanism for Smart Health Care System .....	59
5 Summary.....	64
References .....	66
Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis .....	69
Appendix 2-Description and Manual .....	70
Appendix 3 – Source Code.....	71

## List of figures

Figure 1. Smart health care application in android environment .....	11
Figure 2. Interaction with an operating system .....	15
Figure 3. Smart health care application with android tools.....	16
Figure 4. Android system architecture .....	17
Figure 5. Database for smart health care .....	27
Figure 6. Relationship between view, model, and controller .....	28
Figure 7. Checkme Pro Device [13] .....	34
Figure 8. Level 0 or context diagram for the smart health care system .....	37
Figure 9. Level 1 or context diagram for patient interface in the system.....	38
Figure 10. Level 1 or context diagram for doctor interface in the system .....	39
Figure 11. Level 2 data flow diagram for patient and doctor’s interface in the system .	40
Figure 12. Functional block diagram.....	41
Figure 13. Block definition diagram for the smart health care system .....	44
Figure 14. A use case diagram for doctor interface of the smart health care system .....	45
Figure 15. A use case diagram for patient interface of the smart health care system ...	45
Figure 16. Sequence diagram for the smart health care system .....	46
Figure 17. Requirement specification of smart health care .....	48
Figure 18. The management system of smart health care .....	49
Figure 19. Doctor interface’s source code and demonstration .....	50
Figure 20. Patient interface’s source code and demonstration .....	51
Figure 21. Demonstration of the dropdown location selection in the patient end.....	52
Figure 22. Parameters for patients code and demonstration.....	53
Figure 23. Set parameter and Write the prescription.....	54
Figure 24. Logic code for getting notification.....	54
Figure 25. Notification view.....	55
Figure 26. Demonstration of doctors all functionality .....	55
Figure 27. Demonstration of patients all functionality.....	56
Figure 28. Security requirements for the smart health care system.....	60
Figure 29. Smart health care data transmission process.....	62



Figure 30. RSA algorithm mechanism in smart health care system..... 63

## **List of tables**

Table 1. Android vs IOS vs Windows.....	17
Table 2. Explanation of application layer services.....	18
Table 3. Explanation of application framework layer services .....	19
Table 4. Explanation of library layer services .....	20
Table 5. Packages to run the SDK.....	21
Table 6. Illustrations of study and aftereffects of the alignment system [14] .....	35
Table 7. Functional block and their description .....	42

# 1 Introduction

A long time back in 1996, the evolution of mobile operating systems has started from the Palm OS. Two years later, Nokia developed the mobile phone operating system named Symbian, which created the modern journey for people. The history of keen advancement has delivered numerous, to begin with, time, and has significantly affected people’s lives.

On the other hand, Healthcare is playing a vital role in people’s life. The number of patients is increasing day by day, but the number of doctors is not proportionally enough for patients. Most developed countries have already followed the family doctor’s method where a doctor is responsible for many patients in an area. But the advancement of innovative technology cannot touch the healthcare system at all, especially in developing countries like Bangladesh.

Tech companies are developing hardware, software, and the operating system continuously. Companies are using the latest technology such as sensors, cameras, chips, processors, etc., in the hardware with the help of big data, artificial intelligence, machine learning, etc. In the operating system, Artificial Intelligence and machine learning have brought unprecedented changes.

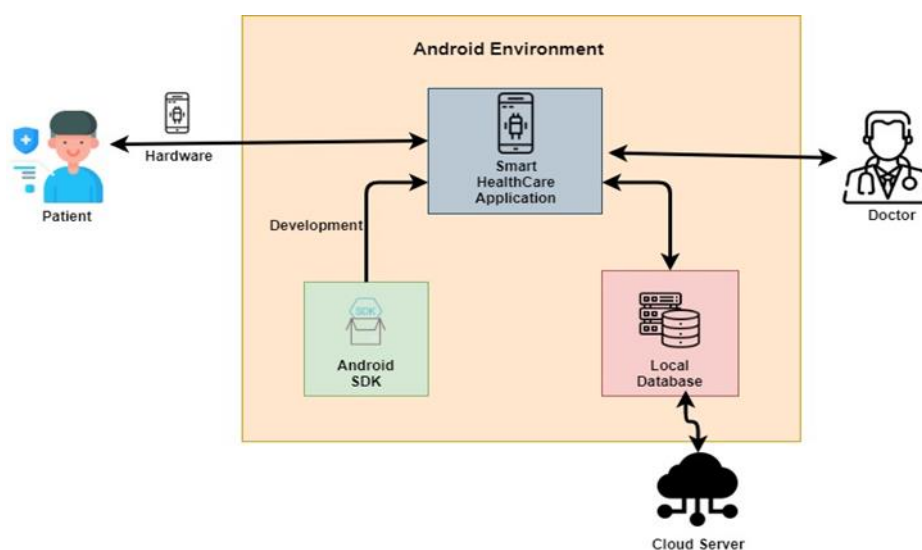


Figure 1. Smart health care application in android environment

Figure 1 shows a clear picture of the smart health care application which runs in the android environment. Patients can use several smart devices (e.g., blood pressure monitor, glucose monitor, heartbeat monitor, Fitbit, Mintti vision health device, checkme pro, etc.). Checkme pro is selected for this paper, and the reason behind the selection of this device will explain in the third chapter. Patients can get health data after using smart devices. Patients will use the android phone as a hardware device for sending health information to doctor and getting the benefits of a smart health care system. Android SDK is used to develop the smart health care application in the android operating system.

Doctors can also use this same smart application to get patients' health status utilizing an android device as hardware. A local database is used for storing all the data. Data will upload to the cloud server after following the encryption and decryption.

In this 21st century, people need to go to the hospital or doctor for regular check-ups physically. They must book an appointment over the call. Most importantly, people from many developing countries do not get proper treatment for the imbalance of the proportion of doctors and patients. For the convenience of the health care system, A smart device application can play a major role in connecting patients and doctors. This smart application will work as an umbrella, and all the entities have joined each other under the umbrella.

This type of centralized system is not available till now. The idea of smart device technology, e.g., different sensors and their accuracy, IoT, cloud computing, database, a centralized application, android environment platform, communication technology, built-in report analysis application, GPS technology, data security, and more technologies will be used combinedly to accomplish this system. This application will play a vital role, especially in the health care sector, and people can get the treatment, and doctors can conveniently provide the treatment.

Smart technology and the development of smart phone's both hardware, software, and OS, have developed rapidly. That urges me to accomplish this task. The skills of mobile development, ongoing research work, and finished research work also helped me achieve this project. On the other hand, the ongoing pandemic situation added another dimension to complete the job to make life more convenient.

## Aim and Objective

The primary focus of this project is to develop an idea of a smart system and smart mobile application named smart health care system that connects the patients and doctors in a single frame. This work aims to monitor the health of the elderly and disabled and ensure the treatment quickly. This application helps the user to keep health data documented and get the treatment from home. Similarly, doctors can see their patients' health information in a single application and prescribe medicines through this application. So, doctors can provide advice or treatment to more patients. The Government can keep tracking every citizen's health information. The executed framework is cost-effective in dealing with health and crisis care, which is also an objective of the proposed work.

With the complexity of the android environment, smart device technology will measure and provide health data, implement health applications and GPS technology to communicate, database, cloud computing, and RSA algorithm for encryption and decryption to ensure data security clearly explained in this paper. This smart application will convert the hectic manual experience to a smart health care experience.

This work contains five chapters. The first section is the introduction, and the rest three are main chapters, and the last chapter is a summary. Every section has contents, and these are as follow:

The introduction describes the basic idea of the work, its background, and its significance. The brief knowledge of the paper and its functionality explains here. This section also describes the primary aim and final goal of this paper.

The second chapter elaborates on the basics of the system and software aspect. This chapter is the technical foundation of this paper. This chapter contains the analysis of the smart android devices and how they work and their accuracy, depicts the idea of the android platform, discusses the android architecture and frameworks.

The third chapter is the main foundation of this paper. This chapter explains state of the art, how software and system aspects will integrate into work. This chapter illustrates the analysis of the smart devices used to measure different health parameters and their accuracy, studies the functional requirements and how the technology is used, and describes how data will store in a database and how data will upload to the cloud. This

chapter also describes how an android environment and our system application will collaborate with the existing tools, and the work will explain step by step. This chapter contains the smart health care application, diagram, and dataflow design that describes how the smart application works and the use cases that will be used.

The fourth chapter is the design and output of the work. The management system, design methods, design principles, market analysis, and implementation will explain in this chapter. Moreover, this chapter will include the security mechanism.

Finally, the conclusion in the last chapter elaborates the summary of this paper and the importance of the smart health care system, especially for developing countries like Bangladesh.

## 2 Android Platform and Tools

The most widely used versatile operating system these days is android. Android is the programming that was established in Palo Alto, California, in 2003 [1]. After that, Open Handset Alliance started developing the Android operating system, then Google and many other giant companies began developing the android platform. This platform provides a better way to create a mobile application that allows the developer to build Android and different devices that support android. Android code is authorized to open source and free programming. As indicated by the GNU General Public License Update, Google delivers a large part of the code with Apache License Update 2.0 [1], and the rest updates the Linux bit. This chapter depicts the idea of the android platform, discusses the android architecture and frameworks.

### 2.1 Android Application Framework

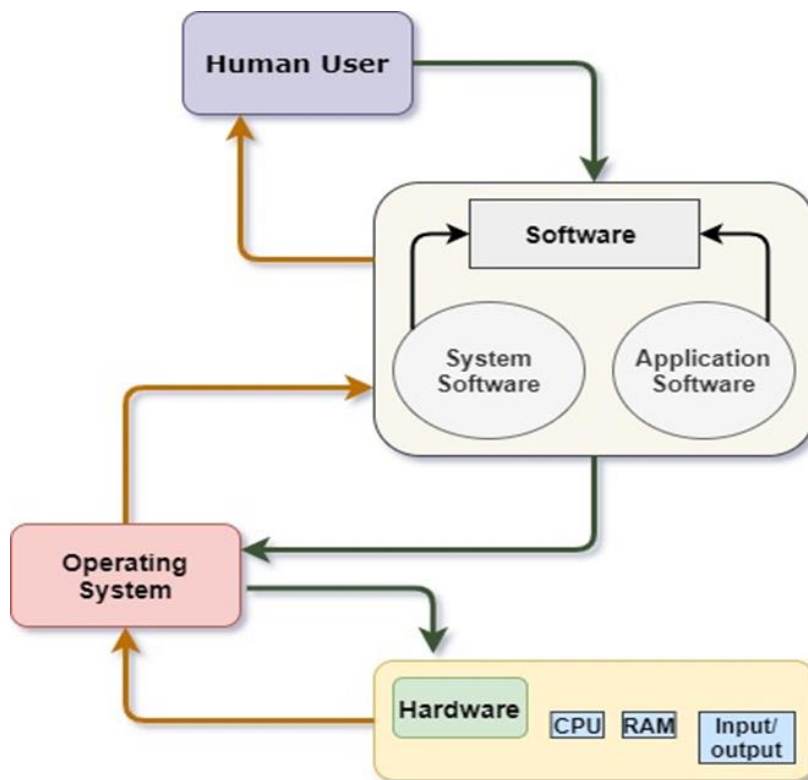


Figure 2. Interaction with an operating system

The interaction with an operating system is described in Figure 2; an operating system can communicate with its underlying level, as shown in the figure. It works like an interface or a bridge between user and hardware. OS allows the user and software to communicate with the hardware and provides the necessary access to perform any task by giving the control.

For instance, when a user wants to play music, the user needs to launch the music player. When the user clicks on the music software or application, the music application sends the message to the operating system that the user wants to open the music application. The operating system immediately takes control of the music application and provides the facility to use the audio driver from the hardware section and play the music.

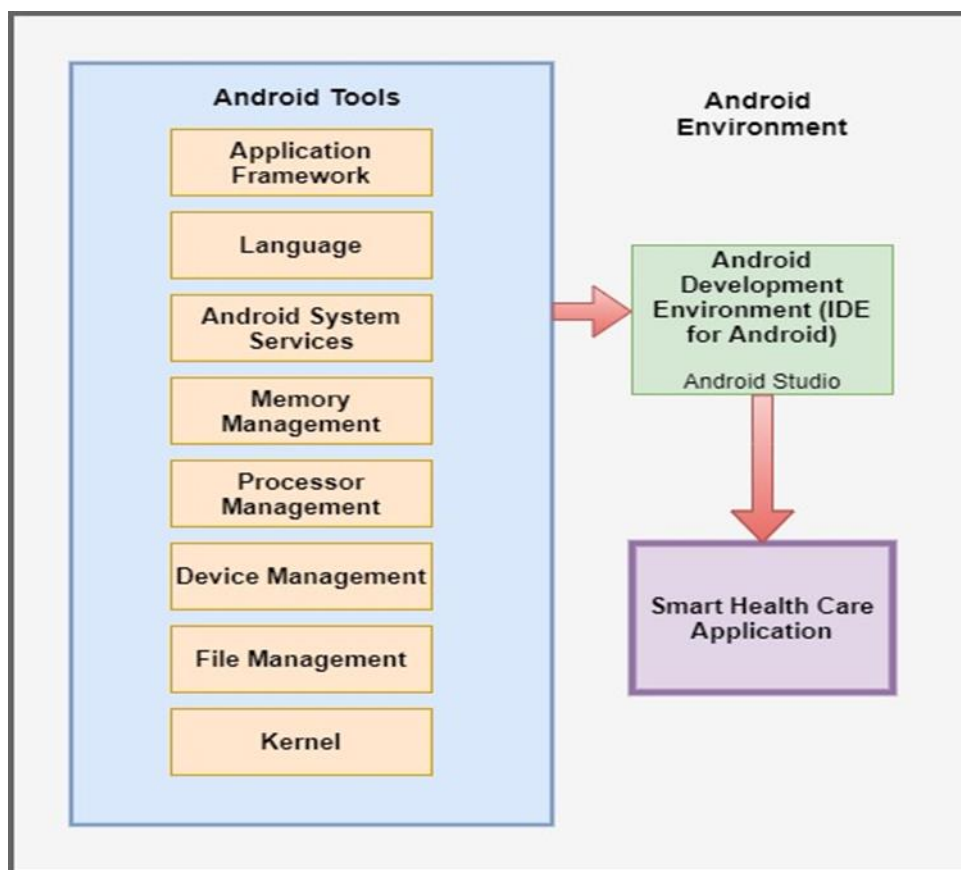


Figure 3. Smart health care application with android tools

Figure 3 clearly shows how android tools and the android development environment are needed to run an android application in the android environment. This environment consists of an application framework, programming languages, system services, and several management systems as a tool, and Android Studio as an IDE to implement the android application in the android environment.



## Comparison IOS, Android, windows

There are three significant working frameworks accessible on cell phones today: Apple uses IOS, Google uses Android, and Microsoft uses Windows.

Table 1. Android vs IOS vs Windows

Category	Android	IOS	Windows
Release	2008	2007	2009
Developer	Google	Apple Inc.	Microsoft
Source	Open	Closed	Closed
Written in	Java, C, C++	Objective c, swift	C, C++
Working state	Current	Current	Discontinued
Operating system	Unix	iOS	Microsoft windows
Package manager	apk-based	Swift PM	Windows phone store
Kernel	Linux	Darwin	Hybrid
Tools	Android studio, flux	X-code, flux	Visual studio
Publisher	Play store	Apple store	Windows store

Table 1 illustrates the comparison between android, IOS, and windows operating systems.

### 2.1.1 Android System Architecture

These product modules primarily contain the portion and set of Libraries that encourage versatile application advancement and structure part of the runtime, application system, and the real application.

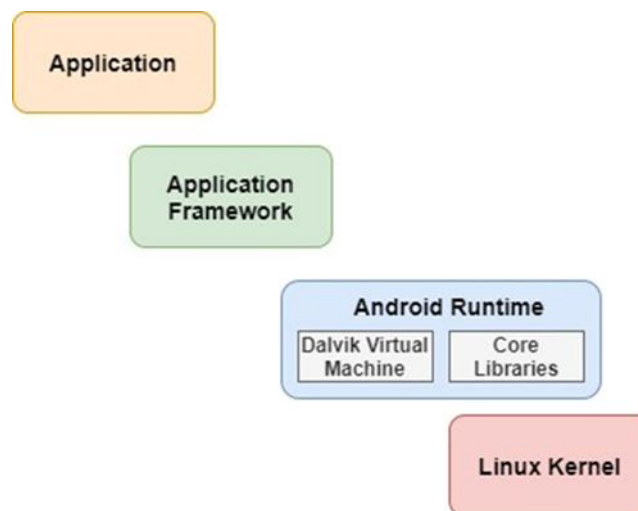


Figure 4. Android system architecture

The android architecture (Figure 4) shows the layers of android. It comprises a few programming modules to help the working of smartphones. These product modules primarily contain the portion and set of libraries that encourage versatile application advancement and structure part of the runtime, application system, and the real application. The application modules are assembled into five segments under four unique layers. Android runtime layer is divided into two pieces: DVM and other is libraries. DVM and libraries are under the runtime layer.

- **Application Layer:** The application layer is the first layer in the architecture model, and it is the front end for the user who will use the android application. Local applications made by using android design alongside outsider applications are presented in this layer. Applications from this layer get executed with the help of the run time layer using the classes and services given by the framework layer. Application parts are the essential structure of an android application. These parts are generally coupled by the application show record AndroidManifest.xml that depicts each section of the application and how they behave.

Table 2. Explanation of application layer services

Services	Description
Activities	They direct the UI and handle the client connection to the cell phone screen. An activity addresses a screen with a customer interface, in-short action performs exercises on display.
Administrations	They handle foundation preparation related to an application. An organization is a portion that runs far out to perform long-running assignments.
Broadcast Receivers	They handle correspondence between Android OS and applications. Broadcast Receivers essentially react to communicate messages from different applications or the framework.
Content Providers	They handle information and data set administration issues. Content providers supply information from one application to others on demand.

- **Application Framework Layer:** The applications Framework layer carries the classes expected to create applications in the Android environment. It empowers

admittance to hardware, manages the UI, and oversees assets for the application. The administrations gave by this layer are made accessible to the application layer for advancement as a class. A portion of the segments in the structure layer are notification manager, activity manager, telephony administration, package manager, and view framework and utilized in application advancement varying. The application system gives a few features, for example, Java classes for application advancement. Engineers use these Java classes during coding. This part provides the accompanying administrations.

Table 3. Explanation of application framework layer services

Services	Description
Action Manager	Directs the chain of importance of the application, deals with the movement, and follows all the exercises.
Content Provider	Facilitates offering information to external applications.
Asset Manager	Enables applications to utilize different assets like color settings, client connections, and strings.
Notification Manager	Manages cautions and notices to clients on the status of use execution.
View framework	Provides different view alternatives for making client communication.

- **Android Runtime:** This is the third fragment of the plan and opens on the second layer from the lower stage. This part provides a key portion called Dalvik Virtual Machine called Java Virtual Machine, particularly arranged and upgraded for Android. The Dalvik VM utilizes Linux because of memory management and multi-stringing, inherent in the Java language. The Dalvik VM empowers to complete the full cycle for every application, including its event of the DVM. Moreover, the Android runtime focuses on libraries that engage android application designers to make android applications using standard Java programming language.
- **Libraries:** This segment joins those Java-based libraries that are unique to android improvement. Examples of libraries in this grouping join the application framework libraries despite those that empower making UI, illustrations plan, and

database access. An abstract of some key place Android libraries open to the Android advancement is according to the accompanying-

Table 4. Explanation of library layer services

Library	Description
android. App	Allows to use of the application model and establishes the android application by developer choice.
android. Content	Encourages content access, distributing, and informing among applications and application segments [1].
android. Database	It is used to get information distributed by content suppliers and incorporate the board classes' SQLite database.
android. OpenGL	Coveys API
android. Os	Gives applications admittance to standard working framework administrations, including messages, framework benefits, and measure correspondence.
android. Text	It is used to deliver and control text on a gadget show.
android. View	The central structure squares of utilization UIs.
android. Widget	A rich assortment of pre-assembled UI segments, for example, catches, marks, list sees, design chiefs, radio catches, and so forth.
android. Webkit	A set of classes expected to permit web-perusing abilities to be incorporated into applications.

- Kernel:** The kernel is the center of the working framework. Android's kernel is worked off the Linux 2.6 kernel for certain design alterations executed by Google outside the typical Linux part improvement cycle. Linux kernel is a solid bit, implying that most of the working framework is found in the part space, for example, gadget drivers, piece extensions. This outcome is an enormous source code. The Linux Kernel is the thing that collaborates with the hardware and contains all gadget drivers utilized by more elevated levels of the product stack to control and speak with the hardware. Such drivers include camera drivers, sound drivers, display drivers, memory drivers, and more user drivers. The kernel is changed for unique requirements in cell phones, for example, power the executives, memory the board, and runtime the board.

## 2.1.2 Development Environment

To build up an android application, there are numerous application suites to create. Yet, the Android studio application is ideal for building up an android application in the simplest manner. Coding in Java inside Android Studio is natural since it gives rich Java features, including context-sensitive assistance and code recommendation. In the wake of gathering Java code neatly, the segments of Android Studio ensure the entirety of the application is bundled appropriately, including the AndroidManifest.xml record [1]. The SDK manager, which ships as a coordinated encounter of Android Studio, deals with downloading the various components of the Android SDK. The SDK manager helps with settling on a decision concerning which SDKs to choose. Developers can download this application for advancement on any stage like Windows, Mac, or Linux. Android Studio combines instruments for managing diverse set express features with the SDK chief. The limit of any application for testing on an emulator or any simple gadget. The third-generation Android studio is taking the highest percentage of choice of the developers nowadays. There are similarly request line gadgets and diverse, consistent consolidation toolchains that permit Android applications. The SDK administrator, which ships as a coordinated encounter of Android Studio, deals with downloading the various components of the Android SDK. The Android devices accessible are somewhat overpowering—android applications on a simple gadget or on the Android Emulator, which sends alongside the Android SDK.

Table 5. Packages to run the SDK

Package	Description
SDK Tools	To run SDK
SDK Platform-tools	This one is used when SDK manager is run for the first time
SDK Platform	To run the application, this package should be installed
System Image	This package is important to use images on the emulator
SDK Samples	Provides basic coding, which will help to understand android development

### **2.1.3 Backend Architecture Possibilities**

Android is a well-known OS for mobile phones today. Java is the famous programming language is utilized to create Android applications. Android applications run on a unique virtual machine called the Dalvik VM that is an immediate motivation from the java virtual machine called JVM. Android application can run on any device where an uncommon Dalvik VM is executed [1].

The most importantly favorable position of utilizing Java for application improvement is that Java. Java upholds OOPS ideas and is more equipped as extensible and adaptable. Java has a broad arrangement of libraries. Android SDK has numerous standard Java libraries included. Android is made to run on various stages, for example, equipment stages. Android code is composed once and to execute necessities to gather and improve local code to manage different devices [2]. Java has stage-free highlights, so it is utilized for android advancement. These give functionalities to information structure, math capacities, illustration implantation, and systems administration capacities, and substantially more. These java libraries assist us with doing all the other things we could need. This way, java creates android applications quick way.

On the other hand, PHP is a programming language for use as a backend for Android applications. It requires android login just as enlistment, which is an extremely normal and fundamental situation. PHP works in a state of harmony, which means the arrangement. You need to finish the initial phase to move to the following one. It executes code in the collection. This encourages the user to be all together and play out an errand in an organized way. PHP interfaces the site's data set to the program, making a smooth, easy-to-use insight. The information base is a basic part of the back end, requirements to function admirably. PHP is conventional, so it performs admirably with MySQL, PostgreSQL, and MariaDB. PHP is the largest standard backend for android applications because of its framework, which is one of the essential clarifications for its thriving. PHP gives an indispensable design for fabricating web applications.

In this paper, on account of the blended programming utilizing Java and PHP, the improvement mostly uses android studio's environment since it tends to be aggregated with the Java language. What is more, the help for different APIs is additionally fully grown.

## 2.2 Android Memory Management

Android utilizes paging and map instead of giving swapping, which implies any memory your application contacts cannot be paged out except if you discharge all references. The Dalvik Virtual Machine's stack size for application measures is restricted. Applications startup with 2 MB, and the greatest portion, set apart as "large heap," is limited to 36 MB [2]. Android keeps background application measures in an LRU cache. When the system comes up short on memory, it will execute processes as per the LRU technique, yet it will likewise consider which application is the biggest memory reserved.

**Garbage collection:** Whether it is Dalvik virtual machine, it resembles a ton of Java virtual machines and has a place with a oversaw memory environment. The oversaw memory environment monitors every memory portion. Once it establishes that the program no longer uses a piece of memory, it will deliver it back to the heap with no intervention from the developer. The system for reusing unused memory in an oversaw memory environment is called garbage collection. Android's memory stack is generational. The length of garbage collection relies upon which age of items it gathers and the number of dynamic pieces in each generation. Although garbage collection is extremely quick, it influences the exhibition of the application. Ordinarily, it would help if you had controls when garbage collection instances are acted in code. The application has a bunch of standards for deciding when to perform trash collection. There are two objectives for garbage collection: One is to discover information demonstrations that cannot be gotten to in the program later. Another one is to reuse the assets utilized by these items.

**Shared memory:** Android can share RAM (Pages) across measures. Every application cycle is forked from a current process called Zygote. The Zygote cycle begins when the framework boots and loads system code and assets (for example, Activity Themes). This framework will fork the Zygote cycle and afterward burden and run the application code in the new process. This strategy permits most of the RAM pages dispensed for system code, and assets are divided between all application processes. This static information incorporates Dalvik code (by putting it in a pre-connected. Odex record for direct memory-planning), application assets (by planning the asset table to be mmap structure), and adjusting APK compress sections), and Traditional task components, for example, local code in .so documents [2].

## 2.3 Android Device and Processor Management

The operating system keeps the communication between devices by using their specific drivers. The operating system maintains the following activities:

- Tracks all devices. I/O controller is responsible for tracking these devices.
- Determines which process of each device and associates the time limit.
- Effectively designates the device.
- De-distributes devices.

Associations that have embraced versatility utilize corporate or by and by claimed gadgets to get to corporate information. To guarantee improved profitability and keep up corporate information security, these associations should convey an answer or programming with a cell phone the executive's highlight.

Mobile phones empower more prominent versatility, quicker reactions, and higher profitability. They also present new difficulties in zones, for example, similarity, security, access, and gadget for the executives.

Associations need to watch out for the gadgets being utilized in their organization, whether they are organization claimed or individual devices. Mobile phone manager plus has all the thorough mobile phones the board includes that assist you with overseeing parts of your mobile phones, from basic setups to further developed security settings. Mobile phone manager plus is a finished cell phone the board instrument that empowers ventures to oversee corporate and by and by claimed gadgets across Apple, Android, Windows, and Chrome OS.

Most of the android applications use their own Linux cycle to run applications. This interaction is made for the application when a portion of its code should be run and will stay running until it is not, at this point, required. The framework needs to recover its memory for use by different applications. These cycle types are (arranged by significance):

- A foreground process is a process that is required to specify what the user is presently doing. Different application segments can cause its containing



interaction to be viewed as a forefront unexpectedly. It has just a couple of such cycles in the framework, and these may be executed if all else fails if memory is low to the point that not even these cycles can keep on running. For the most part, the gadget has arrived at a memory paging state, so this activity is needed to keep the UI responsive.

- A Visible cycle tackles a job that the client is mindful of, so executing it would affect the client's experience. The quantity of these cycles running in the framework is less limited than frontal area measures, yet at the same time generally controlled. These cycles are considered critical and will not be executed except if doing so is needed to keep all forefront measures running.
- A service interaction is one carrying a Service that has been begun with the `startService ()` strategy. Although these cycles are not straightforwardly noticeable to the user, they are, for the most part doing things that the client often thinks about. The framework will consistently keep such cycles running except if there is not sufficient memory to hold all foreground view and visible processes.
- A cached cycle is not presently required, so the framework can terminate it as wanted when assets like memory are needed elsewhere. In an ordinarily acting framework, these are the solitary cycles engaged with some help to the executives. These cycles frequently hold at least one Activity occurrences that are not noticeable to the user (the `onStop ()` technique has been called and returned). Suppose they execute their Activity life cycle effectively (see activity for additional subtleties) [2]. In that case, when the framework terminates such processes, it will not affect the client's experience when getting back to that application: it can re-establish the recently saved state when the related action is reproduced in another interaction.

When concluding how to group a cycle, the framework will put together its choice for the main level found among all the parts at present dynamic. See the Activity, Service, and Broadcast Receiver documentation to detail how every one of these segments adds to the general life pattern. The documentation for every one of these classes portrays what they mean for the available life pattern of their application. So, it is very important to make sure the service of the application.

## 2.4 Database

MySQL is an Oracle-supported open-source RDBMS (Relational Database Management System) considering Structured Query Language (SQL). MySQL is available for developing any system in the Linux, Windows, and UNIX platforms. Notwithstanding how it will generally be used in a wide range of utilization, MySQL is now associated with web applications and web dissemination. MySQL is also known in an enterprise stack as LAMP in any open-source environment. LAMP is a web improvement stage that uses Linux as the working system, Apache as the web worker, MySQL as the social, informational index organization structure, and PHP as the thing arranged scripting language.

**Functional aspects:** The main highlights of MySQL are the following-

- **Relational Database Management System (RDBMS):** MySQL is a general data set administration framework. This database language depends on SQL scripting to deal with the data records in a table.
- **Simple to utilize:** MySQL is not difficult to use. We need to get just the essential information on SQL. We can assemble and communicate with MySQL by using a couple of basic SQL commands.
- **Scalability:** MySQL upholds multi-stringing that makes it effectively versatile. It can deal with practically any measure of information, up to 50 million lines or more. The default record size limit is around 4 GB [3].
- **Secure:** MySQL comprises a strong information security layer that shields delicate information from outsiders who have no authorization. Additionally, passwords are encoded in MySQL.
- **Customer/Server Design:** MySQL follows the working of a customer and servers engineering. There is a database server (MySQL) and subjectively numerous customers (application programs), which speak with the specific server; that is, they can inquire information, save changes, etc.
- **Allowed to download:** MySQL can utilize so the system can download it from MySQL's official site with no expense.

- Speed: MySQL is viewed as one of the exceptionally quick information bases sponsored by an enormous number of benchmark tests.
- High Flexibility: MySQL underpins an enormous number of inserted applications, which makes MySQL truly adaptable.

MySQL data set has a decent help for the use of Android. It is a famous installed information base; it gives an exceptionally straightforward SQL Mouth. It has an exceptionally little memory impression and high-velocity reaction.

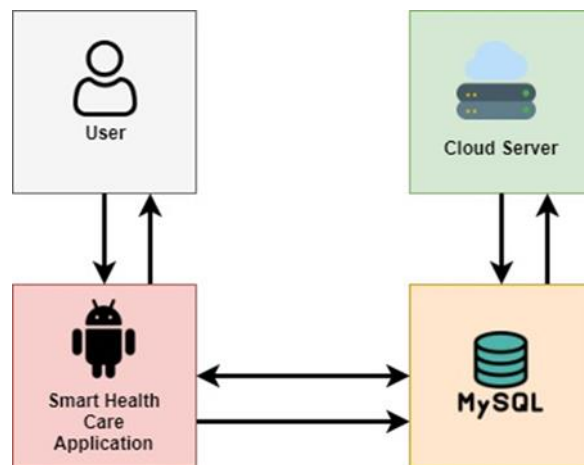


Figure 5. Database for smart health care

From Figure 5, we can see that the user uses the application and the application sends and retrieves data to the MySQL server. The server will communicate with the backup cloud server. It depends on C, java language improvement, and Android advancement is completely viable with the Java language. Thus, MySQL data set has a decent help for the use of Android.

## 2.5 MVC Design Pattern

The primary motivation behind MVC was to present a good division between the information layer and the introduction layer, making the space objects (model) unconscious of the introduction objects. To accomplish that, it utilizes the Observer example to help different introductions (Observers) of one single space model.

- Model: A model exemplifies application-related information and characterizes the rationale and figuring strategies for information preparation. In the Smart Health Care framework, we could peruse patients' data from the information base make

them a model individually as a model. There are balanced or one-to-numerous connections between model items, so the model layer in the application relates to at least one article chart. Model items address explicit issue space information; they can be reused over and over to similar issues.

- **View:** A view object is an article that the user in the application can visualize. A view object realizes how to draw itself and react to the client's activities. The primary motivation behind the view object is to introduce the information in the model object and permit the information to be altered. Notwithstanding this, in the use of the MVC design, the view protests generally dispose of the model items.
- **Controller:** The controller object fills in as a mechanism for at least one view item and model article. The regulator is identical to a core that maintains the whole interface. If the View or Model changes or communicates with the client previously, at that point, the regulator can get this input. The Controller can likewise deal with the existing pattern of different items.

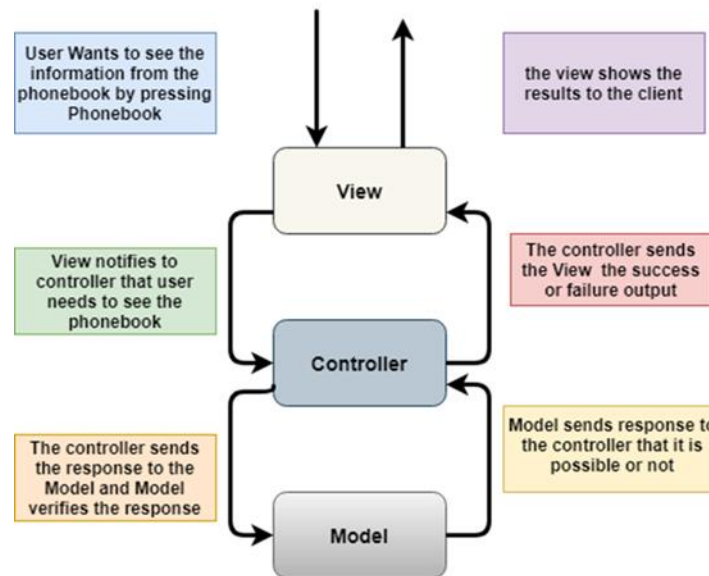


Figure 6. Relationship between view, model, and controller

Figure 6 shows the correspondence and information stream to assemble an application that makes utilization of MVC in a more current manner:

The Activities, Fragments, and Views ought to be the View in the MVC. The Controller and Model ought to be independent classes that do not expand or utilize any android class. The View and the Controller rely upon one another, yet for testing purposes and to

regard the dependency reversal rule, the View should execute an Interface. The Controller should reference that and not the solid class straightforwardly.

## 2.6 Mapkit

These days, there are numerous applications in the market that utilization guide and guide-based arrangements. Now, Huawei gives Map Kit as an answer for engineers. Huawei Map Kit is an SDK for map advancement; with this SDK, we can undoubtedly coordinate guide-based capacities into your applications. Map Kit utilizes the WGS 84 GPS arrange framework [1], which meets most guide improvement prerequisites outside China, including:

- Map collaboration: Controls the connection signals and fastens on the guide.
- Map drawing: Adds area markers, map layers, overlays, and different shapes.

With the Maps SDK for Android, you can add maps dependent on Google Maps information to your application. Likewise, you can utilize API calls to add markers, polygons, and overlays to a basic guide and change the client's perspective on a specific guide region [4]. The API allows you to add these designs to a principle:

- Symbols linked to explicit situations on the direction (Markers).
- Sets of line sections (Polylines).
- Encased units (Polygons).
- Bitmap illustrations secured to clear conditions on the guide (Ground Overlays).

### **3 Smart Device Selection**

There are several works; research has already been completed on the field of Health Care systems. There are many ideas, applications, and devices invented to better the health care system. Recently, “MobSpiro,” “eCAALYX (Enhanced Complete Ambient Assisted Living Experiment),” etc. systems create buzz, and the applications get the attention of the developers and researchers. This chapter contains state-of-the-art, existing smart devices, the selection of smart device and its results and accuracy, dataflow design, and a functional block diagram that describes how the smart application works and the use cases that will be used.

#### **3.1 State of the Art**

An android application has been developed in [5] to provide a viable health care system. The application offers several features like a cabin booking system, the suggestion of choosing hospitals based on the cost and service, the information of hospitals and exact location, information of doctors and their availability, emergency ambulance system, alarm system for taking medicine, and BMI (Body Mass Index) calculator. But there is no central system where the patient and doctor will be connected from home, and there is no work about data security which is a measure concern nowadays.

An e-Health system is designed in [6] to fulfill any emergency needs for older people. The system uses two embedded devices and an algorithm where embedded kits are necessary for patient and ambulance services to get the actual benefits. These kits rely on the IoT (Internet of Things). The algorithm is used for deciding whether the situation is critical or not. Some devices are developed for getting input from the user end. But there is no backup system for data retrieving. One important point is the accuracy of their developed devices, and the system is expensive to develop. There is also no plan about how to secure all the information of users.

A health kiosk [7] is developed for critical patients in the ruler area. An architecture is designed where some monitoring devices (developed by the 3-lead system) are used to get the data; these data will store in the android or tablet device (which will operate by a trained physician) through the diagnostic device. After that, the data will upload from android to the cloud. Finally, the doctor can retrieve the information from the cloud. But

a large-scale workforce is needed to accomplish this task. On the other hand, the android application is only used to get information and send the information to the cloud; here, no communication system, alert system, encryption, and decryption has not been implemented to provide secure service.

An android application in [8] is developed for verifying every user to log in to their account, carrying all users' personal and health information. All the data will be stored in the cloud database. Moreover, the author provides an idea about creating a smart card in the smartphone and a Quick Response (QR code) that will serve as a sharing device work with other health care providers. This system helps generate decisions after evaluating the previous health information before providing any treatment. The doctors assess the QR code of the patient with the android application and get all the information about the patient, medicines, and suggestions that will appear to the user end. This is because doctors and pharmacists can scan the QR code, see the patient's prescription, and provide services conveniently.

The article [9] provides an overview of what exactly Smart Health Care is. On the other hand, the latest technology used to develop smart health care has been described in the article, also provides the idea about which components and considerations are required to establish the framework of smart health care. Some challenges and future work also describe overcoming any existing problem or developing the framework. But there was no specific work related to security.

Things speak IoT cloud [3], which is MATLAB analytics cloud, has been developed to collect data, analyze data considering various conditions, and store the interpreted data to measure the heart rate. Providing the patient's heart is normal or abnormal. Smartwatch, Smart device, and android application are used to get the data of the patient. This system provides the notification to the patient or the patients' relatives if heart rate inclines be abnormal after analyzing all information. But they have no review that what type of smartwatch can provide the least amount of error value.

The work [4] provides an overview of different sensors and how sensors are manufactured in wearable devices to measure and monitor several patient parameters. The paper illustrates collecting information, analyzing, and generating the health report based on the sensor performance. An application is used to collect health information from a

smartwatch after extraction, then create a report inside the application. But there was no section about the accuracy of parameters of the smartwatches.

An IoT system in [10] has been developed to help different sensors like Spo2 sensors, temperature sensors, blood pressure sensors, Bluetooth technology, Arduino board, and android technology. Various sensors have been used to measure patients' different parameters, and Bluetooth technology has been used to synchronize the value to show or upload in the android application. Finally, the android application has been developed to keep the connection between doctors and patients from home. But there is no messaging system so that patients can communicate with the doctor or vice-versa. Moreover, location, additional information updating, and prescription technology are not available in this system. The most important drawback of this system is that the system has been developed for only three patients' parameters. Important parameters such as Blood Pressure, Glucose, and ECG parameters are important to provide smart health care from home.

A system in [11] has demonstrated a manageable random methodology to analyze the effects of dynamic changes in context on the interaction of the smart medical care system, computing infrastructure with its environment to consider the intelligent medical care system security analysis. But there is no analysis, experiment, or initial studies conducted on the security of smart medical care systems. The system has no comprehensive developed security, safety, and sustainability analysis tool.

The security awareness and assurance framework [12] presents a framework for 5G smart medical services that use zero-trust design—driven by the improvement of 5G smart medical care. It is related to security challenges, the first to propose a 4-dimensional security structure for 5G competent medical care thinking about four measurements. Based on this system, the system presented our security awareness and insurance framework's engineering and examined the accomplished security implementations for network virtualization, information cooperation, IoT access, and coordinated 5G organization security. These are helpful to meet the security needs for 5G smart clinical applications. Our framework is actualized, what is more, tried altogether for its functionalities and execution. But there is no norm to survey the development of a security arrangement utilizing a zero-trust design. The issues of inconsistency with existing security structures in the 5G brilliant clinical industry exist.



## 3.2 Existing Technology and Device

The choice of device executed in this examination was based on distributed Journal papers on a very basic level. An orderly content examination uncovered topics that portray the device's vital capacities tended to by the documents. Here are the short points of selection of the device.

### 3.2.1 Existing Smart devices

**Wearable biosensors:** A significant piece of innovation that empowers virtual emergency clinics and different sorts of care, illness anticipation, and sickness location is the wearable biosensor. These gadgets, which are little and lightweight and worn on the body, screen indispensable signs like temperature, pulse, and breathing rate, furnishing health experts with necessary experiences.

**Multi-useful smart monitor:** Mintti Vision is arranged with the circulatory strain estimation module with a MITSUMI high exactness pressure sensor [13]. Not exclusively will it screen and record circulatory strain information. However, it will likewise record essential fundamental signs, for example, pulse, ECG, blood oxygen, and breath. This information represents an individual's particular condition of well-being.

**SmartWatch:** The Apple Watch Series 6 dispatched with blood oxygen, estimating usefulness. Apple is purportedly setting out on a progression of well-being concentrates with research foundations to study how changes in blood oxygen levels can flag the presence of respiratory conditions like asthma. ECG or electrocardiogram is an element that is beginning to be accessible on some smartwatches like the Apple Watch Series 4. It gives more exact pulse estimations and provides more information for identifying possible heart-related issues.

### 3.2.2 Selected Device and Working Method

The Checkme Pro-Health Monitor recently delivered Conformité Européene (CE)-affirmed brilliant across-the-board gadget [13]



Figure 7. Checkme Pro Device [13]

The device estimates four of the five MEWS (Modified Early Warning Score) vital signs in under 25 seconds and can undoubtedly be cared for by patients. Viatom Checkme Pro is a clinical item that can be adjusted to various conditions; It is not just reasonable for home medical services, yet besides for the clinical condition. Checkme Pro has three of the greatest focal points. It depends on a center item grew only by Viatom. It has gotten the FDA, CE, iF Product Design Award and is traded to numerous nations.

The examiner estimations were constantly done quickly previously or after the Checkme assessment to acquire an exact MEWS. The estimation meetings were led in the first part of the day, evening, or night, consistently as close as conceivable to an ordinary medical nurse estimation, inside a maximal time window of 30 minutes for all measures. Patients were not permitted to leave their beds during the measurements. Estimated fundamental signs were systolic BP, temperature, SpO<sub>2</sub>, oxygen organization, and AVPU (Awake, Verbal, Pain, Responsive). A MEWS count was then performed by set up convention. The best quality and medical nurse vital signs were estimated utilizing a BP monitor Dinamap prepared by GE healthcare in Germany, a heartbeat oximeter like GE healthcare, and a tympanic thermometer Genius 2 designed by Medtronic in the USA [14]

The factual examination was performed utilizing Statistical Package for the Social Sciences (IBM SPSS Statistics rendition 20.0, SPSS inc., Chicago, Illinois, USA). The imperative signs were portrayed utilizing mean with standard deviation (SD). Altman plots were made to evaluate intra-spectator changeability and contrast in fundamental signs and determined MEWS estimated by the agent, the medical nurse, and the patient (Checkme). In the plot, each information point addresses the contrast between the two estimation techniques. The strong line addresses the mean contrast, and the dotted lines address the constraints of understanding (1.96 SD) [14]. A one-example t-test was performed on the distinction between the two estimating techniques. A p-esteem < 0.05 was viewed as critical. For each fundamental sign, the clinical satisfaction between the

three strategies for estimations was predestined. Clinically pertinent contrasts were considered as follows: 5+ beats/min (HR); 5+ mm Hg (systolic BP); 0.5+°C (temperature); 2+ breaths/min (RR); 2+% SpO2 [14]. A distinction in MEWS score of 1 point or more between various estimation sets was moreover viewed as clinically significant.

### 3.2.3 Results and Accuracy

Fifty continuous patients were examined for examining, in any event, one bunch of fundamental sign estimations and MEWS figures. Patients' socioeconomics and consequences of the Checkme adjustments are portrayed.

Table 6. Illustrations of study and aftereffects of the alignment system [14]

Category	Total	Men	Women
Gender (%)	50 (100.0)	31 (62.0)	19 (38.0)
Age (years)	56.7 ± 15.8	58.7 ± 14.0	53.4 ± 17.8
Weight (kg)	79.5 ± 18.8	84.0 ± 16.5	72.2 ± 20.5
Length (cm)	171.9 ± 26.7	180.0 ± 7.1	158.7 ± 39.5
Calibration of Checkme			
Successful calibration (%)	41 (82.0)	25 (80.6)	16 (84.2)
Number of successful attempts (%)	30 (73.2)	18 (72.0)	12 (75.0)
1.			
2.	7 (17.1)	6 (24.0)	1 (6.2)
3.	4 (9.7)	1 (4.0)	3 (18.8)

In Table 6, In 41 of 50 patients (82%), starting the Checkme was fruitful, and results were qualified for the examination. Two arrangements of estimations were acted toward the beginning of the day (6:30 AM), 49 sets in the early evening (2:00 PM), and 18 groups at night (8:00 PM). This brought about an aggregate of 69 estimation sets in 41 patients. Nine estimations performed by medical caretakers (13.0%) were not finished.

**General outcomes:** Patient estimations utilizing the Checkme required roughly 30 seconds for every tolerance. Patients also need an extra 6–7 minutes to adjust the gadget. A fruitful first endeavor BP adjustment was acquired in 30 (73.2%) patients. Rehashed alignment endeavors were required in different patients, and adjustment is not acceptable

in 9 patients. Most disappointments were attempted to be because of shuddering or cold hands. Alignment disappointment was not found to associate with patient age, weight, or gender. Estimations of the result were also discovered to be to some degree dissonant, with 14 (24.1%) of all assessments varying by 3–4 breaths/min and 17 (29.3%) of all estimations contrasting more than five breaths/min—differs in determined MEWS [14].

Contrasted and highest quality level MEWS, the medical caretakers' MEWS varied by at least two of every 15 (25.8%) cases. Paradoxically, in 10 (15.4%) cases, MEWS differed two focuses or more between the highest quality level MEWS and Checkme. Most MEWS computations changed by a 0–1 guide between the two strategies. Mean contrasts of determined MEWS were in reach with the predefined acknowledged inconsistencies. Three of 69 MEWS counts by highest quality level, and medical nurses felt external the constraints of understanding and examination results; MEWS counts based on imperative signs acquired by the highest quality level of estimations varied altogether from the MEWS dependent on associated assessments, however, were equivalent to the MEWS gotten from Checkme estimations. All the results and graphs are available in [14].

### **3.3 Data Flow Analysis of the System**

Data flow outlines give a graphical portrayal of the framework to be open to computer experts and non-expert users. The models empower computer programmers, customers to cooperate successfully during the analysis and detail of requirements. Even though this implies that our users are needed to comprehend the displaying strategies and builds, in data flow displaying just a restricted arrangement of developments are utilized, and the principles applied are intended to be straightforward and simple to follow. These equivalent principles and actions apply to all information stream graphs. DFD has regularly been utilized because of the accompanying reasons:

- Coherent data flow of the framework
- Development requirements
- Straightforwardness of documentation
- The framework extension and limits are shown on the outlines

- The fragmentation of undeniable level information stream charts to many more point-by-point graphs gives a general perspective on the comprehensive framework.

All data flow charts are made of similar sorts of models, and the approval rules are the equivalent for all DFDs; there are three fundamental kinds of information stream outline:

- Context diagrams — setting graph DFDs are outlined that present an outline of the framework also, its collaboration with the remainder of the “world.”
- Level 1 information stream graphs — Level 1 DFDs present a more itemized perspective on the context than setting outlines by showing the fundamental sub-cycles and stores of information that make up the framework.
- Level 2 (and lower) database charts — a significant benefit of the information stream displaying strategy is that through a method called “37 leveling.” Certain components of any dataflow outline can deteriorate (“detonated”) into a more detailed model, a level lower in the pecking order [15].

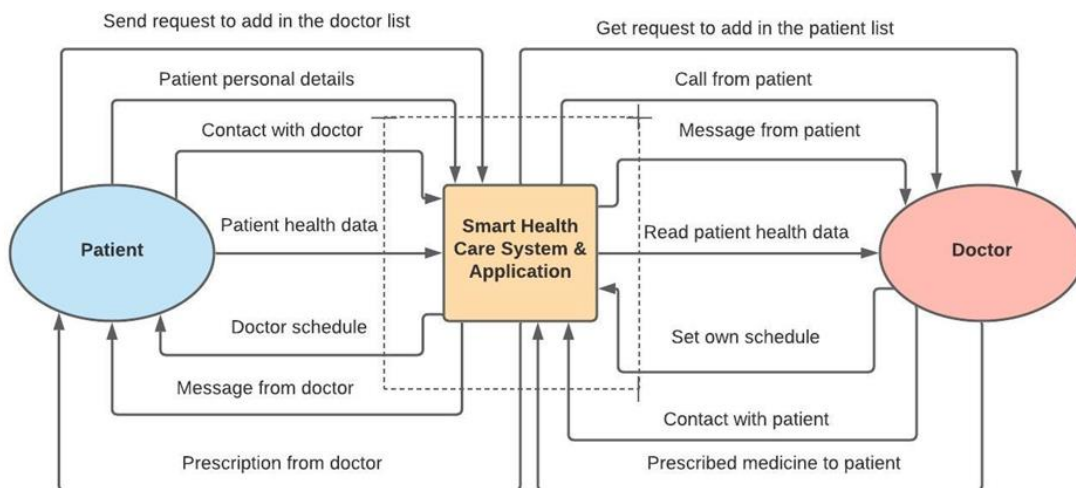


Figure 8. Level 0 or context diagram for the smart health care system

Figure 8 illustrates the level 0 or context diagram of the smart health care system. It shows the basic indication of how the data is going to flow. It also represents the communication of patients and doctors through the smart health care application. There are several functionalities available for patients: register an account, choose a doctor, input personal health information, make appointments for doctor meetings, communicate with doctors,

call an ambulance for any emergency, and get any prescription if doctors provide from home through the system application. This data flow also tells how the system application's backend interface exchanges information from the patient to doctors or vice-versa. Doctors also use the system's application to provide the basic and primary health service from home to help save time and energy.

The accompanying advances are proposed to help develop Level 1 DFD: Firstly, each dataflow into the framework analyses the documentation about the framework and converse with the clients to build up a conceivable cycle of the framework that gets the information stream. Secondly, distinguish information stores by building up where records/information should be held inside the framework. Finally, add information streams streaming among cycles and information stores inside the framework.

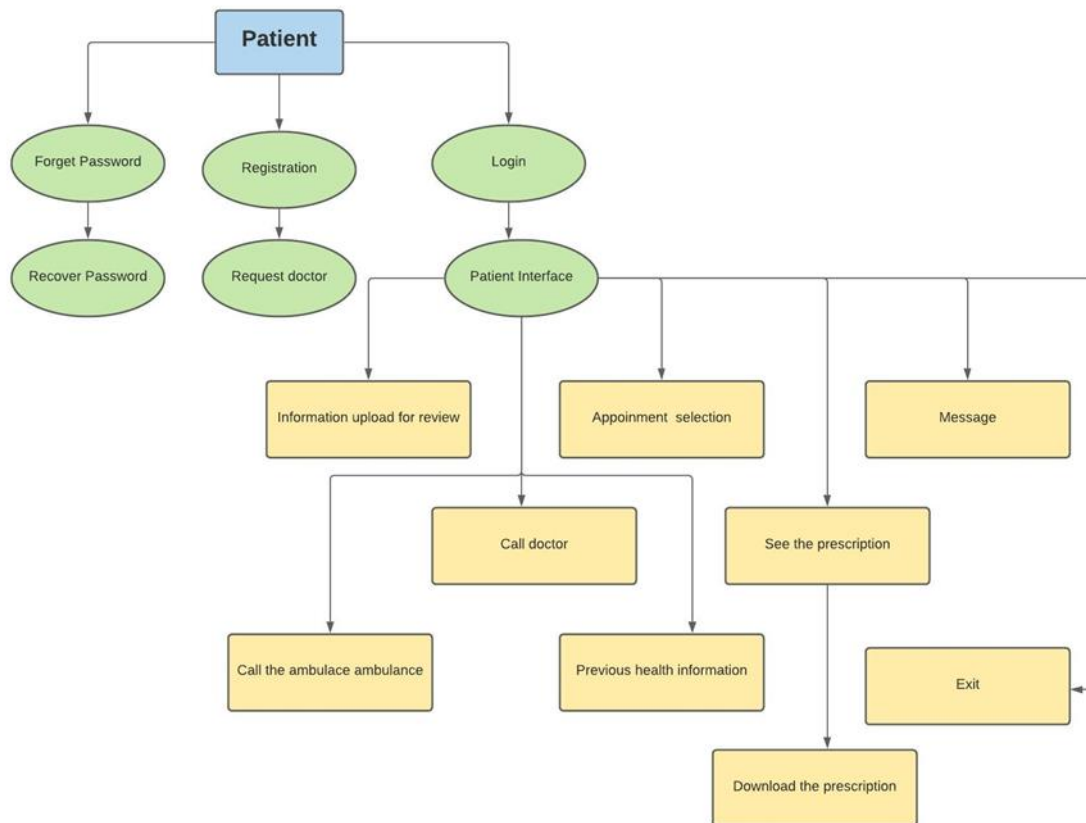


Figure 9. Level 1 or context diagram for patient interface in the system

Figure 9 represents the breakdown of the elements and their functionality. The system's application provides the opportunity to register themselves as patients, access their account, and request to register themselves under the nearest family doctor or hospital doctor depending on their location, and finally enter the system interface. After entering the interface, the patient can have the privilege to share their health data for assessing

their registered doctors; the system also provides the options like a message to the doctor, call the doctor, call an ambulance in an emergency. Patients also get the chance to make an appointment with the doctor according to the provided schedule. They can see their prescriptions or comments (if the doctor offers) and download them for their purposes. Finally, they can sign out from their account.

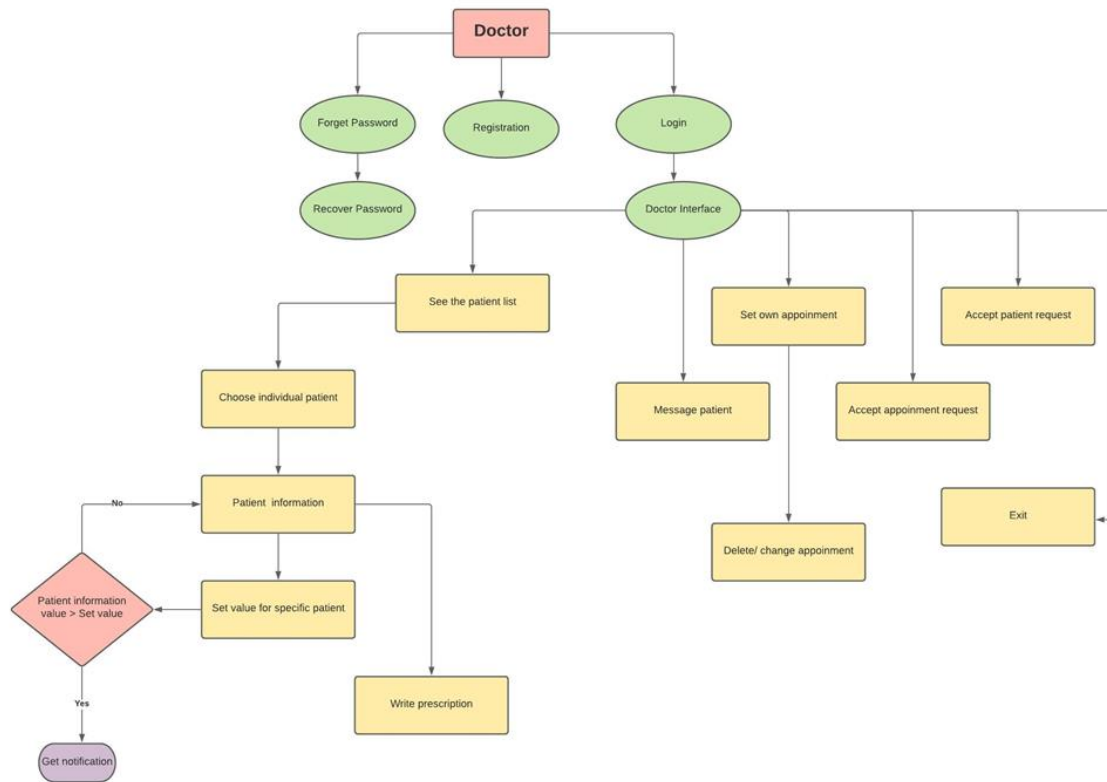


Figure 10. Level 1 or context diagram for doctor interface in the system

Figure 10 illustrates the breakdown of the doctor interface that is the same as the patient at the very beginning. The main interface provides options to the doctor to accept requests from the patient, communicate with the patients, and create the appointment schedule on their own, and have the availability to remove the scheduled time or request from the patient end. Moreover, they have the option of the patient list where they can see all their patients and their uploaded health data for assessment. Doctors have the privilege to set the parameter list of the individual patient for getting an immediate notification (if any parameter is abnormal for a patient) and prescribe medicine or any comments required. Finally, they can sign out themselves from the system.

The numbering of cycles in the previous DFD is gotten from the quantity of the parent interaction. So, all processes in the child DFD of cycle two will be called DFD 2. Take a

gander at the rectangular limit for this level 2 DFD [16]. Outside the boundary is the outer substance “Specialist doctor and Patient.”

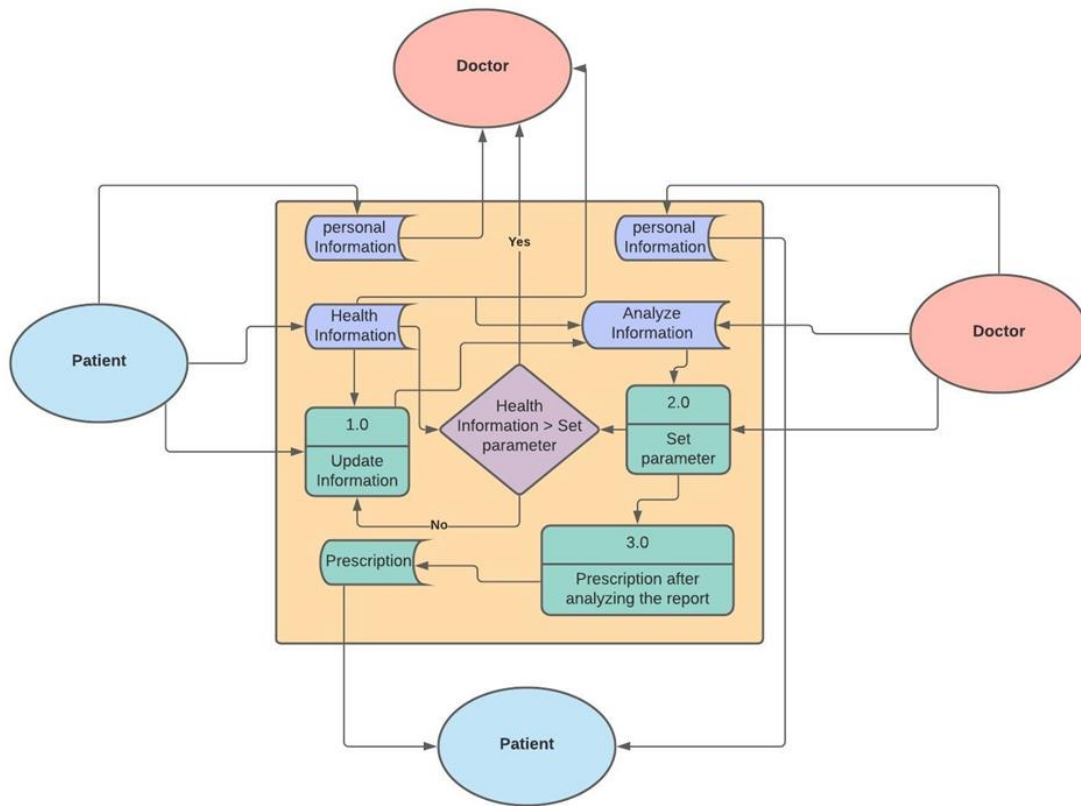


Figure 11. Level 2 data flow diagram for patient and doctor’s interface in the system

Figure 11 portrays a level 2 data flow diagram representing which data are stored and update the data. The system will store all the personal information of the doctors and patients in the system. Here the health information which the patient provides will be stored in the database and can be updated health data if the new data input into the system by the doctor. In Figure 11, a decision gate is used for comparing the input health data from the patient end and the set parameter for the patient by the doctor for getting a notification to avoid any unusual situation and get treatment at the right moment. The system will also store the communication to prevent any exceptional problems. The prescription provides by the doctor will also keep in the system so that patient can download or view the prescription.



### 3.4 Functional Block Diagram

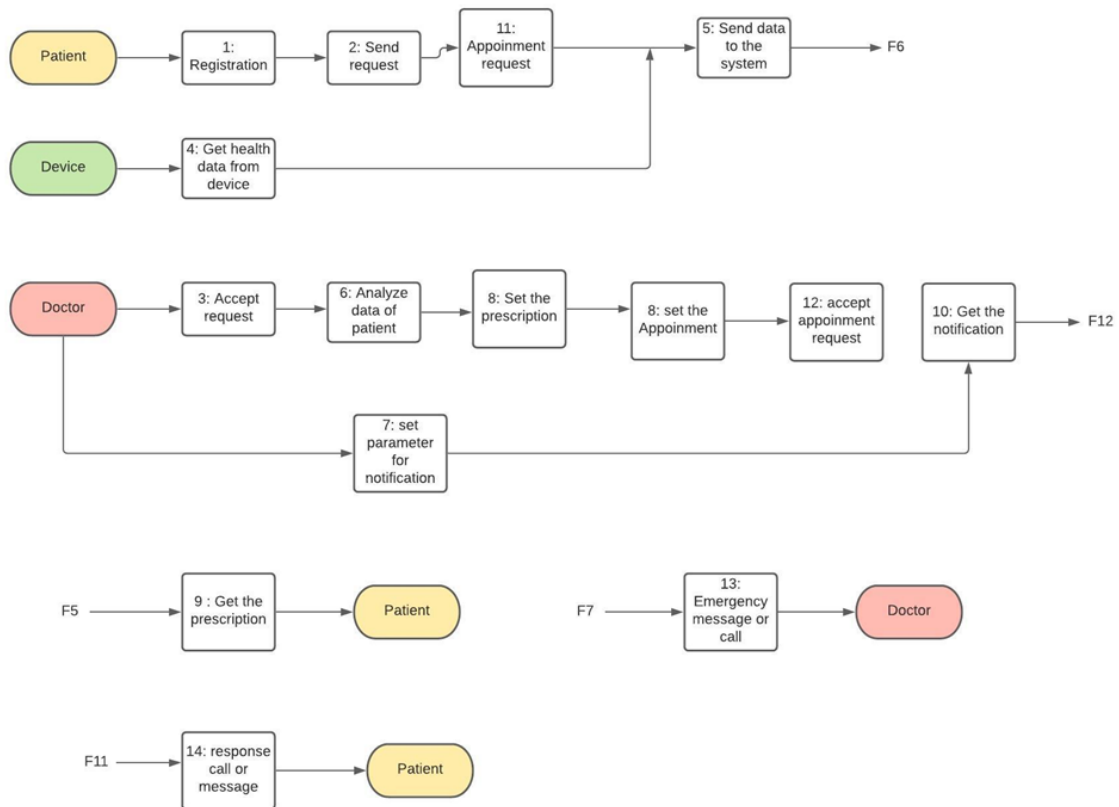


Figure 12. Functional block diagram.

Figure 12 describes all the functional blocks of smart devices, smart applications, and stakeholders will communicate and how all the functionalities work properly. Here, the patient registers in the system and connects with the doctor. Then, the patient uses the smart device to get the value of the necessary parameters. Then, the patient sends the data to the doctor by using this application. The patient can also pick a time for an appointment if the doctors set any appointment schedule. The patient also sees the prescription if any prescription available.

On the other side, a doctor will accept the patient's request. After making the request, the doctor will see the patient's health data for analysis. Then, the doctor will write a prescription and set the parameter for every specific patient. The doctor will also be able to put his suitable time for an appointment.

Table 7 describes all the functional blocks in detail below. There is a clear idea about how the functionality occurs between every block.

Table 7. Functional block and their description

#	Functional block	Type	Description
F1	Register as a doctor	Input	The doctor or family doctor must register by filling in several information, but phone no and location are very important.
F2	Register as a patient	Input	The user or patient must register by filling in the information, and here, the selection of a doctor family physician is most important depending on the patient's location.
F3	Send request	Input	The user or patient has the privilege to send a request to the nearest family physician to get the treatment under his supervision.
F4	Accept request	Input	The doctor can see the patient's request if everything is correct, then the doctor can accept the patient's request.
F5	Receive health data	Output	The user or patient takes health information or data from the device, e.g., blood pressure, glucose, spo2 parameters.
F6	Send commands in the system	Input	After receiving the health information, the patient or the user needs to enter the system to send the information to the doctor.
F7	Send data	Input	The user or patient sends health information by filling all parameters in the system's application for analysis by the doctor.
F8	Forward information or data to the	Transformative	The information from the user interface passes to the doctor interface to evaluate the health status of the patient.
F9	Evaluate the status	Input	The doctor can see the status of the patient and evaluate the condition of the patient.
F10	Set the parameter	Input	The doctor can set the parameter for specific for getting the notification for the critical patient.
F11	Compare the value	Transformative	The system compares the value from the patient with the set value from the doctor
F12	Get notification	Output	If the value exceeds the doctor's set value, the doctor will get the notification by providing which patients which parameter is not under control.

F13	Prepare prescription	Input	The doctor can prepare the prescription for the patient with his signature for the patient if needed.
F14	Forward the prescription	Transformative	The system saves the prescription and sends it to the patient end.
F15	Save the prescription	Output	The patient can see the prescription and saves or download the prescription
F16	Send message	Input / output	The doctor or patient can send messages to each other to keep the communication.
F17	Forward the message	Transformative	The system saves the message and sends it to the patient or doctor end.

Table 7 illustrates the idea about the functionality of every block. The table also shows the block sequence number which is needed for the synchronization and the type. The type also explains that the block is input, output, or transformative. The description section contains the elaboration of the functionality or the working principle of every block.

### **3.5 System Validation and the Relationship**

The Unified Modelling Language (UML) and SysML are the standard languages for indicating, picturing, building, and recording the systems architecture. Each chart is intended to let designers and users see a product diagram from an alternate point of view and fluctuate levels of deliberation. The ordinarily utilized UML and SysML graphs are as per the following.

#### **3.5.1 Block Definition Diagram**

The two significant properties are as per the following: The part properties, which portray the disintegration chain of importance of a square. Secondly, the reference properties describe the connections between blocks that are more fragile than the past ones and worth properties, which depict the quantifiable qualities of a block.

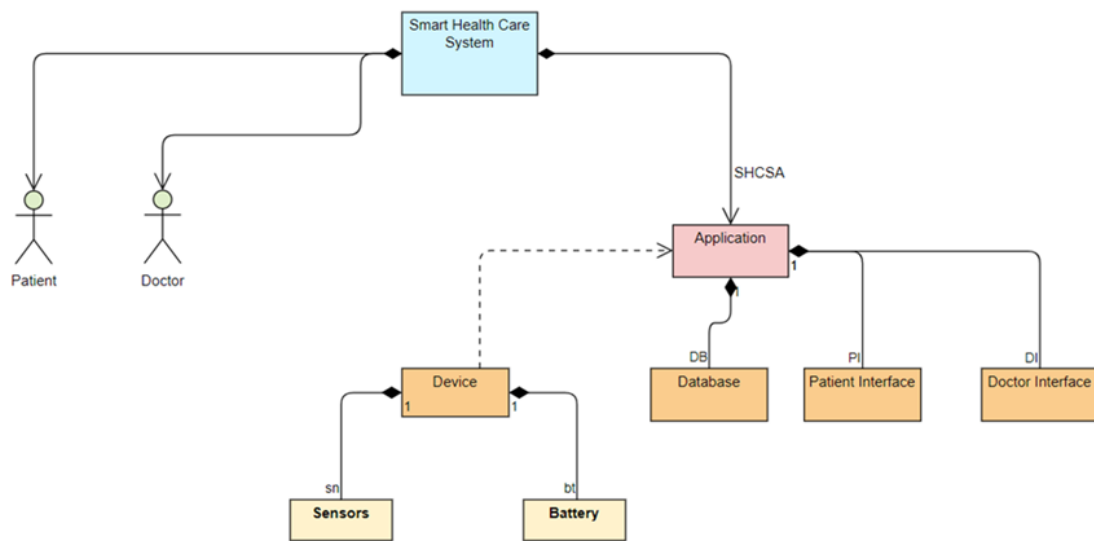


Figure 13. Block definition diagram for the smart health care system

Figure 13 shows different blocks, which break down the design of a block and illustrates the relationship between several blocks, showing the system's dependency. The main smart health care system is decomposed by the patient, doctor, and the main application. There is a dependency between the application and the device because the health information data comes from the device, and several sensors and batteries decompose the devices. On the other hand, the application is decomposed by three important blocks, such as the database used to store several data for patients and doctors. The patient interface and doctor interface decompose several functionalities.

### 3.5.2 Use Case Diagram

A use case chart depicts many situations that uncover the cooperation between the user and an application. A user can be either an individual or another application. A use case alludes to the activities that the framework can perform by collaborating with its entertainers. Subsequently, the use case chart shows a progression of activities and the entertainers that can play out those activities. Use cases are utilized in the underlying period of the advancement interaction. They are valuable in inspiring the product and framework's prerequisites. Most use cases are characterized in the underlying stages, while new ones may be fundamental during the improvement cycle. In addition, use cases are utilized to distinguish the assumptions for a framework, its highlights, and the common conduct among its parts.

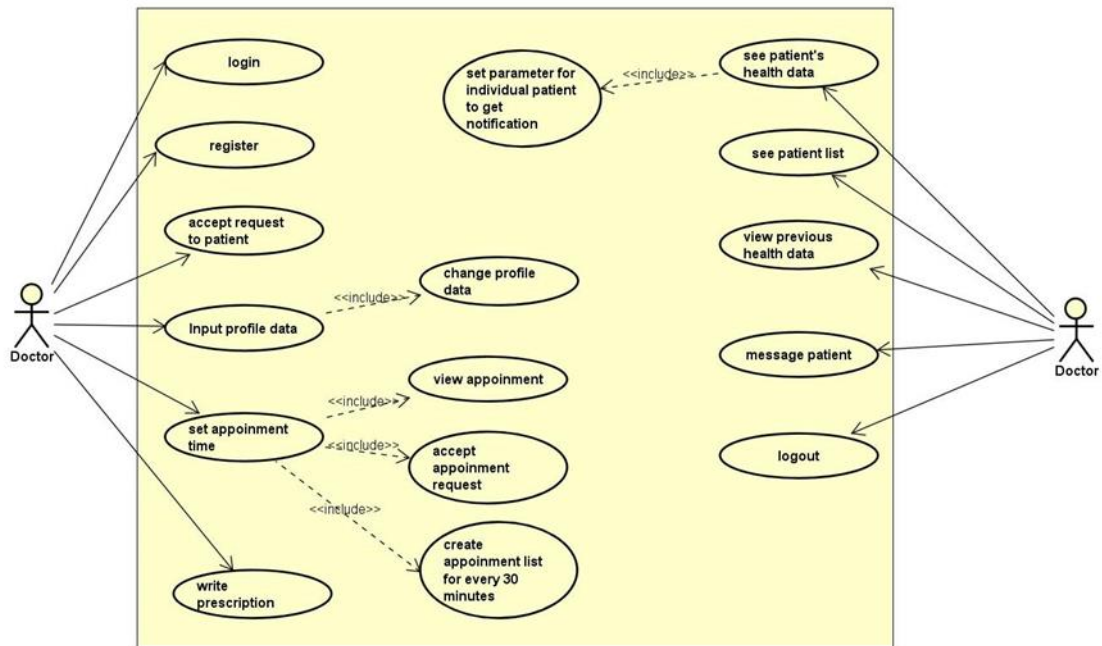


Figure 14. A use case diagram for doctor interface of the smart health care system

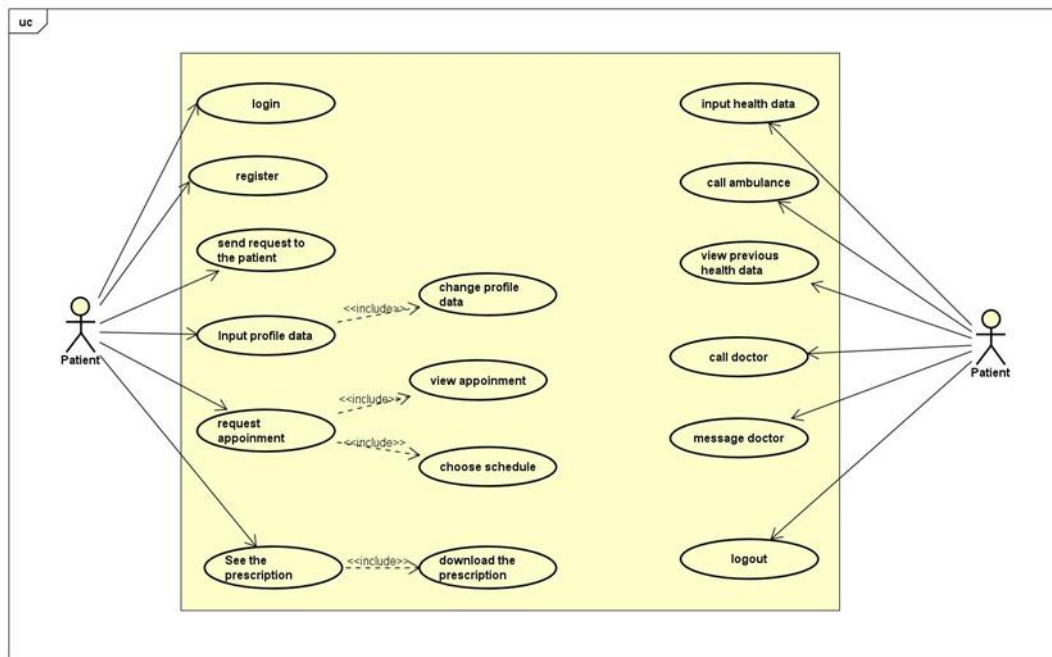


Figure 15. A use case diagram for patient interface of the smart health care system

Figure 14 depicts the use case diagram for the doctor interface. The figure describes the relationship between the functionalities and an actor (doctor). The association rule between doctor and the functionalities represents the establishment of the doctors and the subject in accomplishing the functionality associated with the use case.

Figure 15 depicts the use case diagram for the patient interface. The figure describes the relationship between the functionalities and an actor (patient). The association rule between patient and the functionalities represents the establishment of the patient as an actor and the subject in accomplishing the functionality associated with the use case.

### 3.5.3 Sequence Diagram

Sequence charts depict the connections inside a framework utilizing imparting substances addressed by a square shape that helps plummet. A relationship depends on the trading of

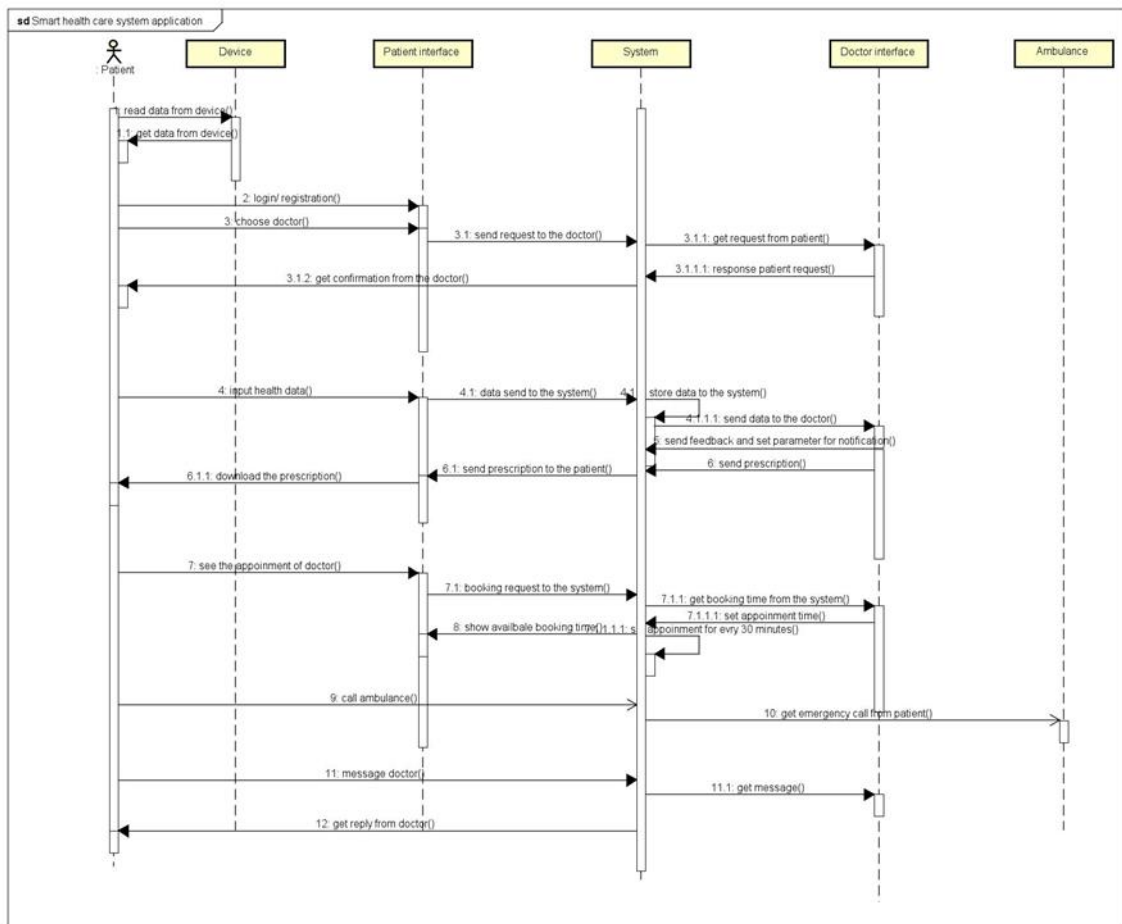


Figure 16. Sequence diagram for the smart health care system

messages as activity calls or signals masterminded in the transient requests. Articles accept the job of such imparting elements. The body of a help addresses the existence pattern of its comparing object. In Figure 16, the sequence diagram of how the system handles identifying the processes is depicted. It showcases the situation where the device, patient interface, and doctor interface, system, and ambulance system Unit identify every step of the happening events and send a request to the nearest block to decide on whether

to send the message or not. The system can either make the decision automatically or manually through input from the patient through the system functionality.

### **3.5.4 Analysis**

SysML has been broadly utilized in the realm of programming, and its use in framework designing was very effective for a long time. Notwithstanding, the need to have a customized variant of SysML that meets the arising prerequisites of the present frameworks designing experts has arisen. Thus, the SysML demonstrating language has been created as a UML profile.

SysML is a fundamental empowering agent for model-based framework designing. It permits a framework's model to store plan choices at the focal point of the frameworks' turn of events measure. This offers an upgraded advancement worldview that improves improvement speed, correspondence, and productivity. SysML is acquiring expanded prominence and numerous organizations from different fields, like safeguard, auto, aviation, clinical systems. Two significant elements catalyze this pattern. In the first place, SysML is lined up with the ISO AP233 information trade standard, and it acquires the XMI exchange from its archetype (UML) [16]. Second, an expanded number of devices effectively offer help for the still youthful demonstrating SysML language. These diagrams give a good idea that this system can be easily implemented. Every diagram illustrates the concept of establishing the communication between the doctors and patients. These diagrams also provide the idea of establishing the relationship between the smart device (which is used for measuring the parameters) and the application (where doctors and patients will be connected).

## 4 Design, Implementation and Result

All the diagrams in chapter 3, Smart Device Selection, describe the functionality and the communication approach between doctor and patient. Here there is written the idea about how the location redirects between doctor and patient, the management system, design methods, design principles, market analysis to explain why the system is important, how the system can be implemented considering the existing and new technology. Finally, a data security mechanism will explain in detail so that health information is safe and secure.

### 4.1 Specification of Smart Health Care

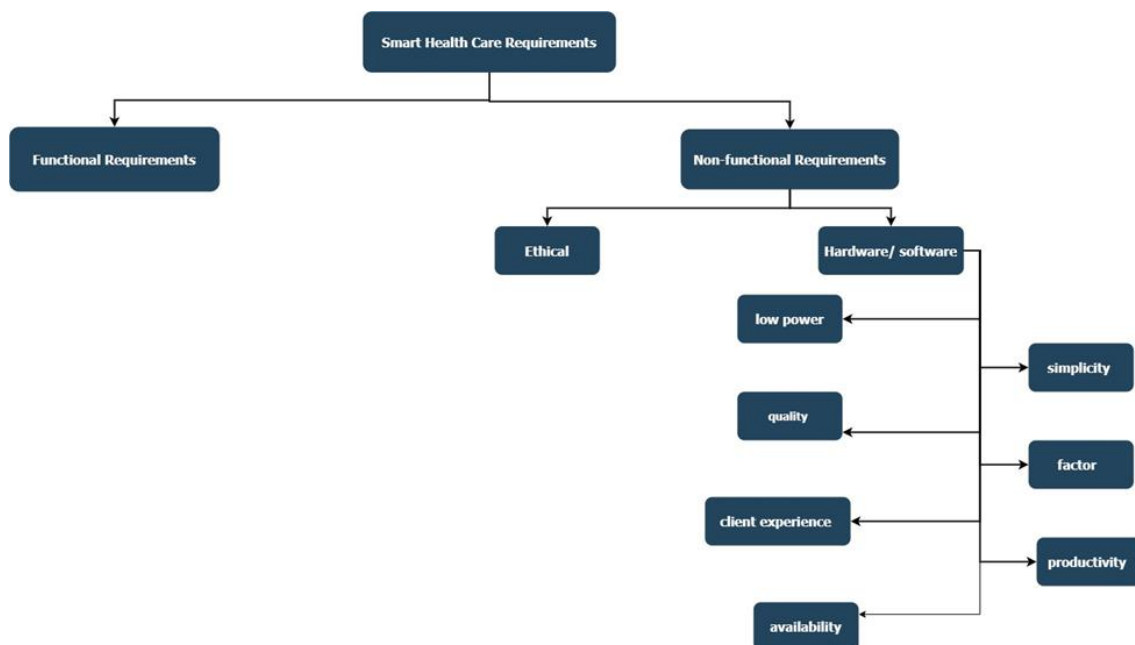


Figure 17. Requirement specification of smart health care

Requirements of smart health care can be extensively ordered into useful functional and non-functional prerequisites, as demonstrated in Figure 17. Operational constraints address explicit requests of smart health care design. The basic models for a productive, strong health care framework are:

- little structure factor
- framework quality assurance
- type of administration



- enhanced patient experience
- capacity to interoperate across various stages
- simplicity of organization
- the universality of the smart health care framework to offer nonstop service
- sufficient availability

## 4.2 The management system of Smart Health Care

Figure 18 clearly shows that the smart health care management system is organized and transparent. The system can ensure the proper medical services to clients (e.g., doctor and patients) from home and connected to the system. An admin panel is also engaged with this management system to provide technical solutions to avoid unethical circumstances. The support assistance team also takes part in this management system to provide services to the clients and management teams, providing all the information and the structure for an intelligent medical services design that incorporates the libraries and the ordered into network stages, registering steps, and administration stages. Organization stages allude to the libraries used to interconnect various structures.

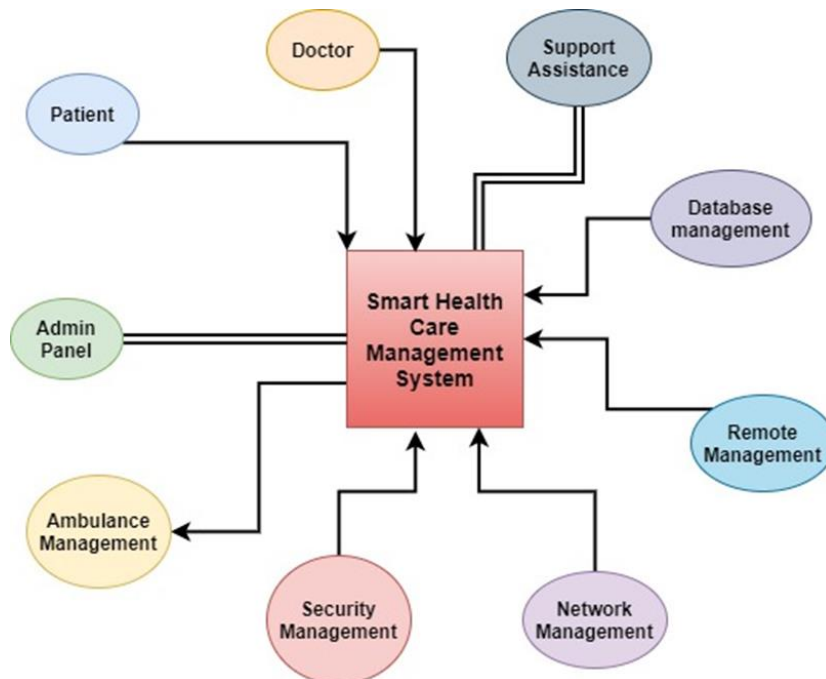


Figure 18. The management system of smart health care

Registering stages can naturally change dependent on the advancements utilized. Because of variations in the application conditions of keen medical services organizations, the designs for processing stages are normally a crossing point of more extensive ideas like information base administration, advancement, human-machine interface, and AI calculations [17].

### 4.3 Implementation and Result of Smart Health Care

All the diagrams in 3.3 Data Flow Analysis of the System describe the functionality and the communication approach between doctor and patient. Doctor and patient are the most important stakeholders in the smart health care system. The application in this system works as a bridge between the doctor and the patient. A doctor can get data of all basic health conditions of a patient, and a patient can observe their health condition regularly from a doctor from home. So, there is a prototype example to reveal the system’s functionality.



Figure 19. Doctor interface’s source code and demonstration

Figure 19 describes the views of the registration of doctors. Here are all criteria like name, email, phone, password, confirm password, valid document (which proves his professionalism), and location. This location is the most important feature that helps a patient save time searching the doctor in a specific region. Here, a phone number is very important because patients can reach the doctor by calling this number in any critical situation. Here all the variable’s value gets from the user converts into string and trim (for removing any space between word or letter). The system will store the values in the

database for future use. Doctors must submit the proof that he/ she is a doctor during registration. This location will appear to the patient view to get the nearest doctors; if one more doctor available in the same area, all doctors will be available in the dropdown menu.

In Smart Health Care System, one patient can register as a patient to the nearest doctor to avoid the pressure of patients to a single doctor, especially in the developing countries. This location feature is depending on the doctor where the doctor register. For example, suppose a doctor register in the “X” location. When a patient wants to register from the “X” areas, the patient can see the location and doctor available in the dropdown menu.



Figure 20. Patient interface's source code and demonstration

Figure 20 represents the code of necessary elements are required to register a doctor. Patients need to put all the data which are required to register. In the location, patients can select the region from where he/she wants to register. Then all the doctors will be available in that region. Here idNumber is used for differentiating the specific patient. Here all the variable's value gets from the user converts into string and trim (for removing any space between word or letter). The system will store the values in the database for future use.

Appendix 3 – Source Code (page 73, retrieve doctor location) talks about the working principles of the retrieveLocation function. Here, the API client is Google API Client object applied to get to the Google Play services' Google APIs.

Call<List<ModelRegisterAsDoctor>>call=apiInterface.retrieve\_doctor\_location (); this line represents that a list of doctor location for a specific region calls if the data is available in the database. At that point call.enqueue sends the ask for and tells callback of its reaction or if a mistake happened to converse with the server, making the request, or preparing the response. If there is a response, then accumulate data in a list and show it in a spinner way (dropdown way). Otherwise, there is an error if it is not responded to.

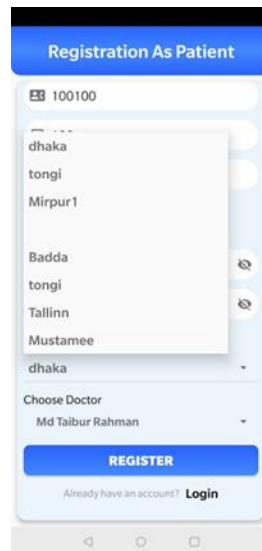


Figure 21. Demonstration of the dropdown location selection in the patient end.

Figure 21 demonstrates the view of the patient registration window. When any patients want to register with his/her location selecting the region from the dropdown menu. If any doctor is available in that region, the list of places and doctors will be available in the dropdown menu for choosing the doctor. If there are several doctors available, all the doctors will appear in the dropdown menu.

Appendix 3 – Source Code (page 77 To select a doctor from the dropdown at Patient End) Spinner is used for getting all the doctors in a specific region for a dropdown way. setOnItemSelectedListener is used for an event listener when an interface in the view selected class contains a single callback technique. These techniques will be invoked by the Android structure when the user sets off the view of all selected items available in the database to which the listener has been enrolled with the thing in the user interface. Public void onItemClick(AdapterView<?> parent, View view, int position, long id) describes the callback technique to be called when a thing in the view has been chosen. This callback is called just when the recently desired position is not quite the same as the newly chosen position or no chosen thing. Implementers can call getItemAtPosition(position) if

they need to get to the information related to the chose thing. Parent AdapterView: The AdapterView where the determination occurred, View: The view inside the AdapterView that was clicked, position int: The situation of the view in the connector, id long: The line id of the thing that is chosen. Then the doctor’s name and location are retrieved from the doctor model by using doctor id and selected location.

For this thesis, blood pressure, body temperature, oximeter, electrocardiogram, and glucose level are used. The patient can measure these parameters using any smart device or recommended device in this paper. After obtaining value from smart devices, the patient can send this information to the doctor.



Figure 22. Parameters for patients code and demonstration

Figure 22 represents the parameter list which sends to the doctor by the patient. All the parameter converts into string or trim and saves to the database. Here, the doctor will see the parameters and able to set parameters for a specific patient. If any value excesses the set value (by the doctor), the doctor will be notified.

When any patient sends all the parameters needed in the application, all the parameters will appear in the doctor interface. Then the doctor will be able to set parameters for each patient to get notifications considering the patients' age, previous health data, and gender. Here, the doctor will be able to write a prescription as well.

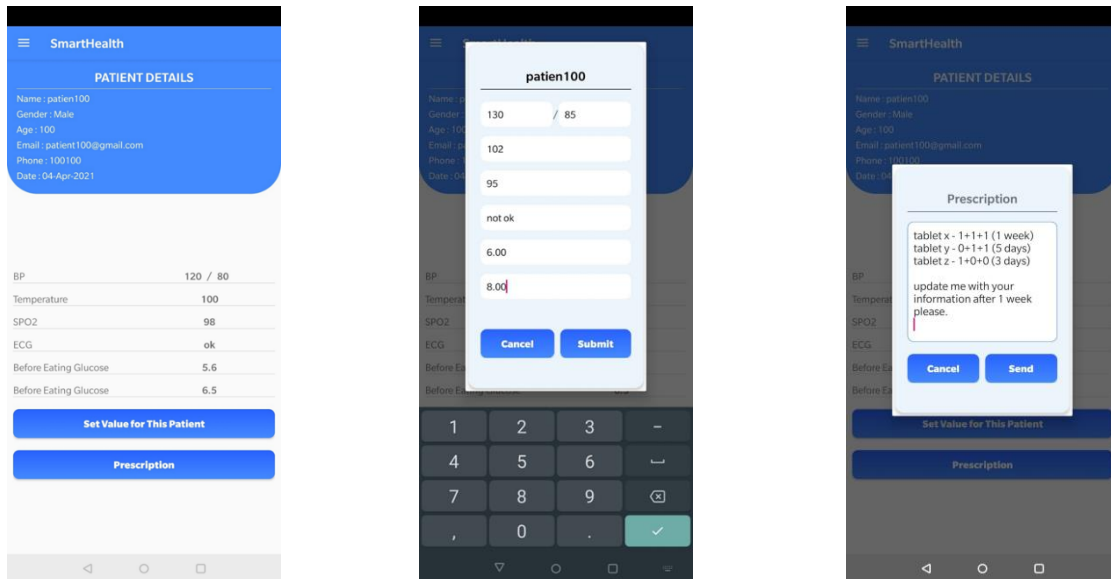


Figure 23. Set parameter and Write the prescription

Figure 23 illustrates. When doctors see the patient's parameters, the doctor can also see two options: set value and prescription. The doctor can set values for the parameters and prescription from the same page for getting notification.

```
#individual notification
if(floatval($p_up) > floatval($d_up) ||
floatval($p_down) > floatval($d_down) ||
floatval($p_spo2) < floatval($d_spo2) ||
floatval($p_bfr) > floatval($d_bfr) ||
floatval($p_after) > floatval($d_aftr) ||
floatval($p_temp) > floatval($d_temp)){
```

Figure 24. Logic code for getting notification

Figure 24 shows the logic formula for getting notification for a doctor. Here, p\_value for patient value which one patient sends and d\_value means doctor set value for a specific patient. These two values always compare, and notification criteria will work when patient values exceed the doctor values.

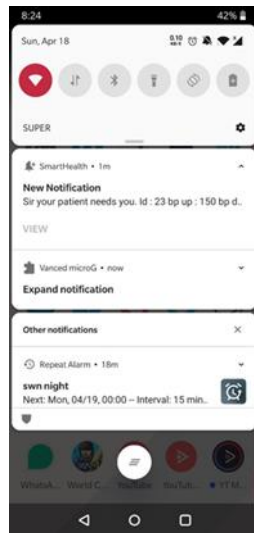


Figure 25. Notification view

Figure 25 represents the notification builder for a doctor. Here there will be an icon, notification title, and context for the doctor interface. When the doctor clicks on the notification doctor redirects towards the specific patient, the doctor will prescribe any medicine or suggestion.

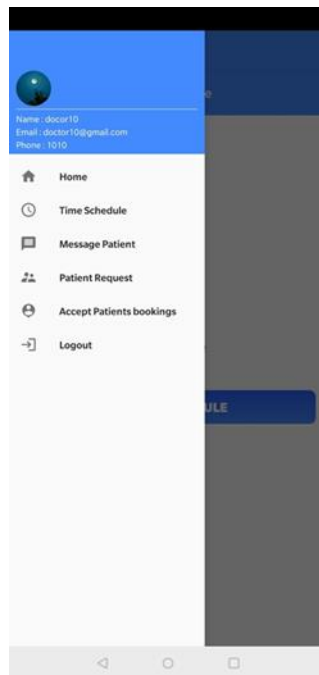


Figure 26. Demonstration of doctors all functionality

In the system, there will be more functionality available for doctors and patients. Figure 26 represents a doctor accepting the patient request, setting parameters for the patient, setting appointment hours, prescribing medicine, seeing the patient history, and communicating with the patient through the patient number or system messaging system.

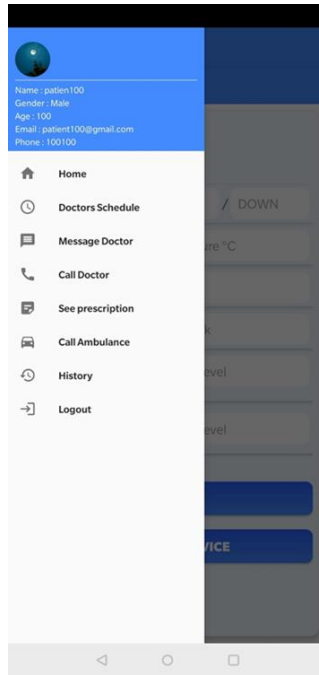


Figure 27. Demonstration of patients all functionality

On the other hand, Figure 27 represents the patient can send data from the Home option, get a response from the doctor through message, book an appointment from doctors' schedule, see the prescription, message the doctor, call, doctor, call an ambulance, and see the history. (Demonstrated clearly in Appendix 2-Description and Manual line 8)

#### 4.4 Design principles and methods of Smart Health Care

The plan of the Smart Health Care framework follows the accompanying standards and strategies: from the very beginning to acceptable standards, techniques for decay and coordination and demonstrating procedures, measures of steady pertinence, standards of normalization and receptiveness, and standards of complete precision.

- The layout configuration starts with the engineering of the framework to set up the general view components, including the itemized hierarchical design, the gathering of elements, and the interfaces between these significant gatherings, inferring the subtleties.
- The association bunches the details of the medical services' basic components alongside the order of the plan. Smart medical care designs need to be interoperable across various innovations. For instance, the sensors utilized in the body would convey among one another through an individual territory



organization or body-region organization. This data would be moved to a mobile phone. Consequently, the association helps talk about the functioning standards and methods engaged with the organization's designs.

- Deterioration coordination technique. Framework deterioration alludes to the disintegration of an intricate framework into a few subsystems. Framework coordination depends on the general design and practical necessities, so every subsystem can help one another and accomplish by and large advancement of the framework through-composed control of interior equilibrium.
- Utilizing the IoT in medical services has driven scientists worldwide to configuration good systems and advancements to give everyone calm, clinical help. Notwithstanding improving the client experience, the IoT additionally asks the business to get computerized, passing more examination across various cross-stages [18].
- Modeling method. Because of the decay of the framework, set up a consistent model of the framework and determine the actual relating item, track down a reasonable designer and work with the division of work and gathering.
- Reliable significance. The framework plan needs to receive a bound-together particular norm, and a brought together record design, and every module ought to have a bound together and good name. On a predictable premise, through correspondence between the interesting work and the rundown of plan modules, check whether each capacity has a relating module to guarantee the relationship between the interesting work.
- Normalization and transparency. This plan keeps the android structure principles and utilizations industry-wide innovation to accomplish this. The essential stage, network engineering, and security principles utilized by the framework additionally fulfill public and global guidelines.

Complete exactness. Another rule of the diagram configuration is to portray the fulfillment and accuracy of the business interaction got from the interest by seeing all the business cycles. We will be able to plan a framework that meets the client's business propensities and is invited.

## 4.5 Market Value of Smart Health Care System

Most mobile phones use any one of these operating systems. Approximately 85% of companies use the Android OS for their smartphone device. Android 10 is holding an 85.41% market share till 2020. [data sources: hostingtribunal.com]. In CAGR Report, Android income is expanded by 2.1% in 2019; additionally, they will anticipate that new income should 1.32 billion every 2023. Average selling costs are increased (5.8%) to 269 dollars from 254 in 2019 compared to 2018 [1].

On the other hand, the extent of the smart medical care system has extended its points of view and has been anticipated by Frost and Sullivan to be a US \$348.5 billion market by 2025 [17]. With many progressing examinations and degrees to address new issues, business visionaries and grounded enterprises rival vivid imagination. Electronic health records are the main results of smart health care, given another viewpoint for tending to huge information issues. These items fall across various verticals, for example, health information and capacity, checking and treatment, and stock administration.

Intel is driving the path with its fundamental foundation in the business [17]. The organization is continually thinking of creative innovations for information investigation, assistive innovation, and improving the home climate for the old populace. IBM's Watson, a falsely wise PC framework, can more proficiently study the substance of the patient's health care record and important health data to give better medical care models.

IBM has banded together with Apple, Johnson and Johnson, and Medtronic to proceed with their advanced health care research for an enormous scope. Google has a day-to-day existence sciences division devoted to creating, what is more, exploring innovations in computerized health care. Qualcomm helps catch the clinical gadget information and conveys it to the close-by data set accomplice through a remote clinical gadget and gets the data.

This stage offered by Qualcomm gives a high reach of framework interoperability and security. Microsoft's associated health stage helps in offering computerized health care administrations through work area structures. Specialists utilize Microsoft Lync to provide clinical benefits to patients in local zones. Samsung has a US\$50 million interest in advanced health care through its computerized health care activity [17], which is a cooperation of intelligent sensors, calculations, and information preparing methods

through open-source equipment and programming stages. Apple has an open-source system, Research Kit, which helps scientists create applications that can work with clinical exploration.

From a retail point of view, Amazon offers to abound together the medical services stage where the clients can get to data, accessibility of the most recent items, medical coverage, and on-request benefits. Wearables, particularly smart watches, or groups have been changing the market. The projected annualized rate is required to arrive at 70 million.

#### **4.6 Security Mechanism for Smart Health Care System**

Smart health care helps give better medical care worldwide, and it also turns out to be more powerless against dangers. Because of the powerful nature and more modest structure factors, the security prerequisites in savvy medical care frameworks fluctuate from the conventional security strategies [19].

With regards to community research exercises, cross-space verification can likewise guarantee the got information sharing. Both calculated models present scope of safety gives that should be deliberately managed in online information-sharing. Smart health care appreciates similar gatherings of advances while intensely depends on admittance to health data. Thus, security and protection safeguarding arrangements ought to be created by mulling overall highlights. This requires the elaborate substances are genuinely associated with smart medical care administrations. For example, Figure 28 diagrams a "Reasonable Picture" of this examination:

Firstly, recognizing brilliant health care applications by consolidating accessible functionalities of existing e-health frameworks; and finally, deciding security and protection prerequisites for smart health care applications.

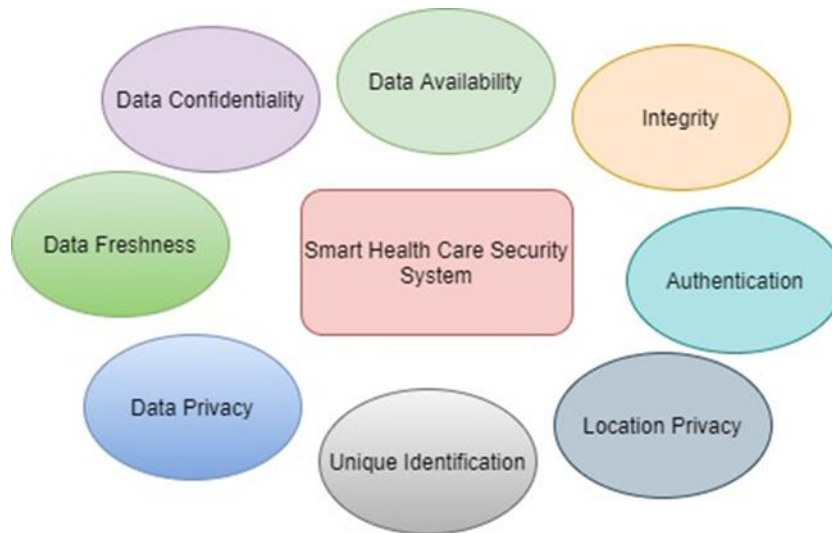


Figure 28. Security requirements for the smart health care system

Figure 28 shows the key security necessities or difficulties in keeping a got shrewd medical care framework since health care networks contain individual data that can be handily controlled. To decrease the expense of the plan, the processors utilized in brilliant medical care frameworks are low-speed and have low on-gadget memory, which cannot oblige extra security components [20]. Health services gadgets are portable, which drives the client to interface with various organizations like those in the home, office, and in broad daylight. This builds the opportunity of assaults on the gadget. Because of the development in the quantity of IoT gadgets in the health services organization, it is an extremely difficult assignment for engineers to give dynamic security refreshes or a good answer for multiprotocol data [21]. The passwords and keys should be refreshed regularly to keep up with information newness in the medical care organization. Assaults focusing on information communicated in the organization can incorporate interference of the help accessibility, change of unique details, fashioning messages, and replaying the messages to disturb the progression of information and make a false impression.

This system will use the RSA algorithm to make sure the privacy and security are intact. The RSA calculation includes four stages: key generation, key distribution, encryption, and decryption.

RSA incorporates a public key and a private key. Everyone can know the public key, and it is used for scrambling messages. The objective is that messages mixed with the public key should be decoded in a reasonable proportion of time by using the private key. The keys for the RSA calculation are created in an accompanying manner:

1. Pick two absolute indivisible numbers,  $p$  and  $q$ .
2. Figure  $n = pq$ .
3. Figure  $\lambda(n)$ , where  $\lambda$  is Carmichael's totient work.  $\lambda(n) = (p-1)(q-1)$
4. Pick a real number  $e$  to such an extent that  $1 < e < \lambda(n)$  and  $\gcd(e, \lambda(n)) = 1$ ; that is,  $e$  and  $\lambda(n)$  are coprime. ( $e$  is delivered as a feature of the public key.)
5. Decide  $d$  as  $d \equiv e^{-1}(\text{mod } \lambda(n))$ , that is,  $d$  is the multiplicative reverse of  $e$  modulo  $\lambda(n)$ . ( $d$  is kept secret as the private key example.) [22].
6. Encryption,  $C = M^e \text{ mod } n$
7. Decryption,  $M = C^d \text{ mod } n$

For example, we have  $p = 3$ ,  $q = 13$ ,  $e = 5$ ,  $M = 10$

1. Figure  $n = pq$ . So,  $n = 39$
2. Figure  $\lambda(n)$ , where  $\lambda$  is Carmichael's totient work.  $\lambda(n) = 24$
3. Pick a real number  $e$  to such an extent that  $1 < e < \lambda(n)$  and  $\gcd(e, \lambda(n)) = 1$ ; that is,  $e$  and  $\lambda(n)$  are coprime. ( $e$  is delivered as a feature of the public key.)
4. Decide  $d$  as  $d \equiv e^{-1}(\text{mod } \lambda(n))$ ; that is,  $d$  is 5
5. Encryption,  $C = M^e \text{ mod } n$ ;  $C = 10^5 \text{ mod } 39$ ;  $C = 4$
6. Decryption,  $M = C^d \text{ mod } n$ ;  $M = 4^5 \text{ mod } 39$ ;  $M = 10$  This is the actual message

Encoding a message or any file for specific people to ensure security and integrity are called encryption. Encryption is the process where an algorithm is used for encoding by generating keys. In encryption, the file or message after encryption is also known as ciphertext. On the other hand, decryption makes the unreadable data or file in a readable

format for the selected authentic person. Any eavesdropper cannot interfere in between sending and receiving processes.

In the RSA algorithm, encryption and decryption are vital to secure data, files, or messages as the sender sends and the receiver receives. In RSA encryption, messages, data, or files are encoded by a code-named public key. This public key is shared by the sender or the system that is encrypting the process. The same process is followed for decryption. At the same time, they are decrypting the same file, message, or data by the specific person or system by using another key known as a private key. This private is not shareable.

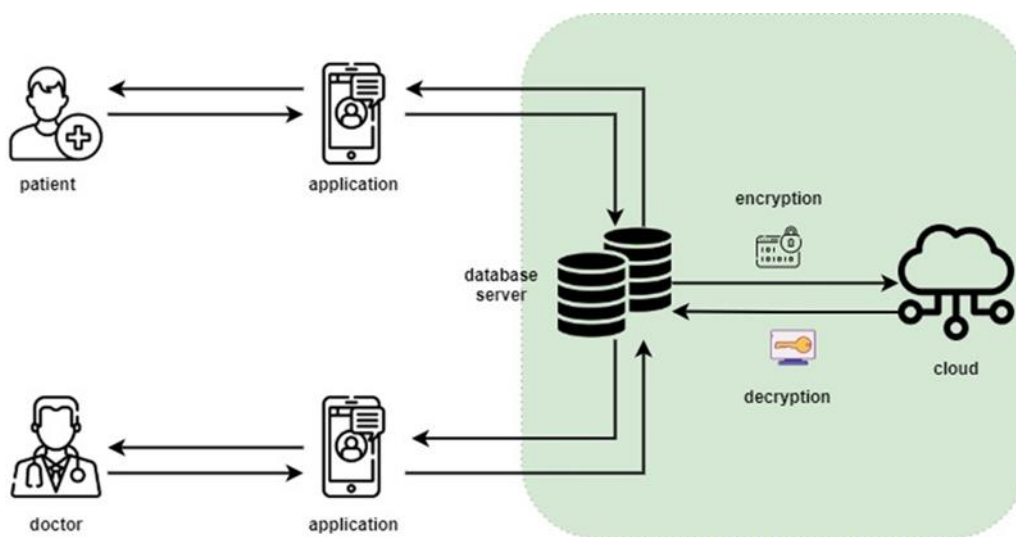


Figure 29. Smart health care data transmission process

Figure 29 illustrates the big picture of the whole smart system, how the system communicates, and the data transmission procedures. The main stakeholders (doctor, patient) can send or receive information or data through the smart health care application. All information is stored in the database server. Doctor or patient or any stakeholder who has permission to access or use the application can send or retrieve data from the database. The database will send all data to the cloud after encryption, and the database can also retrieve the uploaded data if needed by following the decryption steps.

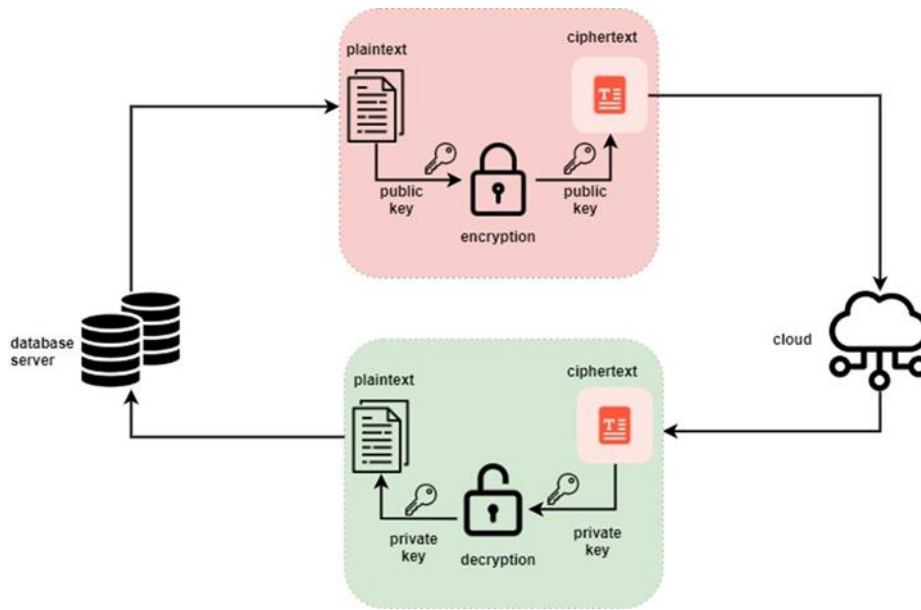


Figure 30. RSA algorithm mechanism in smart health care system

Figure 30 demonstrates the idea about the encryption and decryption procedures in detail. In the smart health care system, the RSA algorithm is used to ensure the security of the data so that any eavesdroppers cannot take unauthorized access to these data for their purposes. When data arrives in the database, the database starts to encrypt the data. One public key is used to make the plaintext to ciphertext. The cloud will get the encrypted data; if hackers want to hack the data, they will get only the data in an unreadable format that is useless.

The cloud sends the encrypted data to the database server; when the database server asks the cloud to retrieve data, the cloud sends encrypted data to the database. The database server knows the private key for decryption which describes on above. The ciphertext is converted to plaintext after the decryption procedures and gets the normal, readable data. Then any stakeholders (doctor, the patient can see the data from the database).

The benefits of encryption and decryption:

The clarification for the wide use of RSA is that in this, both open key and the private key is used, it extends data security. This way is ensured arrangement, believability, genuineness, and non-reputability of data [22]. The delicate key age will make RSA uncommonly feeble against attacks that are the thinking mind should be taken to guarantee that two colossal sporadic indivisible numbers. These are used to calculate the modulus, subsequently contained in a private key and an open key.

## 5 Summary

In this period of science and innovation, individuals are getting more straightforward and advantageous approaches to take care of their common issues. Medical care is also standing out enough to be noticed by the researcher and scientists, and they are building up a supportive framework to save lives and care forever. This paper represents how a brilliant medical services framework can be executed and conveyed by building up a few models. This paper shows a versatile application-based medical care apparatus as a model that can be a companion-like organization for the general population. Utilizing the framework, they can discover numerous accommodations that can change how individuals respond in crisis circumstances. Rather than being alarm, individuals may track down a speedy and viable approach to arrive at the arrangement with the assistance of this application. So, the system discussed in this thesis may change the total picture for any developing countries in the world like Bangladesh because the concept of the health care system in Bangladesh is not as good as developed countries.

Bangladesh has just 0.4 clinic beds per 1,000 population, as indicated by WHO (World Health Organization). It ought to be noticed that the government has an arrangement of building up 1 Community Clinic for each 6,000-population covering rustic Bangladesh. The examination assessed that recognizing half (or 7.1 M) of undiscovered hypertensive individuals matured 35+ years by 2022 (an NCD Plan target) would cost about BDT 1140 M (US\$ 13.8 M). Bangladesh is among the major ten nations with the most elevated number of individuals living with diabetes and hypertension. This investigation assessed that discovering half of the right now undiscovered 3.8 M diabetic grown-ups matured 35+ years would cost around BDT 780 M (US\$ 9.5 M). Treating the at present analyzed 2.33 M diabetic grown-ups developed 35+ years would cost around BDT 9860 M (US\$ 119 M) [23].

So, patients need to examine all his/her own and health information to avoid any unusual situation. In any crisis, the individual must be distinguished rapidly, and he/she needs prompt health data consideration by getting to past health data by utilizing this system.

Because of the analysis that has been done, we can infer that:



- An android-based smart application framework has been created to screen patients' medical issues to complete medical care for patients as right on time as could be expected and improve the health care system.
- The framework can be advantageous for specialists and patients since they can speak with one another and get treatment from home.
- All the data will store in the data set with the goal that anybody can see the historical backdrop of all health data and find out about all patients. Essentially, the public authority will have the possibility of each resident's health data.
- Health data is compassionate data. So, the security of the system has been implemented.
- The government can also get the citizens' health and the country's health care system.

Later, I desire to work broadly on this to create it to another level to improve individuals. I am additionally anticipating utilizing human-made reasoning in aiding individuals in recognizing infections dependent on side effects. Automatic synchronization will be discussed later for correspondence between the framework and smart gadgets in the not-so-distant future. Thus, it is trusted that a versatile-based medical care framework will be a valuable part of regular daily existence.

## References

- [1] Chao Wang, Wei Duan, Jianzhang Ma, and Chenhui Wang, "The research of Android System architecture and application programming," in *International Conference on Computer Science and Network Technology (ICCSNT)*, 2011.
- [2] R. A. Soni, "A Study on Android Application Development," *JTI*, Vols. vol. 1, no. 2, 2013.
- [3] Ananth S, Sathya P and Madhan Mohan P, "Smart Health Monitoring System through IOT," in *International Conference on Communication and Signal Processing*, India, 2019.
- [4] Veena Tripathi, Faizan Shakeel, "Monitoring Health Care System using Internet of Things -An Immaculate Pairing," in *International Conference on Next Generation Computing and Information Systems (ICNGCIS)*, 2017.
- [5] A. Imteaj and M. K. Hossain, "A smartphone-based application to improve the health care system of Bangladesh," in *International Conference on Medical Engineering, Health Informatics and Technology (MediTec)*, 2016.
- [6] A. Elsaadany, A. Sedky, and N. Elkholy, "A triggering mechanism for end-to-end IoT eHealth system with connected ambulance vehicles," in *8th International Conference on Information, Intelligence, Systems & Applications (IISA)*, 2017.
- [7] A. S. Vaidya, M. B. Srinivas, P. Himabindu, and D. Jumaxanova, "A smart phone/tablet based mobile health care system for developing countries," in *35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2013.
- [8] C. Suresh, C. Chandrakiran, K. Prashanth, K. V. Sagar, and K. Priyanka, "Mobile Medical Card' – An Android Application for Medical Data Maintenance," in *Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2020.
- [9] Hongxu Yin, Ayten Ozge Akmandor, Arsalan Mosenia, Niraj K., "Smart Healthcare," *Foundations and Trends in Electronic Design Automation* , vol. xx, p. xx, 2018.
- [10] T. JagannadhaSwamy and T. N. Murthy, "e-Smart: An IoT based Intelligent Health Monitoring and Management System for Mankind," in *International Conference on Computer Communication and Informatics (ICCCI -2019)*, 2019.
- [11] Ayan Banerjee\_ Member, IEEE, and Sandeep K. S. Gupta\_ Senior Member, IEEE, "Analysis of Smart Mobile Applications for Healthcare under Dynamic Context Changes," *IEEE TRANSACTIONS ON MOBILE COMPUTING*, vol. X, p. Y, 2019.
- [12] Baozhan Chen, Siyuan Qiao, Jie Zhao, Dongqing Liu, Xiaobing Shi, Minzhao Lyu, Haotian Chen, Huimin Lu, and Yunkai Zhai, "A Security Awareness and Protection System for 5GSmart Healthcare Based on Zero-Trust Architecture," *IEEE INTERNET OF THINGS JOURNAL*, 2020.

- [13] M. Weenk, H. van Goor, M. van Acht, L. J. Engelen, T. H. van de Belt, and S. J. H. Bredie, "A smart all-in-one device to measure vital signs in admitted patients," *PLoS ONE*, vol. 13, pp. no. 2, p. e0190138, 2018.
- [14] P. A. Ogink et al., "Feasibility of a New Cuffless Device for Ambulatory Blood Pressure Measurement in Patients with Hypertension: Mixed Methods Study," *J Med Internet Res*, vol. 21, pp. no. 6, p. e11164, 2019.
- [15] R. Ibrahim and S. Y. Yen "Formalization of the Data Flow Diagram Rules for Consistency Check," *IJSEA*, Vols. vol. 1, no. 4, p. pp. 95–111, 2010.
- [16] "Software System Design," [Online]. Available: [https://www.cs.uct.ac.za/mit\\_notes/software/pdfs/Chp06.pdf](https://www.cs.uct.ac.za/mit_notes/software/pdfs/Chp06.pdf).
- [17] P. Sundaravadivel, E. Kougianos, S. P. Mohanty, and M. K. Ganapathiraju, "Everything You Wanted to Know about Smart Health Care: Evaluating the Different Technologies and Components of the Internet of Things for Better Health," *IEEE Consumer Electron. Mag*, 2018.
- [18] S. Neelam, "Internet of Things in Healthcare," Faculty of Computing Blekinge Institute of Technology SE-371 79 Karlskrona, Sweden, 2019.
- [19] A. Islam, "Health System in Bangladesh: Challenges and Opportunities," *AJHR*, Vols. vol. 2, no. 6, p. p. 366, 2014.
- [20] Y. Lu. a. R. O. Sinnott, "Security and privacy solutions for smart healthcare systems," *Innovation in Health Informatics, Elsevie*, p. pp. 189–216, 2018.
- [21] Baozhan Chen, Siyuan Qiao, Jie Zhao, Dongqing Liu, Xiaobing Shi, Minzhao Lyu, Haotian Chen, Huimin Lu, and Yunkai Zhai, "A Security Awareness and Protection System for 5G Smart Healthcare Based on Zero-Trust Architecture," *IEEE Internet Things J*, pp. 1-10, 2020.
- [22] S. Agarwal, "RSA Public Key Cryptography," *International Journal of Electrical Electronics & Computer Science Engineering*, no. Special Issue - NCSCT-2018 | E-ISSN: 2348-2273 | P-ISSN: 2454-1222, 2018.
- [23] "Hypertension and Type-2 Diabetes in Bangladesh by World Health Organization and World Bank," [Online]. Available: <http://documents1.worldbank.org/curated/en/914751528271293762/pdf/Hypertension-and-Type-2-Diabetes-in-Bangladesh-Continuum-of-Care-Assessment-and-Opportunities-for-Action.pdf>.
- [24] A. N. A. Yusuf, F. Y. Zulkifli, and I. W. Mustika, "Development of Monitoring and Health Service Information System to Support Smart Health on Android Platform," in *4th International Conference on Nano Electronics Research and Education (ICNERE)*, 2018.
- [25] M. A. Kumar and Y. R. Sekhar, "Android based health care monitoring system," in *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2015.
- [26] M. B. M. K. S. L. a. A. S. K. Navin, "A mobile health based smart hybrid epidemic surveillance system to support epidemic control programme in public health informatics," in *International Conference on IoT and Application (ICIOT)*, 2017.

- [27] Y. Gu, J. Shen, and Y. Chen, "Poster Abstract: Know You Better: A Smart Watch Based Health Monitoring System," in *IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, 2019.
- [28] A. A. a. M. E. Johnson, "Information security and privacy in healthcare: current state of research," *IJIEM*, vol. 6, p. 279, 2010.
- [29] D. D. K. a. D. Jakkan, "A Survey on Smart Health Care System Implemented Using Internet of Things," 2019.
- [30] H. Vahdat-Nejad, Z. Abbasi-Moud, S. A. Eslami, and W. Mansoor, "Survey on Context-Aware Healthcare Systems," in *IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, 2021.
- [31] Tian, W. Yang, J. M. L. Grange, P. Wang, W. Huang, and Z. Ye, "Smart healthcare: making medical care more intelligent," *Global Health Journal*, vol. 3, p. pp. 62–65, 2019.
- [32] A. Sarkar, A. Goyal, D. Hicks, D. Sarkar, and S. Hazra, "Android Application Development: A Brief Overview of Android Platforms and Evolution of Security Systems," in *Third International conference on I-SMAC (IoT in Social, Mobile, Analytics, and Cloud) (I-SMAC)*, 2019.
- [33] "Android Operating System," [Online]. Available: <https://www.elprocus.com/what-is-android-introduction-features-applications/>.

## **Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis<sup>1</sup>**

I Md Taibur Rahman

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Smart Device Application Based on Health Care System”, supervised by Vladimir Viies, PhD, Associate Professor.
  - 1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
  - 1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

09.05.2021

Md Taibur Rahman

---

<sup>1</sup> The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

## Appendix 2-Description and Manual

This section describes the basic functions of how our application interacted. It contains a step-by-step explanation of how the Smart Health Care Application works.

Each model has its separate folder and has some steps.

1. The folders and files for the experiments can be downloaded through this link:  
[https://github.com/trshawonbd/Smart-](https://github.com/trshawonbd/Smart-Health/tree/master/app/src/main/java/com/ppal007)

[Health/tree/master/app/src/main/java/com/ppal007](https://github.com/trshawonbd/Smart-Health/tree/master/app/src/main/java/com/ppal007)

2. The application can be performed on both Android systems.

3. The folder name smarthealth contains seven folders. The activity folder contains message where the messaging system is implemented in ui folder. Then *HomeActivity.java*, *LoginActivity.java*, *RegisterActivity.java* contain respectively home, login, registration functionality.

4. The folder name fragment contains two folders (doctor and patient). Both folders contain several java files for doctor and patient interfaces functionality. The notifi folder contains *MyService.java* file where the notification system is implemented.

5. The folder name model contains several files used to send data to the database in a synchronous way.

6. The folder name retrofit contains several files used to establish API connection and network system handler.

7. The folder name utils contains *common.java* file used to establish the connection between the server and the application.

8. The APK and the manual can be downloaded from this link:

<https://drive.google.com/drive/folders/156Gx23e6PhG8ygBvMx5VLCz6F6t-r4Wc?usp=sharing>

## Appendix 3 – Source Code

### Libraries:

```
//
glide.....
.....
    implementation 'com.github.bumptech.glide:glide:4.12.0'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0'
    //
retrofit.....
.....
    implementation 'com.squareup.retrofit2:retrofit:2.4.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.4.0'
    implementation 'com.google.code.gson:gson:2.8.5'
//    justify text view
    implementation 'com.codesgod:justifiedtextview:1.1.0'

    //firebase
    implementation platform('com.google.firebase:firebase-bom:26.5.0')
    implementation 'com.google.firebase:firebase-database:19.7.0'
    implementation 'com.google.firebase:firebase-analytics'
    implementation 'com.google.firebase:firebase-auth'
    implementation 'com.google.firebase:firebase-core:18.0.3'
}
```

### Permission:

```
<!-- permissions -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"
    tools:ignore="ScopedStorage" />
<uses-permission
    android:name="android.permission.MEDIA_CONTENT_CONTROL"
    tools:ignore="ProtectedPermissions" />
```

### Base URL:

```
public class Common {
//    base url
    public static final String BASE_URL =
"https://smhealth.mdshohelrana.me/";
```

### Doctor Login Interface:

```
ApiInterface apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);
Call<ModelLogin> call = apiInterface.user_Login("d",_email,_pass);
call.enqueue(new Callback<ModelLogin>() { @Override
    public void onResponse(Call<ModelLogin> call, Response<ModelLogin>
response) { if (response.isSuccessful() && response.body()!=null){
        ModelLogin model = response.body();
        if (model.isSuccess()){ //progress bar dismiss
            progressBarAdapter.dismissDialog();
            //sharedPref
            sharedPref.login_status(true);
            sharedPref.setUserId(model.getId());
            sharedPref.setUserName(model.getName());
            sharedPref.setUserGender("doctor_gender");
            sharedPref.setUserAge("doctor_age");
            sharedPref.setUserEmail(_email);
            sharedPref.setUser_pass(_pass);
            sharedPref.setUserPhone(model.getPhone());
            sharedPref.setProfile_img(model.getProfileImgPath());
            sharedPref.setLoginMood(Common.AS_DOCTOR);
            Intent intent = new Intent(getApplicationContext(),
HomeActivity.class);

            intent.putExtra("ex_login_mood_from_login",Common.AS_DOCTOR);
            startActivity(intent); finish(); }
        }
```

### Patient Login Interface:

```
ApiInterface
apiInterface=ApiClient.getApiClient().create(ApiInterface.class);
Call<ModelLogin> call = apiInterface.user_Login("p",_email,_pass);
call.enqueue(new Callback<ModelLogin>() { @Override
    public void onResponse(Call<ModelLogin> call, Response<ModelLogin>
response) { if (response.isSuccessful() && response.body()!=null){
        ModelLogin model = response.body();
        if (model.isSuccess()){
            progressBarAdapter.dismissDialog();
            sharedPref.login_status(true);
            sharedPref.setUserId(model.getId());
            sharedPref.setUserName(model.getName());
            sharedPref.setUserGender(model.getGender());
            sharedPref.setUserAge(model.getAge());
            sharedPref.setUserEmail(_email);
            sharedPref.setUser_pass(_pass);
            sharedPref.setUserPhone(model.getPhone());
            sharedPref.setProfile_img(model.getProfileImgPath());
            sharedPref.setLoginMood(Common.AS_PATIENT);
            Intent intent = new Intent(getApplicationContext(),
HomeActivity.class);

            intent.putExtra("ex_login_mood_from_login",Common.AS_PATIENT);
            startActivity(intent);finish(); }
        }
```



## Doctor Interface

### Retrieve Doctor Location:

```
private void retrieveLocation() { // get doctor location
    progressBarAdapter.startLoadingDialog();
    ApiInterface apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);
    Call<List<ModelRegisterAsDoctor>> call =
apiInterface.retrieve_doctor_location();
    call.enqueue(new Callback<List<ModelRegisterAsDoctor>>())
@Override
    public void onResponse(Call<List<ModelRegisterAsDoctor>> call,
Response<List<ModelRegisterAsDoctor>> response) {
        if (response.isSuccessful() && response.body()!=null){
            List<ModelRegisterAsDoctor> model = response.body();
            AdapterSpinnerLocation adapterSpinnerLocation= new
AdapterSpinnerLocation(RegisterActivity.this,model);
            spinnerLocation.setAdapter(adapterSpinnerLocation);
            progressBarAdapter.dismissDialog();
        }
    }
}
```

### Retrieve Patient Booking at Doctor End:

```
private void retrieveBooking() {
    progressBarAdapter.startLoadingDialog();
    ApiInterface apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);
    Call<List<ModelBookingSchedule>> call =
apiInterface.retrieve_patient_accept_booking(sharedPref.getUserId());
    call.enqueue(new Callback<List<ModelBookingSchedule>>())
@Override
    public void onResponse(Call<List<ModelBookingSchedule>>
call, Response<List<ModelBookingSchedule>> response) {
        if (response.isSuccessful() && response.body()!=null){
            List<ModelBookingSchedule> model =
response.body();
            Gson gson = new Gson();
            String json_str = gson.toJson(model);
            JSONArray jsonArray = null;
            try { jsonArray = new JSONArray(json_str);
            } catch (JSONException e) {e.printStackTrace();}
            List<String> _schedule_booking_time = new
ArrayList<>();
            for(int i = 0; i <
Objects.requireNonNull(jsonArray).length(); i++){try {
            _schedule_booking_time.add(jsonArray.getJSONObject(i).getString("sched
ule_boking_time")); } catch (JSONException e) {e.printStackTrace();}}
            // retrieve patient details
            retrievePatentDetails(_schedule_booking_time);
        }else {progressBarAdapter.dismissDialog();
            showToast("something wrong to retrieve booking schedule");}}
}
```

### Patient List at Doctor End:

```
private void retrieveMyPatient() {ApiInterface apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);
    Call<List<ModelPatientRequestList>> call =
apiInterface.retrieve_my_patient(sharedPref.getUserId());
    call.enqueue(new Callback<List<ModelPatientRequestList>>() {
@Override public void onResponse(Call<List<ModelPatientRequestList>>
call, Response<List<ModelPatientRequestList>> response) {
        if (response.isSuccessful() && response.body()!=null){
            List<ModelPatientRequestList> model = response.body();
            Gson gson = new Gson();
            String jsonStr = gson.toJson(model);
            JSONArray jsonArray = null;
            try {jsonArray = new JSONArray(jsonStr);} catch
(JSOException e) { e.printStackTrace();}
            List<String> _id = new ArrayList<>();
            List<String> _name = new ArrayList<>();
            List<String> _email = new ArrayList<>();
            List<String> _phone = new ArrayList<>();
            List<String> _age = new ArrayList<>();
            List<String> _gender = new ArrayList<>();
            for(int i = 0; i <
Objects.requireNonNull(jsonArray).length(); i++){try {
_id.add(jsonArray.getJSONObject(i).getString("id"));
_name.add(jsonArray.getJSONObject(i).getString("name"));
_email.add(jsonArray.getJSONObject(i).getString("email"));
_phone.add(jsonArray.getJSONObject(i).getString("phone"));
_age.add(jsonArray.getJSONObject(i).getString("age"));
_gender.add(jsonArray.getJSONObject(i).getString("gender"));}
}
```

### To Accept Patient Request at Doctor End

```
buttonAccept.setOnClickListener(v -> {view.setVisibility(View.GONE);
    progressBar.setVisibility(View.VISIBLE);
//
    update request
ApiInterface apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);
    Call<ModelAcceptPatient> call =
apiInterface.accept_patient(sharedPref.getUserId(),id);
    call.enqueue(new Callback<ModelAcceptPatient>() {
@Override public void onResponse(Call<ModelAcceptPatient> call,
Response<ModelAcceptPatient> response) { if (response.isSuccessful()
&& response.body()!=null){ ModelAcceptPatient model = response.body();
        if (model.isSuccess()){
            progressBar.setVisibility(View.GONE);
            view.setVisibility(View.VISIBLE);
dialog.dismiss(); //
            recreate patient request fragment
Objects.requireNonNull(getFragmentManager())
                .beginTransaction()
                .replace(R.id.hostFragmentId,new
PatientRequestFragment())
                .addToBackStack(null).commit()
}
```

## To See the Patient Symptom List at Doctor End

```
private void retrieveSymptomList() {ApiInterface apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);
    Call<List<ModelPatientSymptoms>> call
=apiInterface.retrieve_symptom_list(_id);
    call.enqueue(new Callback<List<ModelPatientSymptoms>>() {
@Override public void onResponse(Call<List<ModelPatientSymptoms>>
call, Response<List<ModelPatientSymptoms>> response) {
        if (response.isSuccessful() && response.body()!=null){
            List<ModelPatientSymptoms> model = response.body();
            Gson gson = new Gson();
            String jsonStr = gson.toJson(model);
            JSONArray jsonArray = null;
            try { jsonArray = new JSONArray(jsonStr);
} catch (JSONException e) { e.printStackTrace();}
            List<String> _id = new ArrayList<>();
            List<String> _bpUp = new ArrayList<>();
            List<String> _bpDown = new ArrayList<>();
            List<String> _spo2 = new ArrayList<>();
            List<String> _ecg = new ArrayList<>();
            List<String> _glBfr = new ArrayList<>();
            List<String> _glAftr = new ArrayList<>();
            List<String> _temperature = new ArrayList<>();
            List<String> _date = new ArrayList<>();
            List<String> _time = new ArrayList<>();
            for(int i = 0; i <
Objects.requireNonNull(jsonArray).length(); i++){try
{ _id.add(jsonArray.getJSONObject(i).getString("id"));
_bpUp.add(jsonArray.getJSONObject(i).getString("bp_up"));
_bpDown.add(jsonArray.getJSONObject(i).getString("bp_down"));
_spo2.add(jsonArray.getJSONObject(i).getString("spo2"));
_ecg.add(jsonArray.getJSONObject(i).getString("ecg"));
_glBfr.add(jsonArray.getJSONObject(i).getString("gl_befr_eat"));
_glAftr.add(jsonArray.getJSONObject(i).getString("gl_aftr_eat"));
_temperature.add(jsonArray.getJSONObject(i).getString("temperature"));
_date.add(jsonArray.getJSONObject(i).getString("submit_date"));
_time.add(jsonArray.getJSONObject(i).getString("submit_time"));} catch
(JSONException e) { e.printStackTrace();}}
```

## Set Schedule Time for Patient at Doctor End

```
// button create schedule click listener
    buttonCreateSchedule.setOnClickListener(new
View.OnClickListener() { @Override public void onClick(View v) {
        Dialog dialog = new Dialog(getContext());
        dialog.requestWindowFeature(Window.FEATURE_NO_TITLE);
        dialog setContentView(R.layout.custom_dialog_doctor_schedule);
        dialog.setCanceledOnTouchOutside(false);
//
        select date click listener
        textViewSelectDate.setOnClickListener(new
View.OnClickListener() {@Override public void onClick(View v) {
```

```

        Calendar calendar = Calendar.getInstance();
        int year = calendar.get(Calendar.YEAR);
        int month = calendar.get(Calendar.MONTH);
        int day = calendar.get(Calendar.DAY_OF_MONTH);
DatePickerDialog dialog = new DatePickerDialog(getContext(),
android.R.style.Theme_Holo_Light_Dialog_MinWidth,mDateSetListener,
year, month, day);

//          select start time
        textViewStartTime.setOnClickListener(new
View.OnClickListener() { @Override
        public void onClick(View v) {
            final Calendar calenderInstance = Calendar.getInstance();
            int hr = calenderInstance.get(Calendar.HOUR_OF_DAY);
            int min = calenderInstance.get(Calendar.MINUTE);

//          set end time click listener
        textViewEndTime.setOnClickListener(new
View.OnClickListener() { @Override public void onClick(View v) {
            final Calendar calenderInstance =Calendar.getInstance();
            int hr = calenderInstance.get(Calendar.HOUR_OF_DAY);
            int min = calenderInstance.get(Calendar.MINUTE);

//    for retrieve schedule
        private void retrieveSchedule() {
            progressBarAdapter.startLoadingDialog();
            ApiInterface apiInterface =
ApiClient.getClient().create(ApiInterface.class);
            Call<ModelDoctorSchedule> call =
apiInterface.retrieve_doctor_schedule(sharedPref.getUserId());
            call.enqueue(new Callback<ModelDoctorSchedule>() {
                @Override
                public void onResponse(Call<ModelDoctorSchedule> call,
Response<ModelDoctorSchedule> response) {
                    if (response.isSuccessful() && response.body()!=null){
                        ModelDoctorSchedule model = response.body();
                        if (model.isSuccess()){
                            progressBarAdapter.dismissDialog();
                            String _date_long_str =
model.getScheduleDate();
                            String _start_time_long_str =
model.getScheduleTimeStart();
                            String _end_time_long_str =
model.getScheduleTimeEnd();

                            schedule_bool = true;

//          convert long to date and time
convert_long_to_date_and_time(_date_long_str,_start_time_long_str,_end
_time_long_str);
            }

```

## Patient Interface

### Patient Send Health Data to Patient

```
ApiInterface apiInterface =
ApiClient.getClient().create(ApiInterface.class);
    Call<ModelPatientSymptoms> call =
apiInterface.send_patient_symptoms_to_doctor(sharedPref.getUserId(),
_bpUp, _bpDown, _spo2, _ecg, glBfr, glAftr, _temperature, _crnt_time);
    call.enqueue(new Callback<ModelPatientSymptoms>()
@Override public void onResponse(Call<ModelPatientSymptoms> call,
Response<ModelPatientSymptoms> response) {
        if (response.isSuccessful() && response.body()!=null){
            ModelPatientSymptoms model = response.body();
            if (model.isSuccess()){
//                hide progress bar
                progressBarAdapter.dismissDialog();
                editTextBpUp.setText("");
                editTextBpDown.setText("");
                editTextTemperature.setText("");
                editTextSpo2.setText("");
                editTextEcg.setText("");
                editTextGlBfrEat.setText("");
                editTextGlAftrEat.setText("");
            }else { progressBarAdapter.dismissDialog();
                showToast(model.getMessage());}
            }else {progressBarAdapter.dismissDialog();
                showToast("something wrong to insert symptoms!");}
        }
    }
}
```

### To select a doctor from the dropdown at Patient End

```
private void initSpinnerLocation() {
    spinnerLocation.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {
            ModelRegisterAsDoctor modelRegisterAsDoctor =
(ModelRegisterAsDoctor) parent.getItemAtPosition(position);

            retrieveDoctorName(modelRegisterAsDoctor.getLocation());

        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {

        }
    });
}
```

## To Show Prescription at patient End

```
ApiInterface apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);
Call<List<ModelPatientSymptomPrescription>> call =
apiInterface.retrieve_symptom_prescription(sharedPref.getUserId());
call.enqueue(new Callback<List<ModelPatientSymptomPrescription>>()
{@Override
    public void onResponse(Call<List<ModelPatientSymptomPrescription>>
call, Response<List<ModelPatientSymptomPrescription>> response) {
        if (response.isSuccessful() && response.body()!=null){
            List<ModelPatientSymptomPrescription> model =
response.body(); Gson gson = new Gson();
            String jsonStr = gson.toJson(model);
            JSONArray jsonArray = null;
            try {jsonArray = new JSONArray(jsonStr);
            } catch (JSONException e) {
                e.printStackTrace();}
        }
    }
});
```

## Download Prescription at patient End

```
//save button click listener
buttonSave.setOnClickListener(v -> {
    //permissions

    ActivityCompat.requestPermissions(Objects.requireNonNull(getActivity()
), new String[]{
        Manifest.permission.READ_EXTERNAL_STORAGE,
        Manifest.permission.WRITE_EXTERNAL_STORAGE
    }, PackageManager.PERMISSION_GRANTED);

    // create pdf
    PdfDocument myPdfDoc = new PdfDocument();
    PdfDocument.PageInfo myPageInfo = new
PdfDocument.PageInfo.Builder(300,600,1).create();
    PdfDocument.Page myPage = myPdfDoc.startPage(myPageInfo);

    Paint myPaint = new Paint();
    //final string
    String myString = "Name : "+name+"\n"+"Email : "+email+"\n"+"Phone
No : "+phone+"\n\n"+pres;
    int x=10, y=25;
    for (String lines : myString.split("\n")){
        myPage.getCanvas().drawText(lines, x, y, myPaint);
        y+=myPaint.descent()-myPaint.ascent();}
    myPdfDoc.finishPage(myPage);

    String myFilePath =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DO
WNLOADS). toString()+"/my_prescription.pdf";
    File myFile = new File(myFilePath);
```

## To Book an Appointment with Doctor at Patient End

```
private void retrieveSchedule() {
    progressBarAdapter.startLoadingDialog();
    ApiInterface apiInterface =
ApiClient.getApiClient().create(ApiInterface.class);
    Call<ModelDoctorSchedule> call =
apiInterface.retrieve_doctor_schedule_from_patient(sharedPref.getUserI
d());call.enqueue(new Callback<ModelDoctorSchedule>() {@Override
    public void onResponse(Call<ModelDoctorSchedule> call,
Response<ModelDoctorSchedule> response) {
        if (response.isSuccessful() && response.body()!=null){
            ModelDoctorSchedule model = response.body();
            if (model.isSuccess()){
                //progressBarAdapter.dismissDialog();
                String _schedule_date = model.getScheduleDate();
                String _start_time = model.getScheduleTimeStart();
                String _end_time = model.getScheduleTimeEnd();
                String _doctor_name = model.getName();
                String _doctorId = model.getDoctorId();
            }
        }
    }
}
```

## Messaging System for Patient and Doctor End

### Sending Message

```
private void sendMessage(String sender, String receiver, String
message) {DatabaseReference reference =
FirebaseDatabase.getInstance().getReference();
    HashMap<String, Object> hashMap = new HashMap<>();
    hashMap.put("sender",sender); hashMap.put("receiver",receiver);
    hashMap.put("message",message);
    reference.child(Common.FIREBASE_CHAT_DB).push().setValue(hashMap);}
```

### Read Message

```
private void readMessage(String uid, String userId) {
    chatList = new ArrayList<>();reference =
FirebaseDatabase.getInstance().getReference(Common.FIREBASE_CHAT_DB);
    reference.addValueEventListener(new ValueEventListener()
{@Override public void onDataChange(@NonNull DataSnapshot
dataSnapshot) { chatList.clear();
        for (DataSnapshot snapshot :
dataSnapshot.getChildren()){Chat chat = snapshot.getValue(Chat.class);
            if (chat.getReceiver().equals(uid) &&
chat.getSender().equals(userId) || chat.getReceiver().equals(userId)
&& chat.getSender().equals(uid)){ chatList.add(chat); }
            messageAdapter = new
MessageAdapter(MessageActivity.this,chatList);
            recyclerView.setAdapter(messageAdapter);
            recyclerView.scrollToPosition(recyclerView.getAdapter().getItemCount()
-1);
            messageAdapter.notifyDataSetChanged();}}
```