Bangabandhu Sheikh Mujibur Rahman Digital University

Faculty of Cyber Physical Systems
Department of Internet of Things and Robotics Engineering (IRE)

mediHelp: IoT based Smart Patient Monitoring System

Course Title: Mobile Platform for IoT Devices Lab
Course Code: IOT 4412

<u>Students Details</u>

Name	ID
Sadikur Rahman Sadik	1801010
Partho Sharothi Chowhan	1801014
Md.Abdul Halim Khan	1801020
Atiqur Rahman	1801025
Md.Shohanur Rahman Shohan	1801033

Submission Date: 6-Nov-23

Course Teacher's Name:
Mahir Mahbub
Lecturer, Dept. of IRE
Md Moshiur Rahman
Lecturer, Dept. of IRE

<u>Lab Project Status</u>	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

The integration of the IoT-Based Smart Patient Monitoring System with an Android application exemplifies an innovative solution that enables users to actively monitor and sustain their health status. The efficacy of this all-encompassing system is predicated on wearable health sensors that consistently track vital signs, with an emphasis on the heart rate and body temperature. The gathered data is transmitted in a secure manner to a cloud server via an IoT gateway.The Android application functions as an intuitive user interface, providing capabilities such as real-time monitoring, data visualisation etc. Furthermore, it incorporates functionalities for the purpose of managing a roster of doctors and patients, thereby streamlining healthcare coordination and communication. By emphasising data security, convenience this novel system has the capacity to revolutionise healthcare by promoting health consciousness and healthier way of life while reducing the need for frequent hospitalisations. It offers a captivating view into the prospective landscape of efficient and readily available health monitoring.

1.2 Motivation

Beyond mere technology, our endeavour, the IoT-Based Smart Patient Monitoring System with the Android application, serves as a pathway to better healthcare and better lives. The potential for this endeavour to significantly affect the welfare of individuals is what drives us to commence this endeavour.

One can only envision the positive impact that can be achieved when individuals are able to receive timely notifications, monitor their health in real-time, and acquire significant insights into their overall well-being. They are capable of being empowered, being encouraged to take command of their health, and identifying health problems in their nascent stages through our initiative. [1]

Moreover, our motivation stems from the understanding that our endeavour facilitates healthcare convenience by diminishing the necessity for frequent hospital visits and optimising doctor-patient communication. This not only improves the accessibility of healthcare but also elevates its quality for individuals from all demographic backgrounds.

In addition, our endeavour is consistent with a future vision in which technology benefits society as a whole by encouraging healthier lifestyles and greater health consciousness. The impetus stems from the conviction that our efforts are not merely instrumental in the development of a system, but rather in the establishment of a foundation that promotes global health, information, and connectivity.

1.3 Problem Definition

1.3.1 Problem Statement

The IoT-Based Smart Patient Monitoring System initiative aims to tackle a series of significant challenges that are presently confronting the healthcare industry. The absence of real-time health surveillance causes significant delays in the detection of health problems for numerous individuals, which may ultimately result in complications. In addition, [2]. conventional healthcare systems are deficient in the capacity to provide individualised health insights, thereby depriving users of knowledge regarding their distinct health trends. The provision of expeditious medical advice and decision-making is impeded by ineffective communication between physicians and patients, whereas exhausted healthcare facilities contend with the demands of continuous monitoring. Moreover, in numerous regions, the dearth of accessibility to high-quality healthcare continues to be an enduring obstacle. Concerns regarding data privacy and security have also increased in tandem with the proliferation of connected devices. The objective of the project is to deliver a holistic resolution by incorporating timely surveillance, individualised analyses, streamlined doctor-patient correspondence, enhanced healthcare availability, and resilient data protection. Through tackling these concerns, the initiative seeks to transform the healthcare sector into one that is more effective, proactive, and inclusive, ultimately augmenting the welfare of people from various demographic backgrounds.

1.3.2 Complex Engineering Problem

In the context of the IoT based smart Patient Monitoring System , we can analyze the attributes mentioned in Table 1.1 to better understand the complex engineering challenges involved:

1.4 Design Goals/Objectives

The primary aims of the IoT-Based smart patient Monitoring System initiative are to develop a resilient system for monitoring health-related data that eliminates the need for intricate decision-making processes. The principal aim of our development is to create an Android application that effectively integrates with wearable health sensors in order to offer continuous surveillance of critical health parameters, specifically heart

Table 1.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributess	Explain how to address
P1: Depth of knowledge required	Utilize IoT sensors for data collection, Android app development using dart,flutter, firebase for ease of use.
P2: Range of conflicting requirements	Balance user experience and data accuracy, address compatibility issues between IoT devices and Android app. Sensors send data to the firebase .Firebase acts as cloud storage .
P3: Depth of analysis required	Implement algorithms for data interpretation and trend analysis.
P4: Familiarity of issues	Ensure the app and IoT devices are user-friendly and align with established health monitoring standards.
P5: Extent of applicable codes	Adhere to relevant coding standards and protocols for IoT communication and data transmission.
P6: Extent of stakeholder involvement and conflicting requirements	Collaborate with healthcare professionals and end-users to gather requirements and feedback.
P7: Interdependence	Ensure seamless integration of IoT devices with the Android app for real-time data monitoring.

rate and body temperature. Following this, our objective is to develop an Android application featuring a user-friendly interface that facilitates effortless access to and interpretation of health data. This will enable users to remain informed about their vital signs without necessitating extensive analysis. In addition, we will ensure that healthcare professionals have access to vital health data in a timely manner, thereby facilitating remote patient monitoring and enabling secure and efficient communication between patients and providers of healthcare.Furthermore, our system will guarantee scalability and compatibility, enabling widespread adoption among a wide range of users, all the while preserving the accuracy of health data representation. Our initiative places utmost importance on safeguarding user health information in terms of privacy and security. To achieve this, we have implemented strong measures to protect the data and ensure adherence to pertinent regulations. In conclusion, the project will facilitate remote patient monitoring and reduce the frequency of hospital visits, thereby enhancing healthcare accessibility, especially in regions where healthcare infrastructure is scarce. By means of these aims, our endeavour endeavours to deliver an easily navigable, secure, and streamlined health monitoring system that empowers individuals to remain apprised of their overall health. [3]

1.5 Application

This Internet of Things-enabled intelligent smart patient monitoring system has numerous real-world applications, including but not limited to:

- Remote patient monitoring :enables healthcare professionals to oversee the health of individuals with chronic maladies from the convenience of their residences, thereby facilitating the provision of timely interventions.

- Elderly Care: Carers can use the system to monitor and assure the well-being of elderly family members.

- Fitness and Wellness: Individuals who prioritise their health have the ability to monitor their vital signs and obtain tailored suggestions to help them sustain a healthy way of life.

- Healthcare Institutions: The system can be utilised by hospitals and clinics to provide more efficient and cost-effective care by remotely monitoring patients.

- Analysis of Collected Health Data: The gathered health data can be utilised for research purposes, thereby contributing to the formulation of novel treatments and healthcare policies.

- Emergency Response: In the event of critical health events, the system can be seamlessly incorporated with emergency services to deliver prompt assistance.

Chapter 2

Design, Development and Implementation of the Project

2.1 Introduction

The transformative project known as the IoT-Based Smart Patient Monitoring System aims to provide individuals with the means to effectively manage their health. The objective of this project is to offer secure communication with healthcare providers, real-time health monitoring, and personal health insights through the integration of wearable health sensors with an Android application. This section will provide an in-depth analysis of the exhaustive project details, which will be divided into several subsections to facilitate comprehension. [4]

2.2 Project Details

The IoT-Based Smart Patient Monitoring System is an innovative undertaking that seeks to empower users to assume responsibility for their health by utilising data-driven insights and real-time monitoring. Through the integration of wearable health sensors and a specialised Android application, this project effectively tackles multiple critical elements, each of which is indispensable in attaining its overarching objectives.

The proposed system for the project is depicted in the following figure. A user registers with the system initially and then logs in. After that, the user locates a diagnostic button and presses it to see three possibilities. Patient data, patient information, and doctor information. "Doctor's info" refers to the list of doctors that are available. Additionally, "patient info" refers to the registered patient's specifics. Two sensor values are present in the patient data. Heart rate and body temperature are the values. The MCU is connected to the Firebase via the WiFi network, and the sensors are connected to the nodemcu ESP8266. Next, the patient data field displays the Firebase data. [5] [6] [7]

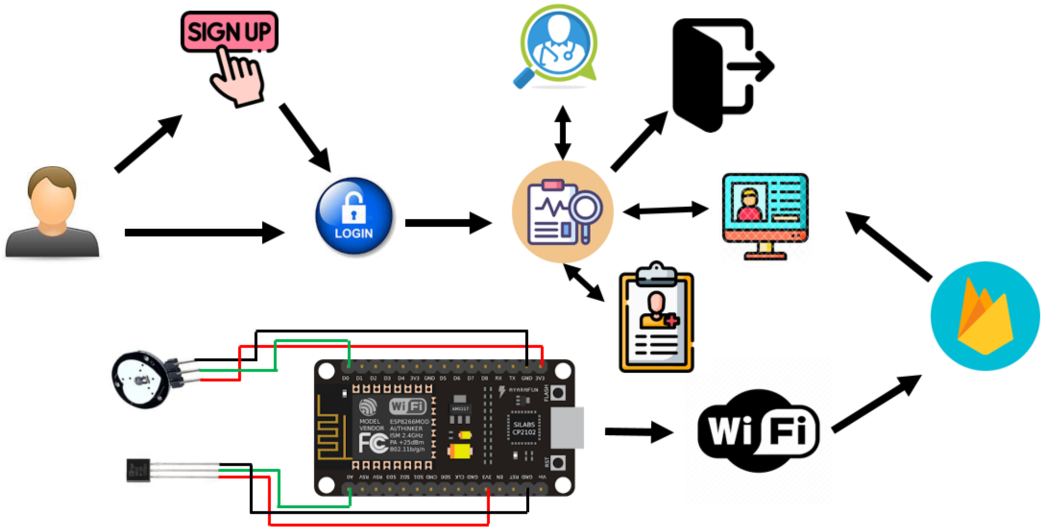

Figure 2.1: Proposed Architecture of System

2.2.1 Sensor Integration

The core of this project revolves around the seamless integration of wearable health sensors. These sensors continuously monitor vital health parameters such as heart rate and body temperature. Ensuring their compatibility with the Android application is a multifaceted engineering challenge. Standardizing data formats, achieving interoperability, and implementing data calibration are essential steps to guarantee data accuracy and consistency. This section dives into the complexities involved in sensor integration, emphasizing the need for robust engineering solutions. [8] [9]

2.2.2 Data Management

Effective data management is the backbone of the IoT-Based Smart Patient Monitoring System. This section focuses on the strategies and methods employed to collect, store, and process health data.As a storage firebase is used here . Firebase stores data which is collected by sensors and also stores registration authentication of users. Topics covered include data standardization to ensure uniformity, data calibration for accuracy, time synchronization for real-time monitoring. Precise data management is imperative to provide users with reliable and meaningful health insights.

2.2.3 User Interface and Experience

User experience is at the forefront of this project. Creating an intuitive and user-friendly interface within the Android application is essential for ensuring that users can easily access and understand their health data. This section delves into the design principles, data visualization techniques, and features such as health alerts and goal tracking. The user interface is pivotal in empowering users to make informed decisions about their well-being. [10]

User interface requirements:

- The design or layout of every form/page will be very clear and very interactive to the user.
- When the user opens the software, the welcome window will appear.
- In the login window the user can easily entered the desired password and login name.
- Then it will give the successfully login message.
- From each window the user can easily go to any desired window that is there is will be a absolute and relative linking.
- In every window is help and support option is present for the ease of user.
- There will be a proper collection of GUI interface, which will provide better look and feel.
- In the screen layout the background color is very light and the graphics and font style will be in proper manner and well organized.
- If the user will print any error statement, then it will give the proper error message display.
- In each window there will be alert, confirm etc message box for displaying message.
- This will provide the better security data because the menu window will be displaying according to the login.
- User can easily save its data into the database and keep track of the records.
- This software will be easily understandable and operable by the user.

Front-end software: Windows 10 or newer
Back-end software: Firebase

Hardware interface requirements:

- Windows.
- 8 GB RAM, 20/40 GB Hard disk.

Software interface requirements:

Following table 2.1 are the software used for the mediHelp system.

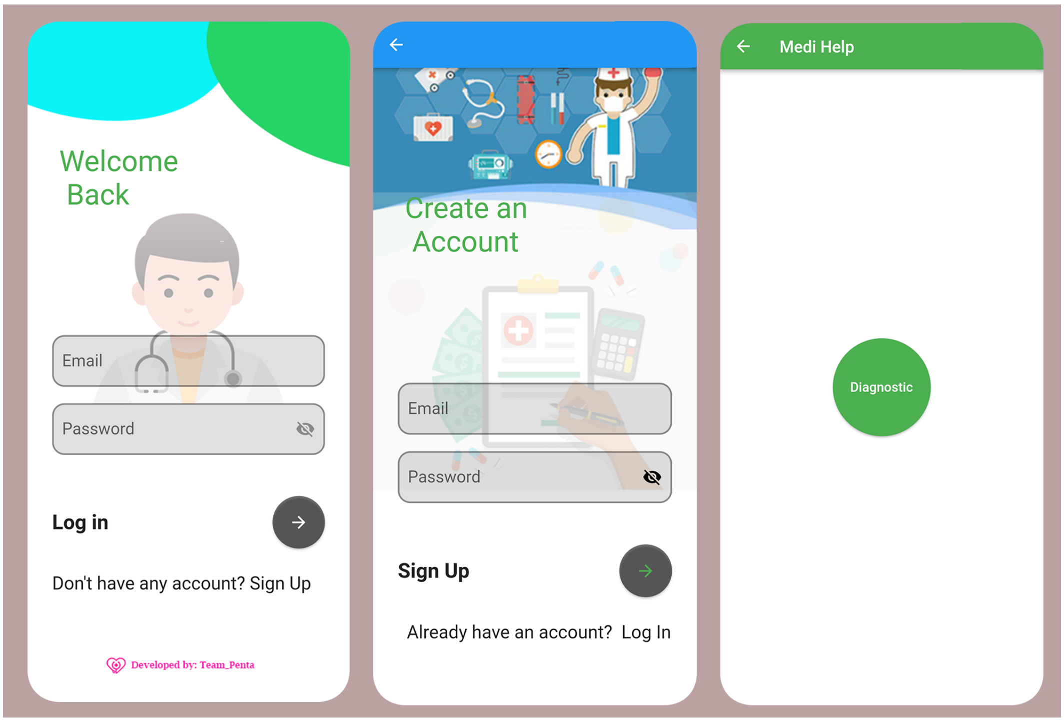

Figure 2.2: Login page, SignUp page, and Main page

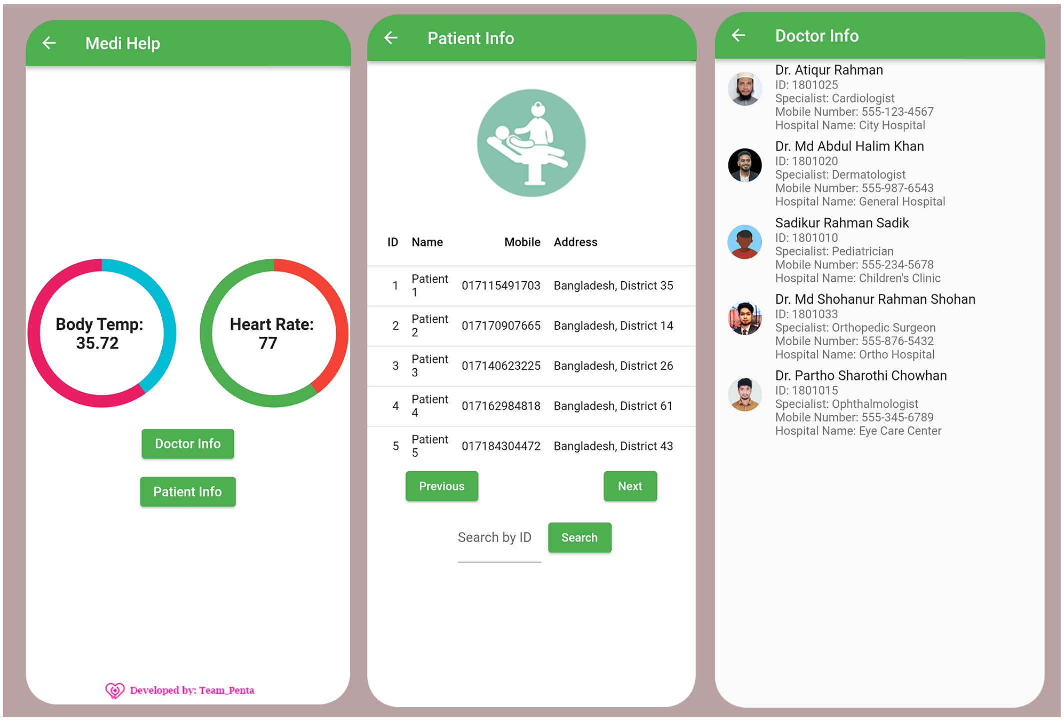

Figure 2.3: Diagnosis page, Doctor and Patient page

Communication interface requirements

- Wi-Fi and UART communications protocols used in the system.

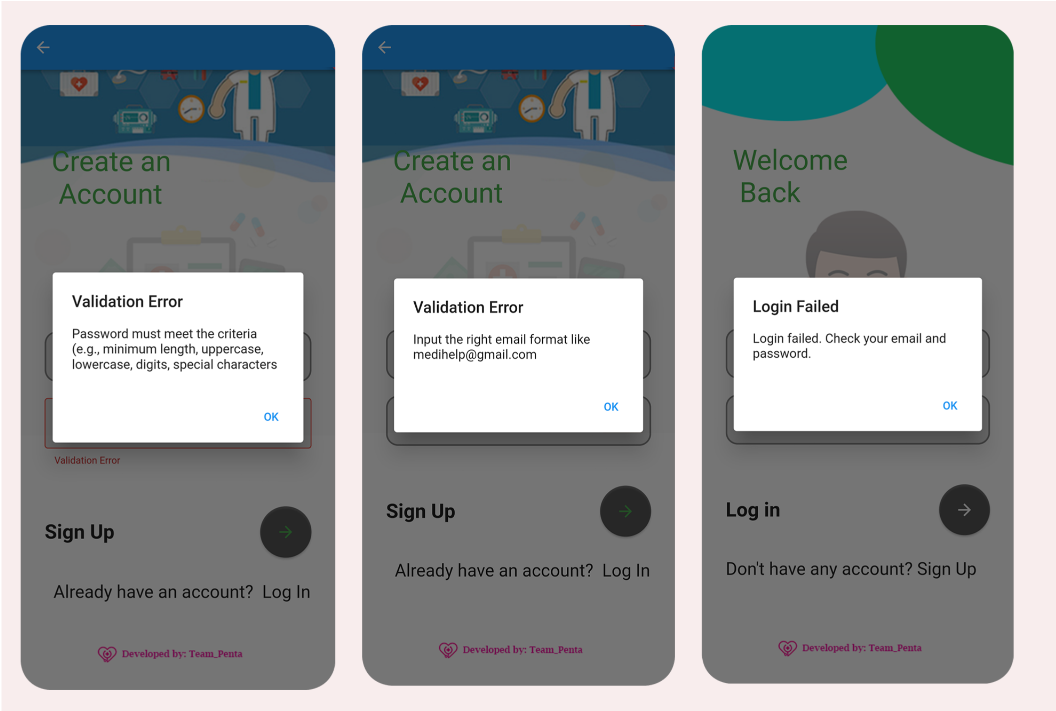

Figure 2.4: Check validation error

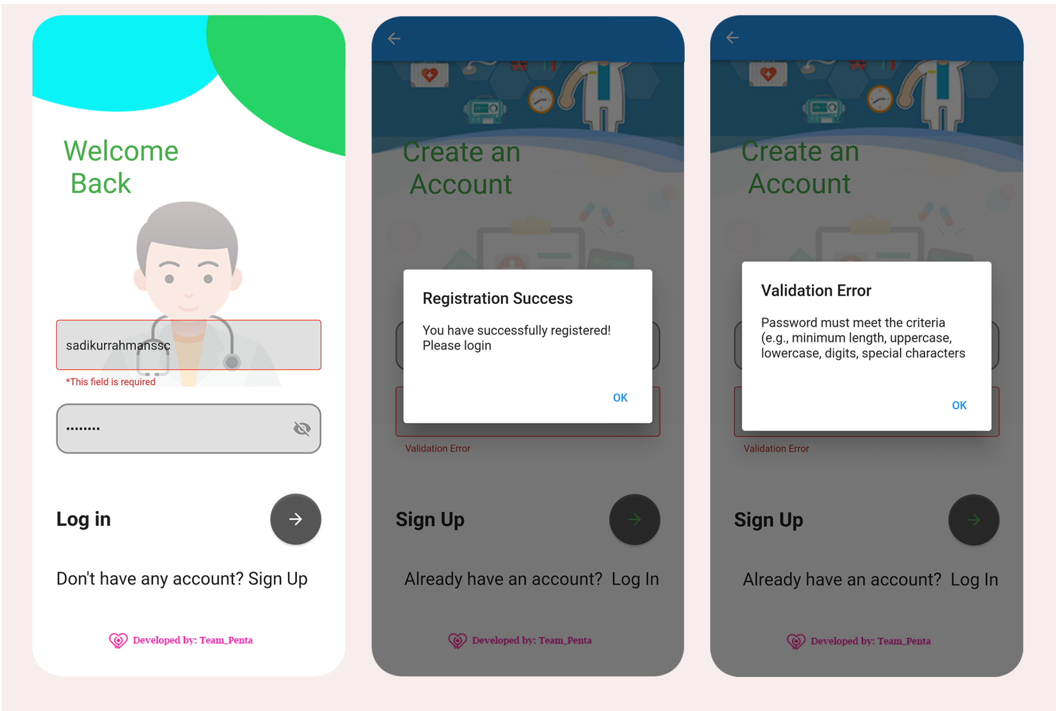

Figure 2.5: Check validation error

Table 2.1: The software used for mediHelp

Software used	Description
Operating system	We have chosen Windows 10/11
Database	To save the flight records, passengers record we have chosen Firebase database.
Android Studio, Arduino IDE, VS Code	To implement the project, we have chosen Flutter Dart Language for its more interactive support and C++ .

2.2.4 Security and Privacy

Data security and privacy are paramount in a project dealing with sensitive health information. This section outlines the security measures in place, including encryption techniques and regulatory compliance, to safeguard user health data. Building trust by ensuring the confidentiality and integrity of personal health information is a top priority. [7] In the subsequent subsections, we will explore each of these aspects in greater depth, providing a holistic view of the meticulous engineering and design efforts that underpin the IoT-Based Smart Patient Monitoring System.

2.3 Use Case Diagram

The following diagram shows the Use Case Diagram [1].

Actors:

Doctor

Use Cases:

Doctor Login: The Doctor logs in to the system using their credentials.

Sign Up: The Doctor can sign up for the system (assuming they are new users).

View Patient Health Data: The Doctor can view patient health data by clicking the "Diagnostic" button.

View Patient Information: The Doctor can view patient information by clicking the "Patient Info" button.

View Doctor Information: The Doctor can view their own information by clicking the "Doctor Info" button.

Search Patient by ID: The Doctor can search for a patient by ID in the patient table.

Relationships:

- The Doctor actor is associated with all use cases.
- The Diagnostic System actor is associated with the "View Patient Health Data" use case.

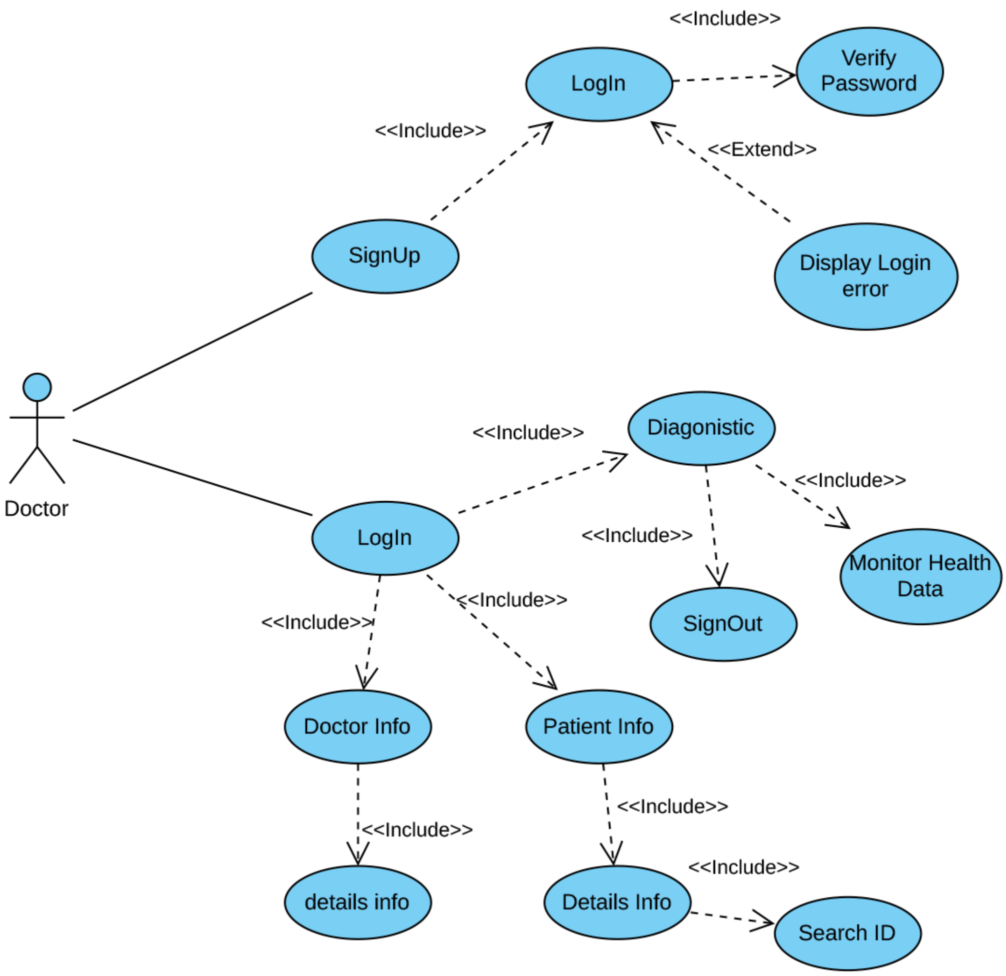

Figure 2.6: Use Case Diagram

2.4 Activity Diagram

The following figure 2.7 shows the Activity Diagram. [2]

Start: This is the starting point of the activity diagram.

Authentication: Here authorization is done with user's email id and password. After authorization of user then user is allowed to enter the system.

Diagnostics: After the authentication process, users are directed to a dashboard page where a central feature is the "Diagnosis" option, This button serves as a gateway to access comprehensive information, combining patient information, doctor information, and real-time sensor values.

View Patient Health Data: it represents the patient's health related value which is collected by sensors and sends to the firebase.

Patient info: Patient info represents data such as patient id ,mobile number , address. Here patient can be found by searching their id . Here patient's informations are stored. After logging in, the doctor is presented with the main menu. The doctor can choose different activities, such as "View Patient Health Data," "View Patient Information," "View Doctor Information," or "Search Patient by ID."

View Patient Health Data: The doctor selects this option. The system retrieves and displays patient health data. View Patient Information: The doctor selects this option. The system retrieves and displays patient information. View Doctor Information: The doctor selects this option. The system retrieves and displays the doctor's information.

Search Patient by ID: The doctor selects this option. The doctor enters the patient's ID. The system searches for the patient in the database. If found, it displays the patient's information. End: This is the endpoint of the activity diagram.

2.5 Data Flow Diagram

The following figure 2.8 shows the Data Flow Diagram. [11] [12]
Level 0 DFD: The Level 0 DFD provides an overview of the entire system, showing the external entities and the main processes. In your case, the external entities are the Doctor and the Diagnostic System, and the main process is the "Manage Patient Information" process.

Level 1 DFD: The Level 1 DFD provides more detail by decomposing the main process from the Level 0 DFD into sub-processes. In this case, we will focus on "Manage Patient Information."

2.6 Entity Relationship (ER) Diagram

Entities

Doctor

Attributes:

- DoctorID (Primary Key),
- Name,
- email,
- Password,
- specialist

Patient

Attributes:

- PatientID (Primary Key),),
- Name,
- mobile number,
- address.

Health Data

Attributes:

- HealthDataID (Primary Key)),
- PatientID (Foreign Key)
- Diagnosis Test Results

Relationships

- A Doctor can have many Patients (One-to-Many relationship).
- A Patient can have multiple Health Data records (One-to-Many relationship).

2.7 Other Nonfunctional Requirements

2.7.1 Performance Requirements

- The development of the software will be based on the object-oriented model.
- The performance of the functions and every module must be well.
- At every step the output of the one phase is the input of the other phase, and it will be reliable and accurate.
- For individual function the performance will be well.
- For login to the software password and username will be matched to the password, Name and email.
- saved in the database and thus only authenticated users are allowed to the login.
- This software will be well supported to the other embedded software such as Irrigation Monitoring Software etc.
- The overall performance of the software will reliable and enable the users to work efficiently. The development of the software will be based on the object-oriented model.

2.7.2 Security Requirements

- There will be proper security regarding to the accessing of data.
- The external security can be provided by given the login authentication.
- The data that are stored in the database must be private.
- There is also required a user authentication.
- The whole software is secure from the outside accessing.

2.7.3 Software Quality Attributes

Our software has many qualities attribute that are given below-

- **Adaptability-** This software is adaptable by any organization.
- **Availability -** The availability of the software is easy and for everyone.
- Correctness- The results of the function are pure and accurate.
- **Flexibility-** The operation may be flexible, and reports can be presented in many ways.

- **Maintainability-** After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
- **Portability-** The software can be deployed at any machine.
- **Reliability-** The performance of the software is better which will increase the reliability of the software.
- **Re-usability-** The data and record that are saved in the database can be reused if needed.
- **Robustness-** If there is any error in any window or module then it does not affect the remaining part of the software.
- **Usability-** To performs any operations and to understand the functioning of software is very easy.
- **Productivity-**This software will produce every desired result with accurately.
- **Cost effective-** This software is less in cost and bearable by any organization .

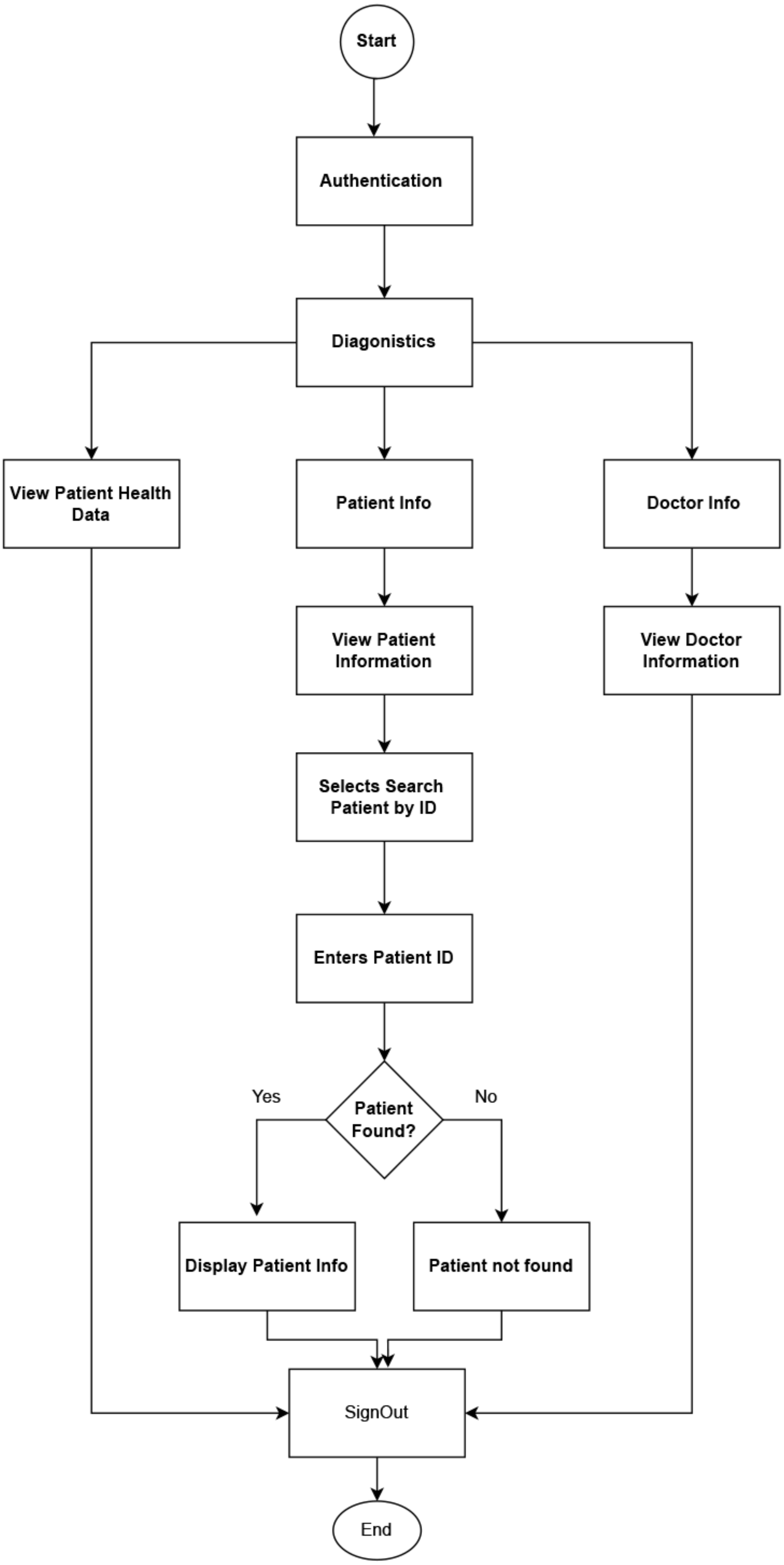

Figure 2.7: Activity Diagram

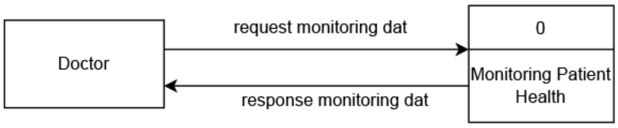

Fig: DFD Level 0

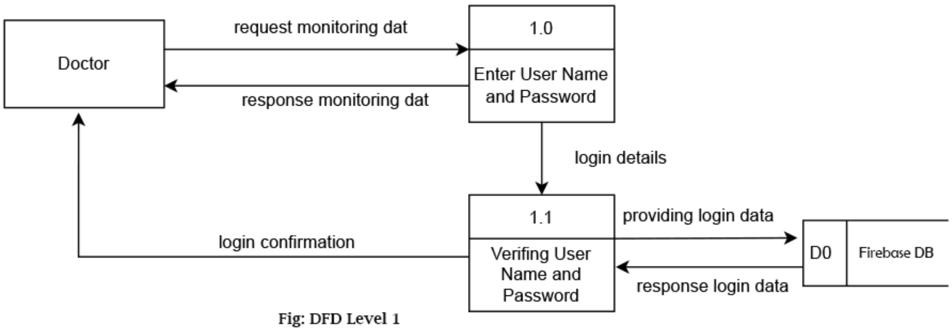

Fig: DFD Level 1

Figure 2.8: Data Flow Diagram

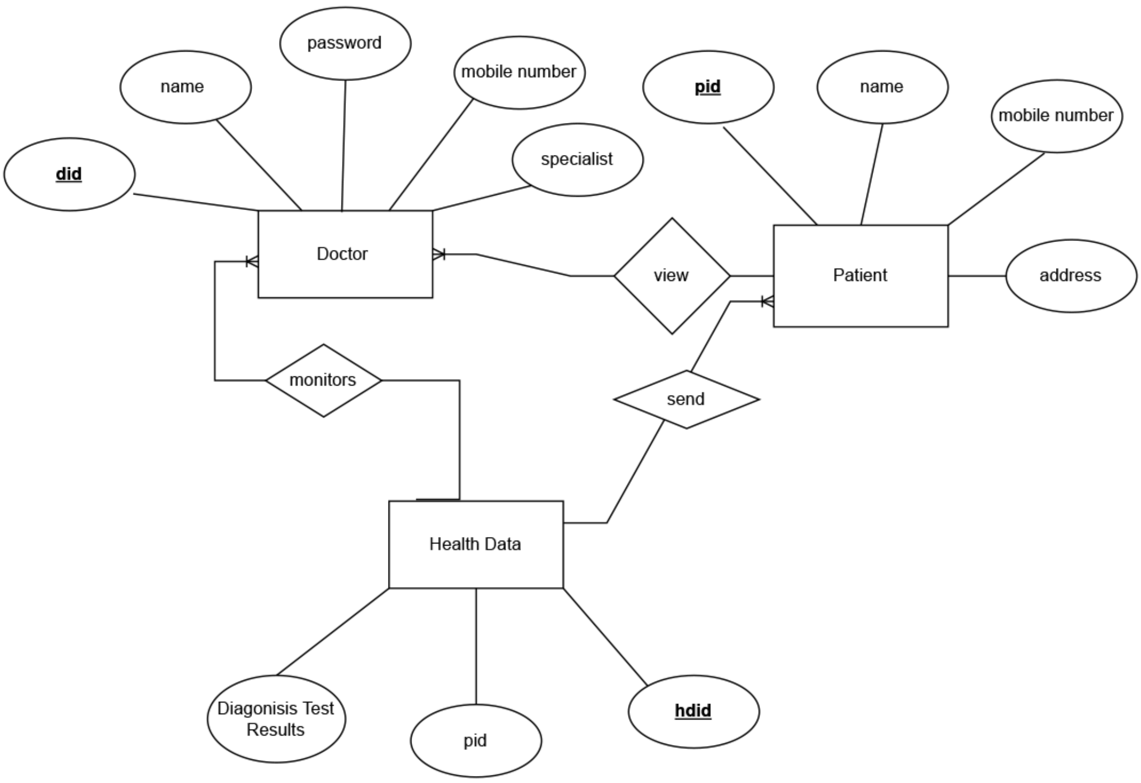

Figure 2.9: Entity Relationship (ER) Diagram

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

The experimental setup and environment installation needed for the simulation of IoT based Smart Patient Monitoring.

3.1.1 Android Emulator:

To test the Android app developed in Dart, we will use an Android emulator or device for app testing.

3.1.2 Development Environment:

We will require a development environment for Dart programming and Android app development. Common choices include Android Studio .

3.1.3 IoT devices:

Simulated IoT devices with sensors for health data, such as heart rate monitors, temperature sensors.

3.2 Results Analysis/Testing

This section delves into specific results in subsections, focusing on the comprehensive analysis and testing of various aspects of the IoT-Based Health Monitoring System.

3.2.1 Back-end Details

Authentication API for Firebase

Sign up here: Users can sign up with their email address and password, or with their phone number.User credentials are safely stored by Firebase.
Sign Up: Returning users can sign in using their previously saved credentials. Allows for several sign-in methods, including social networking.
User Administration: Users' accounts can be managed, passwords reset, and email addresses verified by developers.
Authentication via Tokens: Firebase generates a token for safe API access after successful authentication.

Realtime Database API by Firebase

Structure of Data: To store and organise data, a JSON-like data structure is used.
Real-time synchronisation: Changes made by one client are promptly mirrored on all other clients who are connected.
Data Access Regulations: Developers can build security rules to control data access.
Listeners: Allows listeners to be configured to receive updates anytime data changes.
Queries: Data retrieval is made easier by the ability to query and filter data.

3.2.2 Validation Check

The user must provide the necessary information in the box. The password must be eight characters long and contain at least one capital letter, one small letter, one special character, and a number value. The email address must also have the @ symbol.

3.2.3 Diagnosis

A user interface to access services will then display.To take the user's body temperature and heart rate, push the button.fig 3.2

3.2.4 Read Sensor Value

This screen appears after pressing the diagnostic button.The body temperature and heart rate are displayed on this page based on sensor data.Here, patients may also obtain information on doctors and facilities.

3.2.5 Doctor Info Page

When the user selects the "Doctor Info" button, a page with the doctor's ID, contact information, and hospital data appears.

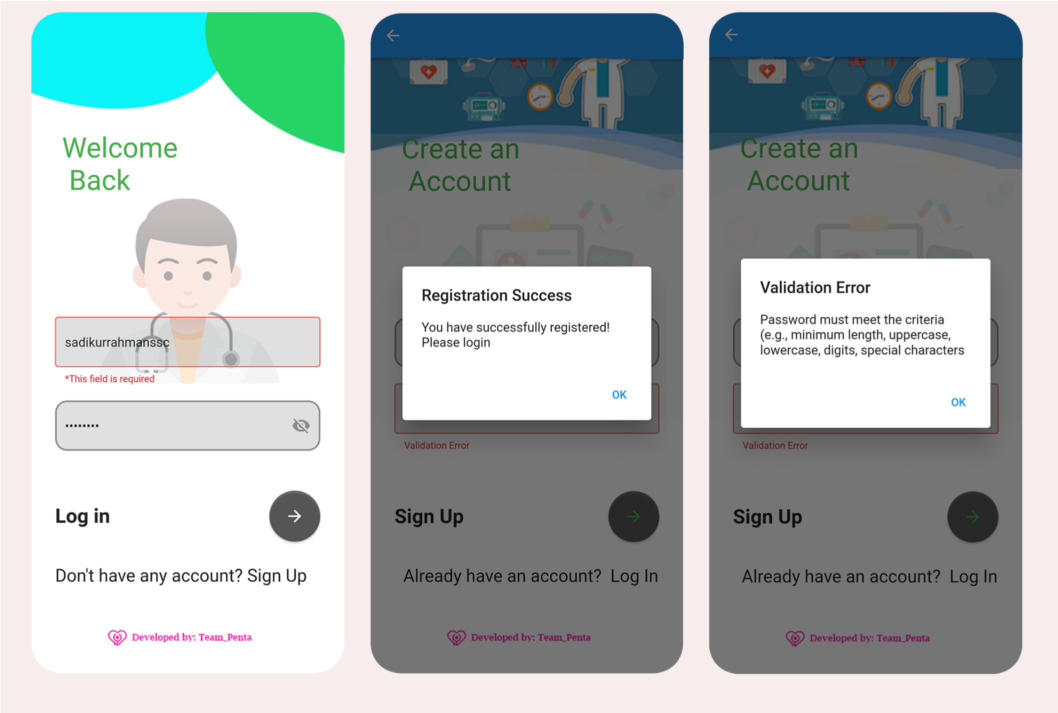

Figure 3.1: Check validation error

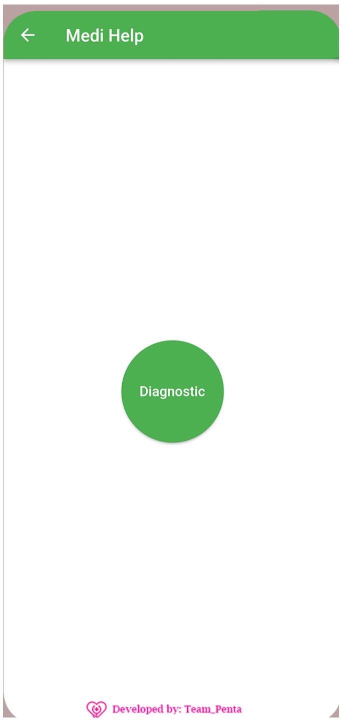

Figure 3.2: Main Page

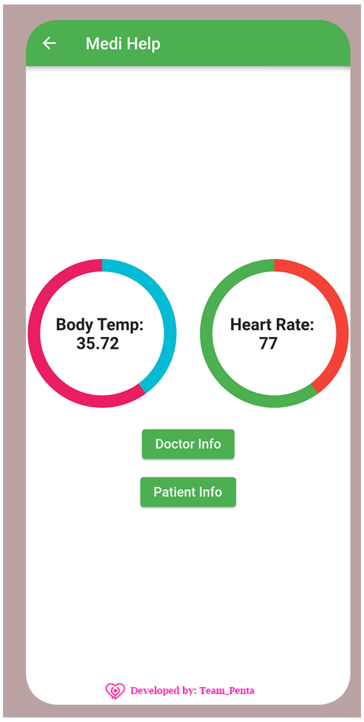

Figure 3.3: Health data page

3.2.6 Patient Info Page

The user can get information from this page by pressing the "Patient Info" button.then the user obtains from this website the patient's address, phone number, etc. [3]

3.2.7 Patient Info Page

Patients can be found by using their unique ID number. If the ID number is incorrect, the software displays the message "No Patient Found" in a popup window.

3.2.8 Firebase

For several compelling reasons [13], Firebase was chosen as the backend for Medihelp, our medical app. First and foremost, Firebase delivers a strong and scalable cloud architecture, reducing the need for complex server configurations. Its real-time database enables easy and rapid data synchronization between devices, guaranteeing that medical information for both healthcare providers and patients is always up to date. Furthermore, Firebase provides a complete authentication mechanism that enables secure access control to sensitive medical data. Because of its ease of connection with many platforms, including online and mobile, it is a versatile solution for a healthcare application that may need to adapt to numerous user devices. Furthermore, Firebase's serverless architecture minimizes development time and expenses, allowing us to focus more on developing features that improve Medihelp's overall user experience. Overall, Firebase simplifies backend development, providing a solid basis for our medical software. [14]

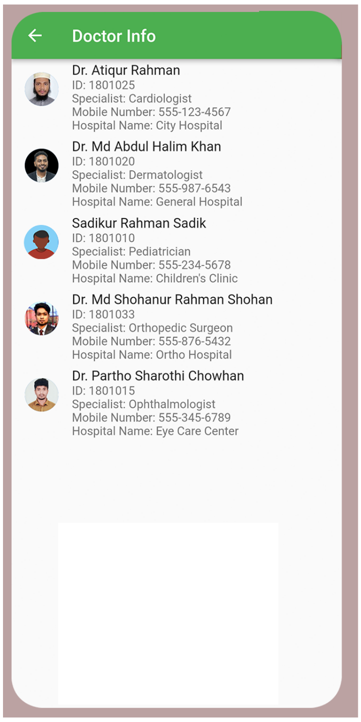

Figure 3.4: Doctor info page

Figure 3.5: Patient info page

3.2.9 Realtime Database

Firebase's Realtime Database acts as the backbone for combining important health data from sensors such as heart rate and body temperature in the context of Medihelp, our medical app. The selection of Firebase is consistent with our objective of providing real-time, reliable information to both healthcare professionals and patients. We build a secure and stable connection between the sensors and the cloud-based database by initializing Firebase within the Medihelp project and configuring it with the app's credentials. [15]

This interface allows for real-time updates, giving healthcare providers access to the most recent heart rate and body temperature information for effective patient monitoring. The ease of use of Firebase's SDK connection guarantees a smooth deployment process across multiple platforms, including web and mobile. Additionally, Firebase's strong authentication features add to the app's security foundation, guaranteeing that only authorized staff may access and manage sensitive medical data.

The features of the Realtime Database are perfectly aligned with Medihelp's mission of providing a user-friendly and responsive healthcare experience. This technology not only improves data synchronization efficiency, but it also contributes to the overall stability and performance of our medical app, underscoring Medihelp's dedication to delivering cutting-edge healthcare solutions. [16]

3.2.10 Realtime Database page

Firebase Authentication is effortlessly integrated into Medihelp, our medical app, to protect the security and privacy of user accounts. We used a simple email and password authentication system that allows users to easily register and log in. Users' email ad-

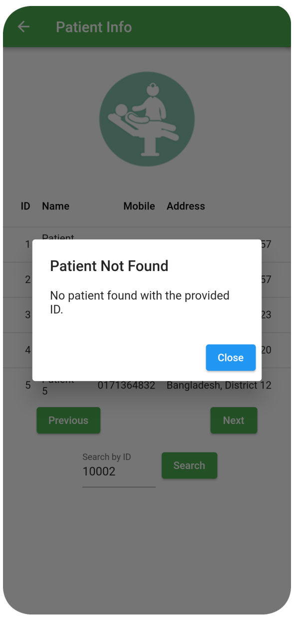

Figure 3.6: Patient info validation page

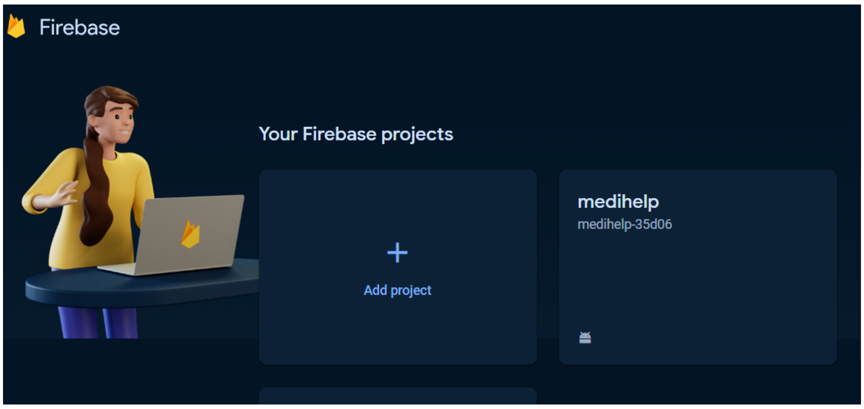

Figure 3.7: Firebase home page

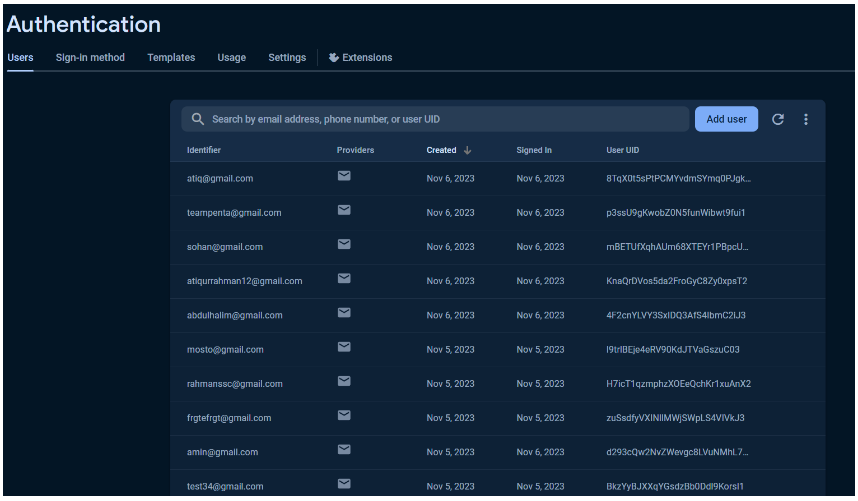

Figure 3.8: Firebase authentication page

dresses and passwords are securely transferred to Firebase Authentication for account creation when they sign up. Firebase implements strong security measures, such as password hashing using industry-standard techniques, to ensure sensitive data is stored safely. Returning users are prompted to enter their registered email and password, and their credentials are verified using Firebase Authentication. The connection features real-time authentication state monitoring, allowing Medihelp to adjust its interface based on whether or not a user is currently logged in. Medihelp prioritizes the confidentiality of user data with Firebase Authentication, boosting the overall trustworthy and personalized experience within the medical app.

3.3 Results Overall Discussion

This section provides a thorough examination of the results obtained during the testing and analysis of the IoT-Based Health Monitoring System. It evaluates the overall performance of the project, identifies strengths and areas for improvement, and recognises the complexities of the engineering problems encountered.

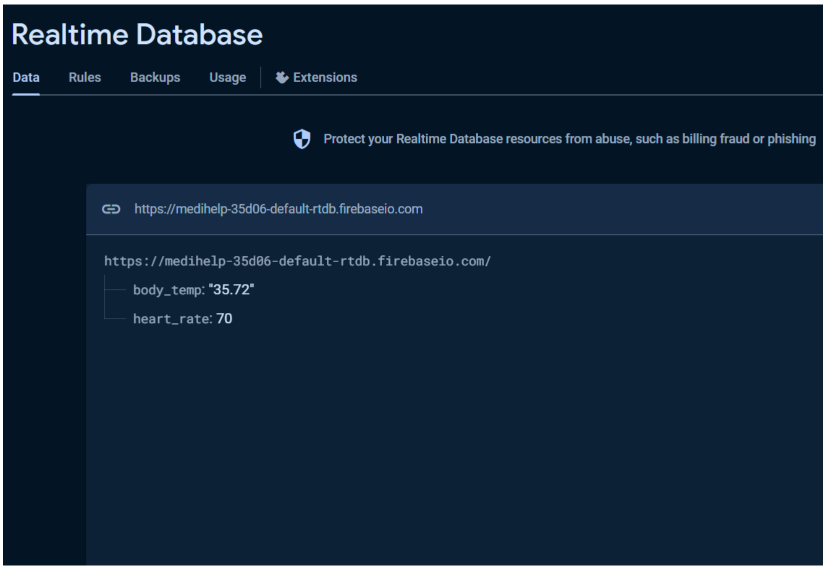

Figure 3.9: Firebase authentication page

3.3.1 Complex Engineering Problem Discussion

We delve into the specific attributes impacted by the complex engineering problems encountered during the project in this optional subsection. This section examines in detail the challenges encountered and the strategies used to address them, referencing the attributes mentioned in the project problem. This discussion provides a more in-depth understanding of the project's engineering complexities.

Throughout this section, the identified results and challenges are analysed to provide a comprehensive view of the project's outcomes. We can gain valuable insights into the system's performance and potential improvements by addressing the problems and complexities encountered.

Chapter 4

Conclusion

4.1 Discussion

Finally, this project has successfully developed an IoT-based smart Patient monitoring system that is integrated into an Android app built with the Dart programming language and has data fetching capabilities from Firebase. The project addressed the growing need for accessible and efficient health monitoring, particularly in remote healthcare settings . The feasibility and effectiveness of the system have been demonstrated through data transmission simulation and testing, health parameter analysis, and user interface evaluation, paving the way for a more convenient and user-friendly approach to health monitoring. The addition of Firebase for data retrieval improves the system's ability to securely and efficiently store and retrieve health data, making it a strong solution for modern healthcare needs.

4.2 Limitations

The limitations include the requirement for continuous and dependable internet connectivity for real-time monitoring, potential data security issues, and reliance on simulated IoT devices. Furthermore, for real-world scenarios, the accuracy of the health parameter interpretation and trend analysis algorithms may require further refinement. These limitations highlight the importance of continuous improvement and rigorous testing prior to deployment in a clinical or real-world setting.

4.3 Scope of Future Work

Future work on the project will include several avenues for improvement and extension. To begin, integrating real-world IoT devices and addressing the challenges of real-world data transmission and security will be critical. Furthermore, user feedback and input from healthcare professionals will guide the Android app's iterative development to ensure alignment with evolving healthcare standards and user needs. Extending the range of monitored health parameters and introducing predictive analytics for early health issue detection will also improve the system's capabilities. Finally, future research will look into the possibility of incorporating artificial intelligence for personalised health recommendations, making the system even more comprehensive and valuable in terms of improving users' overall well-being.

References

[1] Wu-Jeng Li, Chiaming Yen, You-Sheng Lin, Shu-Chu Tung, and ShihMiao Huang. Justiot internet of things based on the firebase real-time database. In *2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE)*, pages 43–47. IEEE, 2018.

[2] Chunnu Khawas and Pritam Shah. Application of firebase in android app development-a study. *International Journal of Computer Applications*, 179(46):49–53, 2018.

[3] Omid C Farokhzad and Robert Langer. Impact of nanotechnology on drug delivery. *ACS nano*, 3(1):16–20, 2009.

[4] Mohammad Monirujjaman Khan, Turki M Alanazi, Amani Abdulrahman Albraikan, Faris A Almalki, et al. Iot-based health monitoring system development and analysis. *Security and Communication Networks*, 2022, 2022.

[5] Vaneeta Bhardwaj, Rajat Joshi, and Anshu Mli Gaur. Iot-based smart health monitoring system for covid-19. *SN Computer Science*, 3(2):137, 2022.

[6] Mazin Alshamrani. Iot and artificial intelligence implementations for remote healthcare monitoring systems: A survey. *Journal of King Saud University-Computer and Information Sciences*, 34(8):4687–4701, 2022.

[7] Dhiraj Kapila, Prince Tehseen Ganai, G Sreenivasa Raju, Lims Thomas, Manisha Girase, and José Luis Arias Gonzales. Iot research on healthcare data to control illness of oncology. In *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pages 2229–2233. IEEE, 2022.

[8] Quan Zhang, Tao Jin, Jianguo Cai, Liang Xu, Tianyiyi He, Tianhong Wang, Yingzhong Tian, Long Li, Yan Peng, and Chengkuo Lee. Wearable triboelectric sensors enabled gait analysis and waist motion capture for iot-based smart healthcare applications. *Advanced Science*, 9(4):2103694, 2022.

[9] Pooyan Shams Farahsari, Amirhossein Farahzadi, Javad Rezazadeh, and Alireza Bagheri. A survey on indoor positioning systems for iot-based applications. *IEEE Internet of Things Journal*, 9(10):7680–7699, 2022.

[10] Amit Kishor and Chinmay Chakraborty. Artificial intelligence and internet of things based healthcare 4.0 monitoring system. *Wireless personal communications*, 127(2):1615–1631, 2022.

[11] Pramod Mathew Jacob, Joyce Jose, Josna Jose, et al. An analytical approach on dfd to uml model transformation techniques. In *2016 International Conference on Information Science (ICIS)*, pages 12–17. IEEE, 2016.

[12] Rosziati Ibrahim et al. Formalization of the data flow diagram rules for consistency check. *arXiv preprint arXiv:1011.0278*, 2010.

[13] Mohamed Abdalla Mokar, Sallam Osman Fageeri, and Saif Eldin Fattoh. Using firebase cloud messaging to control mobile applications. In *2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pages 1–5, 2019.

[14] Manal Fadhil Younis and Zainab Salim Alwan. Monitoring the performance of cloud real-time databases: A firebase case study. In *2023 Al-Sadiq International Conference on Communication and Information Technology (AICCIT)*, pages 240–245, 2023.

[15] Wolfram Wingerath, Felix Gessert, and Norbert Ritter. Invalidb: Scalable push-based real-time queries on top of pull-based databases. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1874–1877, 2020.

[16] Lakshmi Goswami and Piyush Agrawal. Iot based diagnosing of fault detection in power line transmission through google firebase database. In *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, pages 415–420, 2020.