Developing a Personalized Health Monitoring Index on Android using Flutter

Md. Rifat Ahmed
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
rifatahmedritu@gmail.com

Fardin Bin Islam
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
islam.fardin03619@gmail.com

Md Sadman Shouviq
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
shouviq.sadman@gmail.com

Rafiad Rahman Badhon
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
rafiad400@gmail.com

***Abstract*—Lack of information is a major problem in medical diagnosis because of the increase in complex disease patterns. However, the solution is relatively simple, given the way technology has developed in the past decade. People need something easy to use that will not take much time, like an Android application that stores various health factor data in under a couple of minutes. The system developed in this work serves precisely that purpose, improving the quality of the healthcare system as a whole. With a combination of first-hand experience and research, we saw a gap in monitoring daily well-being. Thus, regularly collecting key health factors allows patients and doctors to understand, predetermine, and cure ailments more accurately and efficiently. These data open the gates to open-source data, allowing this information to be used for further research in healthcare, where data is under-resourced and fragmented. Patients can also quickly get the contact information and location of hospitals, doctors, and pharmacies and make appointments through the app. Furthermore, the articles on the home page will give them better access to health-related information. Considering all the functions available in the proposed application, it is like having a complete healthcare package inside a phone.**

Keywords—mobile-health, health monitoring system, Android application, eHealth, Flutter

I. INTRODUCTION

Using smart devices like Android, iOS phones, tablets, or computers to keep track of one's health information and access various health services is known as "Mobile Health" or M-Health [1]. With the development of technologies, people rely on smart devices in their daily lives more than ever, especially mobile phones. The absence of data on patient information is one of the oldest problems in the medical sector [2]. Individuals often give a rough idea of what and when they have been through an ailment, which leads to misdiagnosis. With the proposed app's help, people can keep track of their various health factors like blood pressure, heart rate, diabetes level, and even weather information to help understand external factors. The primary feature of our app is monitoring the health index; thus, we named it 'HoMie,' which gives an intimate and friendly vibe for an app that the user will frequently use.

In [3], A. N. A. Yusuf et al. developed an intelligent health monitoring system using the rational unified process (RUP) framework to create an Android application. In this application, the three primary actors, i.e., patients, doctors, and health staff, are connected to the database through an API because directly accessing the database can lead to many complications. Using the application, the patients will be updating their health information data in the database, and the doctors and health staff can access the data using a website where the back-end is connected to the database using an API. The proposed application's features are divided into multiple segments based on the flowchart design of several activity diagrams, such as inserting data, viewing data, setting an appointment, calling the medical emergency number, and a few more functionalities. The authors followed the black box evaluation process to test out the application, for which they ran a total of 11 tests. In all of them, the result came "As expected," confirming that the features were working as intended.

In [4], A. N. N. Chamim and his colleagues developed a healthcare system using the Android platform. In this paper, the primary attention was the doctors and patients and how they can be connected using the help of an Android app. Using this app, the user can get information about heart rate and average body temperature of people and various other information regarding health. This work demonstrated how Bluetooth was utilized as a conduit between the app and the Arduino to keep track of the outcomes. Using the developed system, the authors measured the body temperature and heart rate with 98.45% and 95.67% accuracy, respectively.

John Torous and his team created LAMP, an acronym for the words "Learn," "Assess," "Manage," and "Prevent." They created a transparent system where patients had complete control of their data, which was data-driven and built for the community based on their feedback. They provide relevant content to help patients "Learn." They collect data in a future-proof scheme in several ways, i.e., intelligent devices and surveys, which they "Assess," suggest ways to "Manage" their problems, and provide the data back to the patients in a simple graphical way to help patients deal with the problem. Built on its powerful data server, protocol, and automation framework, the app is naturally designed and constructed for clinical insights and research methodologies in digital medicine. The data of their developed system can be accessed from the app or a web portal. They ran a total of four activity tests—two cognitive tests and two surveys, among which they got stable results only on the cognitive tests [5].

In [6], X. Kong et al. presented the designation and implementation of a wireless communications system that uses medical sensor detection technology and wireless communication technology to enable remote medical monitoring of various physiological parameters. The central server maintains the patient's medical records and offers the system's other subsystems a data-sharing service. The server and mobile data collection terminal for the Android system are connected via WiFi, and data services are implemented using an HTTP web server interface. The external terminal connects to the webserver to access medical data. Using SQLite databases for client data storage and MySQL databases for server data storage, they create and install a health system using Android's Bluetooth technology and several physical sign detection tools. The system has excellent scalability and flexibility. This can also increase the functionality of the remote server so that the doctor has more data to work with, diagnosing patients accurately and identifying changes in their bodies more quickly. The project's applications are numerous, it can provide a wealth of experience, and it has promising prospects.

Our project has four goals to achieve. First, collect the data from the user via daily report entries. Second, let the user upload their previous medical documents. Third, let the user print or save their reports, allowing the users' history to be shared for better assessments by the doctors. Finally, sharing the data with patient consent allows data-driven systems to be developed and improved. Users can also quickly look for doctors, hospitals, and pharmacies through the developed system. Moreover, the articles on the home page will allow the users to learn more about their well-being through the content.

So, many things had to be covered for the development of the system to be done systematically. In methodology, the front-end, back-end design, and database management are explained, followed by a flow chart showcasing the system's whole development cycle. Then, some activity diagrams are used to show the working process of some functions. Then, in section III, several system use cases were evaluated and discussed. Lastly, section IV discusses how the app could benefit the user, how it can contribute to a better healthcare system, and the plans are with the developed system.

II. Methodology

A. *Front-end Design*

The Flutter framework was chosen for its excellence in creating responsive user interfaces. The initial design phase utilized Figma, a robust online UI design tool, to ensure a meticulous user experience. Leveraging prebuilt Flutter widgets not only expedited the development process but also allowed for seamless modification to align with the UI prototypes created in Figma. Dynamic scrollable pages were implemented to accommodate varying screen sizes effectively. Additionally, the integration of SVG files adhered to high visual design standards.

B. *Back-end Design*

The back-end design showcased a thoughtful integration of various Flutter packages. Notably, the "url_launcher" package efficiently retrieved articles, demonstrating a meticulous approach to content delivery. Then, the "flutter_svg" package was used for the banners and framing designs. The app has many animations; among them, the loading animations were built with the "flutter_spinkit" package, and the "animations" package was used for others. Then the "flutter_phone_direct_caller" package was used to make emergency calls and book appointments. Then, on the doctors' profiles, pharmacies', and hospitals' pages, the integration with Google Maps API showcased a sophisticated approach to location-based services, utilizing the "google_maps_flutter" package. The "path_provider" package lets the user upload files into the database by pointing at the file's location, i.e., a user uploading a medical document from any folder on their device. Now, coming to one of the system's major parts, the "syncfusion_flutter_pdf" package played a pivotal role in adhering to industry standards for generating PDF reports. As for the database, several Firebase-related packages were used for authentication and data entries. The data in the database was entered in JSON format, meaning every piece of data has two parts – keys and values. The values a user enters are bound to a key to store in the database.

C. *Database Management*

The database used for this system is Firebase, a NoSQL database that stores and synchronizes data in real time, meaning it saves the documents instantaneously. The cloud-based Real-time database ensures instantaneous data synchronization, fostering a collaborative and responsive user experience. Each connected client receives a real-time synchronization of the data, which is stored in the JSON format. All the clients will share a real-time database instance and automatically receive updates of the most recent data when using the application. Real-time Database Data Sync by Firebase replaces standard HTTP queries. Every few days, updates are delivered to each connected device in milliseconds. Without considering network code, it offers a collaborative and exciting experience. The Real-time Database SDK for Firebase maintains data on disk to ensure that the Firebase app is responsive even when the user is offline. Upon reconnecting, the client computer retrieves any missing updates and synchronizes them with the server's most recent state.

D. *Use Case Diagram*

The use case diagram in Fig. 1 effectively illustrates user interactions within the proposed Android application, covering a spectrum of functionalities, including account management, health-related information access, and report generation, aligning with industry-standard system analysis and design practices.

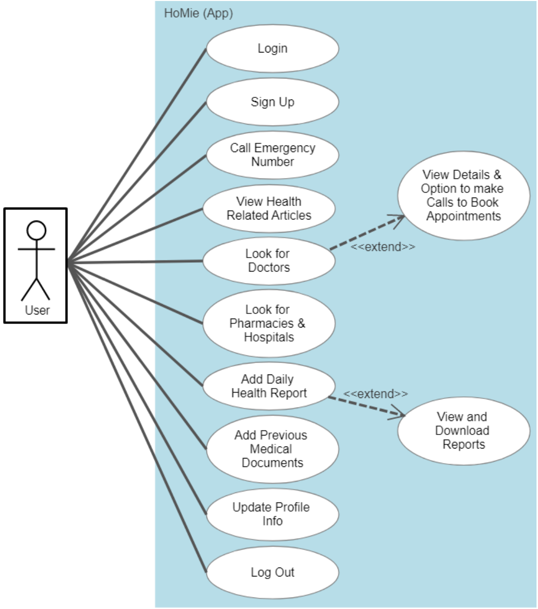

HoMie (App)
Login
Sign Up
Call Emergency Number
View Health Related Articles
Look for Doctors
Look for Pharmacies & Hospitals
Add Daily Health Report
Add Previous Medical Documents
Update Profile Info
Log Out
View Details & Option to make Calls to Book Appointments
View and Download Reports
<<extend>>
<<extend>>
User

Fig. 1. Use case diagram of the proposed Android application.

E. Activity Diagrams

The activity diagrams provided detailed insights into the flow of functions outlined in the use case diagram, offering a comprehensive understanding of the processes involved in logging in/sign-up, adding a new report, and viewing/downloading a report.

In Fig. 2, the activity diagram shows the procedure behind logging in and signing up. It starts with the user trying to log in using their email and password or going to the sign-up page. In the first case, the system verifies the email and password given by the user. If they are correct, the user is directed to the Home Page and notified if they do not match. As for signing up, after the user clicks on sign-up, the system redirects the user to the Sign-Up page, where they fill out the form, and upon verifying if the data is valid or not, the user is notified. From there, they can access the login page to get into the system.

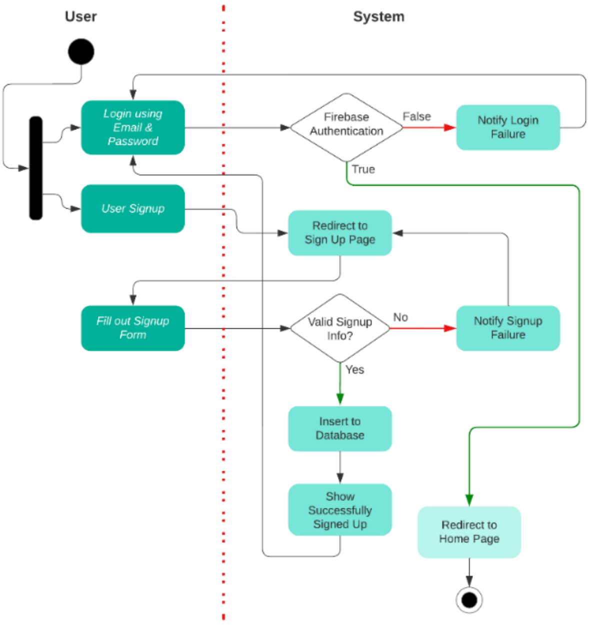


Fig. 2. Activity diagram of login and sign-up Process

The following activity diagram in Fig. 3 gives the process of adding a new report. The user logs in to the system, lands on the home page, and then goes to the daily report tab from the navigation bar, where they fill out the data. After clicking "Add New Report," the report is added to the database, and the user is notified.

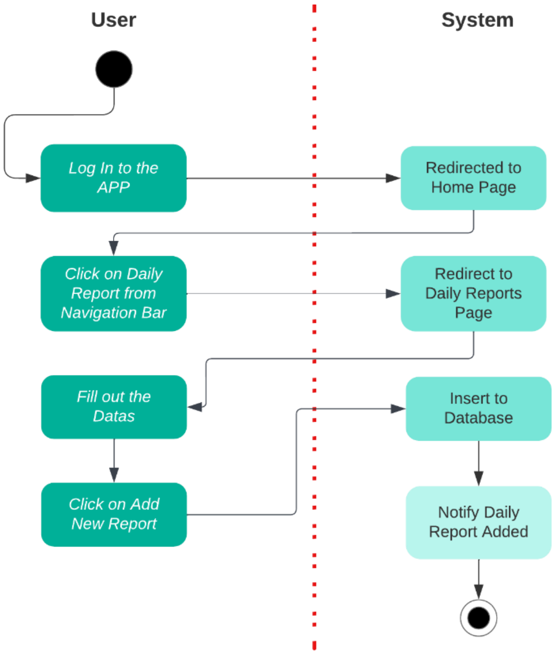


Fig. 3. Activity diagram of adding a report.

The final activity diagram here shows viewing and downloading a report in Fig. 4. First, the user must go to the profile page using the navigation bar. On the profile page, by clicking "Daily Reports," the user can view the reports added to the database. In order to download a report, the user needs to open it by clicking on it, and the report will be generated in a downloadable PDF format.

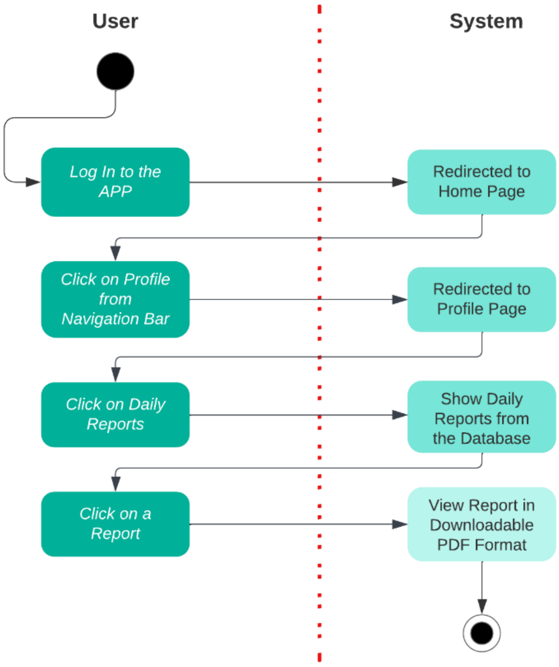
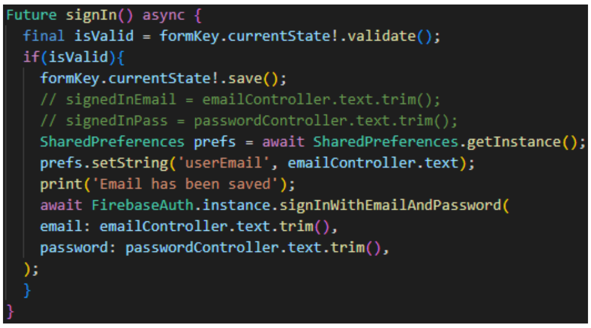

User
System
Log In to the APP
Redirected to Home Page
Click on Profile from Navigation Bar
Redirected to Profile Page
Click on Daily Reports
Show Daily Reports from the Database
Click on a Report
View Report in Downloadable PDF Format

Fig. 4. Activity diagram of viewing and downloading a report.

III. Results and Discussion

The black box testing method was used for the developed "Health Monitoring Index" system, meaning several test cases were run and checked to see if the system gave the desired output.

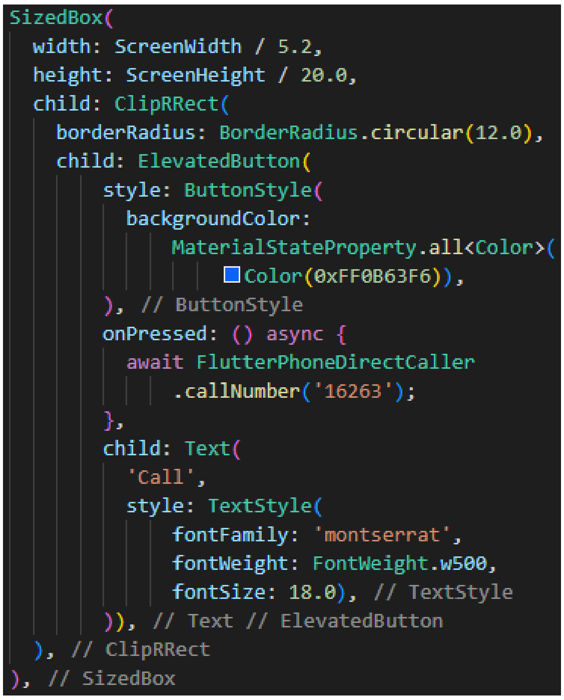
Starting the tests with the sign-up feature, it was checked if the system lets the user create multiple accounts using the same email address. However, it did not, meaning only one account could be created with an email. Then, on the login page, it was checked if the system lets the user log in with fake login credentials, which it did not but with valid data already existing in the database, it redirected the user to the home page. We can see from the code snippet given in Fig. 5 that the login credentials are verified through Firebase authentication when a user tries to log into the system.



```
Future signIn() async {
  final isValid = formKey.currentState!.validate();
  if(isValid){
    formKey.currentState!.save();
    // signedInEmail = emailController.text.trim();
    // signedInPass = passwordController.text.trim();
    SharedPreferences prefs = await SharedPreferences.getInstance();
    prefs.setString('userEmail', emailController.text);
    print('Email has been saved');
    await FirebaseAuth.instance.signInWithEmailAndPassword(
    email: emailController.text.trim(),
    password: passwordController.text.trim(),
  );
  }
}
```

Fig. 5. Firebase authentication of the login process.

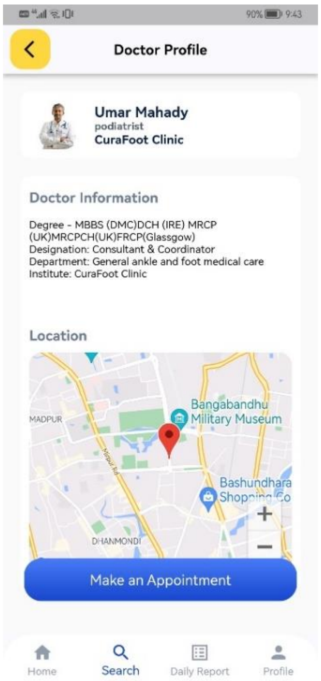
Then, on the home page, the emergency health care centre call button was checked, and upon clicking on it, it asked for permission to call. After giving permission, the call was made, meaning that the emergency button was working. A snippet of the code behind this process is shown in Fig. 6.



```
SizedBox(
  width: ScreenWidth / 5.2,
  height: ScreenHeight / 20.0,
  child: ClipRRect(
    borderRadius: BorderRadius.circular(12.0),
    child: ElevatedButton(
        style: ButtonStyle(
          backgroundColor:
              MaterialStateProperty.all<Color>(
                  Color(0xFF0B63F6)),
        ), // ButtonStyle
        onPressed: () async {
          await FlutterPhoneDirectCaller
              .callNumber('16263');
        },
        child: Text(
          'Call',
          style: TextStyle(
              fontFamily: 'montserrat',
              fontWeight: FontWeight.w500,
              fontSize: 18.0), // TextStyle
        )), // Text // ElevatedButton
  ), // ClipRRect
), // SizedBox
```

Fig. 6. Code behind the emergency call button.

Then, clicking on an article redirected to the article site, meaning the API of the given site was connected successfully in the back-end. Coming to the search tab, clicking on different doctors opened up their details pages; from there, we could see their locations, meaning the Google Maps API was successfully connected. As we can see from the screenshot in Fig. 7, the doctors' chamber location is shown using Google Maps.



Doctor Profile

Umar Mahady
podiatrist
CuraFoot Clinic

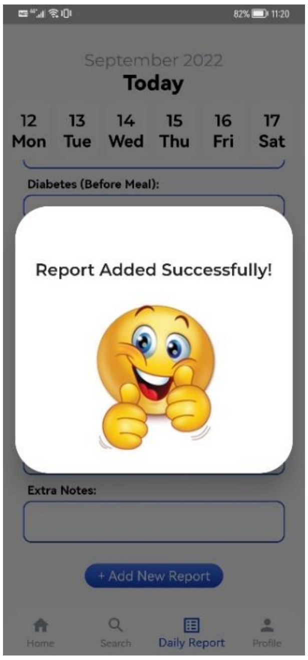
Doctor Information

Degree - MBBS (DMC)DCH (IRE) MRCP (UK)MRCPCH(UK)FRCP(Glassgow)
Designation: Consultant & Coordinator
Department: General ankle and foot medical care
Institute: CuraFoot Clinic

Location

Fig. 7. Showing location using Google Maps API.

After clicking the "Make an Appointment" button, a popup of the doctors' chamber details came up, and clicking on the "Call" button sent a call to the number shown there, which was the expected result. On the daily report page, filling out the data and clicking the "Add Report" button showed a "Report Added Successfully" dialogue box, meaning it worked as expected, as shown in Fig. 8.



September 2022
Today

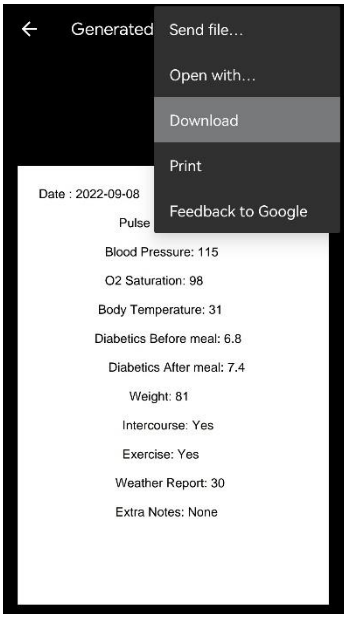
Diabetes (Before Meal):

Report Added Successfully!

Extra Notes:

Fig. 8. Successful report entry notified using dialogue box.

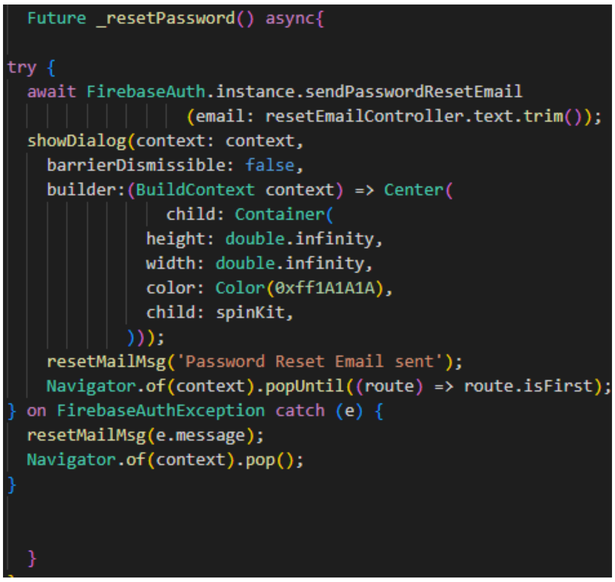
Then, on the profiles page, clicking "Daily Reports" showed the daily reports added to the database; after opening them, there was an option to download as well, meaning generating downloadable pdf was working as intended, the same way it is displayed in Fig. 9.



Date : 2022-09-08
Pulse
Blood Pressure: 115
O2 Saturation: 98
Body Temperature: 31
Diabetics Before meal: 6.8
Diabetics After meal: 7.4
Weight: 81
Intercourse: Yes
Exercise: Yes
Weather Report: 30
Extra Notes: None

Fig. 9. Generating downloadable PDF report.

Then, in profile settings, changing the profile data updated the profile data in the database, meaning it was also working as expected. After giving the email address and clicking the reset password button on the reset password page, it sent a password reset email, meaning this feature also worked perfectly. In Fig. 10, a code snippet of the password reset process is given.



```
  Future _resetPassword() async{

try {
  await FirebaseAuth.instance.sendPasswordResetEmail
                (email: resetEmailController.text.trim());
  showDialog(context: context,
    barrierDismissible: false,
    builder:(BuildContext context) => Center(
            child: Container(
          height: double.infinity,
          width: double.infinity,
          color: Color(0xff1A1A1A),
          child: spinKit,
        )));
    resetMailMsg('Password Reset Email sent');
    Navigator.of(context).popUntil((route) => route.isFirst);
} on FirebaseAuthException catch (e) {
  resetMailMsg(e.message);
  Navigator.of(context).pop();
}


  }
```

Fig. 10. Password reset process.

To add previous medical reports to the database, the user clicks on the add button and selects an image from the file manager; the file gets uploaded, meaning this feature is also working as intended, as given in Fig. 11.

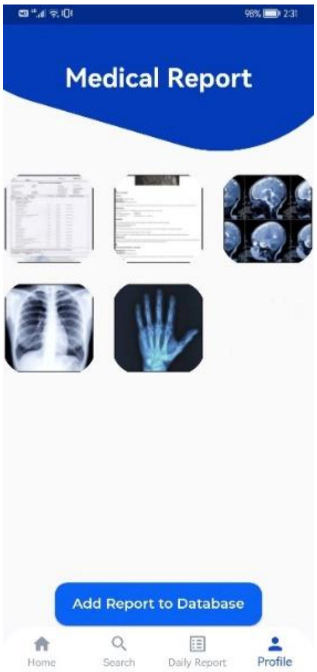


Fig. 11. Adding medical reports to the database.

Finally, clicking the logout button made the user exit the system, which also worked as intended. So, we ran a total of 13 test cases, and in all of them, the results came as intended, as demonstrated in Table I.

TABLE I. TESTING DIFFERENT FEATURES OF THE APP

Test Cases	Test Results
Firebase login authentication	Working as intended
Signing up for a new account	Working as intended
Emergency call button	Working as intended
Home page articles redirecting	Working as intended
Google Maps using API	Working as intended
Calling the number set in the doctors appointment details	Working as intended
Adding new reports	Working as intended
Viewing already added reports	Working as intended
Downloading reports	Working as intended
Updating profile information	Working as intended
Resetting password	Working as intended
Adding medical reports to the database	Working as intended
Logging out of the system	Working as intended

IV. CONCLUSIONS

Keeping track of one's health is something that people often neglect to do, which may cause issues when seeking medical advice since there is little information available. The relationship between a healthcare provider's ability to diagnose patients accurately and the amount of information they have access to is a strong indicator of the need for thorough health monitoring. Our designed system succeeds in filling this crucial gap by emphasizing the consistent archiving of users' health data, promoting an extensive and easily accessed health profile. This facilitates a better-educated healthcare experience by giving customers the option to upload previous medical records in addition to allowing them to get their health reports whenever required.

Our system offers customers a plethora of health-related materials in addition to its many capabilities, which go beyond simple data storage. The system markets itself as a comprehensive health management solution and offers everything from educational resources covering a wide range of health issues to expedited searches for healthcare providers, including physicians, pharmacies, and hospitals. By adding a function that allows users to schedule appointments, the user experience is further improved and timely access to healthcare services is made possible.

In order to enhance the system's capabilities, we have outlined essential projects in our strategic development plan. One planned feature is a messaging network that would facilitate user interactions and build a cooperative health community. Developing a review function, enabling people to exchange opinions and experiences about healthcare providers, will also promote openness and well-informed decision-making.

Our dedication to constant improvement remains the same as the technology develops beyond its present prototype state. The intended course is to achieve both efficiency gains and sustainability, meaning that the system will be able to accommodate changing user requirements smoothly. The intention of this health management application is to transform it from a working prototype into a vital resource that actively supports proactive and knowledgeable personal health and well-being.

References

[1] P. Cipresso *et al*., “Is your phone so smart to affect your state? An exploratory study based on psychophysiological measures,” Neurocomputing, vol. 84, pp. 23–30, 2012.

[2] E. Tantoso *et al*., “Hypocrisy Around Medical Patient Data: Issues of Access for Biomedical Research, Data Quality, Usefulness for the Purpose and Omics Data as Game Changer,” Asian Bioethics Review, vol. 11, pp. 189–207, 2019.

[3] 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,” International Conference on Nano Electronics Research and Education, pp. 1-6, 2018.

[4] A. N. N. Chamim, J. Rinaldi, Y. Ardiyanto, I. Iswanto, and A. Ma'arif, “Heart Rate and Body Temperature Monitoring Based on Android Operating System,” International Conference on Industrial Electrical and Electronics, pp. 143-148, 2020.

[5] J. Torous *et al*., “Creating a Digital Health Smartphone App and Digital Phenotyping Platform for Mental Health and Diverse Healthcare Needs: an Interdisciplinary and Collaborative Approach,” Journal of Technology in Behavioral Science, vol. 4, pp. 73-85, 2019.

[6] X. Kong, B. Fan, W. Nie and Y. Ding, “Design on mobile health service system based on Android platform,” IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, pp. 1683-1687, 2016.