ARM: A Real-Time Health Monitoring Mobile Application

Saeid Pourroostaei Ardakani(✉), Xuting Wu, Shuning Pan, and Xinyu Gao

School of Computer Science, University of Nottingham Ningbo China,
Ningbo 315100, China
{saeid.ardakani,scyxw1,zy22054,scyxg1}@nottingham.edu.cn

Abstract. Considering the risks and difficulties of providing in-patient medical services, this is of great significance to develop mobile applications which are able to provide remote healthcare. This paper proposes a Real-time Health Monitoring Mobile Application that enables patients to remotely report health cues to healthcare teams and receive treatment plans. This allows healthcare team to collect and analyse blood oxygen, heart rate and body temperature which are reported by a Bluetooth interconnected sensory device. Yet, a chat environment is provided that allows patients to submit self-questionaries and communicate with healthcare teams in real time. In addition, this allows healthcare teams to access the patients' health records and profiles and remotely monitor their health status. According to the test results (i.e., functional, system, front-end, back-end and hardware), this mobile application is able to offer a number of promising benefits -mainly remote basic checkup with minimum contact risks especially during COVID-19 pandemic.

Keywords: Wearable sensors · Remote sensing · Healthcare · Mobile communications

1 Introduction

Recent advances and innovations in accessibility and usability of mobile devices and wearables form a new life style. Smartphones can be used almost everywhere and provide users required services including education, healthcare, marketing and logistics. Users stay connected and access to a wide range of information, applications and services as required. Healthcare is one of the key fields that benefits from mobile and digital technologies by interconnecting healthcare stacksholders beyond time and geographical location [1].

Mobile health (M-health) [2] applications offer a number of benefits such as resource conservation, treatment compliance enhancement and healthcare service accessibility improvement to healthcare stakeholders. According to [3], m-health has the potential to significantly reduce healthcare cost and enhance patient convenience. For example, M-health applications provide patients remote consultation and health check-up that results in reduced treatment cost and time [4].

J. A. Marmolejo-Saucedo et al. (Eds.): COMPSE 2021, LNICST 393, pp. 45–59, 2021.
https://doi.org/10.1007/978-3-030-87495-7_4

In-patient treatments, particularly, are usually expensive and resource consuming (e.g. time) as patients need to visit hospitals to receive healthcare services. As [5] outlines, in-patient healthcare is one of the largest expenditures, about one trillion USD in the United States. Hence, researchers and scientists are highly interested in proposing and utilising M-health applications to reduce healthcare risk, decrease cost and conserve healthcare resources especially during COVID-19 pandemic where healthcare resources are restricted and in-patient treatment is highly risk-full.

This article proposes a real-time health monitoring mobile application (ARM) through which healthcare teams would be able to remotely monitor patients and provide them required treatment instructions with minimum physical contact. Indeed, the target is to improve healthcare treatment efficiency and enhance convenience. Patients fill out online healthcare questionnaires and utilise wearable sensors to report body data such as heartbeat, blood oxygen and body temperature. Health records are forwarded to the healthcare team for further monitoring and treatment planning after the patients' profile is updated. In addition, ARM provides a chat channel through which healthcare teams and patients would be able to communicate in real time.

ARM has the capacity of providing mobile healthcare services. By this, healthcare teams would be able to remotely monitor patients, communicate to observe basic disease signs and plan the required treatments. During the pandemic peak, it is repeatedly announced by the health organisations that people should avoid visiting clinic for simple/basic diseases such as flu. Patients are asked to wear masks if they urgently need to visit their GPs. Although wearing masks could slow down the spreading speed of the virus, this could not be assumed as a long-term solution to manage the virus outbreak. Healthcare technologies (e.g. M-health) are promising approaches to solve this problem as they can reduce physical contact. Moreover, they are able to conserve resources -mainly time and expense [6]. For example, it roughly takes 30 min to collect temperature, blood pressure, heart rate, height and weight information for each in-patient visit. Long queues and increased waiting time are unbearable especially when the healthcare services are restricted (e.g. during COVID-19 pandemic) and/or patients are under pain (e.g. stomachache). ARM tackles these issues by allowing healthcare teams to remotely collect patients' basic health data.

This paper is organised as follows: Sect. 2 reviews literatures and outlines M-health applications and technologies. Section 3 introduces ARM and explains design, implementation features and techniques. Section 4 discusses testing results to highlight this mobile application's achievements. Section 5 concludes ARM benefits and outlines the key points of future work.

2 Related Work

There are a number of M-health applications in the market providing users healthcare data collection, communication and monitoring services. However, they still suffer from key drawbacks -mainly online communication and lack of real-time body data collection and analysis. Yet, some of the mobile applications are dedicatedly designed for specific diseases or particular applications.

LabVIEW [7] proposes a healthcare mobile application that measures user's physical data by using wearable sensors to monitor health fitness. In addition, this system alarms users via text/email in the case of health risk or hazard. However, this provides users no communication, feedback or interaction to approach healthcare teams.

[8] introduces a mobile healthcare application dedicatedly designed for diabetes patients. This platform is interconnected to Electronic Medical Record (EMR) system to report and collect patients' records. It analyses the latest patient's blood sugar records which are reported by EMR and offers the patients treatment plans.

BiliCam [9] proposes a mobile application through which newborn jaundice can be detected and reported. For this, smartphone's camera scans infant's skin and the image result is compared with a colour sample card. It extracts the skin colour features and detects how close the skin colour is to the samples, which diagnoses the level of jaundice. Yet, a machine learning technique is used to train datasets and optimise the results. HemaApp [10] is another mobile application that utilises smartphone's camera to measure haemoglobin levels using a similar technique.

[11] focuses on a mobile application that collects and reports behaviours of patients with Schizophrenia. For this, smartphone's accelerometer, microphone and global positioning system are used to collect patients' location, activity and speech during daily life. This provides patients fitted feedback according to their collected behaviour.

Seismo [12] presents a smartphone application through which blood pressure is collected and monitored. For this, smartphone accelerometer and camera report the heart vibration and pulse from the user's fingerprint. By this, the user is able to monitor blood pressure if any treatment is required.

[13] proposes a mobile application design and implementation that is used to monitor physical activity for patients with chronic heart-failure in real time. It utilises smart-phone's embedded sensors to collect body data and forwards them for an off-line analysis.

ClinTouch [14] focuses on a M-health mobile application which is designed for mental issues and diseases. This is implemented and evaluated as a pilot to provide symptom tracker, diary function and appointment reminder services. However, it matches no National Health Service (NHS) principals due to the lack of solid safety and security.

[15] proposes a mobile application which provides doctor-patient real-time communication. However, it supports no data collection of patients and remote healthcare monitoring.

ARM aims to combine the key healthcare features to propose a mobile application which is able to manage healthcare records and treatment plans, collect body data with minimum physical contact and provide real-time communication with doctors. According to the literature, it is learned that existing mobile applications lack such combination to provide healthcare services if real-time communication, on-demand data collection and online monitoring are required.

3 The Proposed Approach

This section introduces a healthcare mobile application that allows patients and healthcare teams are kept interconnected. By this, body data, such as body temperature, blood oxygen and heart rate is collected and reported to healthcare teams for further analysis and utilisation. This M-health mobile application is able to automatically report patients' health status using a sensory platform and online questionnaire. In addition, this allows patients and healthcare teams to manage their profiles and provides them personalised healthcare applications. Yet, this allows the healthcare stakeholders to communicate and supports real-time feedback through an interactive social environment.

Android Studio (3.5.3) is used to implement the mobile application platform as this supports portability and utility. Android is a well-known open-source, flexible and adaptable programming platform that is widely used to implement mobile applications [16]. Yet, this version of android studio supports several permission protocols including internet and WRITE_EXTERNAL_STORAGE that are usually used to provide security and user privacy. This allows users to access information once required permissions are granted.

ARM is implemented by Java to support portability and cross-functionality on desktops, mobiles and/or embedded systems. Java is object-oriented, therefore codes are easy to extend and fix. Java supports multithreading and network computation and provides multimedia. Besides, this enhances project development by separating the characteristics using object-oriented programming.

The proposed system is designed as three components including sensory module, core application and database. The Sensory module stays on the duty of body area data collection including blood oxygen, heart rate and temperature. The Core application is responsible for data analysis. This utilises body data from the Bluetooth-connected sensory module to offer healthcare functions. If some body data is out of the normal range, a message with automatic diagnosis and abnormal data will be sent to both the user and the healthcare team. Database component manages the collected data to response to either user or healthcare team enquires. Figure 1 depicts ARM conceptual diagram.

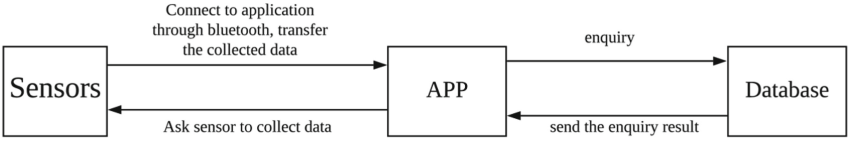

Fig. 1. ARM system architecture.

ARM's Core Application. The core application is designed and set-up for both stakeholders including doctor and patient. As Fig. 2 shows, users need to register and fill out a questionnaire to access the mobile application. There are three major functions: HOME, CHAT and MY page. HOME allows patients to make a health appointment, visit a doctor online, observe health report and monitor the treatment process. Healthcare teams use HOME to check patients' information and give them treatment/advice. CHAT is also available for both patients and healthcare teams when they need to communicate in real-time by using text messages. Yet, MY page allows the users to update profiles, set-up sensory communication through Bluetooth, view history records and log out of the system.

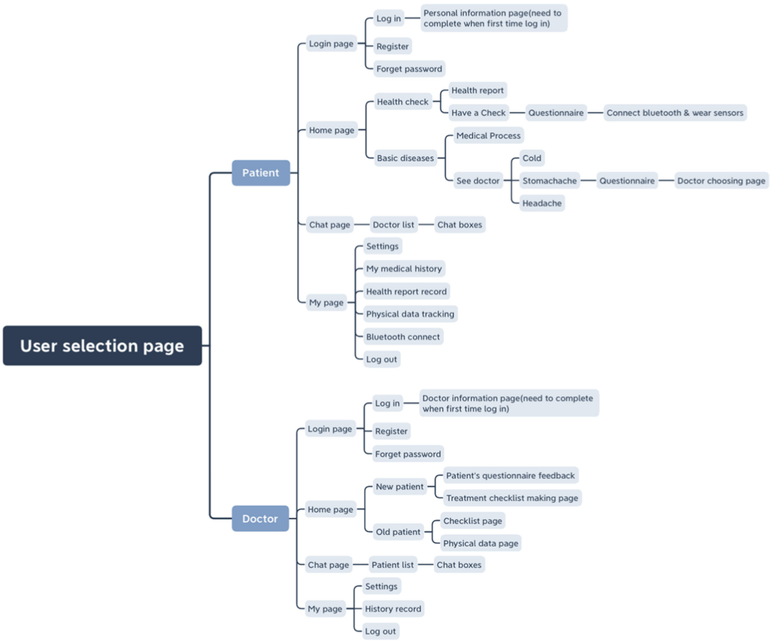

Fig. 2. ARM's Core Application Chart.

According to Fig. 3, ARM's sequence diagram outlines how the users interact with the system from a patient's perspective. By this, first, the patient chooses a disease type and waits until the system returns a questionnaire focusing on the disease to collect data. ARM provides a list of registered doctors with their availability hours/days once the patient submits the questionnaire. This forwards the questionnaire data to the doctor as this is selected. The corresponding doctor makes a treatment checklist and sends this back to the patient to collect the required information -mainly body area data such as temperature, blood oxygen and heart rate. Yet, the patient and doctor are kept interconnected to monitor the treatment procedure until good results are achieved.

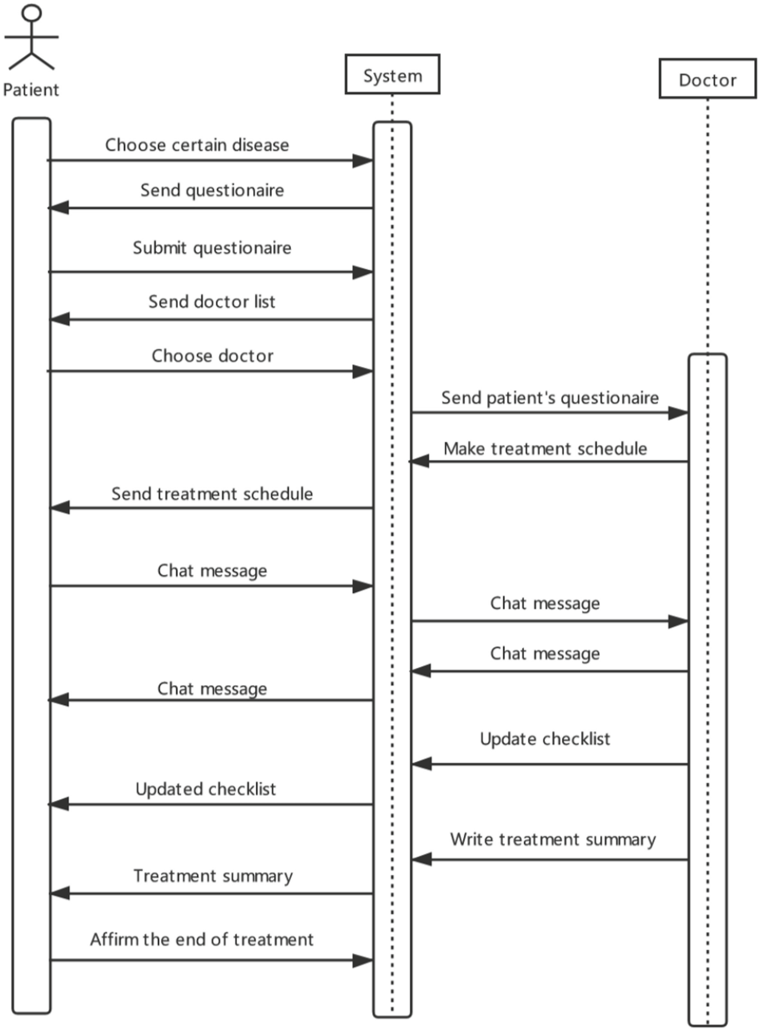

Fig. 3. ARM's sequence diagram.

ARM's Sensory Module. ARM's sensory module is set-up to collect and forward body area data. As Fig. 4 shows, this is comprised of three parts: sensors, Arduino mini controller and Bluetooth. MH-ET LIVE_MAX30102 Arduino chip [17] is used for body area data collection. This integrates a body temperature sensor, heart rate sensor and blood oxygen sensor. This is a small and easy to set- up sensor with a size of (2 cm × 1 cm × 1 mm). Arduino board MEGA 2560 [18] is a micro-controller that is interconnected to the MAX30102 sensory device. This is able to process data -mainly in-aggregation and transmit information through various communication technologies, such as Bluetooth, USB, GSM and I2C. ARM utilises Bluetooth to provide a communication tie between MEGA 2560 and MAX30102. The reason to choose this technology is that Bluetooth reduces costs as compared to infrastructure-based networks -mainly GSM. MEGA 2560 board is programmed using Arduino studio in C programming language. Arduino studio provides many useful functions and libraries, such as serial port monitoring and drawing.

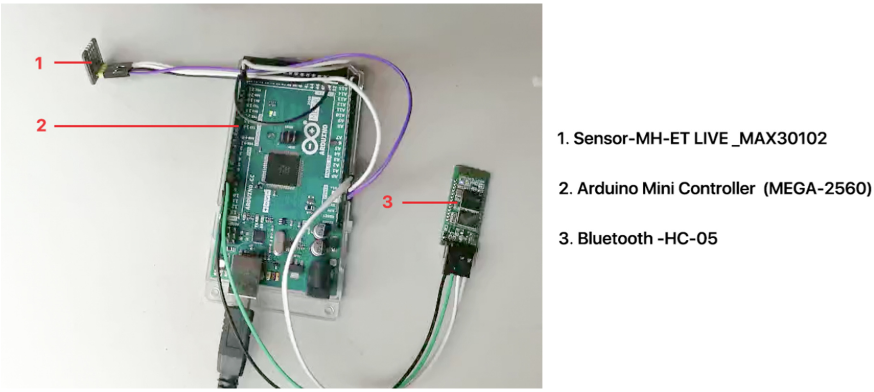

Fig. 4. ARM's sensory module.

ARM's Database Design. ARM database conceptual diagram is depicted as Fig. 5. According to this, DoctorAccount, DoctorBasicInformation, PatientAccount, PatientBasicInformation are the key database tables that store the user information for both doctors and patients. Each user is provided by a particular user account (DoctorAccount and PatientAccount) which updates user profile information. Yet, doctors' and patients' basic information including name, speciality, degree, age and gender is stored in DoctorBasicInformation and PatientBasicInformation respectively. In addition, doctor-patient communication information such as date and title are recorded at the ChattingMessage table. This would help the healthcare teams to re-consider past appointments and patients' healthcare records. Patient_Questionnaire keeps the questionnaire information which is provided by the patients regarding the health issue. This returns the patient's health self-report which can be used by doctors to figure out the health issue and/or disease. DoctorTimetable provides the doctor availability time-slots and is used to show the list of available appointments.

HealthCheckRecord table records the results of health checkups and/or treatments according to the appointment given by AppointmentList. Then, a checklist (CheckList table) is provided for patient patients to update their healthcare records accordingly. FinishAppointment is used to record the appointment timing values and keep the start and end times of the treatment.

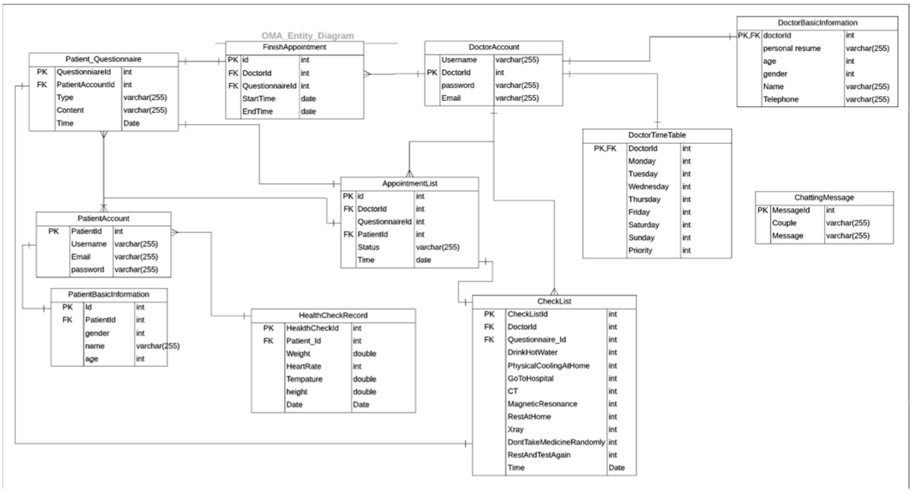

Fig. 5. ARM's Database design.

4 Results and Discussions

The proposed android application aims to provide remote monitoring/treatment, patient/doctor interconnection and body area data collection. Figure 6 shows ARM's user interface. To evaluate ARM's performance, comprehensive testing is conducted for functional, system, front-end, back-end and hardware testing.

Functional Testing. Functional testing aims to check whether ARM is able to perform the designed operations or not. However, ARM is not tested by real patients and doctors. Non-professional users (e.g., students) used this app to test its functionality. It can be addressed as future work to test this app by professional users such as healthcare teams.

Functional testing is conducted to test the following operations as Table 1. According to the test plan and achieved results, the following responses are recorded for each particular test operation/scenario.

System Testing. System testing focuses on system recovery which aims to check how ARM is able to recover in the case of any crash or hardware failure. For this, a hardware crash is simulated through a shut-down mechanism using which ARM server is suddenly re-started and the database is re-built. This evaluates ARM performance and database recovery in the case of a server crash. A registration operation is used to test the system's functionality. By this, the database should be able to successfully retrieve the registration information if the system is re-started after a sudden shut down (or crash).

To evaluate system scalability, ARM functionality is tested when multiple users simultaneously log in. For this, a test scenario is designed through which

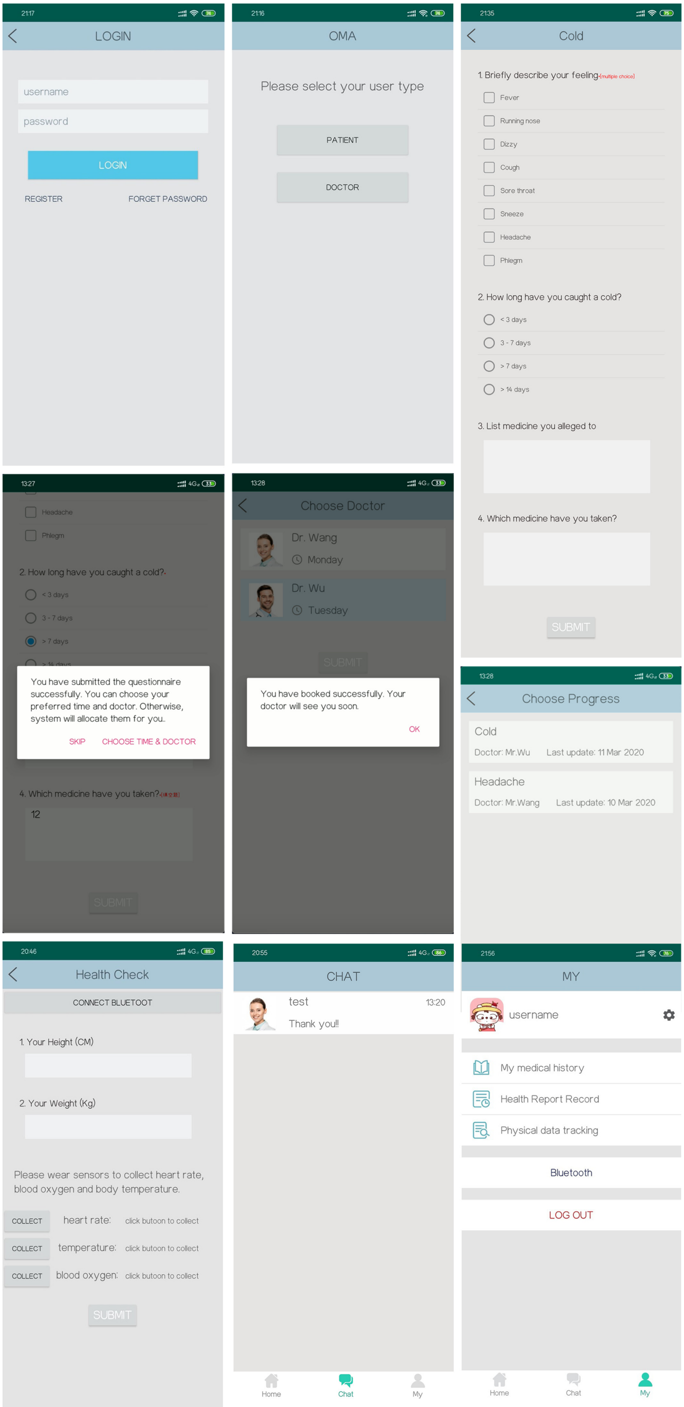
LOGIN
username
password
LOGIN
REGISTER
FORGET PASSWORD
OMA
Please select your user type
PATIENT
DOCTOR
Cold
1. Briefly describe your feeling
Fever
Running nose
Dizzy
Cough
Sore throat
Sneeze
Headache
Phlegm
2. How long have you caught a cold?
< 3 days
3 - 7 days
> 7 days
> 14 days
3. List medicine you alleged to
4. Which medicine have you taken?
SUBMIT
You have submitted the questionnaire successfully. You can choose your preferred time and doctor. Otherwise, system will allocate them for you.
SKIP
CHOOSE TIME & DOCTOR
12
Choose Doctor
Dr. Wang
Monday
Dr. Wu
Tuesday
You have booked successfully. Your doctor will see you soon.
OK
Choose Progress
Cold
Doctor: Mr.Wu
Last update: 11 Mar 2020
Headache
Doctor: Mr.Wang
Last update: 10 Mar 2020
Health Check
CONNECT BLUETOOT
1. Your Height (CM)
2. Your Weight (Kg)
Please wear sensors to collect heart rate, blood oxygen and body temperature.
COLLECT
heart rate:
click butoon to collect
temperature:
blood oxygen:
CHAT
test
13:20
Thank you!
Home
Chat
My
MY
username
My medical history
Health Report Record
Physical data tracking
Bluetooth
LOG OUT

Fig. 6. ARM's UI.

Table 1. Functional test plan and results

Test operation/scenario	ARM response
Register	Verification code is sent
Log in 1) registered 2) unregistered	1) User's profile appears 2) Ask to register
View medical progress	Medical records are displayed
Patient chooses disease (e.g. cold)	Forward to relevant questionnaire
Filling out the questionnaire 1) Completed 2) Uncompleted	1) Questionnaire submitted 2) Error message
Choosing doctor 1) Automatic allocation 2) Manual allocation	1) Free doctor is automatically allocated 2) the selected doctor is allocated
Health history	Health profile
Have a check	Health check page
Bluetooth connection	Connect to Bluetooth
Collect body data	Sensor measures and reports
Tracking	View health records
Update personal information: 1) True information 2) Wrong information	1) Successful updated 2) Error
Doctor visits patients 1) New 2) Existing	1) New profile 2) Past records
Doctor monitors patient	Updated patient's records
Doctor provides feedback	Feedback form filled out
Doctor edits questionnaire/checklist as required	Questionnaire updated
Real-time communication	Private text-chat room

60 test users try to work with ARM at the same time for 10 min as a simple check-up. During the testing period, no obvious delay and system failure was detected and all operations went on smoothly.

Front-End Testing. Front-end testing focuses on evaluating the Graphical user interface, functionality and application usability. The test results are outlined as below:

- All designed buttons work.
- All designed pages are correctly interconnected.
- Layouts appear the same as what has been designed.
- User receives alarm "Please enter correct email address" if the email address is not valid.
- User receives alarm "Username and password cannot be empty" if any field of the user account is left empty.

- User receives alarm ‘Passwords are inconsistent’ if ‘Password’ and ‘Confirm Password’ are different.
- User cannot enter letters in fields that only accept numbers like ‘Height’ and ‘Weight’.

Back-End Testing. Back-end testing examines whether the system is able to correctly respond the user requests or not. This is conducted as follows: one test user, android phone ARM, the corresponding sensors and the database server. By this, the user submits a number of new transactions to modify database records. The objective is to study how the ARM database is dealt with the new transactions according to the system requirements. Table 2 summarises back-end results for each operation which is designed according to the testing scenario.

Table 2. Back-end testing results

Test operation	ARM response	Database response
Mismatched username and password	Error	Username and password not exists
Update profile	Update message	Database record is updated
Submit questionnaire	Submission message	Database record is updated
Wrong checklist/question provided	Error	No database update

Hardware Testing. Hardware testing evaluates system functionality, sensory device stability, system/sensor compatibility and measurement error rate. Hardware testing results are outlined as below:

System Functionality

- If the users frequently move their fingers or wrongly use the sensor, data collection fails or returns inaccurate results.

Sensory Device Stability

- Particular factors (e.g. weather and skin sweat) may address wrong results.
- Users may lose their patience while data collection as this requires to have 10 s finger-touch to collect multiple sensory data.

System/sensor Compatibility

- ARM bluetooth connection may fail if any other application is used during the Bluetooth connection.
- ARM can’t read sensory data if Bluetooth is not successfully interconnected.

Table 3. The comparison of ARM with the literatures

Features	ARM	LabVIEW [7]	[8]	BiliCam [9]	HemaApp [10]	[11]	Seismo [12]	ClinTouch [14]	[15]
Real-time health data collection/analysis	Y	Y	N	Y	Y	Y	Y	Y	N
Body data collection	Y	Y	N	Y	Y	Y	Y	N	N
(Offline) Communication	Y	Y	Y	Y	Y	Y	Y	Y	N
Realtime chat	Y	N	N	Y	Y	Y	N	Y	Y
User profile	Y	N	Y	N	N	N	Y	Y	Y
Healthcare Service management	Y	N	Y	N	N	N	Y	Y	Y
Multi-functionality	Y	Y	N	N	N	Y	Y	Y	Y

Measurement Error Rate

– The accuracy of the measurements is about 92%. This calculates as 55 out of 60 test users correctly report the sensory data.

5 Conclusion and Further Work

ARM is comprised of three key modules questionnaire, chatting environment, and sensory design. The questionnaire are designed to collect health self-report from the patients. For this, the questions are designed according to the user health issues and data requirement. For example, a patient who suffers from stomach should be provided a questionnaire which asks for stomach symptoms and required data collection. For this, doctors, specialists and healthcare teams would select/design the required questions from a questionnaire checklist to collect the required information from the patients. Indeed, the designed questionnaires should be able to capture key information of patients' symptoms and health issues. Chatting module is used to make a real-time communication between the healthcare team and patients. This allows both user-sides to share additional required data and/or discuss about the treatment plan. This allows the patients to report their uncommon symptoms or disease signs which are not covered by the provided questionnaire. Sensor design is used to collect body area data. This allows doctors to remotely collect patients body data with no physical interaction. This allows doctors to use accurate and realtime data to investigate the health issue and plan the required treatment.

ARM provides online health monitoring that offers a number of benefits to enhance the quality of healthcare services. This aims to keep patients and healthcare teams interconnected and collect and analyse physiological data in real-time. ARM's remote and real-time body data collection/analysis would result in reduction of healthcare monitoring cost and risk especially during COVID-19 pandemic. ARM is multi-functional which has the potential to be utilised for various healthcare scenarios and applications. This provides real-time links for healthcare stockholders (doctors and patients) to communicate through. Yet, ARM allows the users to create and update their profiles. Patients' profiles are linked to the healthcare records which can be accessed by the doctors during treatment procedures, whereas doctor's profiles provide their updated experiences, availabilities and specialities. Moreover, ARM allows healthcare teams to manage/schedule healthcare services and inform the patients online. Table 3 summarises the advantages of ARM as compared with the literatures.

To evaluate ARM performance, five test plans are conducted: functional, system, front-end, back-end, and hardware testing. The results of functional testing shows that ARM is able to manage the user transactions in general. System testing results show that ARM is scalable and able to manage multiple simultaneous. Moreover, this has the capacity of full system recovery in the case of sudden hardware crash. Front-end testing supports that ARM user interface is fully implemented and correctly work. Back-end testing shows that ARM is able to manage data records and correctly forward and reply database transactions and queries. Hardware testing results study the influence of hardware factors on sensory data collection and communication. This shows that the accuracy and reliability of collected body data, disease recognition and treatment plan is highly dependent on the hardware factors.

There are three issues that can be addressed as further work. First, to enhance the functionality and operation of ARM's chat environment. The chat environment supports only text communication which is comparatively restricted. For this, providing audio/video communication between the doctors and patients would be a new feature which allows them to communicate easier and more convenience. Second, a chatbot can be designed to improve ARM functionality and user-friendly. This chatbot should be able to automatically recognise the user's questions and provide them relevant answer and/or help/signpost them to find the required information. Text mining approaches should be used to recognise the keywords and link them into the answer pools. Yet, machine learning techniques are required to continuously update the answers based on users' feedback. According to [19], the chatbot can be fed by key healthcare information centres and/or key references to provide best-fitted answers based on the user's questions. With the help of collected sensor data and the context of user input, the implementation of machining learning algorithm and clinical decision-making helps users obtain their expected answers quickly. If the patients are satisfied with the automatically generated answers, they could choose to stop the service thus save both patients and doctors time. Third, this is required to consult with COVID-19 pandemic specialists to re-design this application dedicatedly for COVID-19. During COVID-19 pandemic, this is very risky for patients to visit hospitals and clinics in person [20]. For this, ARM would be able to offer a number of benefits such as patients distance monitoring, remote treatment which reduces the outbreak and conserve the resources. Indeed, ARM would be able to recognise COVID-19 symptoms (e.g. such as cough, fatigue and diarrhoea) and provide required treatment plans with minimum physical contact between doctors and patients.

References

1. Mariakakis, A., Wang, E., Goel, M., Patel, S.: Challenges in realizing smartphone-based health sensing. IEEE Pervasive Comput. **18**(2), 76–84 (2019)
2. World Health Organization: mhealth: new horizons for health through mobile technologies: based on the findings of the second global survey on ehealth (2019). https://www.who.int/goe/publications/goemhealthweb.pdf. Accessed Jan 2020
3. Banga, B.: Global mobile medical apps market to reach usd11.22 billion by 2025, reports BIS research (2020). https://www.prnewswire.com/news-releases/. Accessed Aug 2020
4. Cosco, T.D., Firth, J., Vahia, I., Sixsmith, A., Torous, J.: Mobilizing mhealth data collection in older adults: challenges and opportunities. JMIR Aging **2**(1), 1–8 (2019)
5. Elflein, J.: Hospital care expenditure in the united states from 1960 to 2020 (2020). https://www.statista.com/statistics/184772/us-hospital-care-expenditures-since-1960/. Accessed July 2020
6. Hanrahan, C., Aungst, T.D., Cole, S.: Evaluating mobile medical applications. American Society of Health-System Pharmacists Inc., Technical report (2014)
7. Abdullah, A., Ismael, A., Rashid, A., Abou-ElNour, A., Tarique, M.: Real time wireless health monitoring application using mobile devices. Int. J. Comput. Netw. Commun. **7**(3), 13–30 (2015)
8. Jung, E.Y., Kim, J., Chung, K.Y., Park, D.K.: Mobile healthcare application with EMR interoperability for diabetes patients. Clust. Comput. **17**(3), 871–880 (2013)
9. de Greef, L., et al.: BiliCam: using mobile phones to monitor newborn jaundice. In: ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2014), Seattle, Washington, US, 13–17 September 2014 (2014)
10. Wang, E.J., Zhu, W.L.J., Rana, R., Patel, S.N.: Noninvasive hemoglobin measurement using unmodified smartphone camera and white flash. In: 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Seogwipo, South Korea, 11–15 July 2017 (2017)
11. Ben-Zeev, D., et al.: Mobile behavioral sensing for outpatients and inpatients with schizophrenia. Psychiatr. Serv. (Washington, D.C.) **67**(5), 557–561 (2016)
12. Wang, E.J., et al.: Seismo: blood pressure monitoring using built-in smartphone accelerometer and camera. In: ACM Conference on Human-Computer Interaction (CHI 2018), Montreal, QC, Canada, 21–26 April 2018 (2018)
13. Aranki, D., Kurillo, G., Yan, P., Liebovitz, D.M., Bajcsy, R.: Real-time telemonitoring of patients with chronic heart-failure using a smartphone: lessons learned. IEEE Trans. Affect. Comput. **7**(3), 206–219 (2016)
14. Hollis, C., et al.: Technological innovations in mental healthcare: harnessing the digital revolution. Br. J. Psychiatry **206**(4), 263–265 (2015)
15. Nwabueze, E.E., Oju, O.: Using mobile application to improve doctor-patient interaction in healthcare delivery system. E-Health Telecommun. Syst. Netw. **8**(3), 23–34 (2019)
16. Shah, K.: How benefits of android app can help businesses (2019). https://www.rishabhsoft.com/blog/5-advantages-of-android-app-development-for-your-business. Accessed May 2020

17. Maxim Integrated: High-sensitivity pulse oximeter and heart-rate sensor for wearable health (2020). https://www.maximintegrated.com/en/products/interface/sensor-interface/MAX30102.html. Accessed Jan 2020
18. Arduino: Arduino mega 2560 and genuino mega 2560 (2020). https://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardMega2560. Accessed Jan 2020
19. Divya, S., Indumathi, V., Ishwarya, S., Priyasankari, M., Devi, S.K.: A self-diagnosis medical chatbot using artificial intelligence. J. Web Dev. Web Design. **3**(1), 1–7 (2018)
20. Centers for Disease Control and Prevention: People with certain medical conditions (2020). https://www.cdc.gov/coronavirus/2019-ncov/need-extra-precautions/people-with-medical-conditions.html. Accessed Sept 2020