Proc. of the International Conference on Electrical, Computer and Energy Technologies (ICECET)
9-10 December 2021, Cape Town-South Africa

Black Box Testing with Exploratory Approach of a Software for Remote Monitoring of Patients with COVID-19 and Other Infectious Diseases

2021 International Conference on Electrical, Computer and Energy Technologies (ICECET) | 978-1-6654-4231-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICECET52533.2021.9698734

Mag. Kocho Hrisafov
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
Sofia, Bulgaria
khrisafov@hotmail.com

Mag. Eng. Angel Botev Ivanov
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
Sofia, Bulgaria
angel.b.ivanov@gmail.com

Assoc. Prof. Nayden Chivarov
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
Sofia, Bulgaria
nchivarov@gmail.com

Mag. Eng. Stefan Chivarov
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
Sofia, Bulgaria
schivarov@gmail.com

***Abstract*— Covid-19 pandemic has set the world on fire. It has impacted every aspect of our society. One of the worst affected parts is the countries' health systems. Our goal is to provide a proof of concept for cost effective telemedicine system which can be used by hospitals for monitoring of thousands of patients at once, thus medical personnel exposure to the virus is minimized. In this paper we aim to put this system through the best practices of software testing so we can determine if it achieves its primary functions, keeping in line with the general idea of cost effectiveness. Black box testing with the exploratory approach was used for functionality and feature testing as it gives the most accurate real-world results. During the tests, several bugs were found and fixed. Because of the fact, that the testers were actual medical personnel, we have received valuable information on how to improve the quality of the product, so it can be even more ease to use and provide the necessary robustness and data visualization.**

***Keywords*— telemedicine, software testing, black-box testing, exploratory testing, digital stethoscope, Web GUI**

I. Introduction

The ongoing Covid-19 pandemics has severe impact in the world. As it is extremely contagious, it spread quickly through the world. Human lives were lost, healthcare was unable to cope with the influx of patients, jobs were lost as a result of the lockdowns, countries' economies shaken. In order to preserve the lives on the front line, the use of telemedicine rocketed [1].

The telemedicine [2] is not a new field in science. However, in the time of this pandemic situation it has proven to provide crucial impact on saving lives both for patients and healthcare professionals. For our team, to be able to assist the medical personnel across the globe, it was crucial to design and implement a telemedical system that will have the ease of use of the commercial grade systems and be available to as many people as possible even in the developing countries.

Our goal was to create a proof of concept for cost effective, yet fully functional telemedical system. The target users of this system would be low budget hospitals and developing countries. The system would limit the exposure of medical practitioners to patients as they will monitor patients' vitals remotely. Only open-source tools would be used and cost-effective hardware.

With all those requirements in mind, we developed a proof-of-concept tele-medical system for remote monitoring of patients with Covid-19 and other infectious diseases.

The goal of this paper is to prove that developed telemedicine system performs well and accomplishes its primary function when used by non-technical users and medical personnel. We will minimize the cost of the testing using approach which does not require expensive software testing tools. Medical students were asked to test the software – so they can judge the software functionality from a medical perspective.

II. Brief Description of the Tele-Medical System

One of the classic tools used for monitoring lungs and heart state is the stethoscope. The use of the stethoscope is one of the simplest nonintrusive methods to measure the Respiratory Rate, which is also an important vital sign for coronavirus monitoring. As described in [3], the stethoscope is an acoustic device that transmits the sounds from the chest piece through an air-filled hollow tube to the listener's ears.

In the digitalization era, the stethoscope has received an updated digital version. The digital stethoscope usually uses piezoelectric sensor and a microphone and can transform the acoustic sound to an electrical signal, which can be further amplified for optimal listening. The main advantage of the digital stethoscope is its ability to be a network device and to be connected via various network protocols to other devices and/or control center. The current versions of digital stethoscopes offer connectivity through Bluetooth (BLE) and ZigBee.

So far it became clear, that for respiratory rate monitoring and cough detection to be easy and non-intrusive for multipatient 24/7 monitoring, the digital stethoscope is the better choice. This is what we are going to use in our concept design.

A digital cost-oriented stethoscope from the brand Eko [4] was used. It has Bluetooth LE [5] module which uploads data to Raspberry PI which acts as a BLE collector. Linux operating system is installed on the Raspberry PI.

The control center of the telemonitoring solution will consist of easy-to-use GUI [6]. frontend, adapted for non-technical medical personnel and backend which collects data from connected IoT devices and stores them into a SQL database. The whole solution will be developed in .NET C# in client-server architecture.

Client-server architecture, architecture of a computer network in which many clients request and receive service from a centralized server. Clients are often situated at workstations or on personal computers, while servers are located elsewhere on the network, usually on more powerful machines. In hospital data processing, a client computer can be running an application program for entering patient information while the server computer is running another program that manages the database in which the information is permanently stored. Many clients can access the server's information simultaneously, and, at the same time, a client computer can perform other tasks, such as sending e-mail [7].

.NET is a free, cross-platform, open-source developer platform for building many different types of applications. It supports multiple languages, editors, and libraries to build for web, mobile, desktop, games, and IoT .NET apps can be coded in C#, F#, or Visual Basic [8].

SQL Server is a relational database management system, or RDBMS, developed and marketed by Microsoft. Like other RDBMS software, SQL Server is built on top of SQL, a standard programming language for interacting with the relational databases. SQL server is tied to Transact-SQL, or T-SQL, the Microsoft's implementation of SQL that adds a set of proprietary programming constructs [9].

Mosquitto [10] message broker installed on the Linux converts the .wav files produced by the digital stethoscope into a binary stream and loads them into a SQL Server Express database [11]. The database schema can be seen in Fig 1.

The Web GUI frontend is built in Microsoft Visual Studio Community Edition 2019 using ASP.NET C# [12], [13].

The database and the frontend can be hosted either on premise, in the cloud or as a hybrid solution depending of the existing hospital IT infrastructure – so costs can be minimized.

The frontend is WEB based so it can be accessed from different devices (computers, tablets etc.) without the need to install separate client software. Compatibility problems are minimized (OS independent) – it requires just a web browser.

The frontend is consisted of several list and entry forms. Fig. 2, Fig. 3, Fig 4 shows several of them.

The central part of the system is a form where patients' vitals are displayed in real-time as the database is updated from the message broker. The form is shown on Fig.5

The data in all forms can be sorted, filtered and exported to Excel.

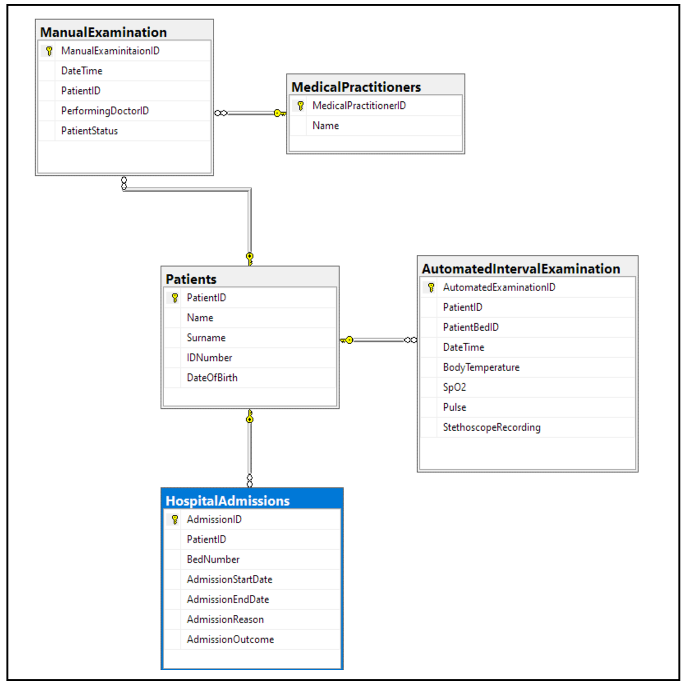

Fig. 1. Database schema

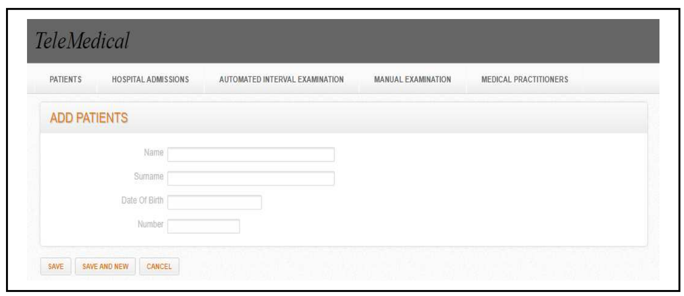

Fig. 2. Patient's entry form

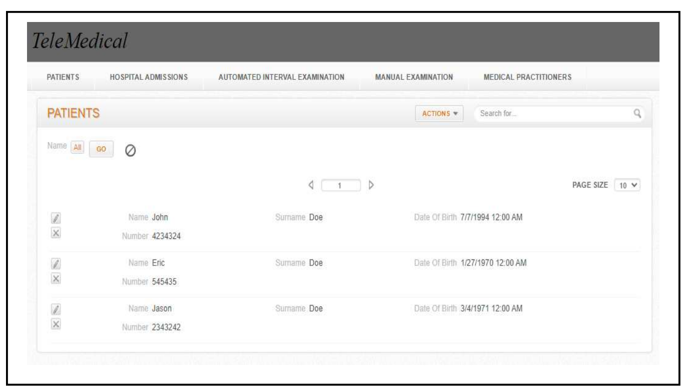

Fig. 3. Patient's list

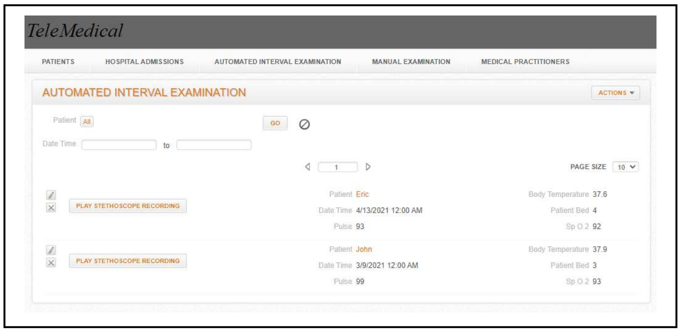

Fig. 4. Hospitalization admissions list

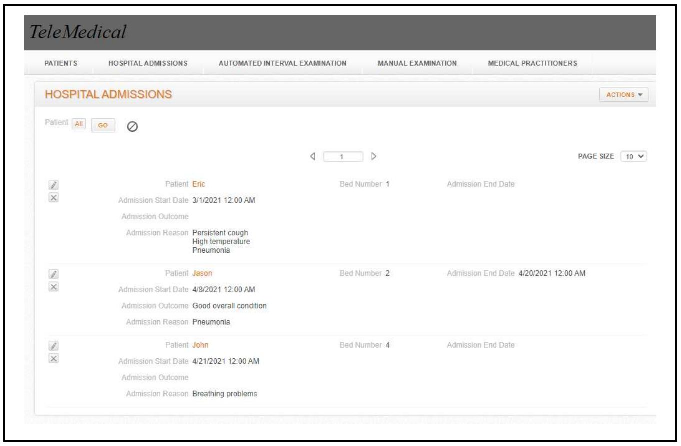

Fig. 5. Automated interval examination screen

The data in all forms can be sorted, filtered and exported to Excel.

We are working to extend the functionality with adding support for pulse-oximeter devices and thermometers with BLE support.

Most of commercially available pulse oximeters are attached on the finger. It monitored two vital signs - heart rate (pulse) and the concentration of Oxygen in the blood (SpO2). Both vital signs can be easily monitored by the process called pulse oximetry.

The principle of pulse oximetry is based on dual-wavelength illumination of arterial blood. This results in an absorption contrast that depends upon the proportion of hemoglobin that is chemically combined with oxygen. The color of blood varies depending on the oxygen content and the hemoglobin molecules reflect more red light when they are oxygenated. The reflection of infra-red light increases with de-oxygenated hemoglobin molecules.

More details can be found at the article "Application of IoT for telemonitoring patients diagnosed with coronavirus COVID-19" [14].

III. Test Data

Doctor - Head of the Clinic of Lung Diseases and her medical student from the Military Medical Academy in Sofia are using the digital stethoscope in addition to established diagnostic methods. Patients were informed and their consent taken. Related anonymized patient data, information about the course of their sickness and health parameters were provided in an Excel sheet. Audio recordings from the digital stethoscope were also included. The provided data would be used by the testers in the testing process.

IV. Testing Methodology

There are several different testing methodologies [15]. Some of them use specialized testing tools, other use scripted tests.

We decided to use black box testing approach [16] as we need to assess functional level of the application. The concept of this approach is that test users are un-aware of the inner workings of the application. They just provide input to application, observe the output and comment their findings.

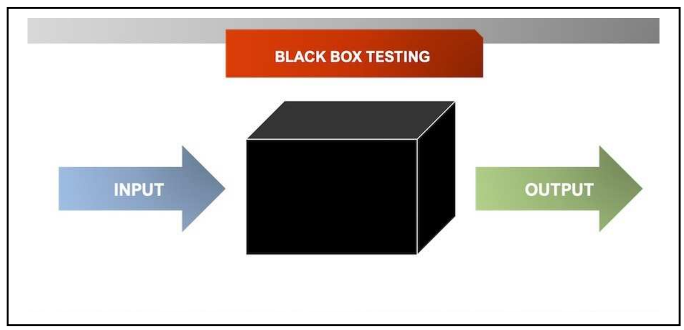

Fig. 6. Black box testing diagram [16]

The current trend is to automate testing process by using pre-defined actions, but automation has its limits as seen in Fig 7.

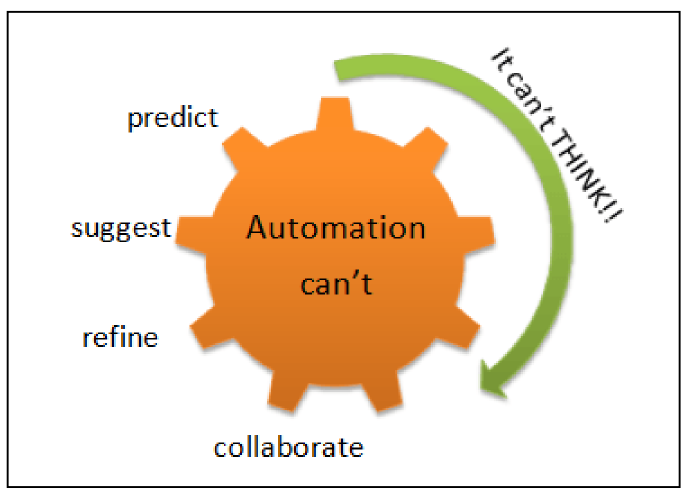

Fig. 7. Limits of automated testing [17]

There are several approaches of black box testing. We chose exploratory testing [17] because we thought that it will be best suited to our case:

- It doesn't require specialized testing software – costs are minimized
- Testers are just given brief overview of the software
- They are not trained extensively into the software usage
- There are no structured tests.
- They are left to explore all aspects of the software freely
- The tests they do rely on their creativity and understating of the function of the software

Usually, exploratory testing is performed by experienced testers. But our goal was software to be tested by non-technical users – medical personnel. The software would be tested by 3 medical students helping in a hospital. In that way we can achieve two points:

- cost savings not engaging paid testers
- functional assessment of the software by its target users

There are several well-known drawbacks of exploratory testing [18]:

- This testing depends on the tester skills
- Limited by domain knowledge of the tester
- Not suitable for long execution time

In our case we think that these drawbacks are countered:

- The medical student's experience using web sites and applications is at acceptable level so they should be able to notice bugs
- Their domain of expertise is medicine so they are not limited like ordinary testers would be
- The aim of the testing phase is to be short so it can be cost effective

V. Testing Envirovment and Process

The software was hosted on a cloud hosting provider, so it can be accessed from every device and place. Collected test data was shared with the testers. They were asked to use it when testing the application.

A short demonstration of the software was provided. We asked the testers to fully explore and "play" with the software for 5 days.

Testers weren't asked to document the performed tests as it contradicts with exploratory testing and is time consuming – it may take more time than actual tests.

Provided was an excel file to the testers so they can document found bugs. We asked to classify the severity of the bugs according to their understanding (high, mid, low), the location where the bug is found and the steps to reproduce it.

At the end of the fifth day testers were asked to fill in a questionary with the following content:

- Does the software fulfill its functions?
- Areas for improvement
- Additional notes

VI. Preliminary Results and Analysis

The results of the testing were collected from the testers. Bugs were summarized and grouped by priorities. They are shown in Table 1

TABLE I. PRELIMINARY RESULTS

Results	**Number of Bugs by classification**	
	Classification	***Number of bugs***
	High	3
	Medium	5
	Low	13

The detected bugs were very unexpected to our team. During our time spent developing and working with the system we didn't encounter them. This validates using exploratory testing approach and non-technical personnel as a tester. Their way of thinking and usage patterns provided unique perspective which resulted in improvement of the software after fixing the bugs.

As for the results of the questionary:

- 2 of 3 students answered that it fulfilled its primary function
- They all pointed areas where the software can be improved
- In the additional notes some of them provided ideas how the system can be extended with more functionalities

VII. Limitations of the Testing

We are aware of several limitations of the performed software testing:

- Small number of testers – In order to do more thorough manual testing, more people are needed, as everyone has its own viewpoints.
- Not using professional testers – Since the testers were not qualified QA engineers, it is possible that not all possible test cases were checked.
- Testing the software using just one method – There's no such thing as perfect software. Therefore, multiple testing methods should be used to find the most bugs.
- Test were done by medical students, not doctors – Although the medical students were highly educated in the medical field, the doctor's knowledge and experience might provide even more useful insights, bugs or feature requests.

VIII. CONCLUSION

The whole testing process was very cost effective – in sync with the idea to develop system for lower income hospitals and countries.

Although there were limitations to the performed testing of the system, the results it provided were very valuable. We can summarize them in few points:

- Bugs were fixed
- Several of the application forms were improved
- The testers recommended interesting functionalities which will be added in a future version of a software

Generally, the testers considered that the system fulfilled its purpose.

Our plan is to improve the testing process for the next major version of the system which will add pulse-oximeter devices and thermometers with BLE support [19]:

- Engage doctors in testing
- Assign new testing method to medical students
- Include technical students in the testing with open-source testing software

ACKNOWLEDGMENT

This research was carried out as part of the bilateral BAS-SAS project: "Cyber-Physical System for smart monitoring and tele-medicine for patient with COVID-19".

REFERENCES

[1] B. Calton, N. Abedini, and M. Fratkin, "Telemedicine in the Time of Coronavirus". Covid-19 articles fast tracked articles, volume 60, issue 1, e12 - e14 . Elsevier. Online at https://www.jpsmjournal.com/article/S0885-3924(20)30170-6/fulltext

[2] goe_telemedicine_2010.pdf (who.int) , (2020)

[3] S. Swarup, A.N. Makaryus, "Digital stethoscope: technology update", Medical Devices, vol. 11, pp. 29-36, (2018)

[4] Eko CORE Digital Attachment - Stethoscope Amplifier | Eko (ekohealth.com)

[5] Bluetooth Technology Overview | Bluetooth® Technology Website

[6] Encyclopedia Britannica, graphical user interface | computing | Britannica

[7] Encyclopedia Britannica, https://www.britannica.com/technology/client-server-architecture

[8] Microsoft, https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet

[9] Microsoft, https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15

[10] Eclipse Mosquitto

[11] Microsoft, SQL Server 2019 | Microsoft

[12] Microsoft, Visual Studio 2019 Community Edition – Download Latest Free Version (microsoft.com)

[13] Microsoft, Visual Studio IDE with .NET - Develop Any App Using C#, F#, VB (microsoft.com)

[14] A. Ivanov, K. Hrisafov, , N. Chivarov, "Application of IoT for telemonitoring patients diagnosed with coronavirus COVID-19", Journal of Informatics and Innovative Technologies: № 2-3 (2), 2020 ISSN: 2683-0930, pp 3- 8

[15] Software Testing Methods - SOFTWARE TESTING Fundamentals

[16] Black Box Testing - SOFTWARE TESTING Fundamentals

[17] What is Exploratory Testing? Techniques with Examples (guru99.com)

[18] Microsoft PowerPoint - BBSTExploring.ppt (testingeducation.org)

[19] A. Ivanov, K. Hrisafov, N. Chivarov, S Chivarov, I. Budinska, "Tele-Medical System for Remote Monitoring of Patients with Covid 19 and Other Infectious Diseases", IFAC-PapersOnLine, Volume 54, Issue 13, Pages 327-332, ISSN 2405-8963, (2021), https://doi.org/10.1016/j.ifacol.2021.10.468,

https://www.sciencedirect.com/science/article/pii/S2405896321019054