

eHealth and Smart Solutions framework for health monitoring in the course of the pandemic

János Dávid Balogh*, Attila Adamkó²

¹Doctoral School of Informatics, University of Debrecen, Hungary

²Department of Information Technology University of Debrecen, Hungary

*balogh.janos.david.official@gmail.com

Abstract – Our main objective focusing on the different design principles for applications which can receive and store various data from smart devices e.g., smart bands, smart watches, smart homes and could gain more information about its users with some extent to their health. We would like to collect this information, store them, then create well understandable diagrams and show them to the user in a modern, responsive environment. We placed great emphasis on the architecture which holds the necessary features (scalability and extensibility). We have followed the standards, recommendations, and protocols during the design process. The result is an application framework which fulfills the duty of an XXI. century smart solutions application that could monitor the health of people and even help to live a healthier life avoiding chronic disorders e.g., obesity and/or diabetes. We focused on predicting and classifying on COVID-19 disease too according to the collected information and research results. We achieved this project using our own devices like a Samsung Galaxy Watch 2 which comes integrated Tizen OS and gave the opportunity to develop a data sending application on it. On the other hand we worked with smartbands like the Xiaomi Mi Band 6 but it does not have programable OS so we needed to find a solution to collect data from it but this case is discussed in the paper later. We would like to implement this project live with the Clinic of our university to get more real cases and data. We are working on a solution where everyone could have their own medical assistant to predict and prevent healthcare issues.

Keywords – Smart Solutions, Smart Devices, Healthcare, Architecture, Data Gathering, Data Analysis, Machine Learning, Covid, Data Model

I. DATA COLLECTION

The first problem is finding a solution to gather health data from various IoT devices and sending them to the main application. Every device has its own structure so we had to find a way to handle that.

A. Smartwatch with SDK

Many smartwatches have more precise sensors to measure heart rate, distance, stress level and many more health information. These watches are coming out with operation systems from the factory and the manufacturers provide SDK. We could develop an application which collects the above

mentioned health information over time into one file and we could send it to our architecture via REST API endpoint. Also these devices have default smartphone applications where we can export data easily without own application and the users could share manually with us



date	time	heart_rate	distance	stress_level	...
2023-06-05	12:30	60.0	12.20	12.54	...
2023-06-05	13:40	60.0	10.03	17.51	...
2023-06-05	14:50	60.0	10.03	17.51	...
2023-06-05	16:00	60.0	10.03	17.51	...
2023-06-05	17:10	60.0	10.03	17.51	...
2023-06-05	18:20	60.0	10.03	17.51	...
2023-06-05	19:30	60.0	10.03	17.51	...
2023-06-05	20:40	60.0	10.03	17.51	...
2023-06-05	21:50	60.0	10.03	17.51	...

Fig 1. Health information from a smartwatch

This figure shows the data we collected from our test device which is a Samsung Galaxy Watch 2. This spreadsheet focuses on heartbeat rate, displays minimum and maximum heartbeat and the current measurements. Beside the measurements informs us the measurement date, device and more useful health information.

B. Smartbands

We talked about the smartwatch solution, but these device still are more expensive than that an average person could afford. That is why we shifted our focus on smartbands which are more affordable for larger crowds.

These devices also equipped with sensors, maybe not the same precision level as the smartwatch sensors, but still could get information about the health information we want to have. The greatest disadvantage of smartbands is the lack of open operation system and SDK. We had to find a solution to collect the data from the devices. Modern Xiaomi and Huawei devices offer a REST API endpoint where we authorize ourselves and get the requested information. We used that endpoint in an application where the user just adds their api key and authorization information so we could get our hands the data and send it to our architecture.

date	time	heartRate
2023.03.14	20:03:00	91
2023.03.14	20:04:00	83
2023.03.14	20:05:00	91
2023.03.14	20:06:00	88
2023.03.14	20:07:00	106
2023.03.14	20:08:00	104
2023.03.14	20:09:00	73
2023.03.14	20:11:00	89
2023.03.14	20:17:00	76
2023.03.14	20:18:00	108
2023.03.14	20:21:00	85
2023.03.14	20:31:00	78
2023.03.14	20:41:00	75
2023.03.14	20:51:00	84
2023.03.14	21:01:00	75
2023.03.14	21:04:00	73
2023.03.14	21:10:00	83
2023.03.14	21:11:00	95
2023.03.14	21:16:00	75

Fig 2. Health information from smartband

As we can see above the smartbands send less information about the heartbeat rate compared to the smartwatch but we still get what is necessary, the date and the measurements which are crucial to our research.

C. Renaming

As we work with many type of devices we are bumping into one great obstacle. Different devices have various data storing structures where they contain the same type of data but the sturcture is different. If we would like to work with them effectively we should get them bring into common structure. The above mentioned figures show what is the problem is, the smartwatch heartbeat rate measurement has a long specific name but the smartband heartbeat rate is a simple one. We dedicated a service which is between the recieving endpoint and the storing to handle these various format and producing into our structure but we do not lose important information.

II. DATABASE AND STORING

A. Database

We were taking account of the difference of smart devices and sensors, they gather and deliver more or less the same data structure, but we wanted to make sure that the varying data structure will not cause problems in storing these data. That is the reason why we decided to use NoSQL database instead of the standard regular SQL databases. The choice of ours was the MongoDB. It offers a flexible document data model database which stores data in JSON format. Ad-hoc queries and secondary indexing are supported which makes truly powerful ways to access our data. The database does not contain any business logic, on how to process the data. It has only one job and it fulfills that perfectly. In our application we implemented a service which handles the repository. The repository has the connection with the database and is responsible for the CRUD operations and various queries This book [4] guided us to implement a data model which will contain our sensitive data and is memory effective.

B. Sorting out false measurements

When we are wearing the smart devices during our life the devices automatically measure our health and sometimes they could get a false information when the sensors are not touching well the user or any kind of problem. If this happens they could record abnormally high heartbeat rate or zero value and we cannot work with these false information, we need to find a way to get rid of these data. We used the Simple Moving Average (SMA) to find out false measurements. SMA attaches a value to every i -th element using the average of sum of all of i elements. When this value is too high or low we label it as false measurements and do not work with anymore.

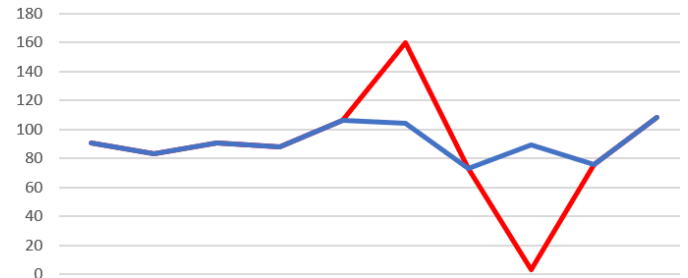


Fig 3. Diagram showing the sorting out

As the figure shows it means a lot when we are using SMA. The red line represents the original measurements with false values and it clearly points out what happens when it contains false information. The big spike is when the heartbeat rate is too high and the downhill when it failed to measure any heartbeat. The blue line is after SMA and it is a more consistent representation of the measurements so we could work with valid and supposedly true values after storing in the database.

III. ARCHITECTURE

A. Goals

As we stated before one of our goals is a flexible data model which could be a base of a general data managing platform. For this purpose, we relied on NoSQL database MongoDB that is a document data model storing data in BSON objects. A BSON object is a binary JSON object, this format helps in data storage and network transfer. We could store sensor data with different structure easily and even process them. In the backend we used the bson package and the Document data type to work with the database. Our other goal was to create an architecture which satisfies our needs. We wanted to design a layered architecture that is scalable, modular, expandable. Scalable, we would like to work with larger data inputs and more users so we must be ready to receive more requests. To do that we must scale our architecture without a problem, or we would suffer heavy data loss.

Modular, anytime we can decide which module is needed or not. We could wire or unwire any module with ease.

Expandable, if any new requirements come in the picture, we could meet them by adding new features and modules to the architecture.

For fulfilling these properties, we decided to implement a microservice architecture. In our project we expanded it to see how it will work out. We studied [1] to get more understand in modern software architecture engineering and [6] to learn developing in Spring Boot which will support our goals.

B. Microservices

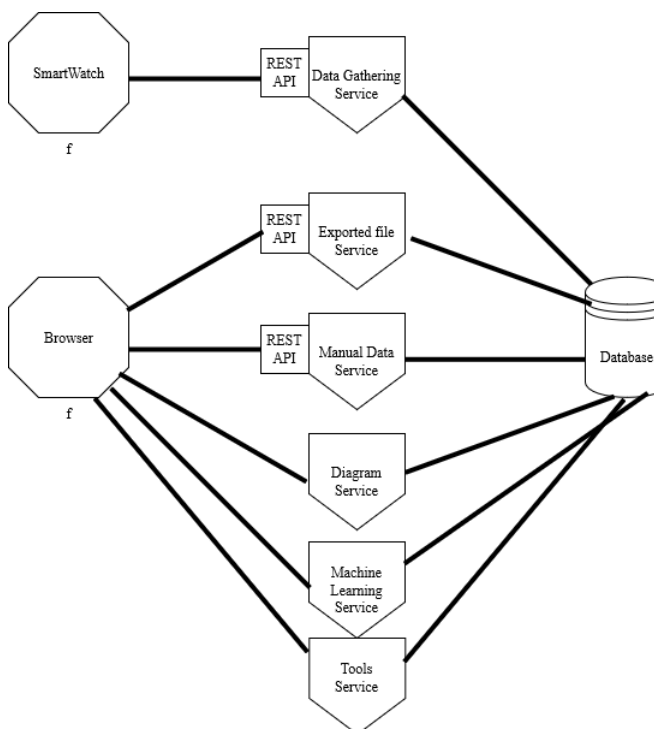


Fig 4. The architecture

The microservices architecture is a collection of services that are highly maintainable and testable and independently deployable. Spring offers an opportunity if we would like to wire many services together. All our services share the same database, but they have different endpoints with REST API interface and business logic. We chose this architecture because it is easy to expand later when we are adding more and more services to the application. As figure shows, we have two external entry point: the smart watch and a browser. The smart watch sends the data through the REST API interface and its service to the database

The browser can send data as exported file or/and manually. We provided different services to them, and they handle the rest to the database. We query the database in the creating diagrams, tools and machine learning services and their results are shown in the browser.

C. Creating graphs

After we got data from the users and they are stored in our database, finally we could work with them. First, we need to check if they are normalized. It means none of them has a different format. If they do, we have to normalize them. The main goal of the application is to provide value added services, give feedback and motivation about the lifestyle change. For this reason, our illustrations are easily readable. To create diagrams, we used the R script language. R is famous for its data processing and plotting abilities, it is widely used in analytic, statistic works, and Spring can invoke third party programs. Our script needs a dataset to work with it. For this purpose, we wrote a query which collects the user's relevant data from the database. The script waits for a csv file, so a simple CSV writer class writes the result into a csv file. The newly created csv file is passed into the script as argument and runs. Creates diagrams such as calorie burn by date, heart rate rating, graph of steps, calorie intake and burn by date and many more. These diagrams are exported as images, and they are shown to the user by the frontend part of the application.

D. Machine learning

The greatest feature should be our machine learning classification which will be using our collected data to classify the user or the patient if they are infected with any disease. Our personal approach was from our experience with the COVID-19 which caused lots of loss and trouble in our world. [5] helped our research to get started to develop a model in machine learning in Python. The architecture will pass the input values to the script and call it and will be waiting for its output.

We used the most popular algorithms to train our model and test it with the data that we had and it looks like it was satisfying. According to this research [2] we saw that if a person with COVID-19 produces more heartbeat even if the person is sitting or in resting position. Because we collected heartbeat rate and more health information we could classify if the user is dealing with COVID-19. This works as a binary classification where we import the user's data which contains heartbeat rate and activity position and we calculate average heartbeat by day when they are resting or sitting. The average heartbeats by days will be used as input data and if they show increasing values day by day and they are out of normal person heartbeat rate in resting position interval we could suppose they are infected with the disease, we label them with true value. If the average values within the interval and shows no increasing values they are healthy and we label them with false value. In the future we would like to work with infected people to wear smart devices to collect more precise information about their heartbeat and symptoms to improve our machine learning model to predict their condition.

E. HL7

Because our project works with healthcare, we found it worthwhile to use the Health Level Seven standards. The Health Level 7 standards are a set of international standards for transfer of clinical and administrative data between software application used by various healthcare providers. The name comes from using the application layer which is layer 7 in the OSI model. The HL7 Standards are produced by Health Level Seven international and used by bodies e.g., American National Standards institute and International Organization for Standards. [3] showed a way to implement and use the HL7 standards to achieve communication between our architecture and healthcare providers. We dedicated for a HL7 a new service which satisfies the standards for com-

munication between healthcare software applications. It is heavily relied on XML documents because HL7 newer versions store the necessary data in it. It includes the patient's ID and status like body weight or other measurements such as heart rate.

We want to develop a service in our general architecture where medical approaches could benefit from it, helping the digitalization of healthcare. We are still working on it to be more precise with the data output from the smart watches, but we have already found a way to process data from them and convert them into XML to use HL7 Standards.

```
<?xml version="1.0" encoding="UTF-8"?>
<Communication xmlns="http://hl7.org/fhir">
  <id value="example"/>
  <text>
    <status value="generated"/>
    <div xmlns="http://www.w3.org/1999/xhtml">Patient has very high serum potassium</div>
  </text>
  <identifier>
    <type value="Paging System"/>
  </type>
  <system value="urn:oid:1.3.4.5.6.7"/>
  <value value="2345678901"/>
</identifier>
  <instantiatesUri value="http://example.org/hyperkalemia"/>
  <partOf>
    <display value="Serum Potassium Observation"/>
  </partOf>
  <status value="completed"/>
  <category>
    <coding>
      <system value="http://hl7.org/fhir/message-types"/>
      <code value="Alert"/>
    </coding>
    <text value="Alert"/>
  </category>
  <medium>
    <coding>
      <system value="http://terminology.hl7.org/CodeSystem/v3-ParticipationMode"/>
      <code value="WRITTEN"/>
      <display value="written"/>
    </coding>
    <text value="written"/>
  </medium>
</Communication>
```

Fig 5. Example XML for HL7

We are trying to build a relationship with the Clinic of the University of Debrecen to see how will work our solution in live environment where doctors take part in our research. They will get 24/7 healthcare information about the patient and if any trouble could occur they could take steps to prevent disease like the COVID or give aid if necessary.

IV. CONCLUSION

While developing this application we gained more knowledge and insight into modern application architecture, responsible websites and even healthcare a bit. We would like to invest more in this project because we are dedicated in our goals to improve people's health using the instruments of technology.

We are planning to invite more people to the project using different smart

devices to collect more data. If we have more data, we can continue the machine learning service because it requires more precise information about diet and service and their results.

REFERENCES

- [1] N. Ford, M. Richards: Fundamentals of Software Architecture: An Engineering Approach, O'Reilly Media, 2020.
- [2] Özkaya, U., Öztürk, Ş., & Barstugan, M. (2020). Coronavirus (COVID-19) classification using deep features fusion and ranking technique. Big Data Analytics and Artificial Intelligence Against COVID-19: Innovation Vision and Approach, 281-295.
- [3] K. S. Mann, E. G. kaur: Generation of CDA/XML Schema from DICOM Images using HL7 Standards, IAEME Publications (2013).
- [4] L. Silverston: The Data Model Resource Book, Vol. 1: A Library of Universal Data Models for All Enterprises Revised Edition, Wiley, 2001.
- [5] P. Sodhi, N. Awashi, V. Sharma: Introduction to Machine Learning and Its Basic Application in Python, Proceedings of 10th International Conference on Digital Strategies for Organizational Success (2019).
- [6] G. L. Turnquist: Learning Spring Boot 2.0, Packt Publishing Limited, 2017.