The architecture design of the health monitoring system

Anja Radulović
Faculty of Organizational Sciences
University of Belgrade
Belgrade, Serbia
ar20210393@student.fon.bg.ac.rs
[0009-0004-2868-8078]

***Abstract*—Smart health devices enable users to create health habits, and maintain or improve their overall well-being. This growing trend increases the demand for systems capable of processing and visualizing collected data in real-time. The focus of this paper is to present the architecture design of a health monitoring system, developed as a full-stack web application using modular MVC-based architecture. The designed system is user-friendly and scalable, offering different types of interactive reports based on the type of the collected health data. The system allows users to register, synchronize their health data with the application, and view personalized reports that help them follow their daily, weekly, and monthly health trends. The result of the implementation provides a functional and adaptable platform for managing personal health data, which can support future integration with real-world wearable devices.**

Keywords—health data, interactive reports, user-centered application, real-time system

I. Introduction

In recent years, healthcare systems have increasingly incorporated smart technologies through wearable devices [1]. These devices allow continuous monitoring of physiological data, such as movement, oxygen levels, blood pressure, and heart rate [2]. Wearables are described as devices that allow users to be passively monitored without interrupting daily routines [3]. This view is widely acknowledged in literature [1,4], showing how such devices support ongoing tracking without interfering with user's day-to-day life. Thus, the wearables have shown effectiveness in healthcare domains, from real-time health monitoring to long-term illness tracking and diagnostic procedures [5, 6].

However, the existing solutions rely on commercial wearable devices to collect health-related real-time data. Accordingly, applying these systems for educational or research purposes, has many challenges, especially in phases before involving physical devices. To address this need, the aim of this paper is to present a simulation-based health monitoring system.

The rest of this paper is organized as follows: Section 2 reviews related work relevant to the topic. Section 3 presents the methodology used in the research. Section 4 describes the system design of the proposed system. Finally, Section 5 concludes the paper and outlines directions for future work.

II. Literature Review

According to current forecasts, digital health systems will grow at an average annual rate of 19.7% through 2032 [7], while wearable health devices will grow by 11.2% annually during the 2023–2028 period [8].
This rising demand for data-driven healthcare implies the need for maintainable and scalable systems. Therefore, the literature is reviewed through three main aspects: usability, essential features and overall implications.

A recent study on the usability of the patient monitoring found a high task success rate (93.8%) and positive user satisfaction (mean score 4.15). However, eye-tracking analysis showed that crucial elements such as alarm sensors were frequently overlooked, showing that usability is directly linked to safety outcomes [9].

Commercial health monitoring systems have different usability profiles according to their target users and the surrounding technological environment. Apple Health and Huawei are known for their smooth design and simplicity, though they rely on their ecosystems. Google Fit is the most accessible and minimalistic platform, yet it offers fewer features compared to more advanced systems. On the other hand, Garmin Connect offers detailed functions and metrics, making it especially suitable for athletes and other advanced users. Because of the complexity of the platform and its dependence on Garmin devices, it is less practical for everyday users.

Beyond usability, key features also vary across these systems. For measuring vital parameters and monitoring physical activity, Apple Health integrates data from Apple Watch and third-party apps (such as Strava) that can send their information into HealthKit. Google Fit also integrates with popular fitness apps and wearables, but compared to other platforms, it is simpler, focusing on basic features like tracking steps, distance, calories and heart-rate, making it well-suited for everyday users.
Huawei Health records health indicators such as heart rate, blood pressure, sleep quality, and others, and on supported wearables (the Watch Fit series) it also provides advanced exercise features, such as guided workouts and running courses. Garmin Connect focuses on tracking performance

with detailed metrics and offers users to create or follow personalized training plans. [10-13]. These examples show platforms targeting different types of users: simple metrics for average users and advanced metrics for athletes.

In terms of benefits and limitations, commercial platforms are user-friendly, but relying on certain devices or ecosystems makes them less accessible. In comparison with other platforms, Google Fit is more open, but it lacks advanced features. Academic systems promote innovation with continuous monitoring, AI-based analytics, IoT platforms and biosensors [14-17], but they are often complex, hard to scale and not easily usable by non-experts.

To conclude, commercial systems are easy to use and widely spread, while academic systems focus on technical innovation. However, there remains an unaddressed gap for a flexible, educational platform that operates of vendor-specific hardware. The proposed system is positioned to focus on adaptability and user-friendliness, with the potential to integrate AI and wearable devices in the future.

III. METHODOLOGY

Following a simplified version of Larman's use-case driven software development methodology, the project was structured in five phases: requirements gathering, analysis, design, implementation and testing [18].

A. Requirements gathering

In the requirements gathering phase, a verbal description of the system was created, along with clearly defined use-cases. A key focus should be placed on this phase in order for the system to meet user requirements and ensure its success [18].

The system is designed for users who interact with smart wearable devices to collect biometric health data. Each user has a unique account and can register, log in, and reset password via email. Users can also edit their personal profile (name, surname, age, weight, height). Once authenticated, the user can synchronize collected data with the application.
The analyzed data can be accessed through a dynamically generated dashboard and through detailed interactive reports. The dashboard presents a daily summary of activity and physiological parameters.
By selecting a parameter, the user can access a generated report on a daily, weekly or monthly, based on the synchronized data. As a result of the requirements gathering process, a use-case model was created to represent defined systems functionalities. Each use-case was also described in detail, outlining specific user interactions based on defined requirements.

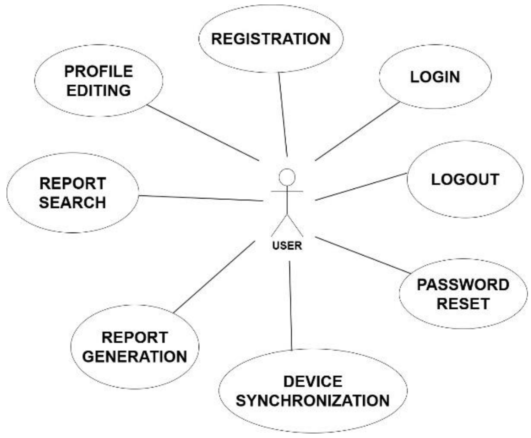

Fig. 1. Use-case model of the proposed system

B. Analysis

The analysis phase describes the business logic of the software system, which is divided into system behavior and system structure. The systems behavior is modeled through UML sequence diagrams and system operations, while the structure of the system is described using conceptual and relational models. [18].

In this project, based on the defined use cases [Fig.1], corresponding sequence diagrams were created to visualize the communication flow and analyze the system's behavior. Furthermore, the system's structure was defined using a conceptual model, which identifies the main domain entities and their relationships. A relational model is also created for later stages of design and implementation, which further formalizes these structures. Examples of the system structure and behavior are given in Fig. 2 and Fig.3 respectively.

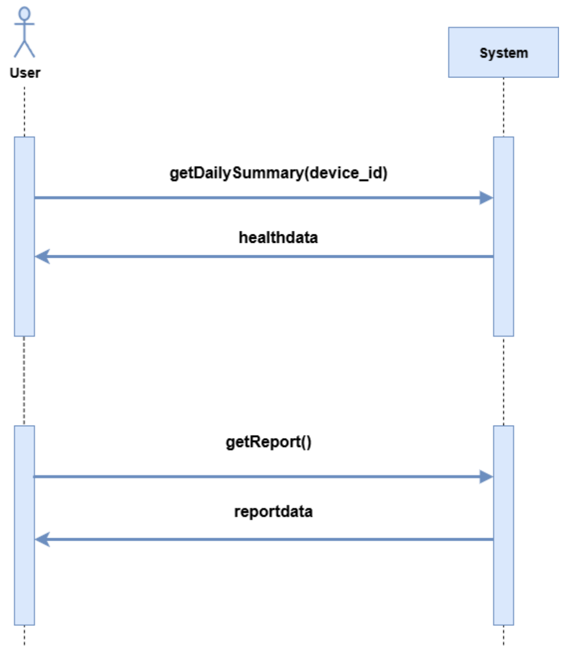

Fig. 2. Sequence diagram illustrating the main scenario of report search functionality

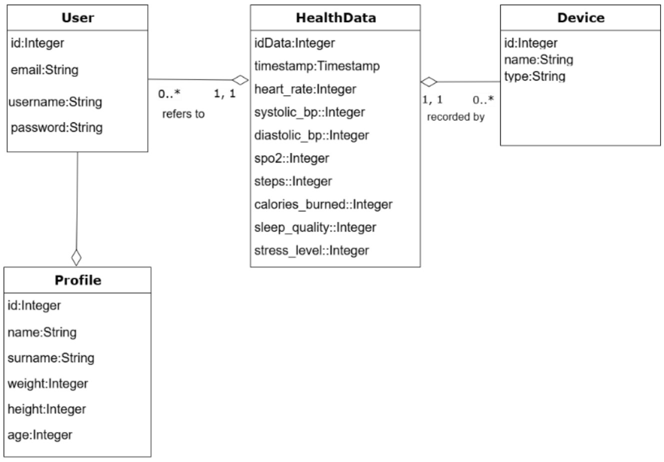

Fig. 3. Conceptual data model of the proposed health monitoring system

C. Design phase

In the design phase, the architecture of the system is defined as standard three-layer architecture, consisting of user interface layer, application logic layer and data layer [18]. Using UML diagrams from the analysis phase, the system was designed according to the requirements.

IV. SYSTEM DESIGN

A. Architecture overview

As defined in the design phase (Section III.C), the system applied modular MVC design using Laravel framework, enabling modifications of one unit without affecting the others. This is valuable in the presented application because it enables further parallel development by independent teams [19].

The research has shown [20-23] that Laravel has proven to be a popular, powerful and developer-friendly tool for building complex and scalable web applications. Thus, it is being a good choice for the backend of the presented web-based health monitoring system.

In this project, Laravel was used to manage authentication, data storage and REST API communication with the React frontend, while React was selected to implement a single-page application, enabling interactive and fast visualization of health reports.

The purpose of the designed system was to enable users to easily access personalized health reports and follow their health condition, while remaining flexible for future upgrades.

B. User Interface

The user interface was designed to be user-friendly and intuitive. Its main components include: login and registration forms, a profile management page, a main dashboard and detailed reports. To improve visual experience, a 3D model animation created in Blender was integrated into profile management page. Examples of the user interface are shown in Fig.4 and Fig.5.

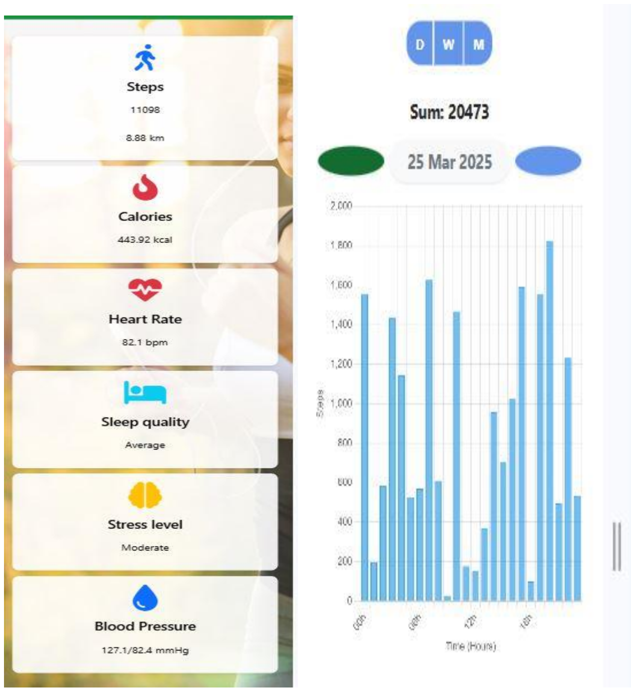

Fig. 4. Dashboard and sample steps report

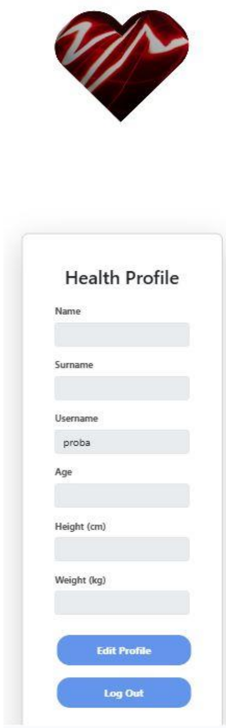

Fig. 5. Profile management page with integrated 3D model

C. Application logic

The communication logic defined through UML diagrams in analysis was used to structure the application logic and connect each use case to the planned system components. The application logic was designed to include planned RESTful APIs, defined HTTP methods and data formats, as well as the roles of controllers and services in request handling. The database structure was planned based on the relation model developed during the analysis phase.

The RESTful API endpoints are designed for key operations such as authentication, data synchronization, profile updates and reporting. An example of the design of a RESTful API endpoint is shown in Table 1.

TABLE I. DEFINED RESTFUL API ENDPOINT

Function description	Generation of random devices
HTTP method	GET
URL	/devices/random
HTTP body parameter	{ }
Format HTTP body parameter	JSON

The proposed architecture (Fig.6) follows a full-stack approach, consisting of client-side (frontend) and a server-side (backend).
In the system, the user interaction with the frontend creates related HTTP request, which is then routed through the RESTful API (as outlined in Table 1) and processed by Laravel middleware before reaching the responsible controller. The controller then interacts with the model, which performs query on the database. After getting the response from model, the controller returns a JSON response through Laravel routing and React frontend renders the requested information to the user.

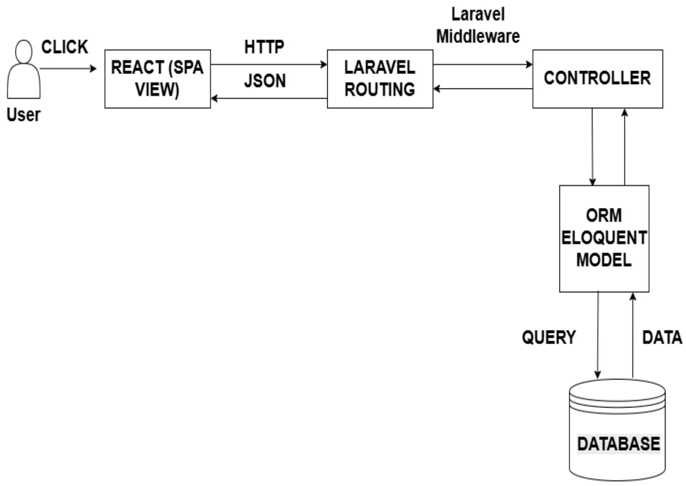

Fig. 6. The architecture diagram of the full-stack system for health monitoring

D. Implementation and Testing

The designed full-stack system is implemented using React for the frontend, and Laravel for the backend.
The frontend React implementation as SPA (single page application) enables updating only necessary data without reloading the entire page, therefore providing dynamic user interaction.

On the backend, Laravel uses token-based authentication with Sanctum to secure that only users with a valid token can access protected parts of the system. Thus, simplifying the communication between the Laravel backend and the React Single Page Application.

The controllers are implemented to follow the principles of Separation of Concerns and the Single Responsibility Principle, so that each has a clearly defined responsibility. Thus, controllers related to the report generating are divided based on the type of the parameters they process: activity metrics or physiological parameters. The activity metrics are aggregated (steps, calories), whereas physiological parameters are over time averaged.

Migrations were used for creating and modifying the database (adding columns, deleting columns, creating columns, adding foreign keys, removing foreign keys, deleting tables and renaming tables). With the help of migrations, one can follow the flow of the edits of the database structure.

For the purpose of testing the system without integrating wearables, a Bluetooth simulation was implemented. By generating data, the simulation mimics the process of collecting data from devices. This generation is initiated through "synchronization" button on the frontend. After the user clicks on this button, the backend function in controller triggers a factory method that produces random values of key parameters within predefined range and for the current day. Based on the generated data, the frontend then renders reports and graphs.

This approach is an efficient way for testing the proposed full-stack system, especially in educational environment where physical devices may not be available.

```php
class HealthDataFactory extends Factory
{
    0 references
    protected $model = HealthData::class;
    0 references | 0 overrides
    public function definition(): array
    {
        return [
            'device_id' => fn(array $attributes): mixed => $attributes['device_id'],
            'timestamp' => fn(array $attributes): mixed => $attributes['timestamp'],
            'heart_rate' => $this->faker->numberBetween(int1: 60, int2: 100),
            'systolic_bp' => $this->faker->numberBetween(int1: 110, int2: 140),
            'diastolic_bp' => $this->faker->numberBetween(int1: 70, int2: 90),
            'steps' => $this->faker->numberBetween(int1: 0, int2: 2000),//koraci na 1h
            'calories_burned' => function (array $attributes): float|int {
                return isset($attributes['steps']) ? round(num: $attributes['steps'] * 0.04, precision: 2) : 0;
            },
            'sleep_quality' => $this->faker->numberBetween(int1: 50, int2: 100),
            'stress_level' => $this->faker->numberBetween(int1: 0, int2: 100),
        ];
    }
```

Fig. 7. Factory method for generating health-related data

V. CONCLUSION

This paper presents a simulation-based, full-stack health monitoring system developed using simplified Larman's methodology. It follows MVC architecture and enables users to generate, store and visualize their biometric data. The use of simulated data allowed testing of the platform without integrating devices, making the solution suitable for educational and research purposes. The main advantage of the proposed system lies in its scalability, flexibility and modularity.

However, the limitations of the current implementation include the lack of integration with real wearable devices, the limited number of monitored parameters and the absence of a large-scale user study. Future work will aim to overcome these limitations with additional modules and functionalities. Planned improvements are based on survey results, which showed that alerts for abnormal values of parameters and personalization options were the most relevant features. Testing with larger user groups is also planned to evaluate the system after adding new modules.

REFERENCES

[1] M. Haghi, K. Thurow, and R. Stoll, "Wearable devices in medical Internet of Things: Scientific research and commercially available devices," *Healthcare Informatics Research*, vol. 23, no. 1, p. 4, Jan. 2017, doi: 10.4258/hir.2017.23.1.4.

[2] A. Erdem, E. Eksin, H. Senturk, E. Yildiz, and M. Maral, "Recent developments in wearable biosensors for healthcare and biomedical applications," *TrAC Trends in Analytical Chemistry*, vol. 171, p. 117510, Dec. 2023, doi: 10.1016/j.trac.2023.117510.

[3] W. Gao *et al.*, "Fully integrated wearable sensor arrays for multiplexed in situ perspiration analysis," *Nature*, vol. 529, no. 7587, pp. 509–514, Jan. 2016, doi: 10.1038/nature16521.

[4] A. Costantino, M. Ferrara, M. Arnesano, and E. Fabrizio, "Off-the-shelf wearable sensing devices for personalized thermal comfort models: A systematic review on their use in scientific research," *Journal of Building Engineering*, vol. 70, p. 106379, Mar. 2023, doi: 10.1016/j.jobe.2023.106379.

[5] S. M. A. Iqbal, I. Mahgoub, E. Du, M. A. Leavitt, and W. Asghar, "Advances in healthcare wearable devices," *Npj Flexible Electronics*, vol. 5, no. 1, Apr. 2021, doi: 10.1038/s41528-021-00107-x.

[6] L. Lu *et al.*, "Wearable Health Devices in Health Care: Narrative Systematic Review," *JMIR Mhealth and Uhealth*, vol. 8, no. 11, p. e18907, Sep. 2020, doi: 10.2196/18907.

[7] J. Rawal, "Digital Health Market Size, Trends, Growth, Analysis, 2025-2032." https://www.fortunebusinessinsights.com/industry-reports/digital-health-market-100227

[8] "Wearable Healthcare Devices Market growth, Drivers, and opportunities,"*MarketsandMarkets*.

[9] H. Choi and W. Jang, "User experience study of the patient monitoring systems based on usability testing and eye tracking," *Healthcare*, vol. 12, no. 24, p. 2573, Dec. 2024, doi: 10.3390/healthcare12242573.

[10] "HealthKit | Apple Developer Documentation," *Apple Developer Documentation*. https://developer.apple.com/documentation/healthkit

[11] "Google Fit | Google for Developers," *Google for Developers*. https://developers.google.com/fit

[12] HUAWEI, "HUAWEI Health - HUAWEI Global," *HUAWEI*. https://consumer.huawei.com/en/mobileservices/health/

[13] Garmin International, "Garmin Connect," *Garmin Connect*. https://connect.garmin.com/

[14] K. Lodewyk, M. Wiebe, L. Dennett, J. Larsson, A. Greenshaw, and J. Hayward, "Wearables research for continuous monitoring of patient outcomes: A scoping review," *PLOS Digital Health*, vol. 4, no. 5, p. e0000860, May 2025, doi: 10.1371/journal.pdig.0000860.

[15] Y. Y. Ghadi *et al.*, "Integration of wearable technology and artificial intelligence in digital health for remote patient care," *Journal of Cloud Computing Advances Systems and Applications*, vol. 14, no. 1, Jul. 2025, doi: 10.1186/s13677-025-00759-4.

[16] S. Rashid and A. Nemati, "Human-centered IoT-based health monitoring in the Healthcare 5.0 era: literature descriptive analysis and future research guidelines," *Discover Internet of Things*, vol. 4, no. 1, Nov. 2024, doi: 10.1007/s43926-024-00082-5.

[17] D.-K. Vo and K. L. Trinh, "Advances in wearable biosensors for healthcare: current trends, applications, and future perspectives," *Biosensors*, vol. 14, no. 11, p. 560, Nov. 2024, doi: 10.3390/bios14110560.

[18] S. Vlajić, Пројектовање софтвера (Скрипта - Радни материјал), Факултет организационих наука, Београд, 2020.

[19] S. Khan and A. T. Khanam, "Study on MVC Framework for Web Development in PHP," *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, pp. 414–419, Aug. 2023, doi: 10.32628/cseit2390450.

[20] P. Garbarz and M. Plechawska-Wójcik, "Comparative analysis of PHP frameworks on the example of Laravel and Symfony," Journal of Computer Sciences Institute, vol. 22, pp. 18–25, Mar. 2022, doi: 10.35784/jcsi.2781.

[21] M. Laaziri, K. Benmoussa, S. Khoulji, and M. L. Kerkeb, "A Comparative study of PHP frameworks performance," *Procedia Manufacturing*, vol. 32, pp. 864–871, Jan. 2019, doi: 10.1016/j.promfg.2019.02.295.

[22] M. K. Ahmed, A. H. Bello, S. S. Jauro, and M. Dawaki, "A comparative analysis of performance optimization techniques for benchmarking PHP frameworks: Laravel and Codeigniter.," *Dutse Journal of Pure and Applied Sciences*, vol. 10, no. 3c, pp. 284–295, Oct. 2024, doi: 10.4314/dujopas.v10i3c.27.

[23] A. Sunardi and N. Suharjito, "MVC Architecture: A comparative study between LaRavel Framework and SlIM Framework in Freelancer project monitoring System web based," *Procedia Computer Science*, vol. 157, pp. 134–141, Jan. 2019, doi: 10.1016/j.procs.2019.08.150.