HEALTHWARDEN – SMART POST-TREATMENT REMOTE MONITORING TOOL

Tomáš Kouřim, Marek Polák, Jan Skála, Anastasia Surikova, Lukáš Forst, Alžběta Kouřimová, Nikola Zadorozhny

Abstract

Rising population, its aging and in the same time increasing live standards all around the world put extreme demands on the entire healthcare system. At the same time, advancing digitalization opens new opportunities also for the healthcare system with telemedicine and remote healthcare on the rise.

One of the areas that can be handled by digital medicine is post-treatment care. It is usually well-defined with standardized guidelines and well-described best practices. HealthWarden is a modern post-treatment remote monitoring tool that allows for early discharge from the hospital, lower readmission rates, and helps patients achieve the best clinical results. The key benefits and features of HealthWarden are introduced in this paper, as well as its technical solution, integrability, deployment, and other technical and medical aspects.

Keywords

healthcare, remote monitoring, application, mobile, communication, hospital, FHIR, telemedicine, remote healthcare, telemedicine

1 Introduction

Rising population, that has recently reached 8 billion people[1], and its rising age (in the year 2022 10% of people is over the age of 65, this number is predicted to reach 16% by the year 2050[2]) lead to increased demands for medicine. This is further amplified by the increasing expectations of the ever more informed patients expecting a smoother experience in all medical encounters, be it a physician's checkup, stay at the hospital, or aftercare treatment. This all puts extreme pressure on the entire healthcare system.

On the other hand, digital technology advancement creates more opportunities for digital healthcare, yet those opportunities remain mostly unused. The digitalization of healthcare thus stays as one of the greatest challenges of our era.

One of the areas where a digital solution could bring a significant advantage is post-treatment care and patient monitoring. During this time the patient often has a clear and straightforward care plan set out by the treatment itself, the patient's condition and comorbidities, and also by the needs of the healthcare institution providing the treatment to the patient. There are also usually well-described potential complications connected with the treatment itself or with the medications the patient is supposed to take. Adherence to the care plan leads to clinically optimal results, with low readmission rates, low complications occurrence, and generally best results for the patients. However, in standard conditions, this adherence is only ensured by the compliance of the patient and only checked during regular but not so frequent checkups with the physician.

The compliance of the patients is generally low. This can be often caused simply by the fact that the patient does not remember the care plan exactly, studies[34] show that the patient only remembers a fraction of what the doctor says. The effectiveness of regular checkups is again affected by the patient's compliance and memory. Non-compliant patients do not tell the doctor the (entire) truth (or even skip the checkups entirely), all patients are then affected by the so-called recall bias when asked about adherence to the care plan, symptoms, complications, etc. This can lead to severe complications or even death of the patient if the problems are not reported and treated fast enough.

One solution to these problems is increasing the frequency of checkups and the time doctors spend with the patients explaining the care plan and possible complications. This is, however, not feasible, especially from an economical point of view. Another option is the digitalization of post-treatment care.

The paper is structured as follows. Section 2 describes the HealthWarden tool, focusing on its technical details, integrability, security, and extensibility. Section 3 presents possible further development of HealthWarden, and the last section concludes this paper.

2 HealthWarden

2.1 Description

HealthWarden is a mobile and web application that guides patients through their post-treatment[5] care. The mobile application is designated for patients while the web-based part is operated by doctors or other healthcare professionals. The application is given to the patient similarly as medications are given and contains a specific care plan, which depends on the treatment, providing institution/doctor, and the patient himself. HealthWarden then precisely exercises the care plan, collects relevant data, and checks it for potential problems. The care plan consists of two major components. Continuous vitality monitoring and virtual ward rounds. A simple diagram of the tool is depicted in Figure 1.

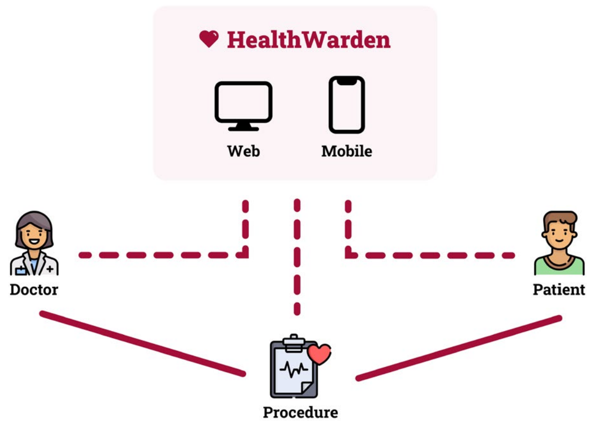


Figure 1 – HealthWarden diagram

2.1.1 Vitality Monitoring

The application can monitor several vitality indicators. By default, 5 indicators are monitored (steps, hydration, sleep, heartbeat, and blood pressure) while others (oxygen saturation, glycemia, calorie intake, weight, stool, etc.) can be added on demand. The indicators can be collected automatically from wearable devices (smart watch, heart rate monitor, …) or other smart devices (bluetooth enabled blood pressure monitor, smart scales, …) or collected manually by the patient.

The care plan contains the required frequency of data collection. If the data is not collected automatically and the patient does not input it on time, the application prompts the patient to fill in the required information. The care plan also contains desired ranges of the controlled indicators, which are shown to the patient for motivational reasons, and alert rules with different severity that can be triggered if the patient does not stick to the care plan. For example, if the patient does not fill in hydration for 3 consecutive days, it can trigger a low-severity alert.

If the heart rate obtained from a smartwatch is over 180 bpm for more than 10 minutes, it can trigger a high-severity alert. The doctor can, based on the clinical conditions, specify what each alert means. It can be an in-app notification, an email sent to designated addressees, or an SMS sent to the treating doctor.

2.1.2 Virtual Ward Rounds

The application introduces a concept called virtual ward round. Similarly to a regular ward round, it contains a set of checks, questions for the patient, and actions to be taken. Virtual ward rounds can be assigned to patients. There are two types of virtual ward rounds, one-time and regular. As the name suggests, a one-time ward round has only one occurrence, while a regular ward round can repeat multiple times. Virtual ward rounds offer an ideal tool for healthcare professionals to prepare and schedule predefined examinations which the patients can perform simply by using their mobile phone.

When a virtual ward round is due, the application notifies the patient to start the virtual ward round. The virtual ward round can contain an arbitrary number of checks, questions, and actions.

The checks consist of vitality checks performed additionally on top of the regular vitality monitoring. An example of such a check can be an extra blood pressure measurement or a heart rate measurement in a specific position (lying down, standing), etc.

Questions are grouped into questionnaires. The goal of the questionnaire is to perform a basic interview with the patient and find possible changes in the patient's condition. A questionnaire is defined by a set of questions. The healthcare professional managing HealthWarden can choose one question as a starting point. Each question can contain one or more answers. A transition can be defined between an answer and another question from the same questionnaire. This manages the way the patients navigate through the questionnaire when answering it.

Actions are a special type of user interaction. With actions, patients can track medication intake, be prompted to exercise or take a photo of an injury for further inspection by the doctor.

Similarly as in vitality monitoring, virtual ward rounds can also contain a set of rules to trigger alerts with different severity. These rules can be linked to the checks (for example specific check results can trigger specific alerts), questionnaires (for example certain answers to certain questions can trigger an alert), and actions (for example failure to fulfill an action can trigger an alert). Again, the managing healthcare professional has to specify what follow-up actions are connected with each alert based on the clinical conditions, as well as whether the doctor or the patient is alerted.

2.1.3 Web Application

The web application is designed to be used by doctors and other healthcare professionals. Its main purpose is to administer the care plan of each individual patient and check their progress. The main page contains a list of patients ordered by the severity of the alert triggered (the ordering can be changed), i.e., patients needing attention are listed first.

Doctors can then navigate to the patient detail page where they can see more information about the patient, results of vitality monitoring and virtual ward rounds. This view also serves as the general setting point for each individual patient. Tracked vitality indicators as well as their target values can be set and altered here, virtual ward rounds can be created, added, or edited from this patient detail page either using a ward round wizard or selecting from templates.

The last part of the web application is a side panel containing a patient's care overview. Here the patient's progress in an organized summary can be observed. It starts with the daily vitality, which shows whether the patient met the required goal or not, followed by events that are ordered chronologically. The events contain information about which questionnaires have been answered by the patient or which actions have been taken.

Screenshots from the web application can be seen in Images 1 and 2.

2.1.4 Mobile Application

The mobile application is the only part of HealtWarden that interacts with patients. We designed it with the often elderly and

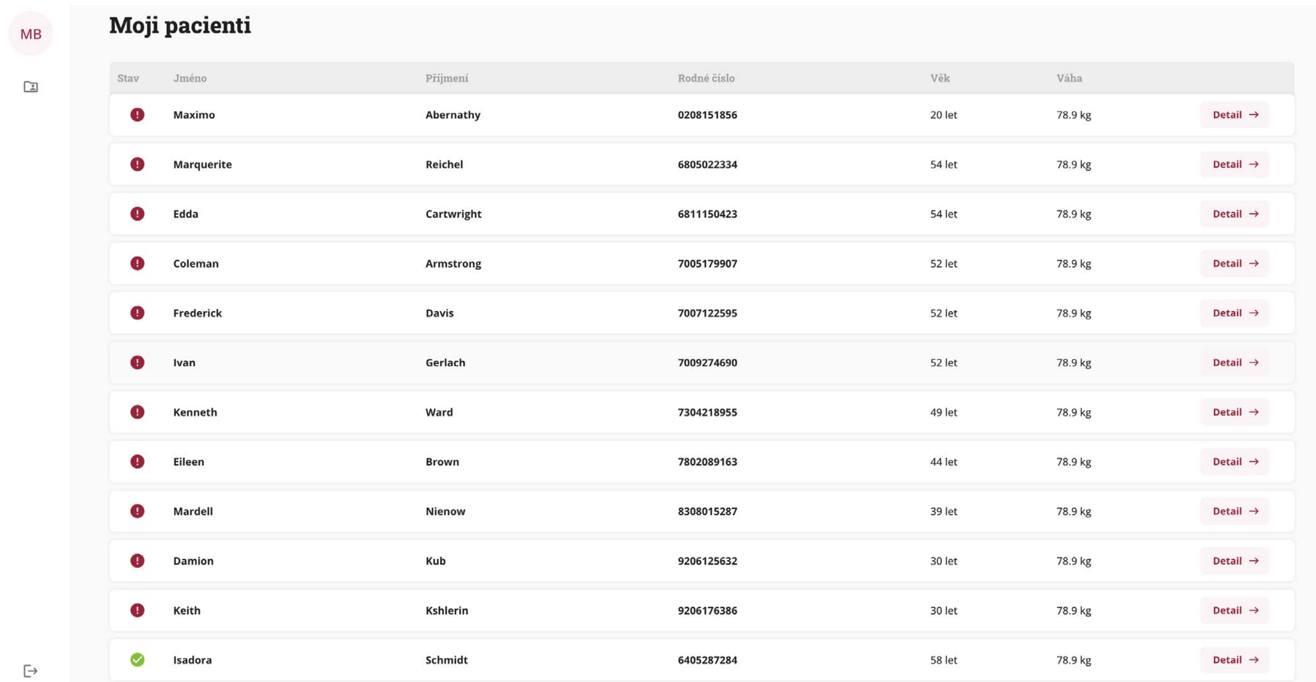


Image 1 – Web application homepage with a list of patients

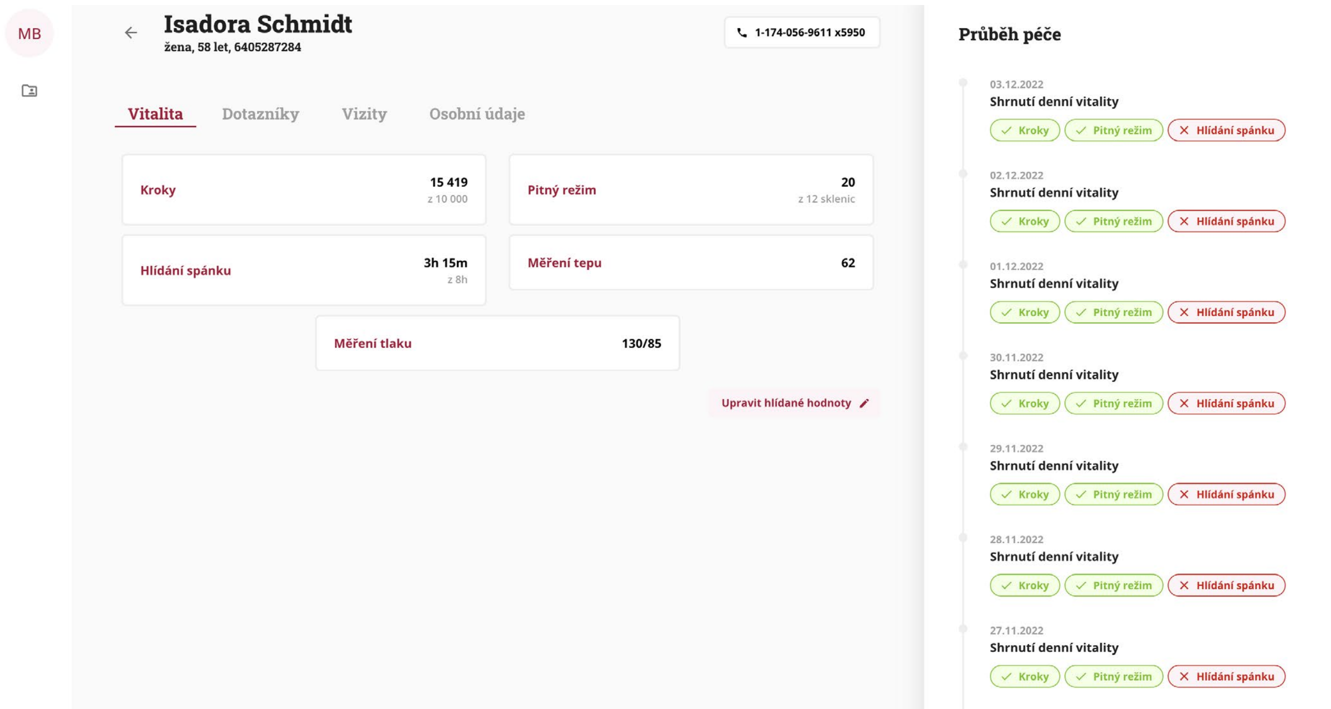


Image 2 – Detail of a patient

less digitally literate users in mind and it thus contains only the core functionality for achieving the required results.

After logging in, patients can see the daily vitality monitoring overview. The charts inform them about how much is left to meet their targets. Automatically collected as well as manually inputted data are shown here.

The top section of the homepage is dedicated to notifications. Users are reminded about pending virtual ward rounds that need attention. Patients who decide to undergo a ward round are navigated to a ward round wizard. After that, they are guided through all questionnaires that were selected for the ward round and requested to perform the required actions and checks.

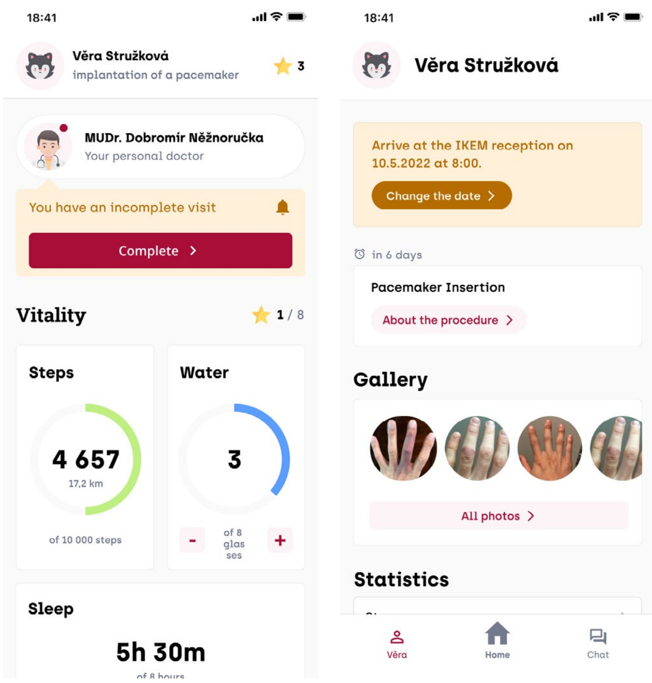
On the profile page of the application, the patients can also view the statistics about their vitality monitoring progress, past virtual ward rounds, their answers to questions, or actions taken. An informational/educational section is also available in the application, where users can find useful information about their care plan, the treatment they have undergone, or general information from healthcare professionals.

Images 3, 4, 5, and 6 show screenshots of the application.

2.2 Key Benefits

HealthWarden is a tool that helps doctors to apply the best care for their patients by guiding the patient through that care step by step. It also controls patients' progress in the post-treatment phase and helps identify those who might develop complications. This allows the doctors to discharge their patients from the hospital earlier and on the other hand lowers the readmission rate. This then lowers the costs of the hospital and the healthcare system in general. Healthwarden also helps to ease the administrative load connected with post-treatment care, further reducing hospital expenditures.

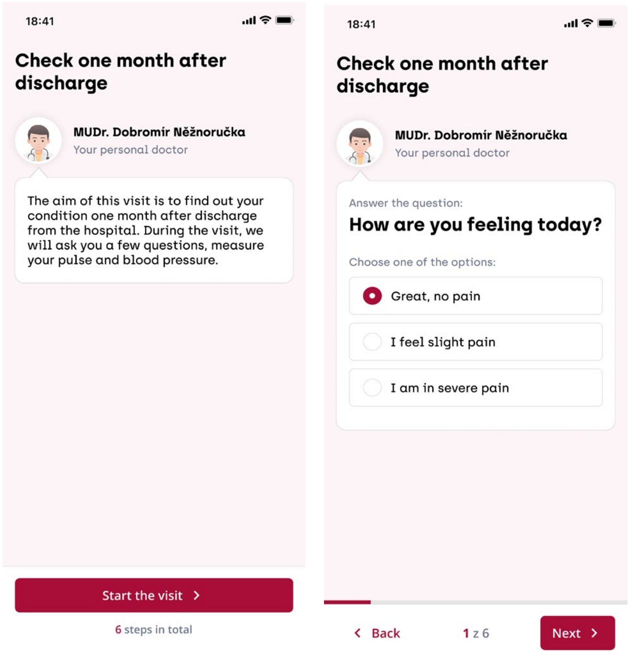
The key benefit for the patient is the improved and optimized care delivered to the patient. By lowering the time spent in hospital and the readmission rates, HealthWarden increases the capacity of hospitals, which then again benefit the patients as the care becomes more accessible



Images 3, 4 – Mobile application homepage and patient detail

2.3 Architecture

Application architecture follows common standards and best practices widely used in modern application design. We designed the solution as a client-server model, where the server is a REST-based web service for processing requests from the clients. The first client is a web application that is used by doctors while the second one is a mobile application that is intended for patients. The interface of the server is well documented and other clients can thus be added with ease. It also allows hospital systems to directly and easily communicate and integrate with the server.



Images 5, 6 – Virtual ward round

The design also respects principles of dependency inversion, which means that no external dependency is referenced directly, but abstraction is used instead. This makes all dependencies interchangeable if the need arises.

The architecture schema of the application can be seen in Figure 2.

2.4 Technical Solution

2.4.1 Backend

The application backend is written in Kotlin[6] using Spring framework[7] with PostgreSQL[8] as a database. It follows the principles of n-tier architecture. The code base is divided into three layers: data access layer, business layer, and presentation layer. Every layer communicates with the adjacent one only.

The data access layer contains database entities and repositories, together with a set of predefined queries. The communication between the database and the application is realized through Java Persistence API[9] and Hibernate[10].

The business layer contains a set of services that handle most of the application logic. All service methods are executed within a single database transaction and thus enforce atomicity and consistency[11] for each performed operation. Communication between the data access layer and presentation layer is achieved using data transfer objects (DTOs), plain objects without any inner functionality, that only carry data.

The presentation layer is a Web API with a set of controllers and endpoints following the principles of REST[12]. Every endpoint is documented with a summary, the expected format of a request, and the expected format of a response. The application also provides OpenAPI Specification[13] that documents the application interface. The documentation is also machine-readable and client code can be easily generated.

The codebase is also covered with automatic tests written in JUnit[14] and AssertJ[15]. The dominant part of testing are so-called integration tests. During these tests, the application is started together with a database. Tests contain three parts: arrange, act

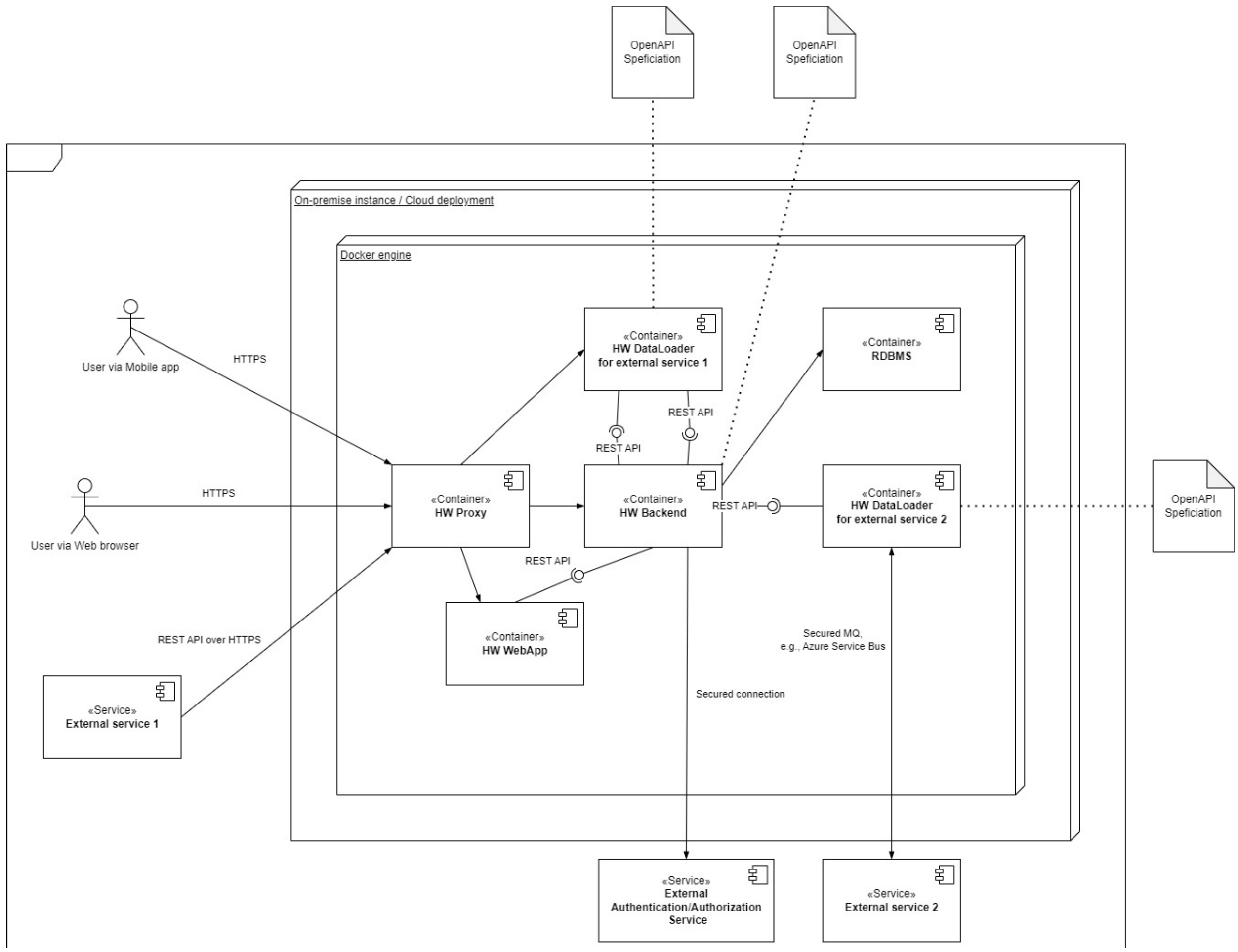


Figure 2 – HealthWarden architecture

and assert. During the arrange part, the system is prepared for testing by seeding the database with initial data. In the act part, the functionality being tested is executed. Finally, during the assert part, it is verified that the system ended up in the desired state. Tests are executed during every pull request. A branch cannot be merged unless all tests pass and at least one reviewer approves it.

2.4.2 Web Application

The web application is written using the React[16] framework. It communicates with the mentioned application backend through HTTPS protocol. The application is optimized to run on Chrome, Firefox, Microsoft Edge, and Safari in their last major versions, however, it should run without problems in any browser with a market share of over 0.2%, excluding Opera Mini and deprecated browsers.

2.4.3 Mobile Application

A hybrid development approach was chosen, which allows to build a multiplatform mobile application using a single codebase. The application is thus available for both Android and iOS devices. The application is built upon React Native CLI[17] for easier integration with the smartphone's native features. HealthWarden application can read data from the health data providers like Apple HealthKit[18], Google Fit[19], Garmin Connect[20], and Samsung Health[21]. This integration is crucial for automated vitality monitoring.

2.5 Security

Security attacks are on the rise and personal data is one of the most valuable targets. HealthWarden was designed from the ground up with security and privacy in mind. Industry-standard authentication mechanisms were employed for authentication and authorization, especially keeping OWASP Top Ten[22] in mind.

The basic version of the application, without any integration, offers user authentication against the internal database. User passwords are never stored in plaintext, but rather hashed using a state-of-the-art bCrypt[23] algorithm that prevents password leakage even in an unlikely event of a database compromise. Optionally it is possible to enable multi-factor authentication using email, SMS, or using any application that is able to generate time-based codes. MFA codes are then generated by the standard TOTP algorithm (RFC 6238)[24].

After successful authentication, the server issues a signed JWT (RFC 7519)[25], that allows a specific user to access the application data.

On the implementation layer, the server utilizes best practices by using industry-standard libraries such as Spring Security framework[26]. This ensures the correctness of all cryptography and security-sensitive operations.

The HealthWarden architecture was designed with interoperability in mind, allowing it to employ a wide range of authentication and authorization techniques and to seamlessly integrate into any authentication system that is used by the end client (typically a hospital). Moreover, the design allows for adding additional security measures when needed.

2.6 Integrability

As mentioned above in the Architecture section, thanks to the initial application design, it is possible to integrate the entire solution into various types of environments, such as internal hospital networks, publicly available web servers, or cloud providers.

Each service of the application is distributed as a standalone Docker[27] container and/or library or OpenAPI specification. This solution introduces the possibility to implement clients that communicate with applications in various technologies and programming languages.

The common problem for almost every application is the integration with other applications, such as hospital information systems. When two systems model the same concept, they often achieve so with a different set of entities or a different set of attributes. This introduces incompatibility between these two systems. One of these systems then has to introduce an integration code that overcomes this incompatibility.

The integration mechanism used in HealthWarden is called Data Loader. It is a standalone web service separated from the main application code. It has a defined interface that adds a level of indirection between HealthWarden and abstracts away the particularity of the integrated information system (typically an electronic healthcare record system). This approach allows multiple data loaders at the same time without the need to change the main application code when deploying to the new environment.

The solution is well integrated with a healthcare data format called HL7 FHIR[28]. Application data are synchronized incrementally in frequent intervals from the application database into FHIR resources on the FHIR server. This allows seamless integration between HealthWarden and any institution that uses FHIR already. Resources that are transferred during the synchronization are Patient, Questionnaire, and Questionnaire Response. Every Questionnaire Response is then interlinked with the Patient and parent Questionnaire. The mapping of resources is strictly limited to built-in attributes introduced in the mentioned data standard. Data provided by HealthWarden are identified by the presence of HealtwWarden system identifier[29] in mentioned resources, which is a common practice for third-party solution providers.

Additional FHIR resources, namely Encounter[30], Observation[31], and CarePlan[32], will be added as well. Encounter is an equivalent of HealthWarden ward round occurrence. It can be linked to multiple FHIR Questionnaires signaling that they were answered within a single ward round occurrence. HealthWarden ward round, on the other hand, can be perceived as CarePlan. This is suitable for plans following specific practice guidelines. Last but not least is the Observation. This resource is suitable for storing measured data from wearables, which makes it ideal for storing vital monitoring data of a patient. After implementing all three mentioned resources, all data produced by HealthWarden will be available on the integrated FHIR server.

The architecture design allows adding custom integrations easily. For example, a Czech hospital currently using Healthwarden required following custom integrations. The hospital's employees are using a custom authentication system and thus their private data, including password hashes, are not stored on the HealthWarden's database server, but rather HealthWarden passes authentication requests to the hospital's internal authentication system through a message queue. This way, hospital employees do not need additional accounts and can use the same usernames and passwords as in their other systems.

The second integration used in the hospital is SeaCat[33]. HealthWarden uses SeaCat technology that employs mutual TLS authentication[34] between the application and the on-premise servers through the hospital's proxy. This allows administrators to have a better outlook on which devices are connecting to the servers and be able to restrict or allow access when needed.

These are just two examples of many integrations that can be developed and distributed on a case-by-case basis.

2.7 Deployment and Scalability

The application can be deployed both on-premise and in the cloud. There are no system/architecture limitations as the application can be easily containerized and deployed on any containerization platform, such as Docker.

As of now, the server is distributed as multiple Docker images allowing maximal flexibility. Moreover, using Docker images guarantees that the application behaves the same during each deployment in any environment. However, when there's more flexibility needed, HealthWarden can be deployed directly as a set of WAR deployments35 to existing compatible servers.

Moreover, thanks to HealthWarden's architecture, the server is stateless and all data and state are persisted in the database. This allows the horizontal replication of the server instances in order to ensure high availability across many servers in any situation.

The usage of Docker containers and Traefik reverse proxy for TLS termination[36] and load balancing is recommended.

2.8 Extensibility

The basic focus of HealthWarden is on cardiovascular procedures, especially TAVI[37], electrical cardioversion[38], catheter ablation[39], or even as a follow-up for patients who encountered a myocardial infarction[40]. It has built-in protocol templates and care plans for the post-treatment of these procedures. However, the application is fully modular and customizable, and adding new protocols and care plans is easy and straightforward. It is also possible to extend the number of monitored vitality indicators. The application can read any available data from native smartphone health providers. Thus HealthWarden can be extended for a large number of procedures and treatments and can serve a large number of patients.

Additional parts of the application can be added in the future. Educational materials, community support, or even a social-networking between patients with the same diagnoses can be integrated within the HealthWarden application.

HealthWarden is also extendable in a sense of its seamless integration into hospitals' information systems. Thanks to the modularity of the backend part, it is no problem to add another module that will connect to a hospital's database or other services.

2.9 Direct and Indirect Costs

The direct costs connected with HealthWarden are the cost of the infrastructure, support, and license fee. The infrastructure costs depend on the size of the hospital and the number of patients actively using HealthWarden simultaneously, which then yields the hardware requirements. Deployment type (on-premise or in the cloud) is also crucial when estimating the costs. However, the hardware requirements of HealthWarden are low in general.

Support can be provided on different levels and availability, including a 24/7 L3 support[41]. The cost of the support depends on the specific conditions of the partner institution, with the same holding also for the license fee.

One of the possible drawbacks of tools like HealthWarden are the indirect costs connected with such tools. Those costs consist mainly of the increased demands put on doctors, nurses, and other healthcare professionals and hospital staff in general. This is not the case with HealthWarden.

One of the key features of HealthWarden is giving the doctor full control over the tool when needed while being able to function independently requiring zero or just absolutely minimal intervention from the doctor. Of course, there is some initial setup and fine-tuning of all care plans, alerts, templates, etc., but once everything is set up, HealthWarden can run on its own without putting any extra demands on the doctor. It is recommended that a nurse or an administrative worker actively manages HealthWarden, which creates some additional administrative work, but at the same time, HealthWarden releases much of the administrative workload currently put on nurses and administrative workers, so the net asset on administrative work is positive. Altogether, HealthWarden has minimal indirect costs which are more than balanced by the significant direct benefits of HealthWarden.

3 Current State and Further Development

At the moment, HealthWarden is in a beta testing phase. It is being deployed in one hospital in the Czech Republic for testing and initial fine-tuning of the application, while a clinical study is being prepared. This phase is supposed to take about half a year, after which the first fully working version of HealthWarden should be released.

3.1 Further Development

In the current state, HealthWarden is simply a tool helping doctors exercise their standard care more effectively. The entire application behavior depends on doctors' decisions. All virtual ward rounds, vitality monitoring, alerts, and just the entire workflow of the application are based on doctors' decisions and the doctors are responsible for the correct setup of cure plans. This allows for maximal transparency of HealthWarden's behavior, which is crucial from the legal and patient safety point of view. However, the potential of HealthWarden goes far beyond that. With the help of artificial intelligence, HealthWarden can not only ensure the most effective care by the current standards, but it can also improve those standards.

There are there major areas where AI can help significantly improve HealthWarden. The first is the creation of the cure plan itself. When creating it, the doctors use their best knowledge and optimal methods as applied in clinical practice. However, these methods are, by their definition, rather general and non--personalized. Doctors can use guidelines that are recommended for a given age group, sex, comorbidities, etc., but they are, as humans, not able to create personalized cure plans for each patient. On the other hand, a machine is. Based on historical results, an AI-powered tool can create personalized cure plans for each individual patient that can optimize the results for that particular patient, further increasing the beneficial effect of HealthWarden.

This can be further improved by dynamically updating the cure plan during the treatment using the concept of adaptive testing[42]. This concept examines patients' results on the fly and based on those results updates the future cure plan. It can then for example modify the frequency of virtual ward rounds, add some extra questions, or drop those questions that are not necessary anymore. This can help not only to further optimize the care for each patient but also improve the adherence of the patients, as this dynamic optimization will in most cases lead to a simplified cure plan with fewer demands on the patient.

Last but not least, a concept known as anomaly detection[43] an be extremely useful when treating patients, especially when identifying possible complications. In this concept, a large amount of data is examined and searched for those that can be labeled as anomalous. In the case of HealthWarden, patients with such an anomaly can potentially develop complications. Early detection of such patients and their proper examination by a doctor can then prevent such complications to happen.

The development of the above-mentioned advanced features is non-trivial not only from the technical point of view but also from the medical and ethical point of view. Modern AI methods, such as neural networks, often do not allow for easy and clear interpretation of their results and their recommendations and thus often resemble a black box. It is thus essential to thoroughly test all such methods and perform robust clinical studies to make absolutely sure that no harm to the patients will be caused.

4 Conclusion

In this paper we presented HealthWarden, an innovative solution that helps doctors discharge patients earlier from the hospital, lower the readmission rates, release the administrative load from nurses and non-medical hospital staff, and most importantly, help patients achieve the best possible clinical results.

It focuses on post-treatment care, as this is a well-described and documented part of medicine with clear guidelines and acknowledged best practices. It introduces a concept of a virtual ward round when the patient is guided through simple tasks, checks, and questions similar to a real ward round. This concept helps to increase the adherence to the treatment of the patients while monitoring their progress at the same time.

In the paper, the current state of the application was described and future plans were introduced. The tool has entered into the beta testing phase, which should last about half a year. This time will be used to fine-tune the application and optimize it so that it can bring its full benefits to a number of hospitals not only in the Czech Republic and to improve healthcare in general.

References

[1.] United Nations 2022 Revision of World Population Prospects https://population.un.org/wpp/

[2.] World Population Prospects 2022: Summary of Results https://www.un.org/development/desa/pd/content/World-Population-Prospects-2022

[3.] Sinbukhova, Elena & Shimanskiy, V. & Tanyashin, S. & Shevchenko, K. & Poshataev, Vladimir & Abdurakhimov, F. & Lubnin, A.. (2022). Remembering what the doctor said: how much of medical information will the patient remember?. Russian journal of neurosurgery. 23. 50-60. 10.17650/1683-3295-2021-23-4-50-60.

[4.] Kessels RP. Patients' memory for medical information. J R Soc Med. 2003 May;96(5):219-22. doi: 10.1177/014107680309600504. PMID: 12724430; PMCID: PMC539473.

[5.] In some cases, HealthWarden can be employed even before the treatment to help the patients prepare for it correctly, thus the label "post-treatment" can be a bit confusing.

[6.] Kotlin - programming language https://kotlinlang.org/

[7.] Spring framework https://spring.io/projects/spring-framework

[8.] PostgreSQL https://www.postgresql.org

[9.] Java Persistence API https://www.oracle.com/java/technologies/persistence-jsp.html

[10.] Hibernate https://hibernate.org/

[11.] Database Systems: A Practical Approach to Design, Implementation, and Management by Thomas Connolly and Carolyn Begg - https://www.researchgate.net/publication/228831514_Database_systems_A_practical_approach_to_design_implementation_and_management

[12.] REST - REpresentational State Transfer - https://restfulapi.net/

[13.] OpenAPI Specification - https://swagger.io/specification/

[14.] JUnit - https://junit.org/junit5

[15.] AssertJ - https://joel-costigliola.github.io/assertj

[16.] React https://reactjs.org/

[17.] React Native https://reactnative.dev/

[18.] Apple HealthKit https://developer.apple.com/documentation/healthkit

[19.] GoogleFit https://www.google.com/fit/

[20.] GarminConnect https://www.garmin.com/en-US/p/125677

[21.] Samsung Health https://www.samsung.com/cz/apps/samsung-health/

[22.] OWASP Top Ten https://owasp.org/www-project-top-ten/

[23.] bCrypt - https://www.usenix.org/legacy/events/usenix99/provos/provos_html/node1.html

[24.] TOTP: Time-Based One-Time Password Algorithm - https://www.ietf.org/rfc/rfc6238.txt

[25.] JSON Web Tokens (JWT) - https://datatracker.ietf.org/doc/html/rfc7519

[26.] Spring Security - https://spring.io/projects/spring-security

[27.] Docker - https://www.docker.com

[28.] Fast Healthcare Interoperability Resources https://hl7.org/fhir/

[29.] FHIR identifier https://www.hl7.org/fhir/datatypes.html#Identifier

[30.] FHIR Encounter - https://build.fhir.org/encounter.html

[31.] FHIR Observation - https://build.fhir.org/observation.html

[32.] FHIR CarePlan - https://build.fhir.org/careplan.html

[33.] SeaCat, Cyber-security and data privacy platform for mobile and IoT applications - https://teskalabs.com/products/seacat

[34.] Siriwardena, P. (2014). Mutual Authentication with TLS. In: Advanced API Security. Apress, Berkeley, CA. - https://doi.org/10.1007/978-1-4302-6817-8_4

[35.] JavaTM Servlet 2.4 Specification - https://jcp.org/en/jsr/detail?id=154

[36.] Traefik, The Cloud Native Application Proxy - https://traefik.io/traefik/

[37.] Transcatheter aortic valve implantation: current and future approaches - https://doi.org/10.1038/nrcardio.2011.164

[38.] Electrical cardiversion - https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/electrical-cardioversion

[39.] Barranco, C. High cure rate for permanent atrial fibrillation with catheter ablation. Nat Rev Cardiol 2, 554 (2005). https://doi.org/10.1038/ncpcardio0340

[40.] Myocardial infarction - https://www.ncbi.nlm.nih.gov/books/NBK537076/

[41.] https://www.teceze.com/difference-between-l1l2-and-l3-technical-support

[42.] WEISS, D.J. and KINGSBURY, G.G. (1984), APPLICATION OF COMPUTERIZED ADAPTIVE TESTING TO EDUCATIONAL PROBLEMS. Journal of Educational Measurement, 21: 361-375. https://doi.org/10.1111/j.1745-3984.1984.tb01040.x

[43.] Hawkins, Douglas M. Identification of outliers. Vol. 11. London: Chapman and Hall, 1980

Contact

Ing. Tomáš Kouřim
Faculty of Nuclear Sciences
and Physical Engineering
Czech Technical University in Prague
Mild Blue s.r.o.

tomas.kourim@mild.blue
corresponding author

Mgr. Marek Polák, Ph.D.
Mild Blue s.r.o.

marek.polak@mild.blue

Ing. Jan Skála
Mild Blue s.r.o.

jan.skala@mild.blue

Ing. Anastasia Surikova
Mild Blue s.r.o.

anastasia.surikova@mild.blue

Ing. Lukáš Forst
Faculty of Electrical Engineering
Czech Technical University in Prague
Mild Blue s.r.o.

lukas.forst@mild.blue

MUDr. Alžběta Kouřimová
General University Hospital in Prague
Mild Blue s.r.o.

alzbeta.kourimova@mild.blue

Nikola Zadorozhny
Faculty of Information Technology,
Czech Technical University in Prague
Faculty of Mathematics and Physics,
Charles University
Mild Blue s.r.o.

nikola.zadorozhny@mild.blue