doi: 10.21164/pomjlifesci.1054

The potential of implementing electronic health record applications using cloud-based low-code platforms

Aleksandra Wiraszko, Wojciech Chlewicki[A], Katarzyna Cichoń[B] ✉

West Pomeranian University of Technology in Szczecin, Faculty of Electrical Engineering, Sikorskiego 37, 70-313 Szczecin, Poland

[A] ORCID: 0000-0002-9564-3062; [B] ORCID: 0000-0002-8986-6358

✉ katarzyna.cichon@zut.edu.pl

ABSTRACT
This work attempts to answer how the latest cloud computing technologies and low-code software development tools can support modern health care. The main scope of the work was the design and implementation of an application using the Microsoft Power Platform environment, including Microsoft Power Apps and Microsoft Dataverse. Testing the implemented application formed the basis for discussion and enabled the formulation of potential benefits, limitations, and possible future enhancements of the proposed solution. The development of this project demonstrated great potential for using Microsoft Power Apps. It was considered important for expanding telemedicine systems, improving access to medical information, and optimizing working time.

Keywords: electronic health record; telemedicine; cloud computing; efficient development.

INTRODUCTION

Dynamic demographic changes, and above all, increasing life expectancy, contribute to a surge in the number of patients, which poses a challenge to the healthcare sector in terms of effective management of their data [1, 2, 3, 4]. Technological progress, especially in cloud computing services, is a response to the growing need for information technology (IT) solutions that provide fast, flexible, and secure access to data. On the other hand, there is a need to develop medical informatics software quickly and efficiently. An answer to such a need could be a low-code development strategy.

Low-code tools may reduce time and coding barriers to clinician participation in software development at medical sites. Many low-code development platforms have already been introduced. One of the most popular tools is Microsoft Power Platform, which is a development tool in the cloud [5]. One of the tools available in the Microsoft Power Platform is Power Apps, which can be used to create business applications, including healthcare applications. Power Apps also integrates with other Microsoft 365 tools, such as Dataverse, SharePoint, and Dynamics 365, making it easier to use various data sources and improve information management. The platform offers robust security mechanisms for regulating access rights to applications and data. It also enables flexible application development to keep up with the evolving requirements of the healthcare field. Applications created in Power Apps can be adapted to various devices, allowing easy access to patient data from desktop computers, tablets, and mobile devices.

This work attempts to answer how the latest cloud computing technologies and low-code tools can support modern health care [6]. In this article, existing technologies that use cloud services to implement electronic patient records were analyzed. The work also included the implementation and testing of a sample application with this type of service. The article presents the application implementation process along with the database design, as well as conclusions.

MATERIALS AND METHODS

Choice of the platform

At the beginning of this study, it was crucial to establish the system requirements. The primary requirements are as follows:
- user information management: provide the ability to view, modify, add, and delete personal data, vital signs, diagnoses, prescribed medications, and treatment outcomes;
- access management: ensure that different authorization levels are in place for various medical and administrative personnel groups, such as physicians and technicians;
- prescription management: enable the creation of prescriptions, including dosage and medication descriptions;
- appointment schedule and notifications: set appointments and reminders for patients with upcoming appointments;
- monitoring of introduced modifications;
- backups of data to avoid any loss;
- data security: compliance with national and international regulations for medical data protection, and encryption of stored and transmitted data;
- efficiency: ensuring smooth operation, even under heavy data load;
- scalability: the ability to maintain continuous performance as the number of patients increases;
- availability: providing access to data at any time and implementing fault tolerance mechanisms to minimize the effects of possible unforeseen disruptions in the application's operation;

- usability and intuitiveness: availability on multiple platforms and an easy-to-use graphical user interface (GUI);
- version and update management: implementation of updates without interrupting operation, providing a history of changes, and the ability to restore previous versions.

Microsoft Power Platform was chosen to conduct this study due to its extensive offerings, including a comprehensive collection of existing manuals, simplicity, and many ready-to-use components [7]. The ability to preview changes optimizes the creation and testing process. Additionally, the platform meets the requirements for security, scalability, and performance.

The database

The Microsoft Dataverse database was chosen to store the data, as it integrates well with Power Platform tools like Power Apps and Power Automate. This solution enables the centralization of data that can be stored in a relational structure, which is useful when implementing an electronic health record. Dataverse offers advanced security mechanisms, similar to Power Apps, which allow access control to sensitive data at various levels. It is also flexible enough to adapt to the changing requirements of the medical informatics system, and scalable to accommodate application development. Furthermore, Dataverse enables data analysis by generating reports that can be used to assess treatment efficiency.

The structure of the designed database consisted of 6 tables. Graphically, it is presented in Figure 1.

Implementation of the application logic

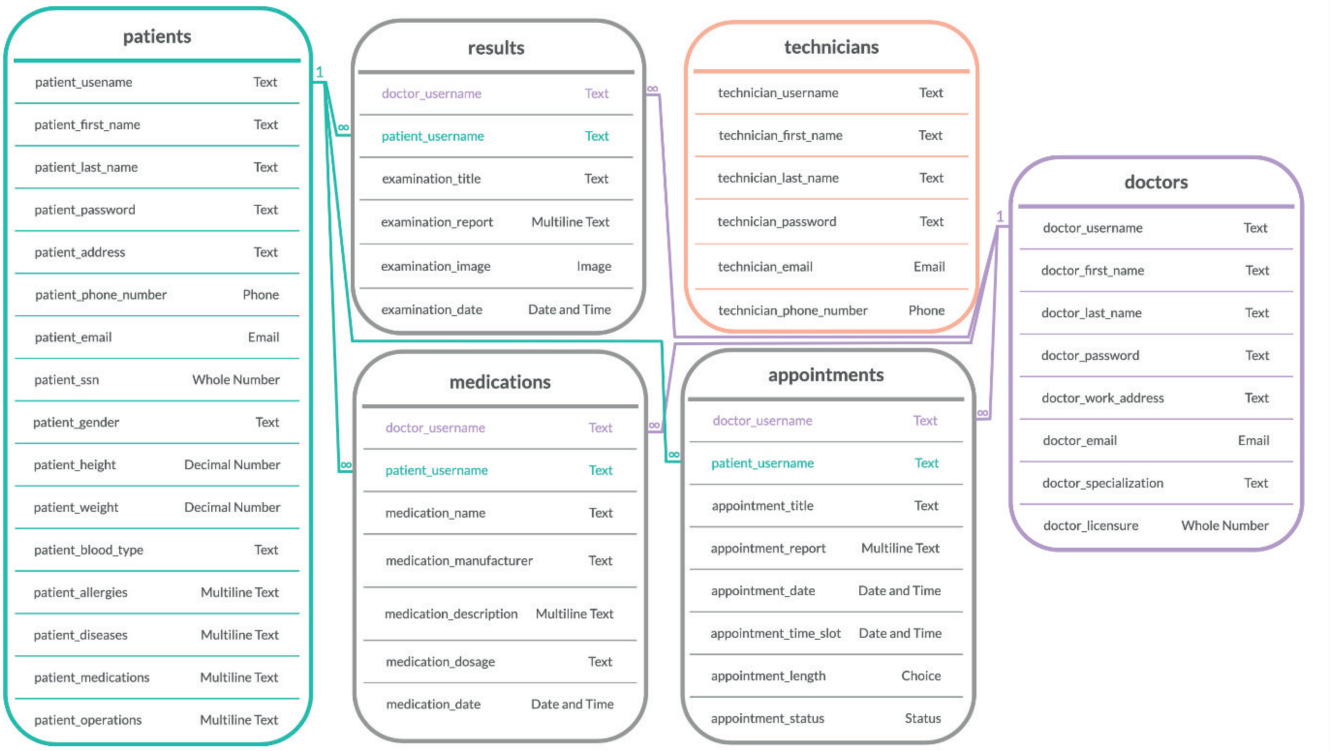
The logic of any Power Apps application is implemented in the form of functions [5], and consequently, this strategy was utilized in this work. Basic error-handling methods were also incorporated to maintain the consistency of data inputted into the database through the application.

After users enter their name and password to log in, their information is checked for accuracy. If there are any errors or missing information, an error message will be displayed. If the information is correct, a window will appear with a field for entering the verification code that was sent to the email address associated with the account. The "Login" button is responsible for checking the accuracy of the code and will take the user to the main application screen.

When a new account is created by clicking the "Register" button, the system checks if all required fields are correctly filled out. If so, the data is saved to the patient's table. The "Set()" function then assigns names to the variables and generates a 6-digit code, which is sent to the email address provided in the form. Finally, the "Reset" function clears all completed fields. The source code for this process is associated with the click event of the "Register" button. For physician or administrative users, the code is adjusted accordingly.

To update their information, patients can simply click on the "Edit data" button located at the bottom of the main application screen. Once they have made the necessary changes, they can save their modifications by clicking on the "Save data" button.

A user logged in as a patient can make an appointment with a selected specialist at an available date. After clicking the "Register" button, a window with a summary of the registered



patients	
patient_usename	Text
patient_first_name	Text
patient_last_name	Text
patient_password	Text
patient_address	Text
patient_phone_number	Phone
patient_email	Email
patient_ssn	Whole Number
patient_gender	Text
patient_height	Decimal Number
patient_weight	Decimal Number
patient_blood_type	Text
patient_allergies	Multiline Text
patient_diseases	Multiline Text
patient_medications	Multiline Text
patient_operations	Multiline Text

results	
doctor_username	Text
patient_username	Text
examination_title	Text
examination_report	Multiline Text
examination_image	Image
examination_date	Date and Time

medications	
doctor_username	Text
patient_username	Text
medication_name	Text
medication_manufacturer	Text
medication_description	Multiline Text
medication_dosage	Text
medication_date	Date and Time

technicians	
technician_username	Text
technician_first_name	Text
technician_last_name	Text
technician_password	Text
technician_email	Email
technician_phone_number	Phone

appointments	
doctor_username	Text
patient_username	Text
appointment_title	Text
appointment_report	Multiline Text
appointment_date	Date and Time
appointment_time_slot	Date and Time
appointment_length	Choice
appointment_status	Status

doctors	
doctor_username	Text
doctor_first_name	Text
doctor_last_name	Text
doctor_password	Text
doctor_work_address	Text
doctor_email	Email
doctor_specialization	Text
doctor_licensure	Whole Number

FIGURE 1. The structure of the database, including the table and field names, as well as the table relations

visit and the “Confirm registration” button appears. For reference, an example function associated with the “Register” button is shown below:

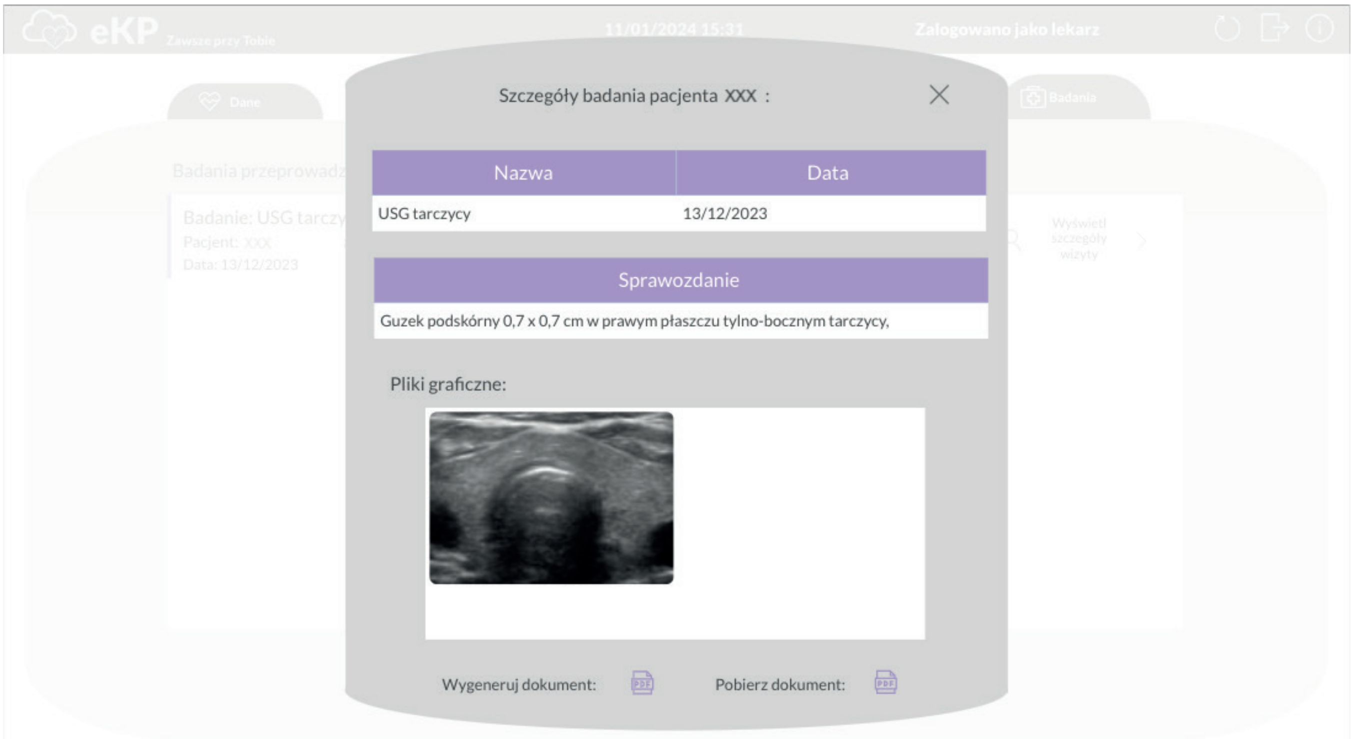
```
Text(
“Specjalizacja: “ & cb_doctor.Selected.doctor_specialization & “
“ & “Lekarz: “ & cb_doctor.Selected.doctor_first_name & “ “ & cb_doctor.Selected.doctor_last_name & “
“ & “Data: “ & Text(
DateTimeValue(available_app_gallery.Selected.appointment_date),
“dd/mm/yyyy”
 ) & “
“ & “Godzina: “ & Text(
DateTimeValue(varSelectedApt.appointment_time_slot),
“hh:mm”
) & “ - “ & Text(
DateAdd(
DateTimeValue(varSelectedApt.appointment_time_slot),
varSelectedApt.appointment_length,
TimeUnit.Minutes
),
“hh:mm”
) & “
“ & “Potwierdzenie rezerwacji wizyty zostanie wysłane na e-mail.”
)
```

```
Patch(
appointments,
LookUp(
appointments,
appointment_date = varSelectedApt.date
),
{
patient_first_name: currentUserFirstName,
patient_last_name: currentUserLastName,
patient_username: currentUserUsername,
appointment_status: varBookedStatus
}
);
Office365Outlook.SendEmail(
LookUp(
patients,
patient_username = currentUserUsername
).patient_email,
“eKP Zawsze przy Tobie: Nowa wizyta”,
“Wizyta u specjalisty “ & varSelectedApt.doctor_first_name & “ “ & varSelectedApt.doctor_last_name & “ w dniu “ & varSelectedApt.appointment_date & “ została potwierdzona!
Zaloguj się na Twoje konto eKP, aby zobaczyć szczegóły.
Do zobaczenia!”
);
   UpdateContext({show_popup_confirm: false});
UpdateContext({show_popup_new_visit: false})
```

A user logged in as a physician can edit visit details, issue prescriptions, and create examination summaries. Windows that allow viewing patient data and details of a visit or examination have icons for triggering the generation of documents in PDF format at the bottom of the screen (refer to Fig. 2). After clicking the first icon, a screenshot of the data displayed on the screen will be created. The second icon functions as a control for downloading the generated file.

The primary responsibility of the administration staff, as implemented in the presented software, is to enter available doctors’ appointments into the “Appointments” table. To complete this task, the employee selects a specialization and a specific specialist and assigns them selected working hours and the duration of the visit. The system allows for half-hourly or hourly visits and displays available times in a table. Furthermore, if the start time of work is later than the end time, a warning notification will be displayed. The “Enter available dates of the selected doctor” button divides the available time into appointment durations. The system then creates records in the database for specific appointment dates with the status “Available”. The implementation of this functionality is as follows:

```
If(
!IsBlank(cb_specialization.Selected.Value) && !IsBlank(cb_doctor.Selected.Value) && !IsBlank(dd_start.Selected.Value) && !IsBlank(dd_end.Selected.Value),
ForAll(
Sequence(
DateDiff(
dd_start.Selected.myDate,
dd_end.Selected.myDate,
TimeUnit.Hours
) * If(
dd_length.Selected.Length = 30,
2,
1
),
0
) As DATA,
Patch(
appointments,
Defaults(appointments),
{
appointment_title: Text(“Wizyta u “ & cb_specialization.Selected.Value),
appointment_date: DateValue(
Text(
dateSelected,
  “dd/mm/yyyy”
)
),
doctor_username: (LookUp(
varSelectedDoctorData,
last_name = varDoctor
).doctor_username),
appointment_status: “Dostępne”,
```



Szczegóły badania pacjenta XXX :

Nazwa	Data
USG tarczycy	13/12/2023

Sprawozdanie

Guzek podskórny 0,7 x 0,7 cm w prawym płaszczu tylno-bocznym tarczycy,

Pliki graficzne:

FIGURE 2. View of a sample summary of a medical examination

```
patient_username: “ “,
appointment_length: dd_length.Selected.Length,
    appointment_time_slot: DateAdd(
dd_start.Selected.myDate,
dd_length.Selected.Length * DATA.Value,
TimeUnit.Minutes
)
}
    )
)
    )
```

The technique of building user interface

When designing a GUI for an electronic patient record application, it is crucial to consider the security of patient data and the availability of specific functions for different user roles. It is essential to ensure that the interfaces for patients, doctors, and administrative employees offer the appropriate level of accessibility and functionality tailored to the specific needs of each role [8]. To make the experience of using these interfaces seamless and user-friendly, each interface is designed to be intuitive and comes with a distinct color palette.

When it comes to building a GUI using Power Apps, 2 approaches can be taken. The first approach is to automatically generate the GUI based on the data structure, while the second approach is to manually assemble all GUI components and then connect them to the data source [5]. A hybrid approach that combines both methods can also be used, where the basic structure of the screens is generated automatically and then adjusted interactively to meet the needs, abilities, and expectations of each group of users.

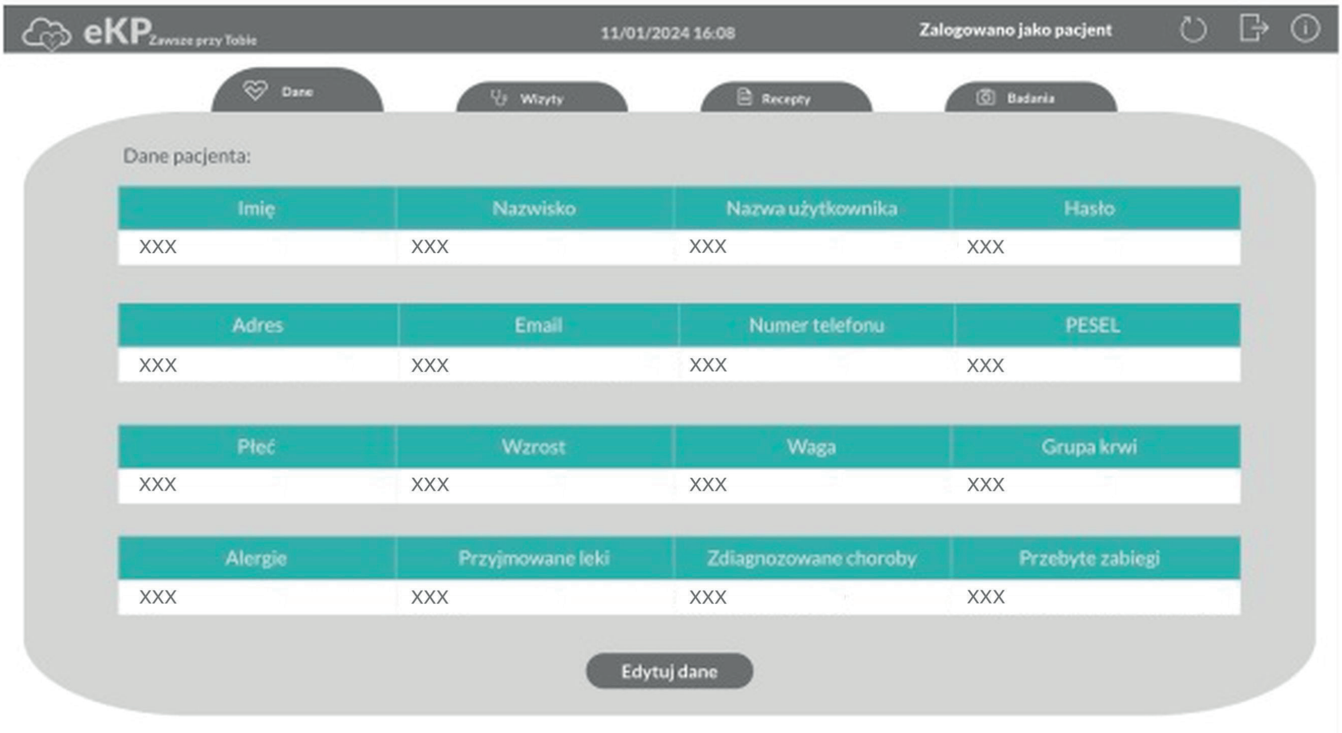
The patient can access their personal information, medical history, test results, and prescribed medications. Additionally, they can view available appointment dates and schedule or cancel appointments. The screenshot of the main user page is presented in Figure 3.

When a physician logs in, he can easily choose a patient from the list and view all of their assigned visits, complete with details. The physician can also check and modify patient data as needed. Any updates or new information must be confirmed, and the patient will receive an email notification of the changes. Additionally, each page includes a convenient option to generate and download a PDF document.

The administrative staff member operates solely within the “Appointments” window. By selecting the appropriate specialization, the specific doctor, start and end times, as well as visit duration, available dates can be inputted into the database (refer to Fig. 4). A dynamic list is present on the page, displaying appointments for the selected date on the calendar, marked with a circle. The present day is highlighted with a square, while small dots beneath the dates signify appointment dates assigned on those days.

RESULTS

The application was tested in terms of its compliance with functionality requirements. The purpose of the tests was also to identify areas that required further improvements. The



Dane pacjenta:

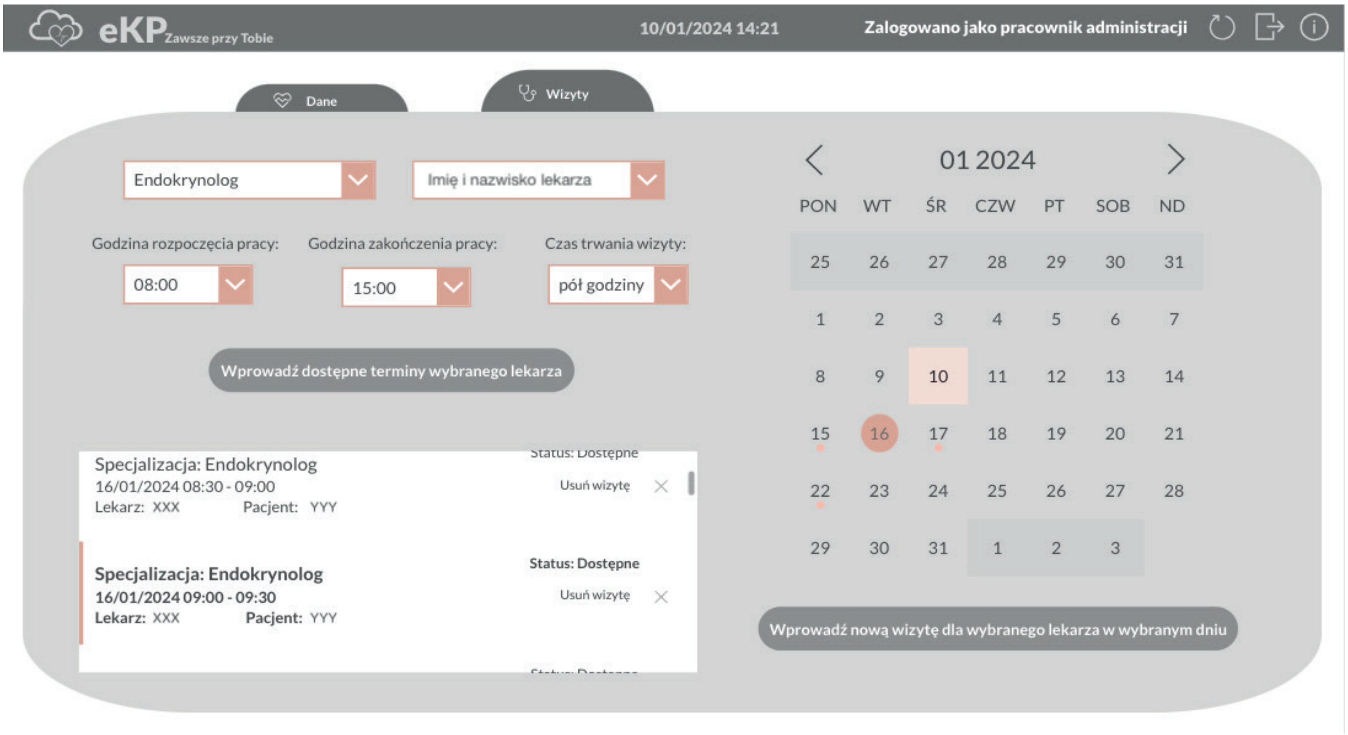
Imię	Nazwisko	Nazwa użytkownika	Hasło
XXX	XXX	XXX	XXX

Adres	Email	Numer telefonu	PESEL
XXX	XXX	XXX	XXX

Płeć	Wzrost	Waga	Grupa krwi
XXX	XXX	XXX	XXX

Alergie	Przyjmowane leki	Zdiagnozowane choroby	Przebyte zabiegi
XXX	XXX	XXX	XXX

FIGURE 3. View of the main page with patient data


eKP Zawsze przy Tobie
10/01/2024 14:21
Zalogowano jako pracownik administracji
Dane
Wizyty
Wprowadź dostępne terminy wybranego lekarza
Wprowadź nową wizytę dla wybranego lekarza w wybranym dniu

Endokrynolog

Imię i nazwisko lekarza

Godzina rozpoczęcia pracy: 08:00

Godzina zakończenia pracy: 15:00

Czas trwania wizyty: pół godziny

Status: Dostępne

Specjalizacja: Endokrynolog
16/01/2024 08:30 - 09:00
Lekarz: XXX Pacjent: YYY
Usuń wizytę ✕

Specjalizacja: Endokrynolog
16/01/2024 09:00 - 09:30
Lekarz: XXX Pacjent: YYY
Status: Dostępne
Usuń wizytę ✕

Status: Dostępne

01 2024

PON	WT	ŚR	CZW	PT	SOB	ND
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	

FIGURE 4. View of the visit schedule management screen

tests included setting up a new account, scheduling appointments, checking test results, and prescribing medications. The criteria for assessing the results were the stability, usability, and speed of the system. The application was tested by a physician and a nurse who use electronic telemedicine solutions daily. Testers were given access to the application without instructions on its functionality and successfully completed all steps necessary to use the electronic health record. Particular attention was paid to the ability to easily add, view, and edit patient data. The importance of the smooth operation of the application and ensuring reliable access to information was emphasized.

The main feedback provided was the removal of all unnecessary steps and additional options that could be unnecessarily distracting and thus complicate and extend the application's usage time. It was recommended to enable the user to go through the available functions in the fewest possible number of clicks to obtain the highest possible efficiency of the system. As a result of the consultations, the "News" tab was removed, where it was planned to post the latest notifications and health articles. All other anomalies found during the testing process have been optimized accordingly. The application should be as intuitive and easy to use as possible, even for users without specialized technical knowledge.

In realistic conditions, medical staff very often start using a given technology without prior training. Additionally, medical visits and check-ups are time-limited, hence the need to be able to use the tool immediately and learn to use the most important options as quickly as possible. Moreover, a crucial element of application development is cooperation between its creator and the implementing user to adapt the service to specific needs. The beginning of using a new system is a key time to gain experience and refine any modifications. The programmer should be open to feedback from medical professionals and respond promptly to suggested corrections.

Throughout the development of the application, it was of utmost significance to prioritize the security of medical data and safeguard against any misuse. However, in-depth tests in this specific area have not yet been conducted, and further investigation is required to ensure that the application meets all necessary security standards.

DISCUSSION

Several authors [9, 10] presented the fundamental advantages and challenges of cloud computing, as well as identified potential areas for application development. Their work recommended improvements and research to enhance telemedicine systems. This paper delves into the specifics and explores the issues surrounding the implementation of the electronic patient record application in Microsoft Power Apps, using the Microsoft Dataverse database. This solution offers numerous benefits to medical staff and patients alike. Some of the advantages of this approach include:

- flexibility and scalability (thanks to integration with Microsoft Dataverse, the application can be adapted to changing needs, new features, and the development of healthcare infrastructure);
- data security (Microsoft Power Platform offers advanced data security mechanisms, which allow you to configure different levels of access for different roles in the application, ensuring the protection of sensitive information);
- integration with other tools from the Microsoft 365 suite (this enables smooth data management and collaboration within various medical organizations);
- speed of testing (Power Apps allows the solution to evolve, which allows for smooth adaptation to current healthcare needs);
- facilitated mobile operation (the application created in Power Apps can be accessed from any electronic device with an Internet connection, which gives high mobility to users).

It is worth noting that despite several advantages, the implemented and tested application in this study has certain limitations and weaknesses that must be taken into account:

- limitations of available functions (in particular regarding the implementation of advanced algorithms or processes);
- possible performance problems (this is especially true for large databases, and performance optimization may require additional work);
- the need for strict management and control of data (to avoid problems related to its security, availability and integrity);
- dependence on the cloud supplier (may lead to decision-making restrictions regarding updates, changes in the pricing policy, or possible changes in the functionalities of the platform);
- licensing and maintenance costs (especially if the number of users is growing or more computing power is needed).

Before commencing such an application, it is crucial for organizations to carefully assess their needs, objectives, and capabilities. This will enable them to mitigate any potential drawbacks through meticulous planning, configuration, and management. Once we had successfully developed and tested the electronic patient card application in Microsoft Power Apps, utilizing the Microsoft Dataverse database, we were able to derive valuable insights about its applicability in the medical field. This methodical approach facilitated comprehension of the application's requirements and potentialities. They are as follows:

- flexibility and speed of implementation (this allows for optimal adaptation of the interface and functionality to the changing needs of the medical environment);
- the need for a thorough analysis of requirements (it is crucial to understand the needs of users, both patients, doctors, and administrative staff);
- the importance of protecting sensitive data (the implementation of the solution requires special attention to the protection of patient data);
- monitoring and optimizing performance (necessary to maintain the appropriate quality of services, especially in a situation where the number of users increases significantly);
- potential for further development of the application (through integration, adding new functionalities, and adapting to evolving needs).

When developing a solution using Microsoft Power Platform, it is important to consider the potential directions of development:

- integrations with other medical systems (ensuring interoperability requires cooperation with other telemedicine service providers, but is a key application improvement that would enable a smooth flow of data between different platforms);

- the extension of functionality (adding new modules, such as the exchange of messages between the patient and the doctor, the incorporation of a payment system for visits, or the management of medical recommendations);
- improvements in the user interface (introducing improvements that increase the usability and intuitiveness of the application);
- improving accessibility for people with disabilities (the ability to modify the contrast of the view and font size, ensuring greater legibility and customization of audio messages);
- data analysis and report generation (development of the ability to create statistics on health status, disease trends or treatment effectiveness);
- improvement of security (implementation of new security protocols and response to new challenges related to cybersecurity);
- personalization of the application (adaptation to the specifics of various medical facilities, taking into account their unique needs and procedures);
- regular testing of performance and scalability (in terms of the number of users and the size of the stored data and analysis of possible ways to optimize the operation);
- conversion of the table from Dataverse to Azure, due to a wider range of services and capabilities (in this case, the Xamarin Forms platform would be used to create the application, and the software in the Microsoft Visual Studio environment).

These conclusions highlight the significance of analyzing data, protecting it, and continuously improving applications that are based on the genuine needs and expectations of users. Implementing such solutions is an ongoing process that demands constant attention and adaptation to changing conditions.

CONCLUSION

The development of telemedicine solutions, like software utilizing an electronic patient card, plays a significant role in the advancement of healthcare systems. While analyzing the available solutions and implementing the application, several benefits were identified, but some challenges were also encountered. The efficiency of the electronic patient record was vital in enhancing access to medical information and optimizing working time. The difficulties faced during the research were mainly due to issues related to the integration of different IT systems, safeguarding sensitive data, and adapting applications to the ever-changing needs and regulations concerning patient data protection.

The developed application with an electronic patient record is an effective tool, but it needs to be continuously improved to provide a higher quality of care. Continuation of this work should focus on further improvement of the systems, implementing solutions in the field of cybersecurity, and optimization of data processing.

This work is the starting point for the low-code development and expansion of the applications utilizing an electronic patient record, taking into account the requirements for the improvement of this type of system in the context of data security and the functionality of the user interface. There is a need to further improve such systems to better serve the needs of patients and medical staff while meeting current safety standards and regulations.

REFERENCES

1. Butpheng C, Yeh KH, Xiong H. Security and privacy in IoT-Cloud--Based e-Health Systems – A comprehensive review. Symmetry (Basel) 2020;12(7):1191. doi: 10.3390/sym12071191.
2. Vaiyapuri T, Binbusayyis A, Varadarajan V. Security, privacy and trust in IoMT enabled smart healthcare system: A systematic review of current and future trends. IJACSA 2021;12(2):731-7. doi: 10.14569/IJACSA.2021.0120291.
3. Sarwar T, Seifollahi S, Chan J, Zhang X, Aksakalli V, Hudson I, et al. The Secondary Use of Electronic Health Records for Data Mining: Data Characteristics and Challenges. ACM Computing Surveys (CSUR) 2022;55(2):1-40. doi. 10.1145/3490234.
4. Hermes S, Riasanow T, Clemons EK, Böhm M, Krcmar H. The digital transformation of the healthcare industry: exploring the rise of emerging platform ecosystems and their influence on the role of patients. Bus Res 2020;13:1033-69. doi: 10.1007/s40685-020-00125-x.
5. Weston M, Bárcena Martín E. Learn Microsoft Power Apps. The definitive handbook for building solutions with Power Apps to solve your business needs – Second edition. Birmingham: Packt Publishing; 2023.
6. Aveiro D, Freitas V, Cunha E, Quintal F, Almeida Y. Traditional vs. low-code development: comparing needed effort and system complexity in the NexusBRaNT experiment. Prague: 2023 IEEE 25th Conference on Business Informatics (CBI); 2023. p. 1-10. doi: 10.1109/CBI58679.2023.10187470.
7. Power Platform. Microsoft 2025. https://www.microsoft.com/pl-pl/power-platform (9.04.2025).
8. Systems and software engineering – Design and development of information for users. International Standard. ISO/IEC/IEEE 26514:2022. Ed. 1. 2022. p. 1–76. doi: 10.1109/IEEESTD.2022.9690115.
9. Molo MJ, Badejo JA, Adetiba E, Nzanzu VP, Noma-Osaghae E, Oguntosin V, et al. A review of evolutionary trends in cloud computing and applications to the healthcare ecosystem. App Comput Intell Soft Comput 2021;1:843671. doi: 10.1155/2021/1843671.
10. Alanazi R. Analysis of privacy and security challenges in e-health clouds. IJACSA 2022;13(9):484-9. doi: 10.14569/IJACSA.2022.0130955.