

Badanie użyteczności Cyfrowej Platformy Zdrowia Młodzieży

Szymon Wiśniewski, Urszula Konopko, Nikodem Szwed,
Eryk Śliwowski, Michał Lipka, Krzysztof Konopko
Politechnika Białostocka, Koło Naukowe Systemów Wbudowanych „W Sieci Rzeczy”
k.konopko@pb.edu.pl

Summary: The paper discusses the role of tests in creating the Digital Youth Health Platform. As a result of its crucial role in ensuring a user-friendly interface, special focus was placed on usability testing. The results of these tests provide important suggestions for improvements that can be made to increase the attractiveness and usability of the platform. The platform for which the tests were conducted is being developed as part of an international project implemented under the Erasmus+ program, in which Bialystok University of Technology is one of the participants. The platform is distinguished by an advanced interface in which artificial intelligence algorithms identify and personalize selected user features, and thus support the work of specialists.

Key words: usability tests, web applications, mobile applications

1. Wstęp

Cyfrowa Platforma Zdrowia Młodzieży (ang. *Digital Youth Health Platform*) to nazwa międzynarodowego projektu realizowanego w ramach programu Erasmus+ [1]. W jego realizację zaangażowane są dwie uczelnie: Ankara Yıldırım Beyazıt Üniversitesi (Turcja) jako lider projektu i Politechnika Białostocka jako partner. Wspierają je również organizacje społeczne: RESHA z siedzibą w belgijskim Berchem oraz tureckie DGH-ARGE Yazılım Danışmanlık Enerji Eğitim İnş. San. Tic. Ltd. Şti. w Malatyi oraz Digital Robotic Life w Wan. Celem projektu jest opracowanie wirtualnego środowiska, które łączy specjalistów udzielających porad z dziedziny zdrowia, sportu czy żywienia oraz użytkowników, do których te porady są adresowane. Wiarygodność tych pierwszych sprawdzana jest na podstawie dostarczanych i weryfikowanych certyfikatów. Docelowe grupy użytkowników platformy to: studenci, doktoranci, młodzież szkolna, lekarze specjaliści, trenerzy, dietetycy, nauczyciele wychowania fizycznego itp.

Platforma umożliwia przeprowadzanie bezpłatnych audiowizualnych konsultacji medycznych, a także realizację zdalnych spotkań z dietetykami czy z trenerami personalnymi. Wyróżnia ją zaawansowany interfejs człowiek-maszyna, w którym algorytmy sztucznej inteligencji oparte na sieciach neuronowych i głębokim uczeniu maszynowym są w stanie rozpoznawać, identyfikować i personalizować wybrane cechy użytkowników i tym samym wspomagać pracę specjalistów. Platforma umożliwia dostęp dla zarejestrowanych użytkowników zarówno z popularnych przeglądarek internetowych (Opera, Microsoft Edge, Google Chrome, Mozilla Firefox), jak i aplikacji uruchamianych na urządzeniach mobilnych.

W projekcie brali udział autorzy artykułu – członkowie Koła Naukowego Systemów Wbudowanych „W Sieci Rzeczy”. Ich zadaniem było opracowanie testów funkcjonalnych [2] i niefunkcjonalnych [3] umożliwiających testowanie aplikacji webowej i mobilnej. Opracowane metody testowania uwzględniały ich specyfikę [4, 5]. W przypadku aplikacji webowych [6, 7] testy skupiały się przede wszystkim na funkcjonalnościach, kompatybilności z różnymi przeglądarkami oraz responsywności. Dla aplikacji mobilnych [8] ważne było również sprawdzenie interakcji z ekranem dotykowym oraz analiza ich działania na różnych systemach operacyjnych i wariantach urządzeń.

Aby móc zacząć korzystać z portalu, konieczne jest testowanie aplikacji przed wdrożeniem, w trakcie i po wdrożeniu. W przypadku testów opisywanych w artykule były one opracowane pod kątem wykorzystania podczas wdrażania platformy. Miało to też znaczenie przy wyborze formy prowadzenia testów [9]. Rozważane było testowanie ręczne oraz automatyczne. Celem tych ostatnich testów jest analiza działania aplikacji realizowanej w jak najkrótszym czasie przy możliwie niewielkiej ilości zasobów. Możliwe jest to dzięki wykorzystaniu odpowiednich programów, do których zaliczyć można: Selenium IDE, Selenium WebDriver, JMeter, SoapUI, SikuliX, AutoIT i Katalon Studio [10]. Celem testowania manualnego jest również wykrycie nieprawidłowości w aplikacji. Bez użycia jakichkolwiek narzędzi do testowania automatycznego sprawdza się istotne cechy aplikacji, wykonuje przypadki testowe czy generuje się raporty. Do podstawowych wad testów manualnych zaliczyć można dużą podatność na błędy, częsty brak precyzyjnego rejestrowania wyników oraz zdecydowanie dłuższy czas, zwłaszcza w przypadku trudnych do wykonania zadań ręcznych. Do zalet natomiast można zaliczyć niskie koszty (brak konieczności ponoszenia kosztów narzędzi i wdrażania procesu automatyzacji) oraz szybką i dokładną informację zwrotną. Ten rodzaj testów bardzo dobrze sprawdza się przy testowaniu małych zmian związanych z procesem wdrażania aplikacji, co przy testach automatycznych może być problematyczne. Lepiej również wychodzi w testach eksploracyjnych i użyteczności [11, 12], które badają głównie subiektywne odczucia użytkownika aplikacji.

W związku z kluczowym testowaniem użyteczności jako formę testów wybrano ich manualną realizację. Ze względu na potrzeby wdrażania platformy opracowano także manualne testy funkcjonalne oraz przeprowadzono testy wydajności. W celu ewaluacji testy te zostały zrealizowane przez członków koła (autorów artykułu), a ich analizę zaprezentowano w kolejnych rozdziałach.

2. Testy funkcjonalne

Celem testów funkcjonalnych [2] jest sprawdzenie realizacji zakładanych funkcji aplikacji. Badanie to bazuje na specyfikacji funkcjonalnej aplikacji, a jeżeli taki dokument nie istnieje, polega na konsultacjach z testerami. Nie wymaga się od nich wiedzy na temat struktury wewnętrznej i znajomości kodu, dlatego często stosuje się technikę czarnoskrzynkową do projektowania przypadków testowych. Testy funkcjonalne były kluczowym elementem wdrażania projektu, ponieważ dzięki nim możliwe było wyszukanie, raportowanie oraz wyeliminowanie defektów istotnych dla poprawnego funkcjonowania platformy. Proces testowania polegał na rozpisaniu przypadków testowych, które sprawdzały określone funkcjonalności aplikacji. Rozpisanie i późniejsza realizacja przypadków testowych dostarczyły szczegółowych informacji związanych z błędami oraz poprawkami wprowadzanymi w oprogramowaniu. Przykład formularza przygotowanego testerom wraz z przykładowymi przypadkami testowymi został pokazany w tabeli 1.

TABELA 1. Przykłady rozpisania przypadków testowych

ID	Opis	Oczekiwany rezultat	Zaobserwowany rezultat	Komentarz	Status
1.	Widok panelu startowego	panel startowy jest widoczny			poprawny niepoprawny
1.1.	Otwarcie aplikacji bez logowania	panel startowy jest widoczny			poprawny niepoprawny
1.2.	Panel startowy powinien być widoczny	panel startowy jest widoczny			poprawny niepoprawny
2.	Więcej informacji o platformie	strona <i>Więcej informacji o platformie</i> jest widoczna			poprawny niepoprawny
2.1.	Widok panelu startowego	panel startowy jest widoczny			poprawny niepoprawny
2.2.	Kliknij <i>Więcej informacji o platformie</i>	jest możliwość wciśnięcia przycisku			poprawny niepoprawny
2.3.	Strona <i>Więcej informacji o platformie</i> powinna być widoczna	strona <i>Więcej informacji o platformie</i> jest widoczna			poprawny niepoprawny
3.	Powrót z <i>Więcej informacji o platformie</i> do panelu startowego za pomocą przycisku wstecz na telefonie	panel startowy jest widoczny			poprawny niepoprawny
3.1.	Strona <i>Więcej informacji o platformie</i> jest widoczna	strona <i>Więcej informacji o platformie</i> jest widoczna			poprawny niepoprawny

ID	Opis	Oczekiwany rezultat	Zaobserwowany rezultat	Komentarz	Status
3.2.	Kliknij przycisk wstecz na telefonie	jest możliwość wciśnięcia przycisku			poprawny niepoprawny
3.3.	Panel startowy powinien być widoczny	panel startowy jest widoczny			poprawny niepoprawny
4.	Powrót z <i>Więcej informacji o platformie</i> do panelu startowego za pomocą przycisku powrót	panel startowy jest widoczny			poprawny niepoprawny
4.1.	Strona <i>Więcej informacji o platformie</i> jest widoczna	strona <i>Więcej informacji o platformie</i> jest widoczna			poprawny niepoprawny
4.2.	Kliknij przycisk powrót	jest możliwość wciśnięcia przycisku			poprawny niepoprawny
4.3.	Panel startowy powinien być widoczny	panel startowy jest widoczny			poprawny niepoprawny

ŹRÓDŁO: opracowanie własne.

Wszystkie błędy, które zostały wykryte w trakcie realizacji testów, były również wysłane do programistów odpowiedzialnych za implementację określonych funkcji aplikacji. W zgłoszeniach każdego błędu znajdował się dokładny opis zauważonych problemów, kroki postępowania, które trzeba było zrealizować, aby zgłaszany problem się pojawił, oczekiwany rezultat oraz zaobserwowany rezultat. Przykładowy raport zgłaszania błędu został pokazany na rys. 1.

<p>Możliwość stworzenia konta bez imienia i nazwiska. Imię i nazwisko nie są wymagane w rejestracji użytkownika, jednak przy wiadomości powitalnej, która jest wyświetlana na stronie „User panel”, nie wyświetla się nawet <i>username</i> użytkownika. Zostaje tylko wiadomość „Welcome, ”</p> <p>Kroki:</p> <ol style="list-style-type: none"> 1. Rejestruję użytkownika ze wszystkimi niezbędnymi informacjami bez podanego imienia oraz nazwiska. 2. Po utworzeniu konta przechodzę na stronę „User panel”. 3. Na ekranie nie wyświetla się poprawna wiadomość powitalna. <p>Oczekiwany wynik: wiadomość powitalna przy braku imienia i nazwiska powinna wyświetlać jego <i>username</i> bądź imię i nazwisko powinny być informacjami wymaganymi przy rejestracji.</p> <p>Zaobserwowany wynik: na stronie głównej wyświetla się wiadomość „Welcome, .”, a informacje takie jak imię i nazwisko nie są obowiązkowe.</p>
--

RYŚ. 1. Przykładowy raport zgłaszania błędu

ŹRÓDŁO: opracowanie własne.

3. Testy użyteczności

Testy użyteczności są przykładem testów niefunkcjonalnych [3], które skupiają się na wszystkich aspektach oprogramowania niezwiązanych bezpośrednio z funkcjonalnościami, np. na wydajności, kompatybilności, użyteczności [11, 12], niezawodności czy bezpieczeństwie. Tego rodzaju testy mają kluczowe znaczenie dla satysfakcji użytkowników. Opracowane testy niefunkcjonalne dotyczyły głównie subiektywnych odczuć zwykłego użytkownika. Najważniejszym aspektem tych testów było sprawdzenie estetyki interfejsu graficznego panelu. Udział w nich wzięło 20 studentów i studentek studiów I i II stopnia studiujących na różnych wydziałach Politechniki Białostockiej. Do każdego zgłaszanego błędu dodawano zdjęcia je odzwierciedlające. Przykładowy raport błędu został pokazany na rys. 2.

Nieprzeskalowana tabelka na stronie „My specialists”
Tabelka znajdująca się w zakładce „My specialists”
przycina się przy ostatniej kolumnie

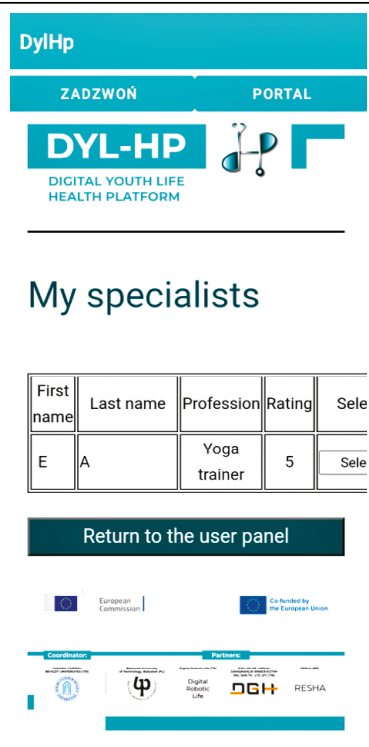
Kroki:

1. Będąc na stronie „User panel”, przechodzę do strony „Specialist search” poprzez wciśnięcie przycisku „Search a specialist”.
2. Wyszukuję jednego z specjalistów, zaznaczając konkretną kategorię, oraz przechodzę na stronę „Specialist search results” poprzez wciśnięcie przycisku „Search” (przykładowe kategorie: Profession: yoga trainer, Country: Poland, Language: polski).
3. Przechodzę do strony „Your selected professional” poprzez wciśnięcie przycisku „Select”.
4. Dodaję specjalistę do swojej listy poprzez wciśnięcie przycisku „Add this specialist to your list”.
5. Będąc na stronie „The specialist was added to your list”, przechodzimy na stronę „My specialists” poprzez wciśnięcie przycisku „Go to your specialist list”.

Oczekiwany wynik: tabelka powinna mieścić się cała na szerokości telefonu.

Zaobserwowany wynik: tabelka nie jest całkowicie widoczna, ponieważ ucina się w połowie ostatniej kolumny.

Zdjęcie do tego problemu oznakowane jest jako Bug1.1.png



RYS. 2. Przykładowy raport błędu

ŹRÓDŁO: opracowanie własne.

Oprócz estetyki w ocenie użytkowej ważną rolę odgrywają też łatwość i wygoda, z jaką użytkownik posługuje się udostępnianym przez aplikację interfejsem. Dlatego też sporządzona została instrukcja, na podstawie której testerzy mogli oceniać dostęp i wygodę realizacji poszczególnych funkcji zarówno w aplikacji mobilnej, jak i webowej.

Ocenianie odbywało się w skali od jednego do pięciu, przy czym ocena jeden uważana była za najgorszą, a pięć za najlepszą. Opracowana ankieta z pytaniami dla testerów została pokazana w tabelach 2 i 3.

TABELA 2. Ankieta z pytaniami dla testerów (pytania 1–6)

Pytania	Średnia (aplikacja mobilna)	Średnia (aplikacja webowa)	Całkowita średnia
1. Strona główna			
1.1. Czy wszystkie przyciski są łatwo dostępne, widoczne oraz czy spełniają swoją funkcjonalność?	3	3,5	3,25
1.2. Czy strona główna wygląda estetycznie?	2	2,5	2,25
Całkowita średnia	2,5	3	2,75
2. Proces rejestracji			
2.1. Czy jasno jest zadeklarowane, jakie informacje powinny zostać umieszczone w formularzu rejestracji?	5	5	5
2.2. Czy strona rejestracji wygląda estetycznie?	4,5	4	4,25
Całkowita średnia	4,75	4,5	4,63
3. Proces logowania			
3.1. Czy jasno jest zadeklarowane, jakie informacje powinny zostać umieszczone w formularzu logowania?	5	5	5
3.2. Czy strona rejestracji wygląda estetycznie?	5	4,5	4,75
Całkowita średnia	5	4,75	4,9
4. Panel użytkownika			
4.1. Czy wszystkie przyciski są łatwo dostępne, widoczne oraz czy spełniają swoją funkcjonalność?	3,5	4	3,75
4.2. Czy strona panelu użytkownika wygląda estetycznie?	3,5	3	3,25
Całkowita średnia	3,5	3,5	3,5
5. Proces edycji profilu			
5.1. Czy wszystkie informacje o profilu są łatwo dostępne?	3,5	4,5	4
5.2. Czy wprowadzanie zmian w zakresie pisemnym jest intuicyjne?	4,5	3,5	4
5.3. Czy wprowadzanie zmian w zakresie multimedialnym jest intuicyjne?	3,5	4,5	4
5.4. Czy strona edycji profilu wygląda estetycznie?	2,5	3,5	2,5
Całkowita średnia	3,5	3,75	3,63

Pytania	Średnia (aplikacja mobilna)	Średnia (aplikacja webowa)	Całkowita średnia
6. Proces wyszukiwania specjalisty			
6.1. Czy wyszukiwanie strony specjalisty jest intuicyjne?	4,5	3	3,75
6.2. Czy filtry, po których wyszukiwany jest specjalista, są przydatne?	3	2,5	2,75
6.3. Czy preferencje wybrane podczas wyszukiwania prezentują się prawidłowo w wynikach?	4,5	4,5	4,5
6.4. Czy strona specjalisty dostarcza jasnych i łatwo dostępnych informacji na temat wyszukanego specjalisty?	3,5	3,5	3,5
6.5. Czy strona specjalisty wygląda estetycznie?	3,5	4,5	4
Całkowita średnia	3,8	3,6	3,67

ŹRÓDŁO: opracowanie własne.

TABELA 3. Ankieta z pytaniami dla testerów (pytania 7–9)

Pytania	Średnia (aplikacja mobilna)	Średnia (aplikacja webowa)	Całkowita średnia
7. Proces umawiania wizyty			
7.1. Czy strona do umawiania wizyt jest łatwo dostępna?	4	4	4
7.2. Czy wybieranie terminu odbycia wizyty odbywa się w intuicyjny sposób?	4	4,5	4,25
7.3. Czy informacje o umówionej wizycie są jasno zadeklarowane na stronie?	4,5	4,5	4,5
7.4. Czy strona umawiania wizyty wygląda estetycznie?	2,5	3,5	3
Całkowita średnia	3,75	4,13	3,94
8. Proces realizacji wizyty			
8.1. Czy dostanie się na umówioną uprzednio wizytę jest intuicyjne?	1,5	2,5	2
8.2. Czy informacje o rozpoczętej wizycie są łatwo dostępne?	4	3,5	3,75
8.3. Czy funkcjonalności związane z przekazem informacji audio oraz przez czat tekstowy są intuicyjne?	4	2,5	3,25
8.4. Czy panel ocenienia specjalisty po spotkaniu w jasny sposób przekazuje kryteria jego oceniania oraz czy ma rzetelne odniesienie do oceny specjalisty?	3	4,5	3,75
8.5. Czy strona realizacji wizyty wygląda estetycznie?	4	5	4,5
Całkowita średnia	3,3	3,6	3,45
9. Proces realizacji rozmów			
9.1. Czy strona do realizacji połączeń audio/video jest łatwo dostępna?	3,5	3	3,25

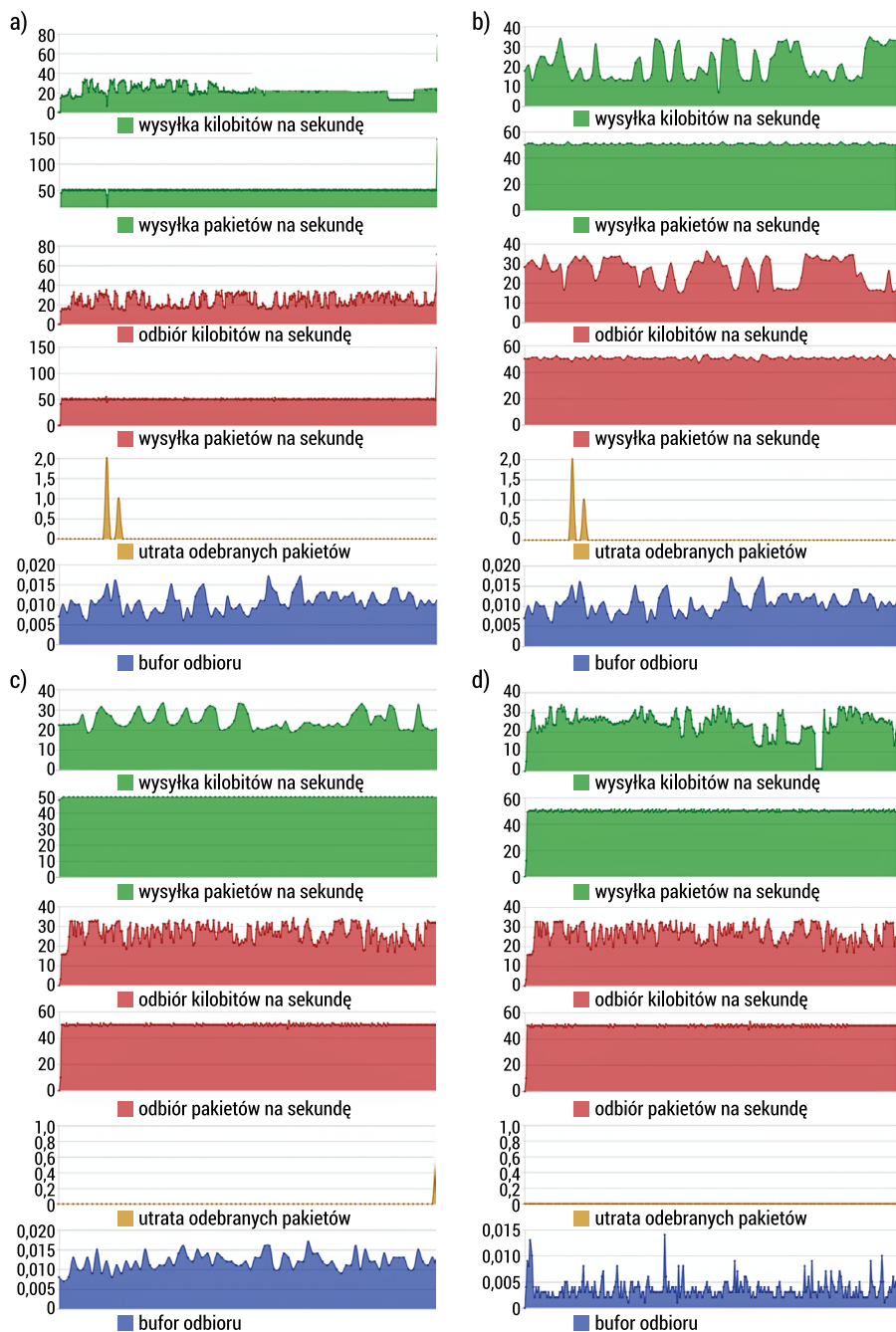
Pytania	Średnia (aplikacja mobilna)	Średnia (aplikacja webowa)	Całkowita średnia
9.2. Czy czat oraz nawiązywanie połączeń są intuicyjne?	3,5	4,5	4
9.3. Czy wszelkie ustawienia dotyczące urządzeń takich jak mikrofon czy słuchawki są łatwo dostępne i intuicyjne?	3	3,5	3,25
9.4. Czy strona realizacji rozmów wygląda estetycznie?	4,5	3,5	4
Całkowita średnia	3,63	3,63	3,63
Całkowita średnia aplikacji	3,75	3,83	3,79

ŹRÓDŁO: opracowanie własne.

Wyniki zawarte w tabelach 2 i 3 można analizować na wiele sposobów. Jednym z nich jest porównanie otrzymanych wartości średnich w poszczególnych kategoriach. Z ankiety wynika, że najniższą średnią uzyskała kategoria „Strona główna”. Jak można zauważyć, średnią zaniżają głównie oceny wyglądu estetycznego strony. Aby podwyższyć tę ocenę, należałoby poprawić oprawę graficzną aplikacji. Uśredniając wszystkie pytania związane z estetyką aplikacji, otrzymano wartość w przybliżeniu 3,6, z czego średnia za estetykę samej strony głównej wyniosła 2,25. Kolejnym aspektem porównawczym są zaobserwowane różnice w ocenie interfejsów użytkownika udostępnianych w wersji mobilnej i webowej. W kategorii strony głównej korzystniej prezentuje się aplikacja webowa. Natomiast w kategorii procesu logowania oraz rejestracji przewagę ma aplikacja mobilna. Główny wpływ na tę różnicę ma estetyka obu interfejsów. Estetyka stron logowania/rejestracji uzyskuje lepsze oceny w aplikacji mobilnej, a estetyka głównego panelu jest lepiej oceniana w aplikacji webowej. W obu wersjach aplikacji jedną z najniższych ocen wśród pojedynczych zapytań otrzymała opcja związana z dostaniem się na uprzednio umówioną wizytę. Zarówno w aplikacji mobilnej, jak i na stronie internetowej nie ma żadnego przycisku ani opcji, która automatycznie przeniosłaby użytkownika na umówioną wizytę, co utrudnia sprawne korzystanie z dostępu do specjalistów.

4. Testy wydajności

Testy wydajności, choć nie były kluczowe w tej fazie wdrażania projektu, zostały przeprowadzone dla aplikacji internetowej uruchamianej w czterech różnych przeglądarkach: Microsoft Edge, Mozilla Firefox, Google Chrome i Opera GX. Obie aplikacje (webowa i mobilna) do zestawiania połączeń głosowych i wideo wykorzystywały standard WebRTC [13, 14]. W trakcie badań analizowano dane dotyczące parametrów transmisji, w tym: wysyłania i odbioru danych, utraty odebranych pakietów i poziomu buforowania danych. Przykładowe wykresy pokazano na rys. 3.



RYS. 3. Przykładowe przebiegi wysyłania i odbioru danych, utraty odebranych pakietów i poziomu buforowania danych dla czterech przeglądarek: a) Microsoft Edge, b) Mozilla Firefox, c) Google Chrome i d) Opera GX

ŹRÓDŁO: opracowanie własne.

Analiza otrzymanych danych wykazała, że testowana aplikacja działała poprawnie we wszystkich czterech przeglądarkach. Potwierdziły to też subiektywne testy użytkowników, którzy nie stwierdzili trudności w zestawianiu połączeń audio i wideo. W ramach testów wydajności przeanalizowano także zużycie zasobów systemu podczas działania aplikacji na poszczególnych przeglądarkach. Wyniki testów zajętości czasu CPU i GPU, ilości pamięci oraz prędkości transmisji przy uruchamianiu aplikacji w różnych przeglądarkach internetowych zostały zawarte w tabelach 4 i 5.

TABELA 4. Wyniki testów zużycia zasobów systemu podczas działania aplikacji realizującej połączenia audio

Rozmowa telefoniczna (połączenie głosowe)				
przeglądarki	procesor	RAM	karta graficzna	sieć
Firefox	2,2–3,6%	630 Mb	0,3–1,8%	0,1 Mb/s
Chrome	1–2,5%	220–280 Mb	0,4–0,7%	0,1–0,2 Mb/s
Opera Gx	2–3,5%	335,5 Mb	0,7–1,9%	0,1 Mb/s
Microsoft Edge	1,5–3%	362,1 Mb	0–0,4%	0,1 Mb/s

ŹRÓDŁO: opracowanie własne.

TABELA 5. Wyniki testów zużycia zasobów systemu podczas działania aplikacji realizującej połączenia wideo

Połączenie wideo				
Przeglądarki	procesor	RAM	karta graficzna	sieć
Firefox	6,5–8,5%	680 Mb	12–17%	4,5 Mb/s
Chrome	7,5–8,5%	340–370 Mb	0,5–1,5%	4,4–6 Mb/s
Opera Gx	7,6–9,2%	361 Mb	3,3–4,3%	4,4 Mb/s
Microsoft Edge	5–7,3%	427,5–429,4 Mb	5–5,3%	4,4 Mb/s

ŹRÓDŁO: opracowanie własne.

Z analizy otrzymanych danych wynika, że najmniejsze zużycie procesora zapewniały przeglądarki Microsoft Edge i Chrome. Najwięcej zasobów spośród testowanych przeglądarek – zarówno jeśli chodzi o użycie pamięci, jak i obciążenie karty graficznej w trakcie realizacji połączeń wideo – wykorzystywała przeglądarka Firefox.

5. Podsumowanie

W artykule zaprezentowano proces tworzenia testów do analizy funkcjonalnej i nie-funkcjonalnej aplikacji webowej i mobilnej umożliwiających dostęp do zasobów wdrażanej Cyfrowej Platformy Zdrowia Młodzieży. Ze względu na kluczową rolę

w zapewnieniu przyjaznego interfejsu użytkownika szczególny nacisk położono na opracowanie testów użyteczności. Ten rodzaj testowania wymaga szczególnego doświadczenia osób kierujących zespołem testerów, użyteczność nie jest bowiem parametrem mierzalnym obiektywnie. Oceny znacznej części elementów list kontrolnych zależą od subiektywnych upodobań i doświadczeń testerów. Poprawne przygotowanie środowiska testowego daje jednak gwarancję spójności i porównywalnych wyników oceny aplikacji dokonywanych przez poszczególnych testerów. Testy użyteczności stanowią kluczowy element oceny platformy, ponieważ skupiają się na tym, jak łatwo i efektywnie użytkownicy mogą z niej korzystać. Oceniają, czy interfejs jest intuicyjny, czy proces rejestracji i nawigacji jest klarowny, czy przy minimalnym wysiłku użytkownicy mogą znaleźć potrzebne im informacje. Ponadto testy użyteczności pozwalają na identyfikację ewentualnych problemów interakcji, które mogą frustrować użytkowników i prowadzić do rezygnacji z korzystania z aplikacji.

Podsumowując, przeprowadzone badania testowe aplikacji webowej i mobilnej należy uznać za ważny element w tworzeniu Cyfrowej Platformy Zdrowia Młodzieży. Wyniki testów umożliwiły sformułowanie listy istotnych problematycznych elementów wraz z rekomendacjami dotyczącymi usprawnień zwiększających atrakcyjność i użyteczność platformy.

Literatura

- [1] *Digital Youth Life Health Problem*, <https://erasmus-plus.ec.europa.eu/projects/search/details/2021-2-TR01-KA220-YOU-000049540> [dostęp: 14.11.2023].
- [2] Smilgin R., *Zawód tester. Od decyzji do zdobycia doświadczenia*, WN PWN, Warszawa 2018.
- [3] Müller T., Graham D., Friedenberg D., van Veedendaal E., *ISTQB certified tester-foundation level syllabus*, 2007.
- [4] Roman A., *Testowanie i jakość oprogramowania: modele, techniki, narzędzia*, WN PWN, Warszawa 2015.
- [5] Myers G.J., Sandler C., Badgett T., *The art of software testing*, John Wiley & Sons, Hoboken 2011.
- [6] Li Y.F., Das P.K., Dowe D.L., *Two decades of Web application testing – A survey of recent advances*, „Information Systems” 2014, nr 43, s. 20–54.
- [7] Superson W., Smyk T., Plechawska-Wójcik M., *Comparative analysis of methods for testing web applications*, „Journal of Computer Sciences Institute” 2023, nr 28, s. 236–241.
- [8] Borys M., Plechawska-Wójcik M., *Badanie użyteczności oraz dostępności interfejsu w aplikacjach mobilnych*, „Nierówności Społeczne a Wzrost Gospodarczy” 2013, nr 35, s. 63–78.
- [9] *Testowanie manualne i automatyczne. Różnice*, 2022, <https://testerzy.pl/baza-wiedzy/artykuly/testowanie-manualne-i-automatyczne-roznice> [dostęp: 14.11.2023].
- [10] Szczepkowicz Ł., Pańczyk B., *Ocena jakości wybranych narzędzi do automatyzacji testów aplikacji*, „Journal of Computer Sciences Institute” 2018, nr 9, s. 324–327.
- [11] Quesenbery W., *Balancing the 5Es of usability*, „Cutter IT Journal” 2004, nr 17(2), s. 4–11.
- [12] Roman A., Zmitrowicz K., *Testowanie oprogramowania w praktyce: studium przypadków*, WN PWN, Warszawa 2017.

- [13] Daoust F., Hazaël-Massieux D., *Web Real-Time Communications Working Group Charter*, 2015, <https://www.w3.org/2011/04/webrtc-charter.html> [dostęp: 14.11.2023].
- [14] Levent-Levi T., *WebRTC Test: Testing Made Easy*, 2021, <https://testrtc.com/webrtc-testing-made-easy/> [dostęp: 14.11.2023].