

UNIWERSYTET PEDAGOGICZNY
im. Komisji Edukacji Narodowej w Krakowie



INSTYTUT INFORMATYKI

Kierunek: INFORMATYKA

specjalność: Multimedia i Technologie Internetowe

Anna Woś

Nr albumu: 137949

Standardy projektowania aplikacji mobilnych
- system wspomagania przeprowadzania audytów eksperckich

Praca inżynierska napisana pod kierunkiem
dr Anny Stolińskiej

Kraków 2021

Streszczenie

Niniejsza praca porusza zagadnienia z zakresu użyteczności oraz projektowania aplikacji, w szczególności aplikacji mobilnych. Tematem pracy było opracowanie standardów projektowania aplikacji mobilnych i stworzenie systemu wspomagającego ich przeprowadzanie.

W pierwszej części pracy przedstawiono obecnie obowiązujące standardy projektowania aplikacji. Opisano najczęściej występujące problemy, jakie napotykają użytkownicy urządzeń mobilnych. Przy pomocy prototypów zobrazowano kilka zasad projektowania aplikacji. Przybliżono także zagadnienia związane z audytem eksperckim i przedstawiono rodzaje audytów eksperckich UX, jak i metody ich przeprowadzania.

W drugiej części pracy skupiono się na przedstawieniu autorskiego zestawu standardów projektowania aplikacji mobilnych. Standardy stworzono w oparciu o heurystyki przedstawione w pierwszej części pracy. Wzięto pod uwagę potrzeby użytkowników oraz ograniczenia wynikające z mobilności urządzeń. Stworzono także webowy system wspierający przeprowadzanie audytów eksperckich.

Abstract

The first part of the Thesis presents the current application design standards. The most common problems encountered by users of mobile devices are described. Several principles of application design were illustrated with the help of prototypes. The issues related to the expert audit were also discussed and the types of expert UX audits and the methods of their conduct were presented.

The second part of the Thesis focuses on presenting the proprietary set of mobile application design standards. The standards were created based on the heuristics presented in the first part of the work. User's needs and limitations resulting from the mobility of devices were taken into account. a web-based system to support expert audits was also created.

Spis treści

Wstęp.....	5
Rozdział 1 Obowiązujące standardy projektowania aplikacji.....	7
1.1. Różnice w projektowaniu aplikacji internetowych i aplikacji mobilnych	7
1.2. Projektowanie aplikacji mobilnych, a wielkość urządzeń.....	8
1.3. Sposób użycia ekranu dotykowego	11
1.4. Standardy systemu Android.....	12
1.5. Zużycie baterii oraz pamięci przez aplikację mobilną	13
Rozdział 2 Audyt ekspercki	14
2.1. Lista kontrolna.....	15
2.2. Analiza heurystyczna.....	15
2.3. Wędrówka poznawcza	20
Rozdział 3 Autorskie standardy projektowania aplikacji mobilnych.....	21
3.1 Pokazuj status i reakcje zwrotne systemu.....	21
3.2 Zapewnij zgodność z naturalną reakcją i wzorcem zachowań	22
3.3 Daj użytkownikom pełną kontrolę	22
3.4 Zachowaj spójność i zgodność z obowiązującymi zasadami	23
3.5 Rozpoznanie ponad przypomnienie.....	23
3.6 Zapewnij minimalistyczny design	24
3.7 Zapewnij odpowiednią kolorystykę i kontrast.....	25
3.8 Zadbaj o odpowiednią typografię	26
3.9 Zadbaj o odpowiednie użycie list	26
3.10 Zapobiegaj błędom oraz dezorientacji.....	27
3.11 Zapewnij obsługę błędów	27
3.12 Zadbaj o pomoc i dokumentację.....	28
Rozdział 4 System wspomagający przeprowadzanie audytów eksperckich	29
4.1 Analiza funkcjonalności aplikacji	29
4.2 Interfejs użytkownika	32
4.3 Testowanie aplikacji	39
Zakończenie.....	42
Bibliografia.....	43
Spis rysunków	45
Spis tabel	45
Spis załączników	45

Wstęp

Historia aplikacji mobilnych zaczyna się w roku 1983. Wówczas na konferencji w Aspen Steve Jobs pierwszy raz wspomniał o możliwości elektronicznego transferu oprogramowania za pomocą linii telefonicznych. Przewidywał on także, że opłata za oprogramowanie będzie odbywać się w sposób zautomatyzowany. W tym samym roku przedstawiono pierwszy telefon komórkowy - Motorolę DynaTAC 800X, z prostą aplikacją do zarządzania kontami [1].

W roku 1993 firma Apple wydała niewielkie przenośne urządzenie PDA (ang. *personal digital assistnat*). Posiadało ono pierwsze podstawowe aplikacje umożliwiające obsługę kalendarza, e-maila oraz książki adresowej. Przełom technologii mobilnych nastąpił w 2007 roku, kiedy na rynku pojawił się pierwszy iPhone. Rok później ukazały się pierwsze wersje konkurencyjnego systemu Android [2]. Od tego czasu technologie mobilne rozwijają się niezwykle szybko. Początkowo telefony dotykowe obsługiwane były za pomocą specjalnych piórek, które wymuszały na użytkownikach dużą precyzję podczas korzystania z ekranu dotykowego [3]. Projektowanie większych przycisków oraz zapewnienie odpowiedniej przestrzeni pozwoliło na obsługę urządzeń mobilnych palcami. Ta zmiana pokazuje, że maksymalne skoncentrowanie na potrzebach użytkowników, poczucie empatii oraz zrozumienia jest niezbędne, aby tworzyć produkty użyteczne (ang. *usability*).

Intensywny rozwój aplikacji mobilnych i wprowadzanie nowych trendów projektowych mogą zwiększyć poczucie zagubienia u początkujących projektantów. Swobodne korzystanie z aplikacji oraz wygoda, jest ważna nie tylko z poziomu użytkownika aplikacji, ale także projektanta oraz eksperta, który daną aplikację analizuje. Opracowanie standardów projektowania aplikacji mobilnych oraz stworzenie odpowiedniego systemu wspomagającego przeprowadzanie audytów eksperckich jest także podejściem skoncentrowanym na użytkownikach - badaczach.

Celem pracy jest opracowanie zestawu standardów projektowania aplikacji mobilnych oraz stworzenie systemu wspomagania przeprowadzania audytów eksperckich. Praca składa się z 3 rozdziałów. W pierwszym omówiono aktualnie obowiązujące standardy projektowania aplikacji. Przedstawiono różnice w projektowaniu aplikacji mobilnych i webowych oraz

wpływ mobilności i wielkości urządzeń na sposób projektowania interfejsów. Zwizualizowano także najczęściej wykorzystywane w projektowaniu mobilnym zasady Gestalt. Opisano sposób użycia ekranu dotykowego oraz techniczne zasady projektowania aplikacji. W drugim rozdziale wyjaśniono, na czym polega audyt ekspercki oraz jakie są metody przeprowadzania analizy interfejsów. Zwrócono szczególną uwagę na analizę heurystyczną. W rozdziale trzecim przedstawiono autorskie standardy projektowania aplikacji mobilnych. Skupiono się na projektowaniu aplikacji z systemem Android. Standardy opracowano na podstawie teorii opracowanej przez Jacoba Nielsena, którą poszerzono o elementy listy kontrolnej. Opisano także działanie aplikacji webowej stworzonej na potrzeby wsparcia podczas przeprowadzania audytów eksperckich.

Praca skierowana jest do projektantów oraz ekspertów przeprowadzających audyty eksperckie UX. Opisuje zestaw norm, które powinny być zastosowane podczas projektowania aplikacji mobilnych. Jest również elementem przygotowania zawodowego, bowiem autorka pracy wiąże swoją karierę zawodową z obszarem projektowania UX/UI.

Rozdział 1 Obowiązujące standardy projektowania aplikacji

Tim Berners-Lee w 1991 roku przedstawił światu pierwszą stronę internetową. Na przestrzeni lat serwisy internetowe zmieniały się, stawały się coraz bardziej interaktywne, z czasem też przeglądarki stały się klientem aplikacji. Równolegle z rozwojem technologii, rozpoczęto badania interakcji użytkowników z projektowanymi interfejsami. Na podstawie tych obserwacji tworzone są dobre praktyki, którymi powinni kierować się projektanci. Obecnie znane i ogólnodostępne standardy pomagają w tworzeniu funkcjonalnych oraz użytecznych aplikacji internetowych.

Mimo, że tworzenie aplikacji mobilnych w pewnym stopniu opiera się na opracowanych standardach projektowania aplikacji internetowych, istnieją istotne różnice, które należy wziąć pod uwagę podczas projektowania aplikacji mobilnych.

1.1. Różnice w projektowaniu aplikacji internetowych i aplikacji mobilnych

Kluczowymi aspektami, które należy wziąć pod uwagę podczas projektowania aplikacji mobilnych jest różnorodność urządzeń przenośnych oraz okoliczności, w których są one wykorzystywane. Użytkownicy aplikacji korzystający z urządzeń mobilnych „spieszą się jeszcze bardziej i czytają jeszcze mniej niż zwykli internauci” [4], czego powodem jest korzystanie z aplikacji w trakcie wykonywania innych czynności np. jazdy tramwajem, spacerowania czy innych aktywności. Brak koncentracji uwagi użytkowników sprawia, że zaprojektowanie odpowiedniej aplikacji mobilnej jest trudniejsze, niż takiej, która jest przeznaczona do wyświetlania na monitorach komputerów stacjonarnych lub notebooków. Mobilność powoduje, że na jakość korzystania z aplikacji mogą mieć wpływ także czynniki zewnętrzne, którymi są: zmienne warunki oświetleniowe, natężenie dźwięku, przebywanie w dużym skupisku ludzi, bądź też konieczność obsługi urządzenia jedną ręką. Wszystko to sprawia, że w zależności od sytuacji zmienia się sposób korzystania z urządzenia mobilnego [5]. Tworzenie aplikacji webowych z tego względu jest dużo łatwiejsze. Użytkownicy aplikacji internetowych zazwyczaj nie borykają się z tego typu utrudnieniami, a ich uwaga jest skoncentrowana na ogół głównie na aplikacji.

Na różnice w projektowaniu aplikacji mobilnych i desktopowych mają wpływ nie tylko warunki zewnętrzne. Tym, co sprawia szczególną trudność podczas projektowania aplikacji mobilnych, jest wielkość urządzeń oraz różna rozdzielczość ekranów. Według przeprowadzonych badań, przyswojenie złożonej treści przy użyciu urządzenia stacjonarnego jest dwukrotnie łatwiejsze niż w przypadku użycia urządzenia mobilnego [6]. Tak duża

różnica powoduje, że konieczna jest odpowiednia optymalizacja treści wyświetlanych na urządzeniach mobilnych, a ich układ musi się różnić ze względu na sposób korzystania z urządzenia.

1.2. Projektowanie aplikacji mobilnych, a wielkość urządzeń

Aplikacje mobilne projektowane są nie tylko na smartfony o różnej rozdzielczości, ale także na urządzenia takie jak tablety. Różnica rozdzielczości tych urządzeń waha się od 128 do 1024 px [7], stąd też często konieczne jest projektowanie kilku wersji aplikacji. Znany portal społecznościowy, jakim jest Facebook, posiada różne wersje swojej aplikacji, które przeznaczone są odpowiednio na urządzenia stacjonarne, tablety oraz smartfony.

Standardy projektowania aplikacji mogą różnić się w zależności od wielkości urządzenia, rozdzielczości, a nawet gęstości pikseli. Wielkość ekranów telefonów wpływa bezpośrednio na ilość treści umieszczanej w aplikacji. Każda dodatkowa ilość tekstu powoduje, że aplikacja staje się nieczytelna, a korzystanie z niej jest uciążliwe dla użytkownika. Projektowanie na mniejsze ekrany sprowadza się do minimalizacji ilości treści przy jednoczesnym maksymalizowaniu przekazywanej informacji.

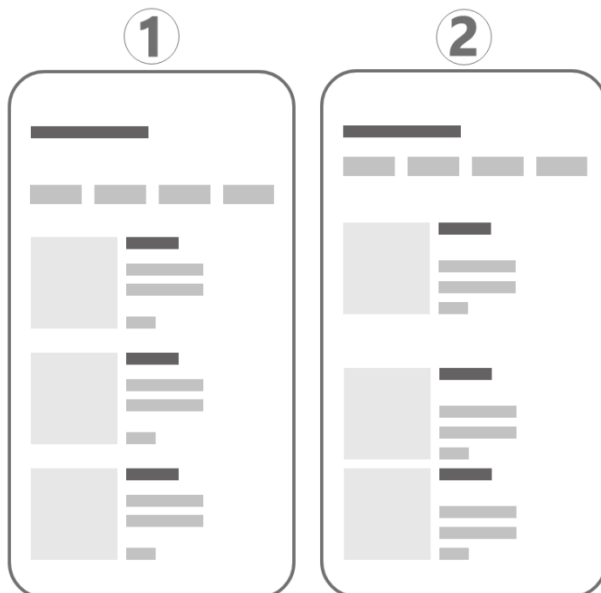
1.2.1. Interfejs aplikacji mobilnych

Podczas projektowania aplikacji na urządzenia o mniejszej rozdzielczości ważne jest, aby rozkład elementów interfejsu dawał użytkownikowi uczucie spójności oraz pozwalał na intuicyjną obsługę aplikacji. Na początku XX wieku w psychologii opracowano teorię Gestalt, która opisuje sposób postrzegania elementów wizualnych. Kanon opracowany w teorii Gestalt jest także powszechnie stosowany w dziedzinie User Experience¹. Przedstawia on prawa: bliskości, podobieństwa, ciągłości, domknięcia oraz figura-tło [8]. Nie wszystkie z wymienionych praw muszą być wykorzystywane w aplikacjach mobilnych. Jednak stosowanie niektórych z nich jest szczególnie istotne dla stworzenia odpowiedniego interfejsu. Poniżej przedstawiono trzy najczęściej wykorzystywane prawa.

Prawo bliskości opisuje zależności między rozmieszczeniem elementów a sposobem ich kategoryzowania przez użytkowników. Naturalnym zjawiskiem jest postrzeganie elementów znajdujących się blisko siebie jako jednej grupy. Wykorzystanie tej zależności podczas projektowania aplikacji sprowadza się do tworzenia interfejsu w taki sposób, aby dzięki bliskiemu osadzeniu elementów użytkownik szybko je grupował. Na rysunku 1 zobrazowane zostały dwie wersje układu elementów. Wersja 1 przedstawia interfejs

¹ Zbiór wszystkich doświadczeń użytkownika korzystającego z produktu.

zaprojektowany zgodnie z prawem bliskości. Dzięki odpowiedniemu ułożeniu, mimo braku treści, wyraźnie widoczne jest, które elementy są ze sobą powiązane. Wykorzystanie odpowiedniego układu elementów podczas projektowania przyspiesza szybkość nauki aplikacji i tworzy ją bardziej czytelną dla użytkownika.



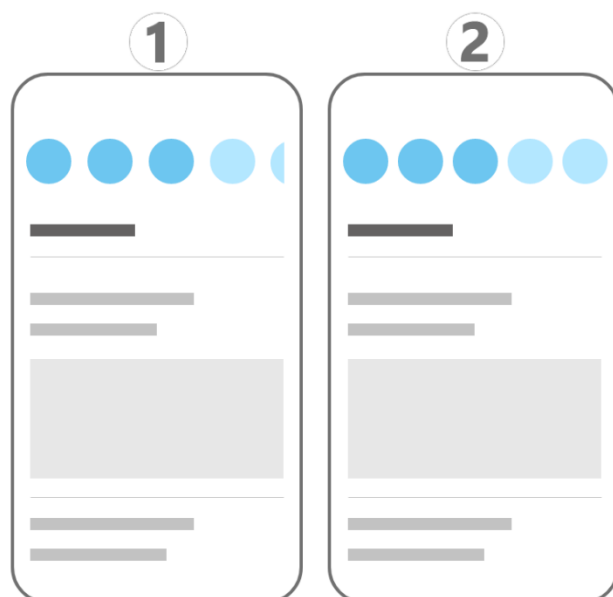
Rysunek 1 Prawo bliskości
Źródło: opracowanie własne

Stosowanie standardów ma na celu stworzenie interfejsów, które zachęcają użytkowników do korzystania z aplikacji. Spójny interfejs uzyskiwany jest poprzez odpowiednie grupowanie oraz wykorzystanie podobieństw. Przedstawione w teorii Gestalt **prawo podobieństwa** w praktyce wykorzystuje się nieustannie. Podobieństwo może opierać się nie tylko na wykorzystaniu tych samych kształtów, ale również barw. Dla przedstawionych na rysunku 2 prototypów widoczne jest wykorzystanie prawa podobieństwa w dwojaki sposób. Aplikacja nr 1 przedstawia wykorzystanie podobieństwa kolorów. Dzięki takiemu zabiegowi użytkownik szybko zauważa zależności kolorystyczne i łączy poszczególne elementy ze sobą. W przypadku aplikacji nr 2 wykorzystano podobieństwa kształtów, co także ułatwia użytkownikowi interpretację aplikacji.



Rysunek 2 Prawo podobieństwa
Źródło: opracowanie własne

Celem zastosowania **prawa ciągłości** jest utwierdzenie użytkowników, że dane pola można przesunąć i zobaczyć kolejne elementy należące do grupy. Pierwsza wersja aplikacji znajdująca się na rysunku 3 wykorzystuje zasadę ciągłości poprzez zastosowanie niepełnego elementu. W efekcie użytkownik aplikacji nr 1 wie, że gestem przesunięcia spowoduje wyświetlenie kolejnych elementów listy poziomej.

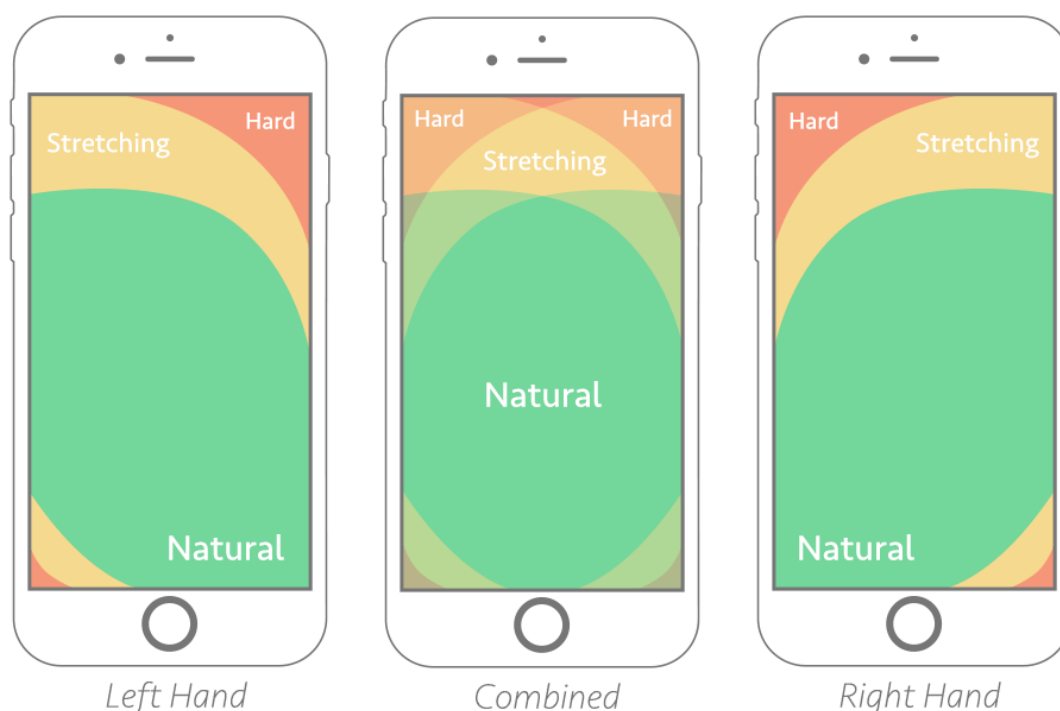


Rysunek 3 Prawo ciągłości
Źródło: opracowanie własne

1.3. Sposób użycia ekranu dotykowego

Spośród wielu czynników mających wpływ na użyteczność aplikacji istotnym elementem jest projektowanie uwzględniające sposób korzystania z urządzenia. Użytkownicy telefonów posługują się nimi w trakcie wykonywania innych czynności, stąd aż 49% użytkowników korzysta ze smartfona jedną ręką.

Obsługa telefonów dotykowych opiera się głównie na użyciu kciuków. W 2013 roku Steven Hooper zaobserwował trzy główne sposoby trzymania telefonów przez użytkowników. Przedstawił on tzw. „strefę kciuka”, która obrazuje najbardziej aktywne obszary ekranu telefonu [9]. Nie jest przypadkiem, że najistotniejsze elementy interfejsu aplikacji znajdują się w centralnych częściach telefonu. Strefy te są najwygodniejsze do użycia właśnie ze względu na wspomnianą wcześniej strefę kciuka. Rysunek 4 przedstawia opracowane przez Hooper’a strefy. Miejsca najwygodniejsze w użyciu różnią się nieznacznie w zależności od sposobu korzystania z telefonu: jedną ręką lub oburącz. W obu przypadkach kciuk używany jest głównie ze względu na duży zakres ruchu, co przekłada się na wygodniejszą obsługę telefonu. Dla użytkowników korzystających z telefonu lewą ręką, naturalnie najwygodniejsza w użyciu jest środkowa oraz prawa powierzchnia ekranu. Zupełnie odwrotna zasada zaobserwowana została przy użytkownikach posługujących się telefonem prawą ręką. W przypadku korzystania z telefonu przy użyciu obu rąk, strefy nachodzą się na siebie poszerzając tym samym obszar wygodnego użytkowania smartfona. Istotne jest więc, aby elementy najczęściej klikane, bądź też takie, które powinny być łatwo zauważalne, umieszczane były w centralnej, bądź też dolnej części widoku ekranu.



Rysunek 4 Strefa kciuka

Źródło: <https://www.smashingmagazine.com/2016/09/the-thumb-zone-designing-for-mobile-users/>

Zastosowanie zasady tworzenia interfejsów opierającej się na strefach kciuka ma miejsce nie tylko w przypadku tworzenia aplikacji na telefony. Mimo znacznie większej rozdzielczości ekranów tabletów, bądź też laptopów z dotykowymi ekranami, urządzenia te także używane są w szczególności przy użyciu kciuków.

1.4. Standardy systemu Android

Sposób używania telefonu oraz samo projektowanie różni się w zależności od systemu operacyjnego. Różne systemy operacyjne posiadają swoje prywatne standardy, na których powinno opierać się projektowanie. Użytkownicy korzystający z urządzeń z systemem Android często nie wiedzą, jaką specyfiką cechują się jego aplikacje. Niewiedza ta nie wyklucza jednak konieczności stosowania pewnego rodzaju schematów, jakimi cechuje się ten system. Aplikacja używana na telefonie z systemem Android nie może cechować się specyfiką iOS, ponieważ mimo wzorowego interfejsu aplikacji użytkownicy przyzwyczajeni do Androida naturalnie by z niej rezygnowali [10].

Dla telefonów z systemem Android charakterystyczne są trzy przyciski sprzętowe znajdujące się u dołu ekranu. System Android stosowany jest na urządzeniach różnych producentów,

gdzie w dalszym ciągu istnieją różnice w projektowaniu samego urządzenia. Ze względu na producentów różnią się one nieco przyciskami funkcyjnymi. Część urządzeń stosuje przyciski dotykowe, inne natomiast przyciski fizyczne. Spotykane są urządzenia z androidem posiadające cztery przyciski funkcyjne. Często wykorzystywanym sposobem, który pozwala uniknąć zagnieżdżenia przycisków systemowych i elementów sterujących aplikacji jest umieszczanie nawigacji w górnej części ekranu. Mimo, że takie rozwiązanie mija się z opisaną w poprzednim podrozdziale strefą kciuka, to jednak nie w każdym przypadku jest ona wiodącą zasadą. Pomyłki wynikające ze źle umieszczonych przycisków aplikacji, które prowadzą do jej zamykania i konieczności ponownego powrotu, mogą być dla użytkownika irytujące i często prowadzą do odinstalowania aplikacji.

1.5. Zużycie baterii oraz pamięci przez aplikację mobilną

Użytkownicy aplikacji zwracają uwagę na to, jak używana aplikacja wpływa na sprawność działania telefonu. Sposób, w jaki projektowana jest aplikacja, ma wpływ na wydajność całego systemu. Źle zaprojektowana aplikacja może powodować wolniejsze działanie innych aplikacji [10].

Duża ilość pobranych aplikacji wraz z ciągłą aktywnością telefonu często powoduje, że konieczne jest jego codzienne ładowanie. Próba optymalizacji zużycia baterii przez użytkowników sprowadza się do śledzenia zużycia baterii przez poszczególne aplikacje, wyłączania aplikacji działających w tle, czy też zmniejszania jasności ekranu. Istnieją także gotowe programy oferowane np. przez system Android, które pozwalają kontrolować, a tym samym zoptymalizować zużycie baterii, dzięki obserwacji wykorzystania procesora [11].

Możliwość kontroli zużywania baterii powoduje, że użytkownicy odinstalowują aplikacje wykorzystujące baterie w największym stopniu. Często pozostawiane są tylko niezbędne programy, bądź też pobierane są zamiennie, które zużywają mniej baterii. Ważne jest zatem, aby przy tworzeniu aplikacji brać pod uwagę poziom zużycia baterii przez aplikację.

Rozmiar aplikacji ważny jest nie tylko ze względu na użytkownika. Biorąc pod uwagę wytyczne sklepu Google Play, skompresowane pliki APK tworzonej aplikacji nie mogą osiągać rozmiarów większych niż 100 MB. Aplikacja posiadająca większy rozmiar plików zostanie odrzucona z powodu przekroczonego limitu rozmiaru [12].

Rozdział 2 Audyt ekspercki

Proces projektowy powinien być skoncentrowany na potrzebach oraz wymaganiach użytkowników. Aby sprawdzić, czy projektowana aplikacja spełnia wymogi oraz oczekiwania potencjalnych użytkowników, przeprowadzane są różnego rodzaju testy. Dzięki temu istnieje możliwość redukcji kosztów oraz czasu poświęconego na prace projektowe.

Ważnym elementem podczas przeprowadzania testów jest badanie użyteczności. Według norm ISO na użyteczność składa się wydajność, efektywność oraz satysfakcja, z jaką może być używany produkt przez określonych użytkowników [13]. Badanie użyteczności pozwala na lepsze poznanie użytkownika oraz zaobserwowanie jego potrzeb. Istnieją dwie metody badania użyteczności: **metody ilościowe** oraz **metody jakościowe**. Każda z metod charakteryzuje się innym sposobem analizy produktu. Analiza użyteczności metodą ilościową ma na celu zebranie oraz zbadanie mierzalnych danych. W ten sposób możliwe jest np. sprawdzenie ilu użytkowników rejestruje się na portalu. Do badań ilościowych należą m.in. testy A/B oraz click tracking². Metody jakościowe odpowiadają na pytanie „jak, dlaczego”. Mają one na celu badanie użytkownika poprzez analizę jego emocji, oraz wrażeń opisywanych podczas korzystania z produktu. Metody jakościowe często polegają także na obserwacji niewerbalnych zachowań użytkownika [14]. Jakościowymi metodami są badania z użytkownikami oraz audyt ekspercki, który zostanie opisany szczegółowo w kolejnej części rozdziału. Podczas badania użyteczności często wykorzystuje się także metody mieszane. Wówczas klasyfikacja zależy od sposobu przeprowadzania badania oraz od liczby respondentów biorących w nim udział [15]. Do badań mieszanych można zaliczyć eye tracking³ oraz card sorting⁴.

Audyt ekspercki jest rodzajem badania użyteczności, który prowadzony jest przez eksperta, bez udziału respondentów. Zadaniem eksperta jest wcielenie się w rolę użytkownika i tym samym zbadanie jego potrzeb oraz wykrycie podstawowych błędów interfejsu. Badanie interfejsu metodą audytu pozwala określić oraz zbadać potencjalne problemy nie tylko na początkowym etapie rozwoju projektu. Audyt pomaga przygotować się do testów z użytkownikami, chociażby poprzez odpowiednie dobranie grupy badawczej. Istnieją różne metody audytu, jednak najbardziej popularnymi są: lista kontrolna, analiza heurystyczna, oraz wędrówka poznawcza. Często podczas wykonywania audytów łączone są różne metody

² Metoda polegająca na zbieraniu informacji poprzez śledzenie kliknięć.

³ Technika rejestrująca położenie oraz ruchy gałek ocznych.

⁴ Metoda służąca do grupowania informacji znajdujących się np. W rozbudowanej aplikacji.

analizy. Dzięki temu możliwe jest wyciągnięcie właściwych wniosków, a w rezultacie znaczna poprawa funkcjonalności.

2.1. Lista kontrolna

Lista kontrolna jest zbiorem zasad użyteczności, według których badany jest interfejs. Poszczególne elementy interfejsu oraz funkcjonalności porównywane są do opracowanej wcześniej listy wytycznych.

Pierwotna i najbardziej znana wersja listy kontrolnej zawiera zbiór 247 pytań, według których ekspert ocenia interfejs. Została ona zapoczątkowana w 2014 roku przez Davida Trávisa. Badacz w oparciu o wspomnianą listę kontrolną weryfikuje kolejne kryteria przypisując analizowanej aplikacji lub serwisowi internetowemu odpowiednią liczbę punktów, która jest sumowana po zakończeniu badania. Ocena 1 przyznawana jest w przypadku, gdy występuje zgodność z podanymi wytycznymi, -1 gdy element nie jest zgodny z wytycznymi oraz 0, gdy dane wytyczne nie są brane pod uwagę w ocenie [16]. Lista kontrolna może być także opracowana przez ekspertów na potrzeby konkretnego projektu. Zazwyczaj zawiera wtedy bardziej precyzyjne wytyczne dostosowane do danego interfejsu.

Wadą listy kontrolnej jest subiektywna ocena audytora. Biorąc pod uwagę różny, indywidualny sposób patrzenia na dany problem eksperci mogą ocenić ten sam przypadek w zupełnie inny sposób. Dlatego ważne jest, aby badanie było przeprowadzane przez kilku niezależnych ekspertów. Takie rozwiązanie wpływa na poprawę jakości badania.

2.2. Analiza heurystyczna

Jedną z najpopularniejszych metod przeprowadzania audytu jest analiza heurystyczna. Została ona zapoczątkowana przez Jakoba Nielsena oraz Ralfa Molicha w 1990 roku i przedstawia zbiór zasad, które powinny być przestrzegane przez projektantów. Według twórców, analiza heurystyczna powinna być przeprowadzana przez więcej niż jednego eksperta. Rekomendowane jest, aby w analizie heurystycznej brało udział od 3 do 5 badaczy, którzy będą analizowali interfejs niezależnie od siebie [17]. Zaprezentowany poniżej zestaw dziesięciu heurystyk jest obecnie jedną z najczęściej wybieranych metod analizy interfejsu. Popularność analizy heurystycznej wynika z jej prostoty oraz niskich kosztów przeprowadzania badania.

2.2.1 Zestaw dziesięciu heurystyk opracowanych przez Jakoba Nielsena i Ralfa Molicha

1. **Pokazuj status systemu.** System powinien informować użytkownika, w jakim miejscu obecnie się znajduje, czego może się spodziewać w kolejnych krokach, np. ile dzieli go od zakończenia transakcji oraz do jakiego kroku może wrócić. Dobrym przykładem pierwszej heurystyki może być także zastosowanie okruszków. Są one wykorzystywane szczególnie w aplikacjach webowych, gdzie informują użytkownika o aktualnym położeniu.
2. **Zachowaj zgodność pomiędzy systemem a rzeczywistością.** Interakcja użytkownika z produktem powinna być prosta i intuicyjna. Ważne jest, aby użytkownik otrzymywał jasne i zrozumiałe komunikaty. Dzięki używaniu prostych zwrotów, które stosowane są w codziennym życiu, użytkownik wie, jak powinien postępować na każdym etapie pracy z aplikacją.
3. **Daj użytkownikowi pełną kontrolę oraz wolność.** Podczas korzystania z narzędzia użytkownik często chce szybko odnaleźć konkretne funkcje błędzi po serwisie. Dlatego ważne jest, aby umożliwić użytkownikowi powrót do poprzednio podjętej akcji. Pozwala to na zmniejszenie irytacji użytkownika w momencie zmiany decyzji.
4. **Zachowaj spójność oraz trzymaj się standardów.** Elementy o tej samej funkcjonalności powinny być przedstawione w ten sam sposób, ważna jest w tym przypadku spójna kolorystyka, użycie zrównoważonej czcionki odpowiedniej dla danego typu serwisu oraz odpowiedni układ elementów. Użytkownicy szybko się uczą i zapamiętują powszechnie stosowane ułożenie elementów. Stosowanie standardów projektowych pomoże użytkownikom sprawnie korzystać z narzędzia bez ryzyka błędzenia oraz zastanawiania się nad poszczególnymi elementami interfejsu.
5. **Zapobiegaj błędom.** Błędy popełniane przez użytkowników dzielą się na omsknięcia oraz pomyłki. W pierwszym przypadku wynikają one z działania użytkowników w sposób automatyczny, bez zastanowienia. Pomyłki natomiast są efektem błędnie obranych celów, przez co nawet poprawnie wykonywane kroki skutkują błędnym rezultatem [18]. Ważne jest, aby podczas projektowania przewidzieć, jakie błędy mogą być popełniane przez użytkowników oraz zmniejszyć ryzyko ich występowania poprzez zastosowanie odpowiednich komunikatów i ograniczeń.
6. **Rozpoznanie ponad przypomnienie.** Interfejs nie powinien zmuszać użytkownika do zapamiętywania poprzednich kroków. Rozpoznanie wymaga od użytkownika mniej czasu i wysiłku niż konieczność przypominania sobie. Jednym ze sposobów wspierania użytkownika jest np. zapamiętywanie przez system poprzednio dokonywanych wyborów i sugerowanie użytkownikom konkretnych akcji.

7. **Zapewnij elastyczność i efektywność.** Ważne jest, aby początkującym użytkownikom dać możliwość spełnienia ich podstawowych celów. Dzięki temu skorzystają oni tylko z prostych funkcji, które są im niezbędne. Użytkownicy doświadczeni powinni mieć możliwość rozszerzenia funkcji o bardziej zaawansowane. Takie podejście daje możliwość korzystania z narzędzia wszystkim użytkownikom, zachowując przy tym przejrzystość interfejsu.
8. **Dbaj o estetykę i minimalistyczny design.** Minimalizm interfejsu pomaga użytkownikowi szybko odnaleźć się w narzędziu i zrealizować zamierzone cele. Nadmiar funkcji oraz chaos spowodowany ilością elementów rozprasza uwagę utrudniając odnalezienie niezbędnych funkcji. Dlatego ważne jest, aby najczęściej wykonywane akcje były szybko odnajdywane a interfejs był wolny od wszelkich rozpraszających, nieistotnych elementów.
9. **Zapewnij obsługę błędów.** Użytkownik powinien być wspierany na każdym etapie współpracy z narzędziem. Niepoprawne interakcje użytkownika powinny być komunikowane przez system na bieżąco. Proste i treściwe instrukcje w szybki sposób pomogą naprawić powstałe błędy.
10. **Zadbaj o pomoc i dokumentację.** Mimo zapewnienia odpowiedniej obsługi błędów, użytkownik powinien mieć możliwość znalezienia odpowiedzi na nurtujące go pytania. Przydatne jest odpowiednie zestawienie najczęściej zadawanych pytań.

2.2.2 Osiem złotych zasad Shneidermana

Heurystyki Nielsena ze względu na swoją uniwersalność są ciągle najczęściej wybieranymi podczas analizy interfejsów. Istnieją jednak także inne zestawy heurystyk, które pomagają w tworzeniu użytecznych interfejsów. Nie mniej znane jest 8 złotych zasad projektowania interfejsu opracowanych przez Bena Shneidermana. Mogą być one wykorzystywane zarówno podczas samego projektowania, jak i podczas oceny systemu [19].

1. **Zachowaj spójność.** Brak odpowiedniej łączności między elementami zmniejsza zrozumienie działania interfejsu, irytuje oraz zniechęca do dalszego wykonywania działań. Należy dążyć do spójności wykorzystywanej terminologii, układu elementów, funkcji, przycisków, czy nawet kolorystyki.
2. **Daj możliwość korzystania ze skrótów.** Zaawansowani użytkownicy powinni mieć możliwość sprawniejszego wykonywania działań przy użyciu np. skrótów klawiszowych.

3. **Oferuj informacje zwrotne.** Użytkownik powinien wiedzieć, gdzie się znajduje oraz być informowany o rezultacie obecnego działania. Zaleca się dostarczać użytkownikowi informacji zwrotnej nawet, przy pozornie prostym działaniu.
4. **Projektuj okna dialogowe.** Brak odpowiednich komunikatów potwierdzających wykonaną czynność pozostawia użytkownika niepewnego. Istotne jest dostarczanie informacji na każdym etapie wykonywania działań za pomocą okien dialogowych. Użytkownik musi być także informowany o powstałych zmianach, które nie są przez niego od razu zauważalne.
5. **Zaoferuj prostą obsługę błędów.** Dopuszczenie do powstania błędu powinno być wyeliminowane np. poprzez bieżące zaznaczenie błędnie wypełnionych pól. W przypadku powstałego już błędu, użytkownik powinien być poinformowany o jego wystąpieniu. Należy także dostarczyć krótkiej i treściwej informacji o powodzie powstałego błędu oraz zapewnić instrukcję dalszego działania.
6. **Zezwalaj na łatwe odwrócenie działań.** Świadomość użytkownika o możliwości powrotu do poprzedniego stanu pozwala mu na swobodne poruszanie się po serwisie/aplikacji. Należy zapewnić możliwość powrotu do poprzedniej akcji.
7. **Wspieraj wewnętrzne poczucie kontroli.** Użytkownik powinien czuć, że ma pełną kontrolę nad systemem. Możliwość decydowania o zmianach daje użytkownikowi komfort podczas korzystania z narzędzia.
8. **Zredukuj obciążenie pamięci krótkotrwałej.** Należy zapewnić prosty, intuicyjny interfejs. Krótkie, treściwe informacje są szczególnie pożądane przez użytkowników. Należy zadbać o zmniejszenie ilości informacji do niezbędnego minimum.

Heurystyki zwane **regułami poznawczymi Gerhardt-Powals**, przedstawiają bardziej całościowe podejście niż heurystyki Nielsena [20]. Do reguł należy:

- 1) automatyzacja niechcianego obciążenia w celu eliminacji niepotrzebnego myślenia,
- 2) redukcja niepewności poprzez wyświetlanie danych w sposób zrozumiały,
- 3) przedstawianie danych od ogółu do szczegółu w celu zmniejszenia obciążeń poznawczych,
- 4) przedstawianie nowych danych w sposób łatwy do przyswojenia,
- 5) używanie nazw, które są koncepcyjnie związane z funkcją,

- 6) grupowanie danych w sposób spójny,
- 7) używanie kolorów, grafik zamiast zadań opartych na danych,
- 8) wykluczanie nieistotnych informacji przy jednoczesnym uwzględnieniu informacji istotnych w danej chwili,
- 9) zapewnienie wielokrotnego kodowania danych,
- 10) praktykowanie rozsądnej redundancji [21].

Wśród najczęściej wykorzystywanych heurystyk wyróżnia się także **30 zasad użyteczności Connella** [22]. Zasady zostały pogrupowane w 7 zestawów: wymagania i zasady funkcjonalności, użytkownik - zasady systemu, zasady dotyczące użytkowników, zasady porównawcze, zasady działania systemu, zasady percepcyjne i motoryczne, zasady obsługi użytkowników. Każdy z zestawów zawiera do 10 zasad, które są odpowiednio wyjaśniane wraz z przykładami.

Heurystyki są często modyfikowane i poszerzane o nowe, na potrzeby specjalistycznych interfejsów. W ten sposób z połączenia heurystyk Nielsena oraz 8 złotych zasad projektowania interfejsu stworzono 14 heurystyk Nielsena-Shneidermana. Modyfikacja ta została stworzona na potrzeby projektowania urządzeń medycznych [23]. Podobna sytuacja zaistniała w przypadku technologii perswazyjnych, gdzie ocena użyteczności na podstawie ogólnych heurystyk nie dawała wiarygodnego wyniku. W oparciu o 10 heurystyk Nielsena stworzono zestaw 10 heurystyk dla technologii służących promowaniu zdrowego stylu życia [24]. Heurystyki Nielsena zostały także poszerzone dla aplikacji wirtualnej rzeczywistości przez A. Sutcliffe i B. Gault. W oparciu o heurystyki Nielsena stworzyli oni 12 heurystyk, które są podstawą do badania użyteczności [25].

Analizując interfejs w oparciu o heurystyki Nielsena zbierane są najważniejsze informacje opisujące błędy weryfikowanego interfejsu. Odpowiednia analiza sprowadza się także do opisanie wniosków i właściwego pogrupowania zauważonych błędów. W przypadku audytu błędy kategoryzuje się w trzy grupy. są to błędy krytyczne, istotne oraz mniej istotne. **Błędy krytyczne** uniemożliwiają użytkownikowi zrealizowanie zamierzonego celu, powodują irytację oraz mogą być głównym powodem rezygnacji z korzystania z narzędzia. Taki rodzaj błędów powinien być natychmiastowo naprawiony. **Błędy istotne** powodują, że mimo możliwości wykorzystywania różnych funkcji, korzystanie z narzędzia jest utrudnione i interfejs wymaga poprawy. Taki rodzaj błędów jest jednym z najczęściej spotykanych podczas badania interfejsu. **Błędy drobne** mają charakter kosmetyczny i nie

powodują problemów w korzystaniu z systemu, jednak takie błędy także warto poprawiać [15]. Dzięki temu użytkownik będzie korzystał z narzędzia z przyjemnością oraz łatwością.

2.3. Wędrówka poznawcza

Jedną z form badań należących do audytu jest także wędrówka poznawcza. Badanie metodą wędrówki poznawczej opiera się na analizie interfejsu z punktu widzenia użytkownika. Podczas badania ekspert wciela się w rolę użytkownika i według ustalonego wcześniej scenariusza realizuje kolejno określone zadania. Tworzone na potrzeby badania scenariusze powstają na podstawie najczęstszych ścieżek użytkownika. W ten sposób ekspert sprawdza, czy jako korzystający z narzędzia użytkownik napotyka problemy podczas wykonywania kolejnych zadań [26]. Ideą wędrówki poznawczej jest sprawdzenie, z jaką łatwością użytkownik może poruszać się po aplikacji oraz jak szybki jest proces uczenia się. Dzięki takim badaniom ekspert może zauważyć naturalny sposób działania użytkownika podczas pierwszego kontaktu z interfejsem.

Wędrówka poznawcza posiada wielu zwolenników. Metoda ta nie wymaga od badacza czasochłonnej analizy, gdyż wnioski nasuwają się podczas wykonywania kolejnych kroków. Zaletą tej metody jest także szybki czas realizacji oraz brak wysokich kosztów badania.

Rozdział 3 Autorskie standardy projektowania aplikacji mobilnych

Poniższa lista składa się z 12 autorskich heurystyk opracowanych na podstawie teorii opracowanej przez Jacoba Nielsena. Zbiór opracowany został na potrzeby projektowania aplikacji mobilnych na urządzenia z systemem Android. Kolejne heurystyki wzbogacone zostały o bardziej szczegółowe zasady w sposób charakterystyczny dla listy kontrolnej. Takie podejście pozwala na bardziej szczegółową analizę interfejsu, skoncentrowaną na najbardziej istotnych elementach aplikacji oraz najczęściej spotykanych błędach.

3.1 Pokazuj status i reakcje zwrotne systemu

Użytkownicy aplikacji powinni być informowani, w jakim miejscu się znajdują oraz co aktualnie dzieje się z systemem. Informacje takie jak tytuły zakładek są szczególnie ważne dla użytkowników, którzy dopiero zapoznają się z aplikacją. Proste ikony z animacją, bądź krótkie komunikaty są niezbędne, aby nie wzbudzać w użytkowniku niepewności podczas procesu ładowania. Nie mniej ważne jest dodawanie powiadomień, które nienachalnie powinny informować o ważnych dla użytkownika aktualnościach w aplikacji, podczas gdy ten aktualnie z niej nie korzysta. Częstym problemem pojawiającym się podczas projektowania interfejsów jest brak wyraźnego zaznaczenia aktywnej sekcji. Zalecanym zabiegiem jest nie tylko pogrubienie ikony i tytułu aktywnej sekcji menu, ale także zmiana barwy, zacinienie lub np. zaznaczenie przy użyciu akcentów kolorystycznych w postaci częściowego obramowania. Użytkownik na każdym etapie pracy z aplikacją powinien wiedzieć, że ta z nim współpracuje. Podczas naciskania przycisku, aplikacja powinna informować użytkownika, że przycisk został naciśnięty poprzez zmianę koloru przycisku, efekt rozlania itp. Podczas kliknięcia użytkownik oczekuje od systemu potwierdzenia o pomyślnym wykonaniu akcji. Krótkie komunikaty potwierdzające poprawność kroku lub animowane ikony zatwierdzające dają użytkownikowi pewność poprawności wykonywanego działania.

- Widoczny tytuł zakładki w której znajduje się użytkownik
- Widoczna animowana ikona ładowania
- Notyfikacja o nowych aktywnościach aplikacji w powiadomieniach
- Widoczny status postępu
- Wyraźne zaznaczenie/ zacinienie aktywnej sekcji
- Wykonane akcje potwierdzane komunikatami

- Animacja/zmiana koloru naciśniętego przycisku.

3.2 Zapewnij zgodność z naturalną reakcją i wzorcem zachowań

Skoncentrowanie na użytkowniku prowadzi się do poznania jego naturalnych reakcji, brania pod uwagę działania zmysłów, skojarzeń itp. Użytkownicy często zamiast czytać jedynie skanują aplikację. W przypadku prostych komunikatów warto dodawać proste grafiki pozwalające użytkownikowi szybciej zrozumieć treść. Naturalnym jest, że elementy większe zauważane są jako pierwsze. Dzięki zastosowaniu skalowania najważniejsze elementy będą szybko zauważane przez użytkownika. Kolejność umieszczanych obok siebie przycisków także ma znaczenie z uwagi na kierunek czytania treści. Stosowanie tła w przycisku zwiększa jego zauważalność, dlatego dodanie tła do jednego z przycisków spowoduje, że będzie on wizualnie wyżej w hierarchii a w rezultacie ważniejszy dla użytkownika.

- Główny przycisk znajduje się z prawej strony (w przypadku dwóch przycisków obok siebie)
- Główny przycisk znajduje się najniżej (w przypadku ułożenia przycisków jeden pod drugim)
- Zastosowana wizualizacja zamiast tekstu
- Widoczna hierarchia ważności/skalowanie elementów
- Zastosowane tło w głównym przycisku.

3.3 Daj użytkownikom pełną kontrolę

Użytkownik powinien mieć możliwość swobodnego poruszania się po aplikacji. Mając na uwadze różne upodobania użytkowników, warto dać wybór stosowania ciemnego lub jasnego trybu widoku. Ważne jest także zapewnienie możliwości wyboru języka, wówczas aplikacja będzie dostępna dla większej grupy użytkowników. Początkujący użytkownicy poszukują nowych funkcji aplikacji, zaawansowani natomiast często bez celu klikają w zakładki. Obu grupom należy zapewnić swobodny dostęp do poprzedniego widoku.

- Możliwy wybór języka
- Możliwy wybór trybu widoku jasnego/ciemnego

- Możliwy powrót do poprzednio wykonywanego kroku.

3.4 Zachowaj spójność i zgodność z obowiązującymi zasadami

Znanych jest wiele zasad projektowania interfejsu, które należy stosować, aby znacznie poprawić jakość aplikacji. Głównymi zasadami, powszechnie stosowanymi podczas projektowania aplikacji mobilnych są: zasada kciuka, zasada teorii Gestalt oraz prawo Fittsa⁵. Obowiązujące zasady projektowania wpływają na pozytywne doznania użytkownika bez względu na platformę, z której dany użytkownik korzysta. Pływające przyciski szczególnie często używane w aplikacjach Androida, powinny znajdować się jak najbliżej menu dolnego. Dzięki temu, będą one znajdować się w strefie kciuka, co jest wygodniejsze dla użytkownika. Spójne, zgodne z zasadami projektowanie znacznie ułatwia korzystanie z aplikacji. Należy być konsekwentnym w stosowaniu kolorów oraz ikon na każdym poziomie aplikacji. Nie mniej ważne jest wyrównanie treści oraz etykiet do krawędzi. Wszystkie te zabiegi mają znaczny wpływ na zachowanie spójności interfejsu aplikacji.

- Interfejs zaprojektowany zgodnie z metodą kciuka
- Zastosowane prawo bliskości
- Zastosowane prawo ciągłości
- Zastosowane prawo podobieństwa
- Zastosowane prawo Fittsa
- Ikony tego samego stylu w całej aplikacji
- Odświeżanie poprzez przeciągnięcie w dół
- Pływający przycisk umieszczony przy menu dolnym
- Etykiety i tekst wyrównane do lewej krawędzi.

3.5 Rozpoznanie ponad przypomnienie

Aplikacja posiadająca funkcje wspomagające sprawne wyszukiwanie jest atrakcyjna pod względem użytkowym. Funkcja zapamiętywania maili, poprzednio wpisanych haseł itp. jest użytkownikom bardzo pomocna. Podobnie podczas wyszukiwania, aplikacja powinna sugerować dokończenie wpisywanego hasła. Dokończenie hasła powinno bazować na najczęściej wpisywanych przez użytkowników frazach.

⁵ Czas potrzebny do osiągnięcia celu jest zależny od rozmiaru obiektu oraz jego odległości od użytkownika.

- Możliwe zapamiętywanie wpisanych haseł
- Podane sugestie podczas wyszukiwania
- Najczęściej wybierane opcje wyświetlane jako pierwsze.

3.6 Zapewnij minimalistyczny design

Podczas projektowania ważne jest skupienie się na głównych funkcjonalnościach, które dana aplikacja powinna posiadać. Dobrym przykładem zastosowania minimalizmu jest przeglądarka Google, gdzie zostawiona jest niewykorzystana biała przestrzeń. Stosowanie takich zabiegów nie jest bezcelowe, chociaż i w takim przypadku należy uważać, aby projekt nie sprawiał wrażenia nieskończonego. Nadmiar informacji wprowadziłby chaos i utrudniał użytkownikowi dotarcie do ważnych elementów. Proste kształty ikon, brak zbędnych szczegółów, czy też nadmiaru animacji czyni interfejs przejrzystym oraz bardziej dostępnym dla użytkownika. Przy zastosowaniu dużej liczby zakładek w menu typu hamburger warto, by dodatkowe ukryte były w zakładce „więcej”. Dzięki temu, użytkownik zainteresowany jedynie podstawowymi elementami nie będzie rozpraszany przez dodatkowe zakładki. Najważniejsze funkcjonalności warto umieścić w menu górnym bądź dolnym. Mimo, że menu dolne jest częściej stosowane w urządzeniach z systemem iOS, coraz częściej stosuje się menu dolne także w urządzeniach z Androidem, głównie ze względu na wynikającą ze strefy kciuka wygodę.

- Brak szczegółów w zastosowanych ikonach
- W menu typu hamburger dodatkowe opcje schowane w zakładce „więcej”
- Brak nadmiaru animacji
- Niewielka ilość użytych obramowań elementów
- Stosowanie wysuwanego menu („hamburera”) przy większej liczbie sekcji
- Stosowanie menu dolnego dla głównych sekcji, gdy ich liczba nie przekracza 5
- Użycie minimalnej liczby kroków koniecznych do wykonania zadania.

3.7 Zapewnij odpowiednią kolorystykę i kontrast

Jednym z najczęstszych problemów podczas projektowania aplikacji okazuje się źle dobrany kontrast. Podczas projektowania na urządzenia mobilne, należy brać pod uwagę możliwość zmiany jasności ekranu urządzenia. Aby zapewnić odpowiednią widoczność, należy dostosować poziom współczynnika kontrastu kolorów. Kontrast powinien być zgodny z wymaganiami WCAG2.1. Można posłużyć się kontrolerami kontrastu dostępnymi online. Niektóre elementy interfejsu nie powinny charakteryzować się wysokim kontrastem. Elementy dodatkowe np. linie oddzielające elementy listy, czy dodatkowe treści mniej istotne dla użytkownika przy wysokim kontraście zostają zbyt szybko zauważalne. Dzięki odpowiedniemu kontrastowi użytkownik większą uwagę skupia na elementach kluczowych, a dopiero w kolejnym kroku na tych mniej istotnych.

Kolorystyka wykorzystywana w aplikacji powinna być zgodna z kontekstem użycia. Nadmiar kolorów i nieodpowiednie zestawienie także nie wpłynie pozytywnie na interfejs. Zaleca się zestawiać ze sobą kolory w sposób analogiczny, monochromatyczny, triadyczny lub dopełniający. Nie należy zmuszać użytkownika do uczenia się znaczenia kolorów na nowo. Barwy czerwone kojarzone z alertami powinny być odpowiednio wykorzystane, np. przy niekorzystnych informacjach spadkowych. Podobnie w przypadku pozytywnych informacji kojarzonych z kolorem zielonym. Aby stworzyć głębię i sprawić interfejs ciekawszym warto używać cieni. Sprawiają one wrażenie podniesienia elementów, dzięki czemu będą się wyróżniać oraz sprawią wrażenie bycia wyżej w hierarchii. Warto stosować cienie o niskim kryciu posiadające wysokie wartości rozmycia, dodatkowo cienie warto zestawić kolorystycznie z kolorem danego elementu, który zostaje zacieniony.

- Kontrast wysoki w celu wygodnego odczytania elementów
- Kontrast niski w przypadku dodatkowych elementów
- Informacje dodatkowe zapisane z użyciem czcionki o mniejszym kontraście
- Poziom kontrastu zgodny z WCAG2.1
- Spójna kolorystyka zgodna z charakterem aplikacji
- Zestawienie kolorów zgodne ze schematami barw
- Wysokie wartości rozmycia oraz niskie krycie cieni.

3.8 Zadbaj o odpowiednią typografię

Zbyt duża liczba użytych czcionek, nieumiejętne stosowanie pogrubienia i brak odpowiednio zastosowanej typografii może znacznie wpłynąć na pogorszenie interfejsu. Należy dbać o odpowiednie zestawienie czcionek oraz ich układ. Nie jest zalecane wyśrodkowanie tekstu, czy też układ kilkukolumnowy. Bezpiecznym rozwiązaniem jest użycie prostych czcionek takich jak: Helvetica Neue, San Francisco czy też zalecana dla aplikacji z systemem Android - Roboto. W przypadku łączenia różnych fontów, czcionki niestandardowe warto używać w nagłówkach. Takie zestawienie zmniejsza ryzyko postawiania chaosu i czyni interfejs ciekawszym. Ze względu na mobilność rozmiar tekstu nie powinien być zbyt mały, aby użytkownik swobodnie mógł przeczytać treść, ani za duży, aby nie zajmował dużo miejsca. Zalecane jest, aby rozmiar czcionek wahał się pomiędzy 16px dla tekstu podstawowego, a 20px dla nagłówków.

- Rodzaj czcionki pasujący do charakteru aplikacji
- Klasyczne fonty dla tekstu podstawowego np. Roboto
- Zastosowane nie więcej niż 3 różne fonty w aplikacji
- Nagłówki sekcji pogrubioną czcionką
- Rozmiar czcionki wynosi nie mniej niż 16px
- Zauważalna hierarchia w typografii
- Tekst wyrównany do lewej
- Układ jednokolumnowy.

3.9 Zadbaj o odpowiednie użycie list

Sposób używania list w aplikacjach mobilnych wpływa na szybkość wykonywanych czynności. Przyciski binarne (ang. *toggle buttons*) kojarzone są z globalnym włączaniem/wyłączaniem ustawień na stałe, dlatego stosowanie ich np. jako przycisk opcji wyboru jest niezalecane. W przypadku przycisków akcji (ang. *radio buttons*) użytkownicy przyzwyczajeni są, że zaznaczenie jednej opcji wyklucza drugą. Stąd też stosowanie tego rodzaju przycisków w listach wielokrotnego wyboru nie jest zalecane. Przyciski akcji zaleca się stosować przy nie więcej niż pięciu opcjach wyboru. Gdy wybór elementów nie jest duży, nie zaleca się stosowania suwaków (ang. *slider buttons*). Szybsze dla użytkownika będzie wybranie elementów z gotowej listy. Podczas stosowania długiej listy warto zastosować opcję

„inne” dla pozostałych niewyróżnionych elementów, aby użytkownik szybciej wybrał interesującą go opcję.

- Przyciski binarne użyte w celu włączenia/wyłączenia danej funkcji
- Przyciski akcji dla jednej możliwości wyboru
- Lista rozwijana przy wielu elementach
- Opcja „inne” dla najrzadziej wybieranych sekcji
- Zastosowany suwak przy dużym zakresie wyboru np. ceny.

3.10 Zapobiegaj błędom oraz dezorientacji

Dezorientacja oraz powstawanie błędów najczęściej są efektem szybkich akcji użytkowników i nieintuicyjnego interfejsu. Należy zmniejszyć ryzyko powstawania błędów dodając krótkie komunikaty wyprzedzające akcje użytkownika, np. informację o długości hasła podczas tworzenia konta. Użytkownicy są przyzwyczajeni do charakterystycznych widoków przycisku, dlatego stosowanie przycisków w formie samego tekstu powinno być używane jedynie w szczególnych przypadkach. Podobnie statyczne elementy nie powinny wydawać się użytkownikowi przyciskami. Wszelkie komunikaty powinny być krótkie oraz zrozumiałe dla użytkownika, tak aby czas potrzebny na ich zrozumienie był minimalny.

- Widoczne informacje wyjaśniające wyprzedzające akcje użytkownika
- Przyciski w formie samego tekstu jedynie w dodatkowych odnośnikach
- Elementy statyczne nie posiadają stylu przycisku
- Brak specjalistycznych zwrotów w komunikatach
- Zastosowane krótkie wyjaśnienia
- Pola do zapisu wypełnione przykładowymi odpowiedziami przy użyciu czcionki o niskim kontraście.

3.11 Zapewnij obsługę błędów

Przyzwyczajenie do częstego przeglądania aplikacji oraz szybkiego przewijania widocznych treści powoduje, że nawet niewielki błąd aplikacji irytuje jej użytkowników. Użytkownicy są wymagający, dlatego w przypadku zaistniałego problemu oczekują

zrozumiałej informacji o błędzie i jego szybkiego naprawienia. Problem może się pojawić jeszcze przed rozpoczęciem działania z aplikacją. W przypadku, gdy aplikacja wymaga dostępu do internetu, podanie odpowiedniego komunikatu informującego o byciu offline jest niezbędne. Komunikaty dotyczące błędnie podjętej akcji powinny być zapisywane bezosobowo - nikt nie lubi być obarczany winą, użytkownicy także. Aby zmniejszyć czas potrzebny na znalezienie rozwiązania problemu, wszystkie komunikaty dotyczące błędu powinny być umieszczane jak najbliżej jego występowania np. tuż nad błędnie wypełnionym polem formularza.

- Widoczna informacja o byciu offline (gdy aplikacja wymaga dostępu do internetu)
- Komunikaty błędów podawane w formie bezosobowej
- Alerty wyświetlane na środku aplikacji
- Komunikaty o błędach wyświetlane na bieżąco w pobliżu zaistniałego błędu.

3.12 Zadbaj o pomoc i dokumentację

Użytkownicy aplikacji mobilnych nieczęsto korzystają z opcji pomocy, mimo to jest ona niezbędna. Zarówno proste jak i zaawansowane aplikacje powinny posiadać proste samouczki wyświetlające się w nowo pobranej aplikacji. Nie powinny być one długie, aby niepotrzebnie nie znudzić użytkownika. Konieczne jest umieszczenie pomocy zapewniającej informacje o bezpieczeństwie oraz prywatności użytkownika. Warto, aby aplikacja zawierała FAQ (ang. *frequently asked question*), który może pomóc użytkownikowi szybko znaleźć odpowiedź na najczęściej zadawane pytania. Nie mniej ważna jest możliwość zgłaszania problemu w aplikacjach, w których może dojść do naruszenia prywatności.

- Zastosowane proste i krótkie samouczki dla nowych użytkowników aplikacji
- Obecne FAQ
- Widoczna opcja pomocy
- Widoczne informacje dotyczące prywatności.

Rozdział 4 System wspomagający przeprowadzanie audytów eksperckich

Efekty badań audytów eksperckich - zaobserwowane błędy oraz wnioski, analitycy często przedstawiają w formie tekstowej. Aby umożliwić audytorom sprawną analizę interfejsów mobilnych stworzono aplikację webową. System ten, dzięki wykorzystaniu opracowanych standardów, pozwala w łatwy sposób określić jakość interfejsu badanej aplikacji. Wykorzystanie wizualizacji w postaci wykresów pomaga lepiej zrozumieć błędy projektowe interfejsu, odpowiednio je skategoryzować oraz wpłynąć na ich szybsze rozwiązanie.

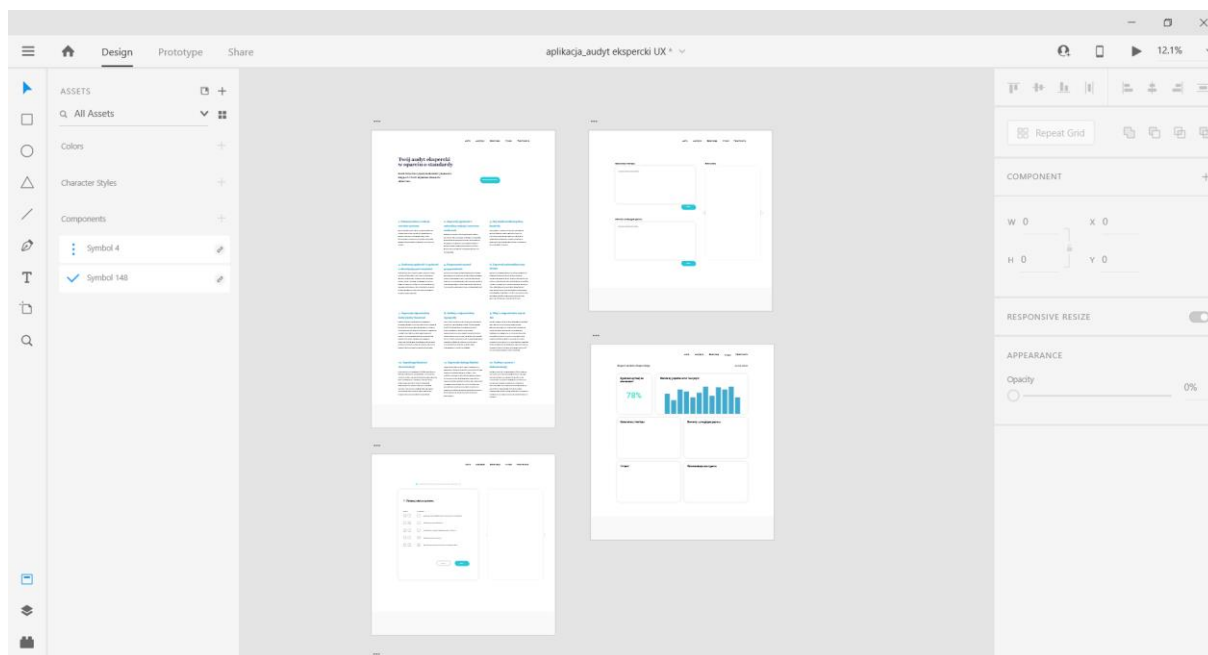
4.1 Analiza funkcjonalności aplikacji

4.1.1 Wymagania funkcjonalne

System ma na celu ułatwienie przeprowadzenia audytu eksperckiego UX dzięki automatycznemu generowaniu raportu. Raport generowany jest na podstawie odpowiednio zaznaczonych elementów listy kontrolnej oraz wypełnionych pól tekstowych. Użytkownikami systemu mogą być zarówno eksperci wykonujący audyt, jak i początkujący projektanci chcący przeprowadzić audyt swojej aplikacji.

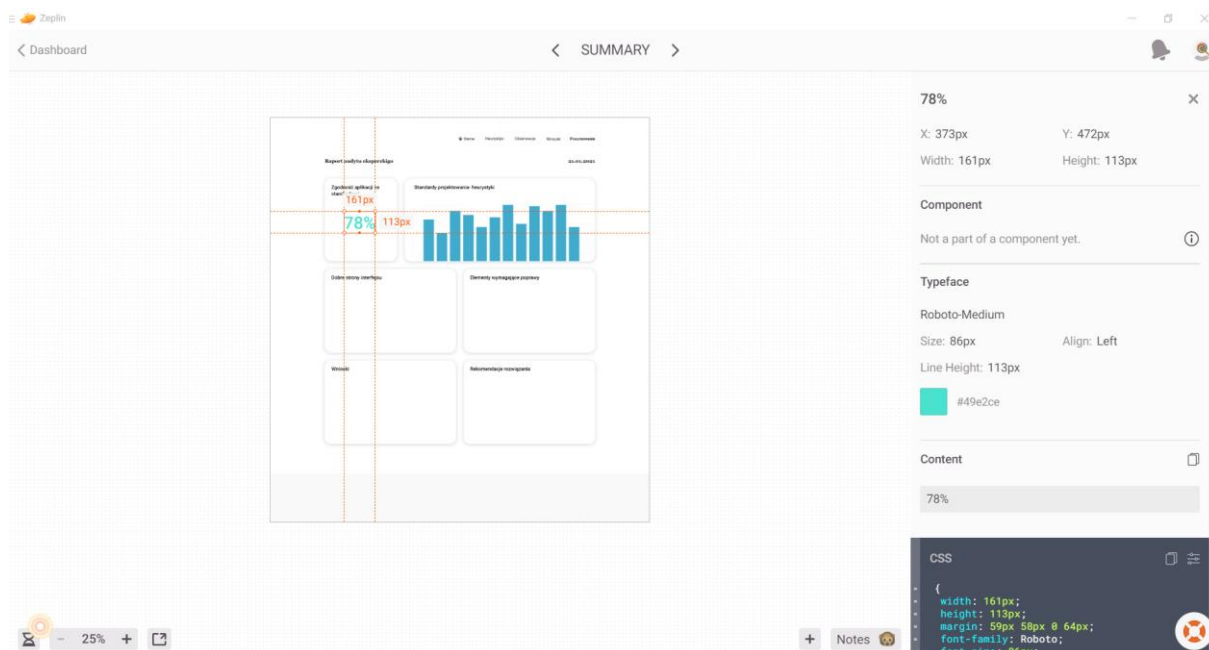
W celu zaprojektowania systemu wykorzystano program Adobe XD, który pozwolił na stworzenie prototypu high-fidelity⁶. Dzięki możliwości tworzenia interakcji między elementami możliwe było zobrazowanie realnego projektu jeszcze przed jego implementacją. Rysunek 5 przedstawia zrzut ekranu aplikacji Adobe wraz z fragmentem zaprojektowanego interfejsu.

⁶ Interaktywny prototyp odwzorowujący finalny produkt



Rysunek 5 Zrzut ekranu prototypu aplikacji wykonany w programie Adobe XD
Źródło: opracowanie własne

Aby sprawnie zaimplementować stworzony interfejs, wykorzystano aplikację desktopową Zeplin. Aplikacja pozwoliła na szybszą implementację kodu oraz dokładne odwzwierciedlenie typografii, jak i kolorystyki. Dzięki wygenerowanym w aplikacji stylom CSS określone zostały dokładne wymiary poszczególnych elementów interfejsu. Zrzut ekranu przedstawiający interfejs zaimportowany do aplikacji Zeplin obrazuje rysunek 6.



Rysunek 6 Zrzut ekranu prototypu w aplikacji Zeplin
Źródło: opracowanie własne

Sama aplikacja stworzona została w oparciu o język HTML5. Aby uzyskać interfejs zbliżony do zaprojektowanego w programie Adobe XD użyto języka CSS3. W celu umożliwienia odbierania plików z zewnętrznych źródeł, wprowadzania animacji na stronie i generowania raportu wykorzystano język Java Script. Kod aplikacji dostępny jest w repozytorium GitHub pod linkiem: <https://github.com/ankaw401/System-przeprowadzania-audytow-eksperckich>.

System umożliwia wgranie widoku interfejsu aplikacji mobilnej poprzez wybór pliku z komputera użytkownika przeprowadzającego audyt. Za pomocą przycisków typu *radio button* możliwe jest zaznaczenie, czy wgrany interfejs spełnia założenia standardów. Na podstawie zaznaczeń generowany jest wykres dla poszczególnych heurystyk oraz procentowa zgodność interfejsu z heurystykami. Możliwe jest poruszanie się pomiędzy kolejnymi kartami zawierającymi heurystyki. System posiada także funkcję pozwalającą zapisywać indywidualne obserwacje i wnioski. Dla wpisywanych obserwacji założono, że długość tekstu nie może przekraczać 880 znaków. W przypadku wniosków możliwy jest zapis 810 znaków. Zapisane obserwacje i wnioski automatycznie umieszczane są w raporcie. Aplikacja pozwala także na zapisanie raportu jako plik PDF, który następnie może zostać przedstawiony klientowi.

4.1.2 Wymagania нефункционалне

Użytkownicy mają możliwość dostępu do aplikacji za pomocą komputerów stacjonarnych i laptopów. System nie posiada połączenia z zewnętrzną bazą danych. Opiera się on na przechowywaniu danych w Local Storage przeglądarki użytkownika. Środowiskiem uruchomieniowym jest dowolna przeglądarka internetowa.

Aplikacja zawiera opis każdej z poszczególnych heurystyk w celu lepszego ich zrozumienia. Wygląd systemu stworzony jest w sposób minimalistyczny, aby wspierać skupienie uwagi audytora na analizowanym interfejsie. System umożliwia także jednocześnie analizowanie wgranej aplikacji i zaznaczanie przycisków określających jej zgodność ze standardami. Bliskie ułożenie elementów przycisków i listy powinno skutkować sprawnym przejściem do kolejnej heurystyki.

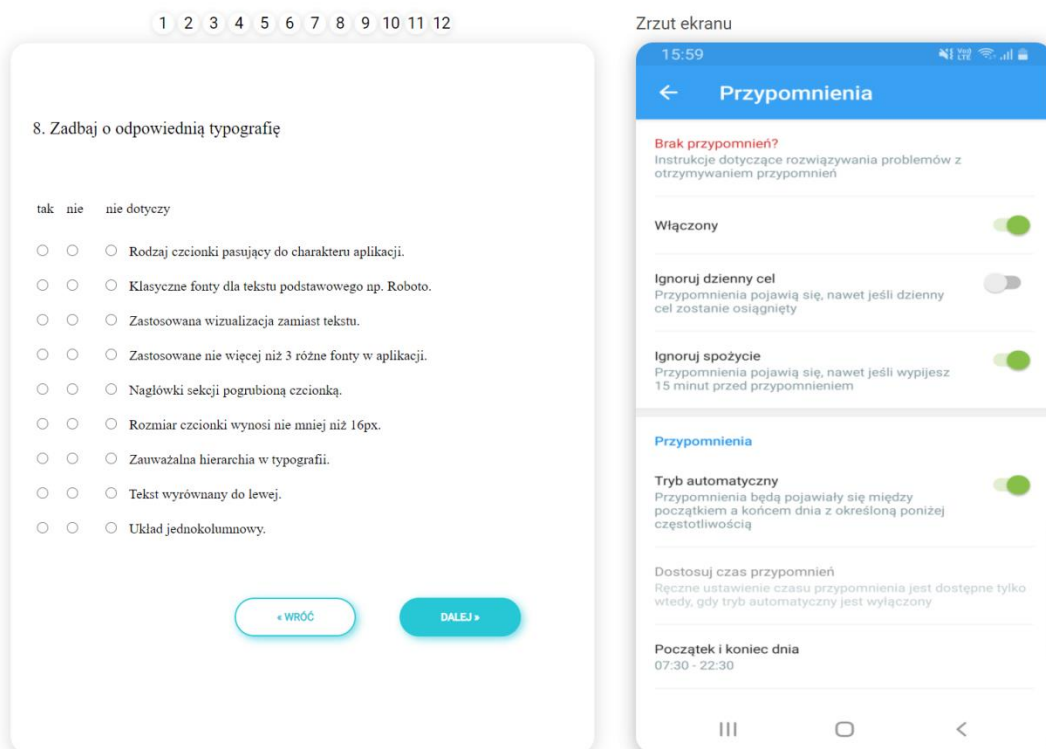
4.2 Interfejs użytkownika

Stworzony system zawiera stronę główną oraz cztery podstrony. Całość aplikacji utrzymana jest w minimalistycznej kolorystyce bieli i błękitu. Na poziomie zakładki „home” system umożliwia audytorowi wgranie zrzutu ekranu aplikacji mobilnej poprzez przycisk „wgraj swoją aplikację”. W widoku blokowym przedstawione zostały heurystyki wraz z krótkim opisem. Zrzut ekranu obrazujący stronę główną przedstawiony został na rysunku 7.



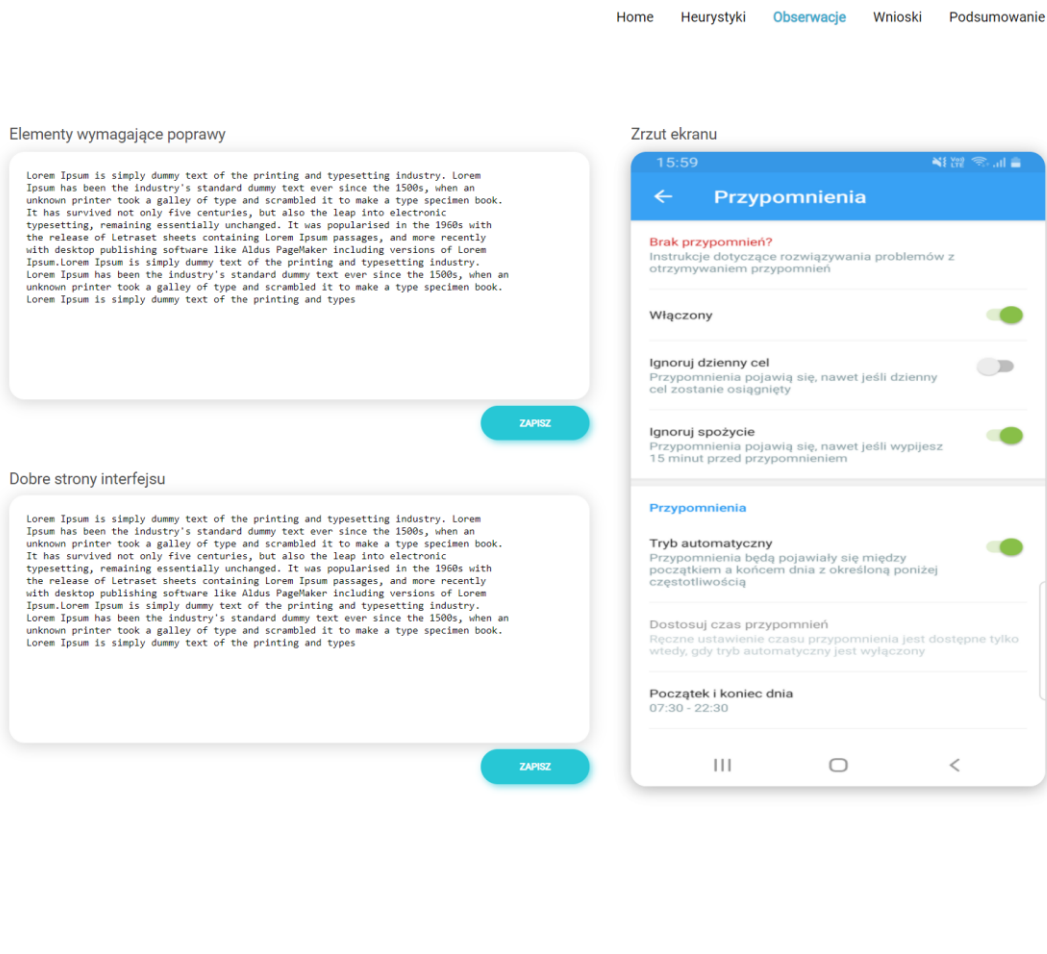
Rysunek 7 Zrzut ekranu strony głównej systemu
Źródło: opracowanie własne

Dzięki wgranemu zrzutowi ekranu zapisującemu się na bieżąco w Local Storage przeglądarki internetowej, możliwa jest weryfikacja wgranego interfejsu ze standardami na kolejnych podstronach aplikacji. Podstrona „heurystyki” umożliwia użytkownikowi przeanalizowanie aplikacji w oparciu o standardy. Za pomocą przycisków typu *radio button* możliwe jest zaznaczenie jednej z trzech opcji widocznych przy każdym punkcie listy kontrolnej: „tak” w przypadku poprawnego zastosowania zasady w aplikacji, „nie” w sytuacji, gdy dany punkt nie jest spełniony lub „nie dotyczy”, gdy aplikacja nie posiada danej funkcjonalności/elementu. Każda z poszczególnych heurystyk znajduje się na kolejnej karcie slidera. Przycisk „dalej” powoduje przejście do kolejnej heurystyki. W tym czasie w zakładce „podsumowanie” generowany jest wykres wyniku dla danej heurystyki. Przycisk „wstecz” pozwala użytkownikowi na powrót do poprzedniej heurystyki. W przypadku zmiany zaznaczeń wykres jest na bieżąco aktualizowany. Zrzut ekranu przedstawiający listę heurystyk wraz z wgraną przykładową aplikacją znajduje się na rysunku 8.

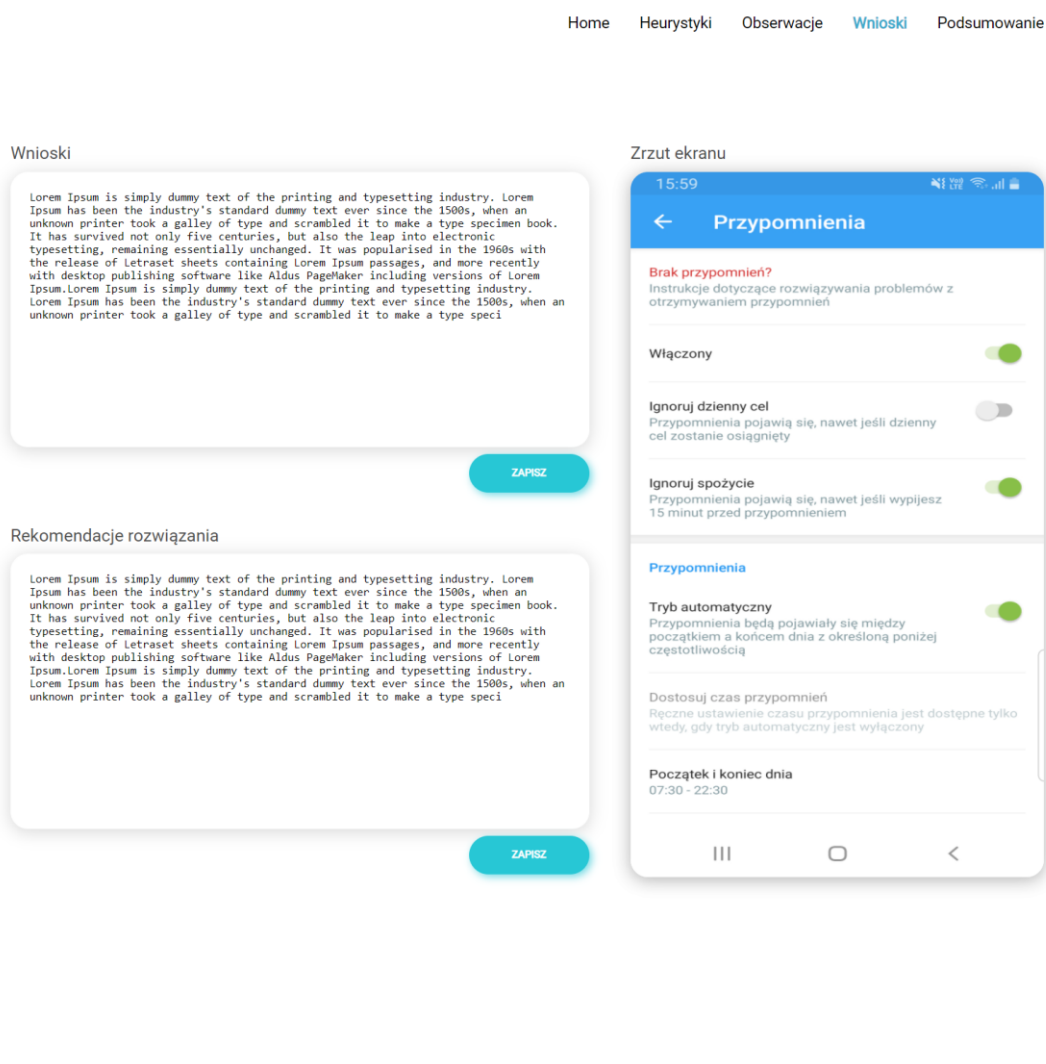


Rysunek 8 Zrzut ekranu strony zawierającej listę standardów i przykładowy interfejs
Źródło: opracowanie własne

Aplikacja poza listą gotowych standardów daje także użytkownikowi możliwość indywidualnej analizy interfejsu. System na podstronach „obserwacje” i „wnioski” oferuje pola tekstowe, w których audytor może opisać swoje obserwacje - złych jak i dobrych stron interfejsu oraz wniosków i rekomendacji. Elementy zapisane przez eksperta generowane są do raportu. Przedstawione na rysunku 9 i 10 zrzuty ekranu obrazują podstrony z obserwacjami oraz wnioskami audytora.



Rysunek 9 Zrzut ekranu strony oferującej pola tekstowe obserwacji
Źródło: opracowanie własne



Rysunek 10 Zrzut ekranu strony oferującej pola tekstowe wniosków i rekomendacji
Źródło: opracowanie własne

Wygenerowany wykres z uprzednio zaznaczonych list, procentowy zapis przedstawiający zgodność ze standardami całej aplikacji wraz z obserwacjami i wnioskami umieszczony jest na podstronie „podsumowanie”. Odznaczające się kolory wykresu i procentowego spełnienia standardów powodują szybsze zwrócenie uwagi na istotne wyniki z raportu. W tym miejscu możliwe jest także wygenerowanie wykresu do pliku PDF. Zrzut ekranu przedstawiający wygenerowany raport na podstronie przedstawia rysunek 11.

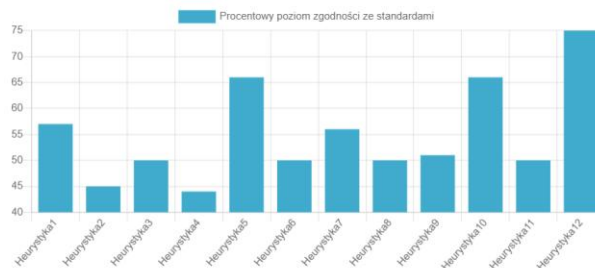
Raport audytu eksperckiego

Zapisz jako PDF

POZIOM ZGODNOŚCI ZE
STANDARDAMI

57%

ZGODNOŚĆ Z HEURYSTYKAMI



DOBRE STRONY INTERFEJSU

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries.

ELEMENTY WYMAGAJĄCE POPRAWY

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries.

WNIOSKI

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type speci

REKOMENDACJE ROZWIĄZANIA

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type speci

Rysunek 11 Zrzut ekranu strony przedstawiającej wygenerowane dane
Źródło: opracowanie własne

4.3 Testowanie aplikacji

Za pomocą opisanych kroków testowych przeprowadzono testy manualne najważniejszych funkcjonalności aplikacji. Przetestowano funkcje wgrywania zrzutu ekranu aplikacji mobilnej, działania przycisków oraz slidera heurystyk, zapisu tekstu w raporcie, jak i generowania wykresów. Dokładne scenariusze testowe zawarte są w tabelach 1-8, opracowanych przez autorkę niniejszej pracy.

Tabela 1 Opis testu pierwszego

Podstawowe informacje		Data utworzenia	Testowana wartość	Opis testu	Typ testu	Dodatkowe informacje	Priorytet testu
Numer testu	1	16.02.2021	Poprawne działanie funkcji wgrywania zdjęć oraz wyświetlania ich na podstronach aplikacji.	Weryfikowane jest prawidłowe działanie przycisku odpowiedzialnego za wgrywanie zrzutu ekranu aplikacji mobilnej do trzech kolejnym podstron stworzonego systemu.	manualny	brak	wysoki
Tytuł testu	Wyświetlenie wgranego interfejsu aplikacji						
Autor testu	Anna Woś						
Cel testu	Weryfikacja prawidłowego działania funkcji wgrywania zrzutów ekranu aplikacji mobilnej						
Rodzaj testu	Manualny						
Nazwa aplikacji	Aplikacja do przeprowadzania audytów						

Tabela 2 Kroki testowe testu pierwszego

Kroki testowe				
Lp.	Opis kroku	Oczekiwany rezultat	Wynik	Uwagi
1	Najechanie kursorem na przycisk "wgraj swoją aplikację".	Efekt zmiany koloru przycisku.	pozytywny	brak
2	Naciśnięcie przycisku "wgraj swoją aplikację".	Pojawienie się okna umożliwiającego wybór zdjęcia.	pozytywny	brak
3	Wybranie obrazu zrzutu ekranu aplikacji mobilnej.	Zaznaczenie wybranego obrazu.	pozytywny	brak
4	Naciśnięcie przycisku "otwórz".	Wyświetlenie się okna dialogowego z komunikatem "zdjęcie zostało wgrane, przejdź do zakładki heurystyki".	pozytywny	brak
5	Naciśnięcie przycisku "OK" w oknie dialogowym.	Zniknięcie wyświetlanego wcześniej okna dialogowego.	pozytywny	brak
6	Naciśnięcie przycisku "heurystyki" menu górnego.	Przejdzie do podstrony "heurystyki" z widocznym wgranym zdjęciem.	pozytywny	brak
7	Naciśnięcie przycisku "obserwacje" menu górnego.	Przejdzie do podstrony "obserwacje" z widocznym wgranym zdjęciem.	pozytywny	brak
8	Naciśnięcie przycisku "wnioski" menu górnego.	Przejdzie do podstrony "wnioski" z widocznym wgranym zdjęciem.	pozytywny	brak

Tabela 3 Opis testu drugiego

Podstawowe informacje		Data utworzenia	Testowana wartość	Opis testu	Typ testu	Dodatkowe informacje	Priorytet testu
Numer testu	2	16.02.2021	Działanie zaznaczania punktów listy oraz przycisków odpowiedzialnych za zmianę widoku heurystyk.	Weryfikowane jest prawidłowe działanie funkcji zaznaczania przycisków typu radio button i przycisków "dalej" oraz "wstecz" odpowiedzialnych za przesuwanie slidera.	manualny	brak	wysoki
Tytuł testu	Poprawne działanie slidera w podstronie heurystyki						
Autor testu	Anna Woś						
Cel testu	Weryfikacja prawidłowego działania zaznaczania kolejnych pól listy kontrolnej oraz slidera						
Rodzaj testu	Manualny						
Nazwa aplikacji	Aplikacja do przeprowadzania audytów						

Tabela 4 Kroki testowe testu drugiego

Kroki testowe				
Lp.	Opis kroku	Oczekiwany rezultat	Wynik	Uwagi
1	Naciśnięcie przycisku "heurystyki" menu górnego.	Przejdzie do podstrony "heurystyki" z widocznym wgrany zdjęciem.	pozytywny	brak
2	Naciśnięcie losowego przycisku z kolumny "tak".	Pojawienie się koloru w naciśniętym przycisku.	pozytywny	brak
3	Naciśnięcie losowego przycisku z kolumny "nie" tego samego wiersza.	Pojawienie się koloru w naciśniętym przycisku. Odznaczenie przycisku z kolumny "tak".	pozytywny	brak
4	Naciśnięcie losowego przycisku z kolumny "nie dotyczy" tego samego wiersza.	Pojawienie się koloru w naciśniętym przycisku. Odznaczenie przycisku z kolumny "nie".	pozytywny	brak
5	Naciśnięcie przycisku "dalej".	Pojawienie się karty z kolejną heurystyką.	pozytywny	brak
6	Naciśnięcie przycisku "wstecz".	Wyświetlenie poprzedniej heurystyki.	pozytywny	brak

Tabela 5 Opis testu trzeciego

Podstawowe informacje		Data utworzenia	Testowana wartość	Opis testu	Typ testu	Dodatkowe informacje	Priorytet testu
Numer testu	3	16.02.2021	Działanie funkcji generowania wykresów.	Weryfikowana jest poprawność działania generowania wykresów poprzez zaznaczenie przycisków.	manualny	brak	wysoki
Tytuł testu	Generowanie wykresów						
Autor testu	Anna Woś						
Cel testu	Weryfikacja poprawnego generowania wykresów do raportu.						
Rodzaj testu	Manualny						
Nazwa aplikacji	Aplikacja do przeprowadzania audytów						

Tabela 6 Kroki testowe testu trzeciego

Kroki testowe				
Lp.	Opis kroku	Oczekiwany rezultat	Wynik	Uwagi
1	Naciśnięcie przycisku "heurystyki" menu górnego.	Przejdzie do podstrony "heurystyki".	pozytywny	brak
2	Kliknięcie wszystkich przycisków z kolumny "tak" w karcie pierwszej heurystyki.	Zaznaczenie wszystkich przycisków kolumny "tak" na kolor niebieski.	pozytywny	brak
3	Kliknięcie przycisku "dalej".	Wyświetlenie karty z drugą heurystyką.	pozytywny	brak
4	Naciśnięcie przycisku "podsumowanie" menu górnego.	Wyświetlenie zawartości podstrony "podsumowanie" z wygenerowanym wykresem.	pozytywny	Na wykresie wyświetlony jest tylko jeden słupek, ponieważ tylko dla jednej heurystyki zostały zaznaczone przyciski.

Tabela 7 Opis testu czwartego

Podstawowe informacje		Data utworzenia	Testowana wartość	Opis testu	Typ testu	Dodatkowe informacje	Priorytet testu
Numer testu	2	16.02.2021	Odpowiednie wyświetlanie wpisanego tekstu w polu tekstowym w raporcie	Weryfikowane jest prawidłowe działanie funkcji zapisu wpisanego przez użytkownika tekstu w raporcie	manualny	brak	wysoki
Tytuł testu	Zapis tekstu w raporcie						
Autor testu	Anna Woś						
Cel testu	Weryfikacja prawidłowego wyświetlania tekstu w raporcie						
Rodzaj testu	Manualny						
Nazwa aplikacji	Aplikacja do przeprowadzania audytów						

Tabela 8 Kroki testowe testu czwartego

Kroki testowe				
Lp.	Opis kroku	Oczekiwany rezultat	Wynik	Uwagi
1	Naciśnięcie przycisku "obserwacje" menu górnego.	Wyświetlenie zawartości podstrony "obserwacje".	pozytywny	brak
2	Naciśnięcie na pole tekstowe.	Pojawienie się kursora w polu tekstowym.	pozytywny	brak
3	Zapis losowego tekstu w polu tekstowym.	Wyświetlenie zapisanego tekstu w polu tekstowym.	pozytywny	brak
4	Naciśnięcie przycisku "zapisz".	Zmiana koloru naciskanego przycisku.	pozytywny	brak
5	Naciśnięcie przycisku "podsumowanie" menu górnego.	Wyświetlenie zawartości podstrony "podsumowanie" z wypełnionym polem tekstowym wpisanym przez użytkownika.	pozytywny	brak

Po przeprowadzeniu testów nie zaobserwowano błędów w działaniu weryfikowanych funkcjonalności aplikacji. Przeprowadzone testy dały wynik pozytywny dla każdego z badanych kroków.

Zakończenie

Celem pracy było opracowanie standardów projektowania aplikacji mobilnych oraz stworzenie systemu wspomagającego przeprowadzanie audytów eksperckich. Dzięki analizie dostępnej literatury i istniejących aplikacji stworzono zestaw 12 heurystyk opisujących zasady projektowania interfejsów mobilnych. Stworzono także aplikację webową, mającą na celu wsparcie w tworzeniu audytu dla aplikacji mobilnych. System pozwala na analizę i ocenę aplikacji mobilnej oraz generowanie raportu z zebranych danych. Praca zawiera charakter innowacyjny ze względu na opracowane heurystyki poszerzone o listę kontrolną.

Niniejsza praca nie wyczerpuje ostatecznie tematu projektowania interfejsów mobilnych. W pracy skupiono się na graficznym interfejsie, ponieważ istnieją normy regulujące techniczne standardy projektowe. Skoncentrowano się na wąskim zakresie urządzeń mobilnych, jakim są smartfony oparte o system Android. Kolejnym elementem, na który należy zwrócić uwagę, są zagadnienia dotyczące dostępności interfejsów mobilnych dla osób ze specjalnymi potrzebami (dysfunkcjami).

Bibliografia

- [1] S. Alfajhan, „Software Development and Mobile Devices,” 2016.
- [2] L. Costello i M. Powers, „Developing In-House Digital Tools in Library Spaces,” IGI Global, Hershey , 2018.
- [3] P. Perea i P. Giner, „Potrzeby związane z naszymi ludzkimi cechami,” w *UX Design Projektowanie aplikacji dla urządzeń mobilnych*, Gliwice, Helion SA, 2019, pp. 26-27.
- [4] S. Krug, „Wielka mobilizacja, czyli świat w kieszeni,” w *Nie każ mi myśleć! O życiowym podejściu do funkcjonalności stron internetowych*, HELION, 2014, pp. 155-158.
- [5] M. P.-W. M. Borys, „Badanie użyteczności oraz dostępności interfejsu,” Wydawnictwo Uniwersytetu Rzeszowskiego, 2013.
- [6] R. B. J. Nielsen, *Mobile Usability*, 2013.
- [7] S. H. E. Berkman, *Designing Mobile Interfaces*, Featured, 2011.
- [8] E. Canziba, *Hands-On UX Design for Developers: Design, prototype, and implement compelling user experiences from scratch*, Birmingham: Packt Publishing, 2018.
- [9] J. Clark, *Designing for Touch*, New York: A Book Apart, 2015.
- [10] J. Lehtimäki, *Smashing Android UI*, 2012.
- [11] „Inspect CPU activity with CPU Profiler,” 28 05 2020. [Online]. Available: <https://developer.android.com/studio/profile/cpu-profiler>. [Data uzyskania dostępu: 12 07 2020].
- [12] „Konsola Play - Pomoc,” 2020. [Online]. Available: <https://support.google.com/googleplay/android-developer/answer/113469?hl=pl>. [Data uzyskania dostępu: 18 07 2020].
- [13] ISO, „Online Browsing Platform (OBP),” 2018. [Online]. Available: <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en>. [Data uzyskania dostępu: 29 sierpień 2020].
- [14] C. Badura, *UXUI Design Zoptymalizowany*, Gliwice: Helion, 2019.
- [15] B. R.-T. I. Mościchowska, *Badania Jako Podstawa Projektowania User Experience*, PWN, 2015.
- [16] D. Travis, „Userfocus,” Userfocus, [Online]. Available: <https://www.userfocus.co.uk/resources/guidelines.html>. [Data uzyskania dostępu: 12 10 2020].

2020].

- [17] R. M. J. Nielsen, „Heuristic evaluation of user interfaces,” April 1990. [Online]. Available: <https://dl.acm.org/doi/pdf/10.1145/97243.97281>. [Data uzyskania dostępu: 13 09 2020].
- [18] D. Norman, *The Design of Everyday Things*, New York: Basic Books, 2013.
- [19] A. Dix, R. Beale, J. Finlay i G. D. Abowd, „Human-computer Interaction,” Pearson Education Limited, Harlow, 2004.
- [20] M. Borys i M. Plechawska-Wójcik, „Badanie użyteczności oraz dostępności interfejsu,” Wydawnictwo Uniwersytetu Rzeszowskiego, 2013.
- [21] J. Gerhardt-Powals, „Cognitive Engineering Principles for Enhancing Human-Computer Performance,” New Jersey.
- [22] I. Connell, „Full Principles Set. Set of 30 usability evaluation principles compiled by the author from the HCI literature,” 2000.
- [23] T. R. J. V. L. P. D. L. P. T. K. J. Zhang, „Using usability heuristics to evaluate patient safety of medical devices,” 2003.
- [24] J. A. Kientz, „Heuristic Evaluation of Persuasive Health Technologies,” Washington, 2010.
- [25] B. G. A. Sutcliffe, „Heuristic evaluation of virtual reality applications,” Manchester, 2004.
- [26] P. W. Jordan, „Cognitive Walkthrough,” w *An Introduction to Usability*, Taylor & Francis Ltd, 2002, pp. 79-80.
- [27] J. Nielsen, „10 Usability Heuristics for User Interface Design,” 1995.

Spis rysunków

Rysunek 1 Prawo bliskości	9
Rysunek 2 Prawo podobieństwa	10
Rysunek 3 Prawo ciągłości	10
Rysunek 4 Strefa kciuka.....	12
Rysunek 5 Zrzut ekranu prototypu aplikacji wykonany w programie Adobe XD.....	30
Rysunek 6 Zrzut ekranu prototypu w aplikacji Zeplin.....	31
Rysunek 7 Zrzut ekranu strony głównej systemu	33
Rysunek 8 Zrzut ekranu strony zawierającej listę standardów i przykładowy interfejs	35
Rysunek 9 Zrzut ekranu strony oferującej pola tekstowe obserwacji	36
Rysunek 10 Zrzut ekranu strony oferującej pola tekstowe wniosków i rekomendacji	37
Rysunek 11 Zrzut ekranu strony przedstawiającej wygenerowane dane	38

Spis tabel

Tabela 1 Opis testu pierwszego.....	39
Tabela 2 Kroki testowe testu pierwszego.....	39
Tabela 3 Opis testu drugiego.....	40
Tabela 4 Kroki testowe testu drugiego.....	40
Tabela 5 Opis testu trzeciego	40
Tabela 6 Kroki testowe testu trzeciego	41
Tabela 7 Opis testu czwartego.....	41
Tabela 8 Kroki testowe testu czwartego	41

Spis załączników

Pliki źródłowe (zamieszczone w repozytorium): <https://github.com/ankaw401/System-przeprowadzania-audytow-eksperckich>