

LABORATORIUM 3. PREDYKCJA KLM I HEURYSTYKI NIELSENSA: ANALIZA ISTNIEJĄCYCH INTERFEJSÓW OPROGRAMOWANIA MOBILNEGO

Cel laboratorium:

Celem zajęć jest nabycie umiejętności analizy interfejsów oprogramowania mobilnego w oparciu o predykcję KLM i heurystyki Nielsena.

Zakres tematyczny zajęć:

- predykcja KLM,
- heurystyki Nielsena,
- analiza interfejsów oprogramowania mobilnego.

Pytania kontrolne:

1. Jakie są korzyści ze stosowania predykcji KLM dla interfejsów oprogramowania mobilnego?
2. Jakie są przykłady dobrych realizacji heurystyk Nielsena w oprogramowaniu mobilnym?

Wprowadzenie

Ze względu na fakt omówienia predykcji KLM oraz heurystyk Nielsena w ramach *Laboratorium 1*, w niniejszym rozdziale przedstawione zostaną głównie przykłady analiz interfejsów oprogramowania mobilnego.

Dla aplikacji mobilnych podobnie jak dla aplikacji webowych i desktopowych, metoda KLM znajduje zastosowanie w ocenie czasu realizacji zadań. Tab. 3.1 przedstawia ogólne zestawienie operatorów KLM dla aplikacji mobilnych. Jak widać, są one powiązane z gestami mobilnymi a ich czasy – uśrednione. W predykcji KLM dla aplikacji mobilnych zastosowanie mają również operatory poznane w ramach *Laboratorium 1* i 2, np.: M, R, K.

Tabela 3.1. Zestaw operatorów KLM dla aplikacji mobilnych

Operatory dla aplikacji mobilnych:			
Oznaczenie	Gesty	Opis	Czas realizacji (s)
V	Scroll, pan, flick, rotate, itp.	Ruchy w górę i w dół interfejsu, wraz z obracaniem, itp.	0,8
T	Tap	Dotknięcie interfejsu	0,1
TT	Double tap	Podwójne dotknięcie interfejsu	0,2
L	Long press	Przytrzymanie, np.: przycisku w interfejsie	1,0



Fundusze Europejskie
Wiedza Edukacja Rozwój



**Rzeczpospolita
Polska**

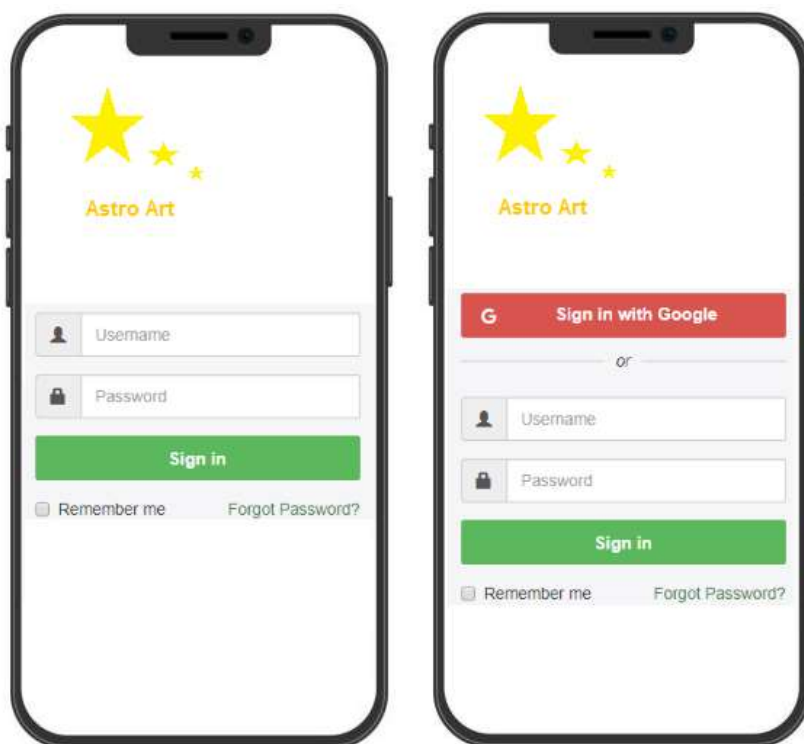
PAGE
VS
MERGE
FORMA
T54

Unia Europejska
Europejski Fundusz Społeczny



Założmy, że w zaprojektowanej aplikacji mobilnej projektant chce przeanalizować formę logowania. Wśród dwóch rozważanych opcji znajdują się: logowanie za pomocą loginu i hasła oraz logowanie przez *Gmail*. Projektant zaprojektował widoki interfejsu aplikacji mobilnej dla obu rozważanych opcji – Rys. 3.1. Chcąc przetestować szybkość realizacji zadania logowania, należy stworzyć scenariusze w oparciu o szkice interfejsu. Przykładowe scenariusze znajdują się na Rys. 3.2. Zliczając czasy poszczególnych operatorów w danych scenariuszach otrzymujemy następujące czasy realizacji:

- scenariusz 1: $T + 3 \times K + T + 8 \times K + T = 3,28 \text{ s}$
- scenariusz 2: $T = 0,1 \text{ s}$



Rys. 3.1. Widoki interfejsu dla logowania poprzez login i hasło oraz konto Gmail Google

SCENARIUSZ 1	SCENARIUSZ 2
Logowanie poprzez login i hasło	Logowanie poprzez konto Gmail Google
<ol style="list-style-type: none"> 1. Dotknięcie pola loginu (T) 2. Wpisanie loginu: xyz (3xK) 3. Dotknięcie pola hasła (T) 4. Wpisanie hasła: abcdefgh (8xK) 5. Dotknięcie pola <i>Sign in</i> (T) 	<ol style="list-style-type: none"> 1. Dotknięcie przycisku <i>Sign in with Google</i> (T)



Rys.3.2. Przykładowe scenariusze badawcze

Samo uzyskanie czasów nie kończy analizy. Warto rozważyć, która z opcji logowania jest opcją bardziej ergonomiczną, bardziej przyjazną dla użytkownika. Czasami rozwiązania szybsze wbrew pozorom mogą okazać się rozwiązaniami gorszymi, ponieważ, np.: pomijają ważne działania, wykorzystują niewiedzę użytkownika, zmuszają użytkownika do działania wbrew sobie. Z analizy scenariusza 1 i 2 wynika, że lepszą opcją jest logowanie przez *Gmail* z racji szybkości, jednak w przypadku osób, które nie posiadają kont na tych platformach lub nie chcą dokonywać powiązań z aplikacją, rozwiązanie logowanie loginem i hasłem staje się cenniejsze. Także wszystko zależy od preferencji użytkownika. Natomiast widać, że czasy obu wersji logowania są niewielkie.

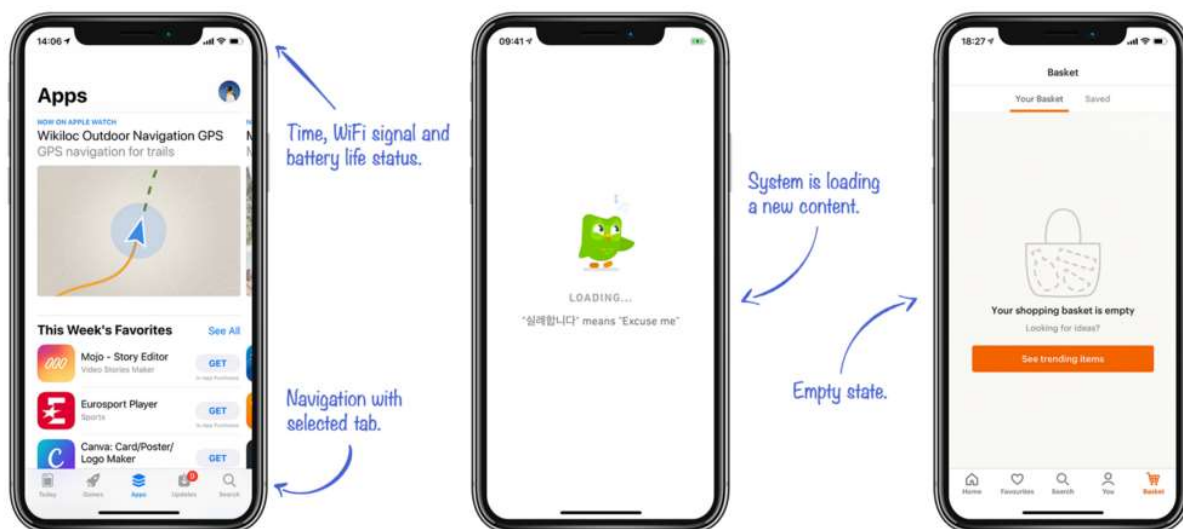
Poznane w ramach **Laboratorium 1** oraz **Laboratorium 2 heurystyki Nielsena** odnoszą się również do aplikacji mobilnych. Ze względu na ich wcześniejsze omówienie w poprzednich laboratoriach, poniżej zapisano wyłącznie wybrane przykłady realizacji heurystyk w oprogramowaniu mobilnym, zaś na Rys. 3.3 – 3.7 pokazano przykłady.

Heurystyki Nielsena a oprogramowanie mobilne:

1. **Widoczny status systemu** – przykłady: paski postępu, breadcrumbs, informacja na ekranie o stanie baterii, połączeniu Wi-fi bądź nieodebranych połączeniach. Dodatkowo animacje pokazujące aktywność systemu, potwierdzenia wykonania akcji, kroki i umiejscowienie użytkownika, informacje o zakończeniu przetwarzania.
2. **Zgodność systemu z rzeczywistością** – przykłady: zrozumiałe funkcje i prosty język, np.: *Dodaj*, *Usuń*, *Kup*; zrozumiałe ikonki, symbolika, np.: koło zębate jako *Ustawienia*, telefon jako *Połączenie*, karteczka jako *Notatka*.
3. **Kontrola i swoboda działań** – przykłady: przyciski lub opcje *cofnij*, *ponów*, *anuluj*. Ponadto możliwość: wyjścia lub przerywania działania, podglądu wprowadzanych zmian, cofnięcia operacji lub zamknięcia okna.
4. **Zachowanie jednakowych konwencji w obrębie serwisu** – przykłady: zgodność ze standardami aplikacji tj. logotyp z lewej strony na samej górze oraz podlinkowany do strony głównej, wewnętrzna wyszukiwarka umieszczona w prawym górnym rogu, a link do regulaminu na dole strony. Ponadto dostosowanie rozmieszczenia elementów w zależności od trendów i ograniczeń fizycznych dłoni. W ramach konwencji systemu zadbanie również o kolory, czcionki, komponenty – by ze sobą współgrały.
5. **Zapobieganie błędom** – przykłady: stosowanie bezpiecznych obszarów, gdzie użytkownik jeśli kliknie przypadkowo to aplikacji nie podejmie działania. Ponadto używanie ograniczeń, które nie pozwolą ustawiać niewłaściwej wartości (np.: zamiast podania cyfry, podano literę a strona reaguje). A także sugerowanie użytkownikom najczęściej wybieranych opcji lub używania okien dialogowych, np.: podczas załączania plików do wiadomości.

6. **Rozpoznawanie a nie zapamiętywanie** – przykłady: minimalizowanie liczby obiektów w interfejsie aplikacji a także ustawianie widoczności opcji. Stosowanie powszechnie kojarzących się ikon (łatwych do rozpoznawania) a także typowego słownictwa. Ponadto przypominanie o dokonanych wyborach, podgląd i wizualizacji opcji.
7. **Elastyczność i efektywność** – przykłady: możliwość customizowania akcji, zmiany ustawień, konfigurowania aplikacji w zależności od swoich potrzeb. Również możliwość dostosowywania opcji – pokazywanie i ukrywanie opcji, ułatwianie wyboru przez rekomendacje.
8. **Estetyka i minimalizm interfejsu** – przykłady: ograniczenie się do elementów, które są w dany momencie niezbędne na ekranie; zachowanie minimalistyczności projektu mimo wrażenia pustego ekranu.
9. **Właściwa obsługa błędów** – przykłady: wysyłanie użytkownikowi komunikatów ostrzegających i wyjaśniających jaki błąd wystąpił, oferowanie możliwości naprawienia błędu lub zdiagnozowania go.
10. **Pomoc i dokumentacja** – przykłady: posiadanie sekcji FAQ lub Pomoc. Również kierunkowskazy i wsparcie w kontakcie z systemem (ang. *onboarding*).

Na Rys. 3.3 – 3.7 przedstawiono przykładowe realizacje heurystyk Nielsena w oprogramowaniu mobilnym.



Rys.3.3. Realizacja zasady widoczności statusu systemu¹

¹ https://miro.medium.com/max/1000/1*mCkfYEEYnsUMKUDaR5bQRqw.png





Fundusze Europejskie
Wiedza Edukacja Rozwój



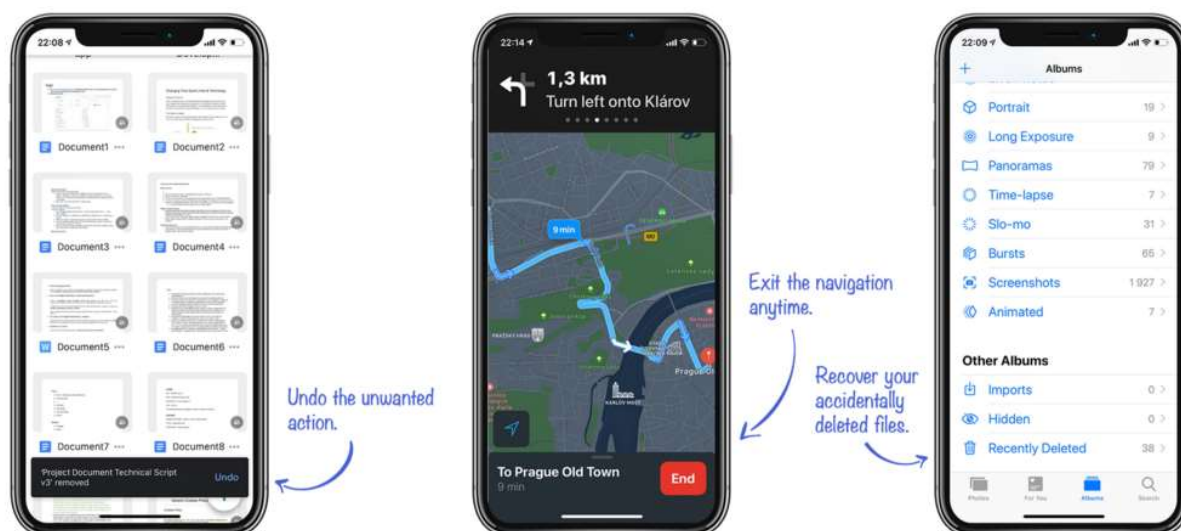
**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz Społeczny





Rys.3.4. Realizacja zasady zgodności systemu z światem rzeczywistym²



Rys.3.5. Realizacja zasady kontroli i swobody działań³

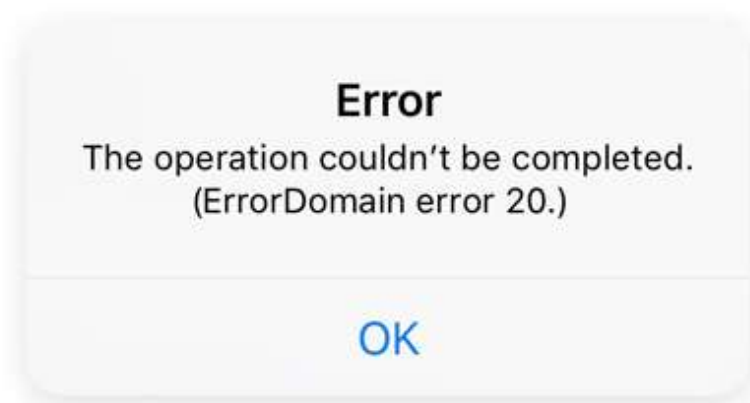
² https://miro.medium.com/max/700/1*x3shJBeY43-V4SW-2QyRBQ.png

³ https://miro.medium.com/max/1000/1*Z-39W0kNfXTvQX2qYmPZqw.png





Rys.3.6. Realizacja zasady zapobiegania błędom⁴



Rys.3.7. Realizacja zasady właściwej obsługi błędów⁵

⁴ https://miro.medium.com/max/1000/1*VvFMhN_s88gMjOTpJus_ng.png

⁵ https://miro.medium.com/max/616/1*61O8Jl6fNTecvK5TAoNFLw.png



Zadanie 3.1. Predykcja KLM dla interfejsów oprogramowania mobilnego

Korzystając z metody KLM, dokonaj oceny jakości interfejsu i realizacji dowolnego zadania w wybranym oprogramowaniu mobilnym lub wskazanym przez prowadzącego. Zaproponuj również alternatywną wersję interfejsu, dokonaj obliczeń i porównania z wersją pierwotną. Całość analiz zamieść w Tab. 3.2.

Zadanie 3.2. Heurystyki Nielsena a interfejsy oprogramowania mobilnego

Sprawdź zgodność dowolnie wybranej aplikacji mobilnej z heurystykami Nielsena. Zamieść zrzuty ekranowe pokazujące realizację heurystyk i dodaj komentarze oceniające sposób oraz poziom zgodność



Fundusze Europejskie
Wiedza Edukacja Rozwój



**Rzeczpospolita
Polska**

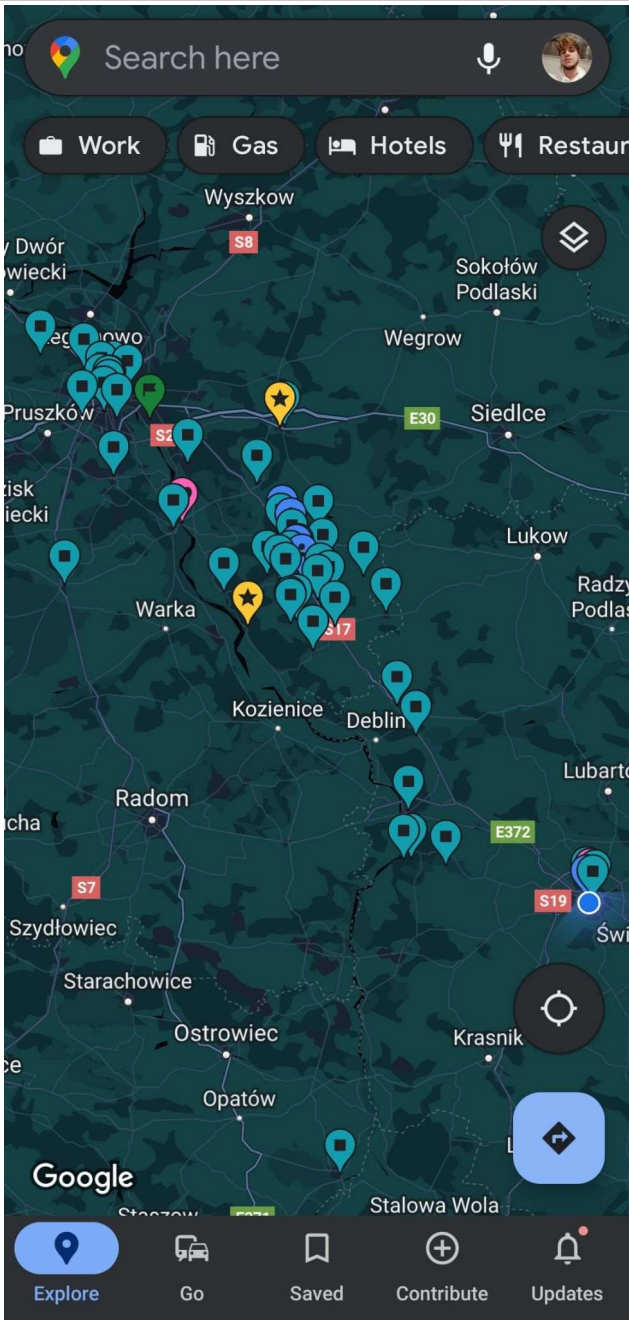
Unia Europejska
Europejski Fundusz Społeczny



Tabela 3.2. Interfejsy oprogramowania mobilnego – ocena i porównanie

Oprogramowanie mobilne	Google Maps
Zadanie	Wyznaczanie trasy do miasta “Warszawa”
SCENARIUSZ 1 (PODSTAWOWY)	
<ol style="list-style-type: none">1. Kliknięcie w pole wyszukiwania (T)2. Wpisanie hasła “Warszawa” (8xK)3. Kliknięcie w przycisk “Prowadź” (T)4. Kliknięcie w przycisk “Start” (T)	
Zrzut ekranowy interfejsu aplikacji mobilnej	



	
Zapis zadania w notacji KLM	
T+8xK+T+T	
Czas realizacji zadania dla scenariusza 1	2.54 s
SCENARIUSZ 2 (ALTERNATYWNY)	
<ol style="list-style-type: none"> 1. Kliknięcie w proponowane miejsce “Warszawa” (T) 2. Kliknięcie w przycisk “Prowadź” (T) 3. Kliknięcie w przycisk “Start” (T) 	



Fundusze Europejskie
Wiedza Edukacja Rozwój

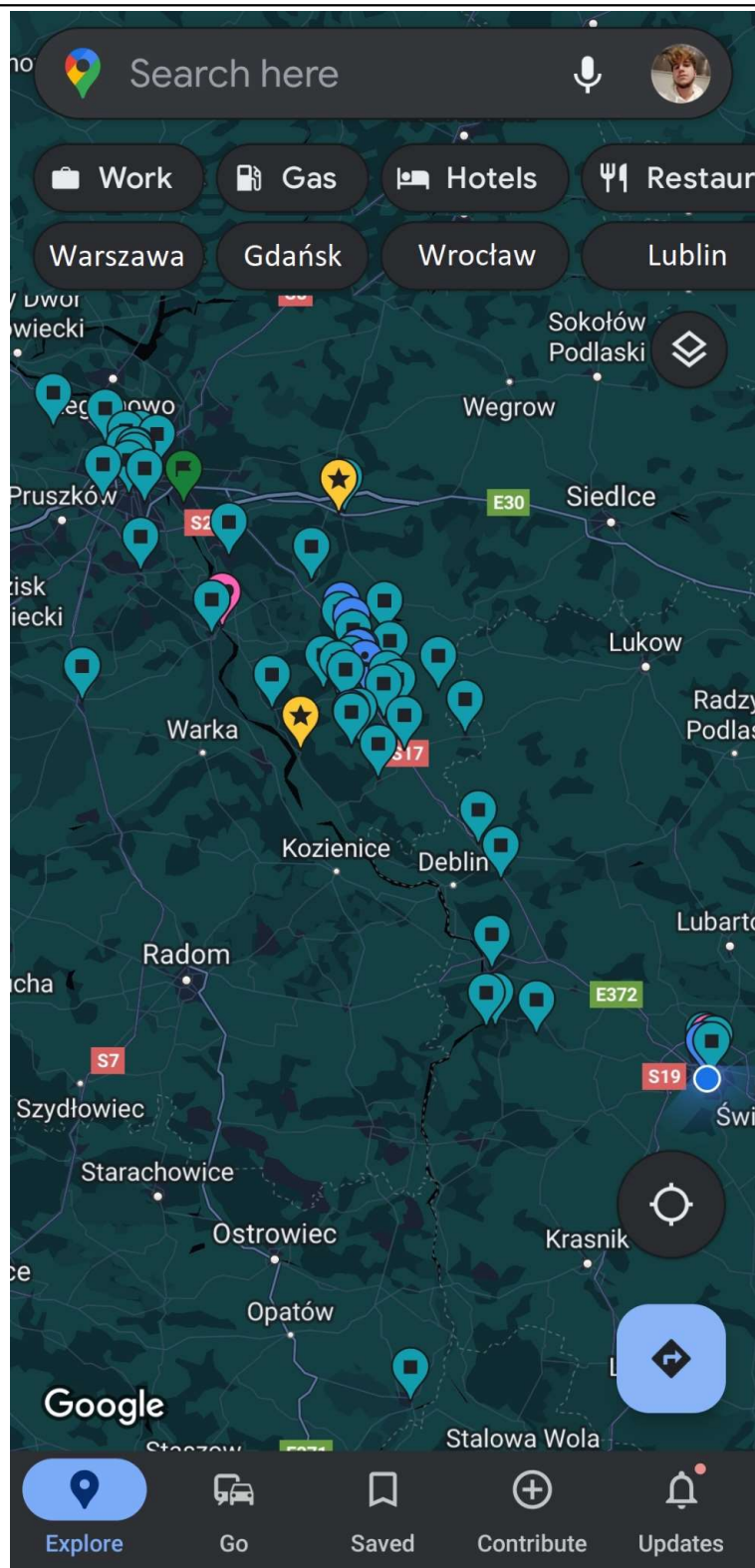


Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Społeczny



Propozycja interfejsu



Zapis zadania w notacji KLM



Fundusze Europejskie
Wiedza Edukacja Rozwój



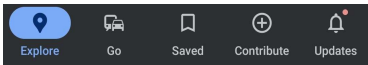
Rzeczpospolita Polska

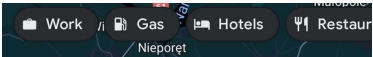
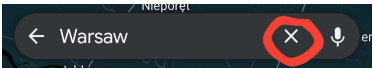


Unia Europejska
Europejski Fundusz Społeczny

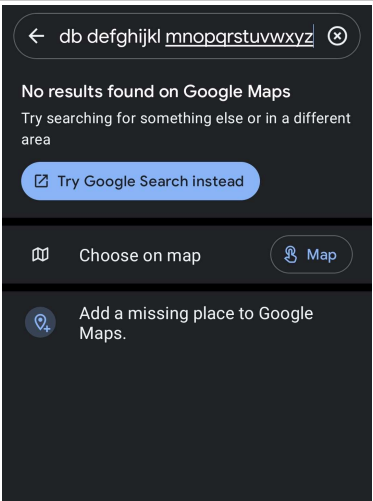


T + T + T	
Czas realizacji zadania dla scenariusza 2	0.3 s
Porównanie wyników scenariuszy i wnioski końcowe	
Wyznaczanie trasy według scenariusza 2 jest szybsze, użytkownik nie musi wyszukiwać miasta, aplikacja sama podpowiada miejsca, co wielokrotnie przyspiesza wykonanie akcji.	

Tabela 3.3. Analiza zgodności interfejsu oprogramowania mobilnego z heurystykami Nielsena

ANALIZA INTERFEJSU OPROGRAMOWANIA MOBILNEGO Z UŻYCIEM HEURYSTYK NIELSENA			
Użytkownik		Cel	Działania użytkownika
Standardowy użytkownik		Nawigacja	Użytkownik może wyszukiwać miejsca, sprawdzać i dodawać opinie oraz zdjęcia, korzystać z nawigacji
Oprogramowanie mobilne: Google Maps			
Lp.	Heurystyka	Realizacja	Przykład – zrzut ekranu
1.	Widoczny status systemu	<ul style="list-style-type: none">Użytkownik widzi w jakim module aplikacji się znajdujePo wyszukaniu miejsca, na mapie pokazuje się dane miejsce <p>Interfejs bardzo dobrze informuje użytkownika o obecnym statusie.</p>	
2.	Zgodność systemu z rzeczywistością	<ul style="list-style-type: none">Mapa wyświetlana jest podobnej formie jak tradycyjne mapyIkony przycisków reprezentują właściwie wykonywane działania <p>Ikony bardzo dobrze opisują ukryte pod nimi funkcje, zgodnie z rzeczywistością.</p>	
3.	Kontrola i swoboda działań użytkownika	<ul style="list-style-type: none">	

		Aplikacja pozwala użytkownikowi na dużą swobodę i kontrolę, lecz nie powoduje przy tym uczucia zagubienia u użytkownika.	
4.	Jednakowa konwencja w obrębie serwisu	<ul style="list-style-type: none"> Wszystkie ikony są zrobione w tym samym stylu graficznym Wszystkie aktywne przyciski i nieaktywne wyświetlają się w tych samych kolorach Wszystkie ważne przyciski modułów wyświetlają się na dolnym pasku, obok siebie Elementy na mapie tego samego typu wyświetlają się w ten sam sposób 	
5.	Zapobieganie błędom	<ul style="list-style-type: none"> Użytkownik po wyszukaniu może cofnąć <p>Interfejs jest tak skonstruowany, że trudno jest popełnić błąd, aplikacja odpowiednio to zabezpiecza.</p>	
6.	Rozpoznawanie a nie zapamiętywanie	<ul style="list-style-type: none"> Ikony przycisków są łatwe do zrozumienia Ważne funkcje są dostępne na głównym layoutcie aplikacji <p>Użytkownik może kierować się intuicją i skojarzeniami, aplikacja wspomaga użytkownika poprzez uzupełnianie i autopodpowiadanie.</p>	
7.	Elastyczność i efektywność	<ul style="list-style-type: none"> <p>Aplikacja posiada odpowiednie skróty i funkcje ułatwiające i przyspieszające pracę.</p>	
8.	Estetyka i minimalizm interfejsu	<ul style="list-style-type: none"> Na głównym interfejsie jest mała ilość funkcji, lecz są to funkcje najważniejsze Interfejs wykorzystuje małą ilość kolorów, przy czym każdy oznacza coś innego <p>Kolory i ikony są łatwe do zrozumienia przez użytkownika. Użytkownik ma możliwość minimalizowania i rozszerzenia interfejsu.</p>	

9.	Właściwa obsługa błędów	<ul style="list-style-type: none"> Jeżeli podczas wyszukiwania nie zostaną odnalezione żadne rezultaty, użytkownik zostanie o tym poinformowany <p>Aplikacja prawidłowo informuje użytkownika o błędach.</p>	
10.	Pomoc i dokumentacja	<ul style="list-style-type: none"> Użytkownik ma możliwość wyświetlenia pomocy opisującej wszystkie funkcje aplikacji Użytkownik ma możliwość zgłaszania problemów <p>Użytkownik ma możliwość skorzystania z szerokiej pomocy.</p>	
Ocena końcowa – wnioski i spostrzeżenia		<p>Interfejs użytkownika jest dobrze przystosowany dla użytkowników początkowych i zaawansowanych. Na głównym layoucie posiada małą ilość najważniejszych funkcji. Użytkownik ma możliwość dostosowywania interfejsu do swoich potrzeb. Kolory i ikony są łatwe do zrozumienia dla nowego użytkownika. Aplikacja wspomaga użytkownika poprzez autouzupełnianie i autopodpowiadanie.</p>	

