



Politechnika Wrocławskiego

Wydział Informatyki i Zarządzania

kierunek studiów: Informatyka
specjalność: Systemy Informacyjne

Praca dyplomowa - magisterska

Wykorzystanie Google Maps API do prezentacji geoznacznikowanych danych multimedialnych

Artur Ptaszek

słowa kluczowe:

Google Maps, geoznacznikowanie, mapy,
nawigacja, fotografie, użyteczność,
multimedia

krótkie streszczenie:

Praca porusza problem składowania danych geoznacznikowanych. Szczególny nacisk został położony na użyteczność prezentowania danych. Zrealizowane zostało to na aplikacji stworzonej w ramach pracy. Głównym narzędziem było Google Maps API.

opiekun pracy	dr inż. Andrzej Siemiński
diplomowej	Tytuł/stopień naukowy/imię i nazwisko	ocena	podpis

*Do celów archiwalnych pracę dyplomową zakwalifikowano do: **

a) kategorii A (akta wieczyste)

b) kategorii BE 50 (po 50 latach podlegające ekspercyzje)

** niepotrzebne skreślić*

pieczętka wydziałowa

Wrocław 2017 rok

W szczególności pragnę podziękować dr inż. Andrzejowi Siemińskiemu za poświęcony czas i cenne wskazówki, które pozwoliły na stworzenie niniejszej pracy. Dziękuję również mojej Narzeczonej i Rodzicom za wsparcie i motywację.

Streszczenie

Systemy GPS przez ostatnie lata zrewolucjonizowały poruszanie się po globie ziemskim. Powstała niezliczona ilość technologii wykorzystujących nawigację satelitarną do stworzenia aplikacji wspomagających życie codziennie. Praca przedstawia wykorzystanie tej technologii w kontekście efektywnej kategoryzacji zbiorów dokumentów.

Abstract

GPS Systems past couple of years revolutionized navigation around the Globe. A lot of technologies has created which uses the satellite navigation to create applications that support an everyday life. Thesis presents usage of this technology in the context of effective categorization of libraries of documents.

Spis treści

Wprowadzenie.....	9
1 Specyfikacja systemów nawigacji.....	11
1.1 Geneza	11
1.2 Przegląd technologii powstających przy pomocy GPS	11
1.2.1 Samochody autonomiczne	11
1.2.2 Mapy	12
1.3 Opis dostępnych formatów pozwalających zapisanie danych geolokalizacyjnych	12
1.3.1 GeoJSON	13
1.3.2 Keyhole Markup Language (KML)	13
1.3.3 GPS Exchange Format	14
1.3.4 EXIF	15
1.4 Problem składowania danych.....	16
1.4.1 Opis problemu.....	16
1.4.2 Grupowanie znaczników.....	16
1.4.3 Indeksowanie	17
1.4.4 GeoHash.....	19
1.4.5 Indeksowanie przy pomocy geohash'ów w aplikacji	22
2 Google Maps.....	23
2.1 Serwis Google Maps	23
2.2 Mapy Google API.....	24
2.3 Konkurencyjne rozwiązania.....	25
2.3.1 Apple Maps	25
2.3.2 Yandex Maps	26
2.3.3 Google Earth	27
2.3.4 Bing Maps	27
2.3.5 Open Street Maps.....	28
2.4 Aplikacje wykonane przy pomocy Google Maps.....	29
2.4.1 Pokemon Go.....	29
2.4.2 Uber.....	29
2.4.3 JakDojade.pl.....	29
3 Aplikacja.....	30
3.1 Podstawa teoretyczna	30
3.1.1 Single Page Application (SPA).....	30
3.1.2 MVC.....	30
3.1.3 Logowanie tokenowe	30
3.1.4 ImageMagick	32
3.2 Technologie	33
3.3 Wymagania funkcjonalne	35
3.4 Wymagania niefunkcjonalne.....	36
3.5 Analiza wykonanych prac.....	36
3.6 Opis aplikacji	36
3.6.1 Strona logowania.....	37
3.6.2 Ekran główny	38
3.6.3 Widok listy	40
3.6.4 Widok szczegółów dokumentu	41
3.7 Porównania podobnych systemów dostępnych na rynku	42
3.7.1 Adobe Lightroom	42
3.7.2 Zdjęcia Google	43
3.7.3 Zdjęcia Apple	43
4 Analiza użyteczności	45

4.1	Podstawa teoretyczna	45
4.1.1	Użyteczność aplikacji internetowych.....	45
4.1.2	Heurystyki.....	47
4.1.3	Śledzenie kliknięć	51
4.1.4	Testy A/B	52
4.1.5	Zdalne badania z użytkownikami	52
4.1.6	Metoda śledzenia oka.....	52
4.2	Badanie przy pomocy heurystyk.....	53
4.2.1	Strona logowania.....	53
4.2.2	Strona główna	55
4.2.3	Widok listy	58
4.2.4	Widok szczegółów	61
4.3	Badania według scenariusza.....	63
4.3.1	Scenariusze.....	63
4.3.2	Badana nr 1	63
4.3.3	Badana nr 2	65
4.3.4	Badany nr 3	66
5	Ocena metod.....	68
5.1	Heurystyki.....	68
5.1.1	Usystematyzowanie	68
5.1.2	Niski koszt.....	68
5.1.3	Natychmiastowe wyniki.....	68
5.1.4	Odpowiednia dla każdego projektu.....	68
5.1.5	Dobra metoda startowa	68
5.1.6	Zasady są bardzo ogólne	68
5.2	Badanie wg. scenariusza	69
5.2.1	Wizja osób korzystających z aplikacji	69
5.2.2	Wymagane przygotowania i znalezienie osób chętnych.....	69
5.2.3	Polityka firmy może nie pozwalać na wykonanie takich badań	69
5.2.4	Problem ze znalezieniem badanych	69
6	Wnioski	70
7	Spis tabel.....	71
8	Spis rysunków	72
9	Bibliografia	74

Wprowadzenie

Branża nawigacji i systemów satelitarnych bardzo szybko rozwinęła się na przestrzeni kilku ostatnich lat. Urządzenie pozwalające na uzyskanie aktualnej pozycji użytkownika znajduje się w każdym smartfonie i jest dostępne dla każdego użytkownika. Wiele firm takich jak Google prowadzi badania na tym polu co otwiera nowe możliwości dla twórcy aplikacji, by tworzyć bardziej rozbudowane systemy informatyczne korzystające z tego podejścia. Wraz z nowym narzędziem pojawia się problem użyteczności wytworzonych rozwiązań. Praca porusza temat użyteczności w kontekście aplikacji zbudowanych przy pomocy Google Maps API do kategoryzacji dokumentów geoznacznikowych.

Została ona podzielona na 10 rozdziałów.

Pierwszy rozdział opisuje znaczenie systemów nawigacji w naszym codziennym życiu. Zostały opisane technologie czerpiące najwięcej z popularyzacji systemów GPS, a także z jakimi problemami stykają się naukowcy w kontekście utrzymania tejże technologii. Przedstawione zostały obecne standardy przechowywania danych geolokalizacyjnych.

Rozdział drugi przybliża technologię Google Maps. Definiuje co oferuje to rozwiązanie i jakie inne konkurencyjne rozwiązania powstały na rynku. Pod koniec jest przegląd systemów, które powstały przy pomocy tej technologii.

W rozdziale czwartym została opisana aplikacja stworzona w ramach pracy. Między innymi przedstawione zostały powody jej stworzenia, wykorzystane technologie, postawione wymagania. Na samym końcu zostały opisane wszystkie ekran aplikacji i porównanie stworzonego systemu z innymi podobnymi rozwiązaniami.

Rozdział piąty testuje stworzoną aplikację w kontekście użyteczności. Wykorzystane zostały 2 techniki badań: heurystyki i scenariusze. Aby przybliżyć temat czytelnikowi problem użyteczności zostały przedstawione różne techniki badań na samym początku rozdziału.

Rozdział szósty przedstawia subiektywną ocenę metod badań użyteczności. Pod uwagę zostały wzięte różne środowiska biznesowe. Ocena przede wszystkim kładzie uwagę na wartość biznesową z użytych metod.

Rozdział siódmy podsumowuje wykonane badania. Przedstawia wnioski z przeprowadzonego eksperymentu.

Rozdziały ósmy i dziewiąty przedstawiają spis tabel i rysunków zawartych w pracy.

Bibliografia została zawarta w rozdziale dziesiątym.

1 Specyfikacja systemów nawigacji

Ostatnie lata były kluczowe dla rozwoju systemów map. Pierwsze implementacje zostały zapoczątkowane dla wojska, a względnie niedawno spopularyzowane przez firmy takie jak Google przez swoje mapy. W związku z nową technologią powstało wiele nowych problemów do rozwiązania, które zostaną opisane w dalszych częściach tej pracy. Otworzyło to wiele możliwości, które pozwoliły na stworzenie niewyobrażalnej ilości produktów. Uber, Airbnb, Google Maps to jedynie przykłady najprostszego wykorzystania systemu nawigacji. Branża jednak zauważała zdecydowanie większe możliwości tj. samochody autonomiczne, które pozwalają użytkownikowi wybrać ścieżkę, a następnie samochód wybiera trasę i sam rusza do zadanego celu.

Nawigacja jak widać redefiniuje pewne problemy m. in. kategoryzację zasobów, które są bardzo blisko związane z tematem tej pracy.

1.1 Geneza

System GPS został zbudowany na bazie innego systemu nawigacyjnego opartego na przekaźnikach radiowych LORAN. W roku 1956 Friedwardt Winterberg zaproponował test relatywistyczny z użyciem zegarów atomowych. Teoria względności w przypadku GPS jest bardzo ważnym elementem, ponieważ mogłoby to doprowadzić do bardzo dużej niedokładności całej nawigacji. Zakłada się, że dla obserwatora na ziemi czas biegnie o 38 mikrosekund szybciej niż na satelicie wystrzelonym na orbitę ziemską. Aby zredukować ten efekt wykorzystuje się teorie względności, aby zredukować błąd powstały na drodze sygnału z satelity.

Pierwszy system nawigacji został zbudowany dla Amerykańskiego wojska w roku 1960. Korzystał on z 5-ciu satelit okołoziemskich do wyznaczenia pozycji. Przy wykorzystaniu metody triangulacji można było wyznaczyć pozycję użytkownika raz na godzinę.

Początek wieku 21-szego był najważniejszym punktem zwrotnym dla rozwoju tej technologii dla cywili. W roku 2004 prezydent George W. Bush podpisał porozumienie z Unią Europejską w sprawie wykorzystania Amerykańskich satelit w celu uruchomienia cywilnej nawigacji satelitarnej. Zapoczątkowało to dynamiczny rozwój tej technologii. Zostały wystrzelone kolejne satelity na orbitę, a od tego momentu rynek zaczął budować układy dla urządzeń mobilnych. Jednak sama zdolność do poznania pozycji użytkownika nie jest wystarczająca. Zaczęło powstawać wiele oprogramowania wykorzystującego spopularyzowany system.

Aktualnie na orbicie jest dostępnych 31 satelit umożliwiających nawigację. W 2017 roku jest zaplanowane wystrzelenie 12 dodatkowych. Podniesie to znacznie dokładność całego systemu i większą dostępność w większych miastach. [1]

1.2 Przegląd technologii powstających przy pomocy GPS

Tak jak wcześniej wspomniałem nawigacja wykłarała nowe technologie. Poniżej przedstawię te bardziej popularne lub innowacyjne.

1.2.1 Samochody autonomiczne

Według mnie najbardziej rewolucyjna branża wykorzystująca dobroci nawigacji satelitarnej. Wyszczególnić można 2 firmy, które rywalizują na tym polu: Tesla i Google.

Oczywiście poza nimi prowadzonych jest wiele badań na uniwersytetach, lecz chciałbym poświęcić więcej uwagi na te 2 wyszczególnione.

Tesla jest firmą dosyć młodą na rynku samochodowym. Zasłynęła na rynku przez pierwsze samochody elektryczne wyglądające podobnie do aktualnych limuzyn niż do samochodów przyszłości. Jednak wygląd nie decyduje o braku innowacyjności. Tesla udowodniła, że jest w stanie wykrzesać z samochodu bardzo wiele. Początkowo samochody te nie posiadały modułu autonomicznego, jednak sprzęt był do tego przystosowany. Wraz z aktualizacją Model S uzyskał w pełni autonomiczny moduł, który zapewnia autonomiczność na poziomie 5 wg. skali SAE [2]. Oznacza to, że samochód Tesli jest w stanie poruszać się bez ingerencji człowieka. Nie byłoby to możliwe bez map do nawigacji i systemu GPS. Dzięki nim samochód wyznacza trasę i za nią podąża. Oczywiście cały system został tak zaprojektowany, aby w przypadku różnic między mapami, a rzeczywistą trasą zachować się właściwie na drodze. Jest to pierwsze rozwiązanie na rynku komercyjnym oferujące tak wysoki poziom autonomiczności. Wcześniej pojawiali się asystenci parkowania, tempomaty adaptacyjne i systemy utrzymywania pasa. Tesla poszła kilka kroków na przód i zapewniła najwyższy dotąd poziom autonomiczności. Oczywiście Tesla testuje swoje urządzenia w rzeczywistych warunkach i nie obyło się bez wypadków. Większość z nich jednak było winą człowieka. Tesla reagowała za szybko na zdarzenia, a człowiek nie był w stanie tak szybko zauważyć niebezpiecznych sytuacji.

Google jest kolejną korporacją inwestującą w samochody autonomiczne. Prowadzi rozbudowane badania z udziałem samochodów autonomicznych. Samochód Google jest na razie w fazie testów. Google Car został wyszczególniony w tym rankingu ze względu na inwestycję tej korporacji w technologie pośrednio związane z działką samochodów autonomicznych, a mianowicie Mapy i ich ciągłe udoskonalanie. Pozwoli to w przyszłości na lepszą adaptację samochodów autonomicznych do istniejącego otoczenia. Na podstawie [3].

1.2.2 Mapy

Komputerowe mapy są najbardziej popularnym zastosowaniem GPS. Większość osób w dzisiejszych czasach nie wyobraża sobie życia bez nich. Odnalezienie adresu sklepu lub najoptimalniejszej trasy nie jest już wyzwaniem. Korzystając z wyszukiwarki jest się w stanie w bardzo prosty i szybki sposób wyszukać żądane medium. Mapy poszły jednak krok dalej. Analizując aktualne pozycje użytkowników prowadzą statystyki natężenia ruchu na drogach, a także w lokalach. Pozwala to na uniknięcie zbędnego oczekiwania na światłach lub na skorzystanie z trasy alternatywnej. Te najbardziej zaawansowane udostępniają wirtualną wycieczkę po wyznaczonej trasie. Google dzięki swoim samochodom oferuje w/w funkcjonalność, która nazywa się Street View.

Tak jak zostało to wspomniane zostały one bazą do popularnych ostatnimi czasy aplikacji: Uber i Airbnb. Wiele miast w Polsce posiada wsparcie dla aplikacji JakDojade.pl, która także wykorzystuje Mapy do optymalizacji komunikacji miejskiej. Miasto Wrocław dodatkowo udostępnia aktualne pozycje autobusów co pozwala wyświetlenie rzeczywistych czasów przyjazdu autobusów i tramwajów. Rozwiążanie takie redukuje ciągłe wystawianie na przystankach i oczekiwanie na autobusy, które nie przyjeżdżają.

1.3 Opis dostępnych formatów pozwalających zapisanie danych geolokalizacyjnych

Składowanie danych geolokalizacyjnych jest jednym z ważnych aspektów każdego systemu nawigacyjnego, dlatego powstało wiele standardów. Pozwalają one na

geoznacznikowanie danych poruszanych w ramach tej pracy. Poniżej przedstawię te najbardziej popularne.

1.3.1 GeoJSON

Jest otwartym formatem dla zapisu kolekcji prostych geograficznych elementów z użyciem JSON. Elementy zawierają punkty (lokalizacja GPS lub jeden konkretny obiekt), linie (całe ulice, autostrady), wielokąty (państwa, dzielnice, miasta) lub kolekcje wyżej wymienionych elementów.

Format ten jest rozwijany przez grupę Internetową i programistów w przeciwnieństwie do innych standardów GIS¹

```
1. { "type": "FeatureCollection",
2.   "features": [
3.     { "type": "Feature",
4.       "geometry": { "type": "Point", "coordinates": [102.0, 0.5]},
5.       "properties": { "prop0": "value0" }
6.     },
7.     { "type": "Feature",
8.       "geometry": {
9.         "type": "LineString",
10.        "coordinates": [
11.          [102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0]
12.        ]
13.      },
14.      "properties": {
15.        "prop0": "value0",
16.        "prop1": 0.0
17.      }
18.    },
19.    { "type": "Feature",
20.      "geometry": {
21.        "type": "Polygon",
22.        "coordinates": [
23.          [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0],
24.            [100.0, 1.0], [100.0, 0.0] ]
25.        ]
26.      },
27.      "properties": {
28.        "prop0": "value0",
29.        "prop1": { "this": "that" }
30.      }
31.    }
32.  ]
33. }
```

Rys. 1.1 Przykładowy plik GeoJSON

1.3.2 Keyhole Markup Language (KML)

Język znaczników oparty na XML-u, otwarty standard zatwierdzony przez Open Geospatial Consortium pozwalający na wizualizację trójwymiarowych danych przestrzennych.

Wykorzystywany jest m.in. w aplikacjach Google Earth, Google Maps, Bing Maps, Flickr, NASA World Wind oraz Wikimapie.

¹ System Informacji Geograficznej (ang. Geographic Information System, GIS)

Słowo Keyhole pochodzi od nazwy programu, który ewoluował do Google Earth, a który nazywał się tak przed przejęciem firmy Keyhole Inc przez Google w 2004 roku.

Istnieje także spakowana forma KML o rozszerzeniu KMZ. Pozwala ona zmniejszyć rozmiar plików przez zastosowanie kompresji ZIP. Przez zastosowanie kompresji dodatkowo w takim pliku można załączyć dodatkowe dane takie jak ikony, obrazy satelitarne, modele 3D.

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <kml xmlns="http://www.opengis.net/kml/2.2">
3.   <Placemark>
4.     <name>My office</name>
5.     <description>This is the location of my office.</description>
6.     <Point>
7.       <coordinates>-122.087461,37.422069</coordinates>
8.     </Point>
9.   </Placemark>
10. </kml>
```

Rys. 1.2 Przykładowy plik KML

1.3.3 GPS Exchange Format

Zaprojektowana XML Schema do zapisu najczęściej używanych danych GPS dla aplikacji. Potrafi opisywać punkty, drogi i trasy. Format jest otwarty i może być wykorzystywany bez opłat licencyjnych. Dane lokalizacji i opcjonalnie czas oraz inne informacje są zapisane przy pomocy znaczników i mogą być wymieniane między różnymi urządzeniami GPS. Najczęściej aplikacje wykorzystują ten format do przedstawiania danych geolokalizacyjnych na różnych mapach i pokazujących różne typy danych, przykładowo geoznacznikowanie obrazów.

```
1. <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2.
3. <gpx xmlns="http://www.topografix.com/GPX/1/1" xmlns:gpxx="http://www.garmin.com/xmlschemas/GpxExtensions/v3" xmlns:gpxtpx="http://www.garmin.com/xmlschemas/TrackPointExtension/v1" creator="Oregon 400t" version="1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.topografix.com/GPX/1/1 http://www.topografix.com/GPX/1/1/gpx.xsd http://www.garmin.com/xmlschemas/GpxExtensions/v3 http://www.garmin.com/xmlschemas/GpxExtensionsv3.xsd http://www.garmin.com/xmlschemas/TrackPointExtension/v1 http://www.garmin.com/xmlschemas/TrackPointExtensionv1.xsd">
4.   <metadata>
5.     <link href="http://www.garmin.com">
6.       <text>Garmin International</text>
7.     </link>
8.     <time>2009-10-17T22:58:43Z</time>
9.   </metadata>
10.  <trk>
11.    <name>Example GPX Document</name>
12.    <trkseg>
13.      <trkpt lat="47.644548" lon="-122.326897">
14.        <ele>4.46</ele>
15.        <time>2009-10-17T18:37:26Z</time>
16.      </trkpt>
17.    </trkseg>
18.  </trk>
19. </gpx>
```

Rys. 1.3 Przykładowy plik GPX

1.3.4 EXIF

Standard metadanych dla plików z obrazkami. Jest on obsługiwany przez większość aparatów cyfrowych i oprogramowania graficznego, a także przez systemy operacyjne.

Pozwala on na przechowanie w pliku graficznym danych o parametrach:

- nazwę aparatu, którym wykonano zdjęcie
- ustawienia aparatu, takie jak czas naświetlenia, wartość przysłony, czułość matrycy w ISO czy ogniskowa obiektywu oraz położenie aparatu (pionowe/poziome)
- datę wykonania zdjęcia oraz przetworzenia na postać cyfrową
- informację o prawach autorskich (przeważnie dodawaną w post-processingu, niewiele aparatów dodaje ją automatycznie)
- miniaturkę obrazka
- rozdzielcość w pikselach
- sposób pomiaru światła przez aparat
- współrzędne GPS miejsca wykonania zdjęcia

```
1. exif:ColorSpace: 1
2. exif:ComponentsConfiguration: 1, 2, 3, 0
3. exif:DateTime: 2014:08:02 13:06:51
4. exif:DateTimeDigitized: 2014:08:02 13:06:51
5. exif:DateTimeOriginal: 2014:08:02 13:06:51
6. exif:ExifImageLength: 1920
7. exif:ExifImageWidth: 2560
8. exif:ExifOffset: 224
9. exif:ExifVersion: 48, 50, 50, 48
10. exif:ExposureBiasValue: 0/10
11. exif:ExposureMode: 0
12. exif:ExposureProgram: 3
13. exif:ExposureTime: 1/1702
14. exif:Flash: 1
15. exif:FlashPixVersion: 48, 49, 48, 48
16. exif:FNumber: 26/10
17. exif:FocalLength: 331/100
18. exif:ImageLength: 1920
19. exif:ImageWidth: 2560
20. exif:InteroperabilityOffset: 706
21. exif:ISOSpeedRatings: 50
22. exif:Make: SAMSUNG
23. exif:MakerNote: 7, 0, 1, 0, 7, 0, 4, 0, 0, 0, 48, 49, 48, 48, 2, 0, 4, 0, 1, 0, 0,
   0, 0, 32, 1, 0, 12, 0, 4, 0, 1, 0, 0, 0, 0, 0, 16, 0, 5, 0, 1, 0, 0, 0, 90, 0
   , 0, 0, 64, 0, 4, 0, 1, 0, 0, 0, 0, 0, 80, 0, 4, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0
   , 1, 3, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
24. exif:MaxApertureValue: 276/100
25. exif:MeteringMode: 1
26. exif:Model: GT-I8260
27. exif:Orientation: 1
28. exif:ResolutionUnit: 2
29. exif:SceneCaptureType: 0
30. exif:Software: I8260XXAMG2
31. exif:thumbnail:Compression: 6
32. exif:thumbnail:ImageLength: 240
33. exif:thumbnail:ImageWidth: 320
34. exif:thumbnail:InteroperabilityIndex: R98
35. exif:thumbnail:InteroperabilityVersion: 48, 49, 48, 48
36. exif:thumbnail:JPEGInterchangeFormat: 866
37. exif:thumbnail:JPEGInterchangeFormatLength: 11594
38. exif:thumbnail:Orientation: 1
```

```
39. exif:thumbnail:ResolutionUnit: 2  
40. exif:thumbnail:XResolution: 72/1  
41. exif:thumbnail:YResolution: 72/1  
42. exif:UserComment: 65, 83, 67, 73, 73, 0, 0, 0, 0, 85, 115, 101, 114, 32, 99, 111, 109,  
    109, 101, 110, 116, 115, 0  
43. exif:WhiteBalance: 0  
44. exif:XResolution: 72/1  
45. exif:YCbCrPositioning: 1  
46. exif:YResolution: 72/1
```

Rys. 1.4 Przykładowe dane EXIF

1.4 Problem składowania danych

Składowanie danych geolokalizacyjnych jest bardzo złożonym problemem. Naukowcy zaczęli prowadzić badania jak przechowywać dane, aby dostęp do nich byłby najoptymalniejszy. Jest to kolejny przykład branży, która musiała się rozwinąć, aby zaspokoić potrzeby biznesowe.

1.4.1 Opis problemu

Posiadając wielką ilość danych geolokalizacyjnych bardzo szybko można doprowadzić do wielkiego problemu wydajnościowego zarówno po stronie serwera, a także klienta. Zapytania do bazy danych będą wydłużały czas potrzebny do otrzymania danych przez co serwer nie odpowie w rozsądny dla użytkownika końcowego czasie. Jeszcze gorzej jest po stronie klienta, gdzie trzeba wymyślić rozwiązanie, które nie będzie wyświetlało każdego dodanego znacznika, ponieważ nie tyle co będzie wpływać na wydajność interfejsu, ale także na komfort korzystania z aplikacji.

1.4.2 Grupowanie znaczników

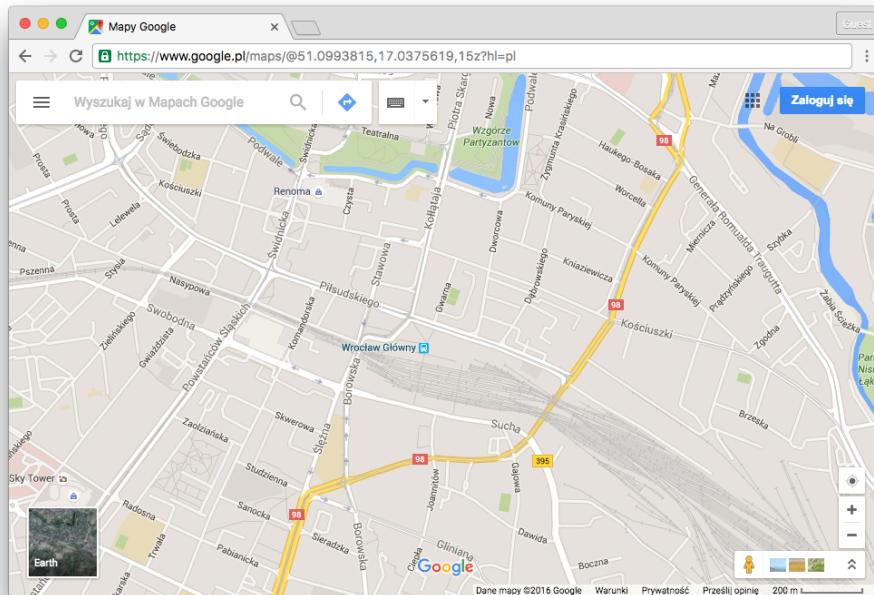
Wielka ilość danych, które powinny być wyświetcone na mapie można grupować. Jest to powszechna praktyka, z której korzysta wiele serwisów przedstawiające dane na mapie.

Przykład pokazany na

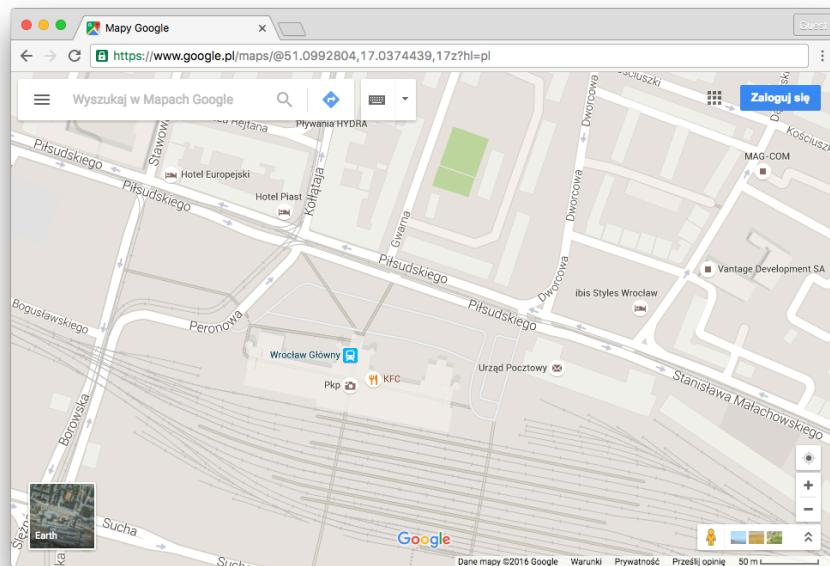
Rys. 1.5 przedstawia mapę Wrocławia oddaloną, która wokół Dworca głównego nie posiada wielu punktów. W miarę zwiększania skali odkrywają się elementy, które przez Google zostały uznane jako mniej atrakcyjne i widoczne jedynie przy większej szczegółowości mapy. Przypadek ten został przedstawiony na Rys. 1.6.

Zabieg ten pozwala na jaśniejsze przedstawienie danych posiadanych przez Google użytkownikowi, dla którego naturalne jest to co się stało, a dla inżynierów łatwiej jest pokazać mniejszą ilość elementów na mapie.

Metoda ta umożliwia użytkownikom biznesowym na pozycjonowanie swoich punktów na mapie np. przez mechanizm bidowania, który pozwala punktom rywalizować. Ten dokument nie będzie poruszał dokładnie tego zagadnienia, ale warto wspomnieć o takiej możliwości.



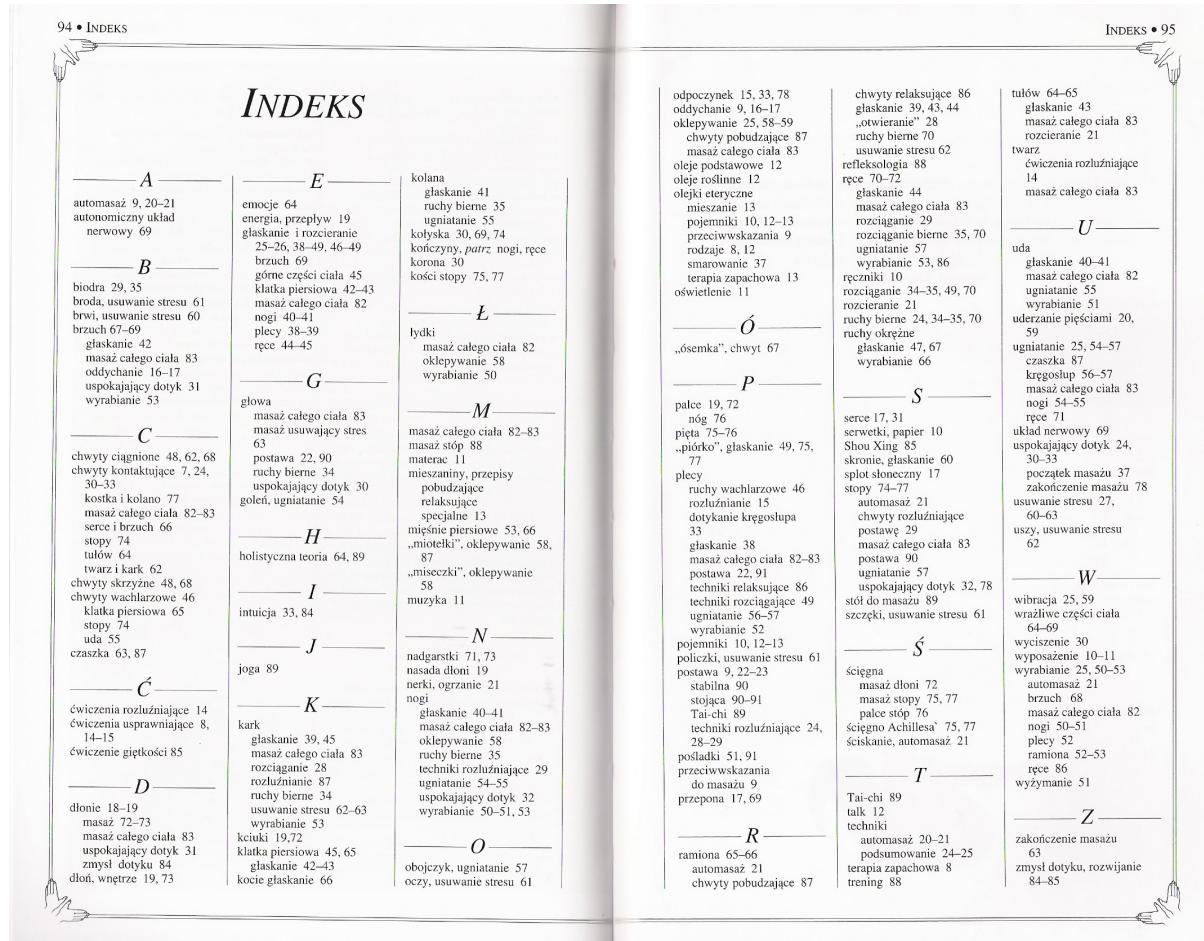
Rys. 1.5 Dworzec główny przy mniejszym powiększeniu



Rys. 1.6 Dworzec główny przy większym powiększeniu

1.4.3 Indeksowanie

Indeks w kontekście materiałów pisanych jest to spis pojęć występujących w danym dokumencie. Głównym zadaniem takiego indeksu jest przyspieszenie wyszukiwania fraz w książce. Rys. 1.7 przedstawia przykład takiego indeksu. Wszystkie pojęcia są posortowane alfabetycznie i obok każdego są podane numery stron, na których one występują. Jak widać jest to bardzo dobrze uporządkowany spis, który podsumowuje ważniejsze frazy w książce.



Rys. 1.7 Indeks książki "Nauka masażu w Weekend" Nitya Lacroix

W jaki sposób użytkownik korzysta z indeksu i jaki jest zysk czasowy?

Najjaśniejszym sposobem będzie przedstawienie w postaci kroków:

1. Szukam frazy *serce* w książce
2. Nawiguję do indeksu i szukam grupy fraz na literę S
3. Patrzę na pierwszą pozycję
4. Dowiaduję się, że interesująca mnie fraza występuje na stronie 17 i 31
5. Nawiguję do tych stron

W ten bardzo optymalny sposób użytkownik dotarł do interesującego go zagadnienia.

Jak by to wyglądało bez stworzonego indeksu?

1. Zaczynam przeszukiwanie od strony nr 1
2. Sprawdzam każde słowo aż do strony 17
3. Znajduje słowo serce
4. Nie mam pewności czy dalej coś istnieje w książce dotyczące tego zagadnienia, dlatego kontynuuję przeszukiwanie
5. Docieram do strony 31
6. Nie mając pewności dalej przeszukuję
7. Docieram do końca książki

Na pierwszy rzut oka widać, że pierwszy algorytm jest mniej kosztowny. Nie zmusza użytkownika do przeszukania wszystkich danych w celu odnalezienia odpowiedzi na pytanie. Co w przypadku, gdy interesuje nas ilość wystąpień danej frazy w książce? Czy istniejący indeks nadaje się do otrzymania tych danych? Na pewno ułatwia on proces wyliczeń, gdyż mamy strony, na których dane słowo występuje, ale wyszukiwanie stałoby się szybsze, gdyby zamiast organizacji alfabetycznej byłaby organizacja po ilości wystąpień.

Warto poruszyć także problemy występujące podczas indeksowania. Są nimi większa złożoność przy operacji wstawiania nowych elementów i rozmiar. Aby zapewnić spójność danych przy każdym nowym zagadnieniu dodanym lub usuniętym z książki trzeba także zaktualizować sam indeks. Jak można zauważyć im więcej posiadamy indeksów tym bardziej kosztowna staje się ta operacja. Także tworzenie takich dodatkowych struktur wiąże się z poświęceniem dodatkowego miejsca w książce, co dodatkowo zwiększa koszty samej książki.

Wszystkie powyższe reguły odnoszą się także do indeksowania wykorzystywanego w informatyce m.in. w bazach danych. Indeksy jednak nie są tak prosto skonstruowane jak wykorzystana powyższa analogia indeksu książki. Wykorzystują one bardziej skomplikowane algorytmy, aby odpowiadać na wyspecjalizowane zapytania w jak najszybszym czasie.

1.4.4 GeoHash

Jest to algorytm indeksowania wymyślony przez Gustavo Niemeyer'a udostępniony publicznie 26 lutego 2008 roku. Głównym jego zadaniem jest podzielenie mapy na kubelki w kształcie siatki.

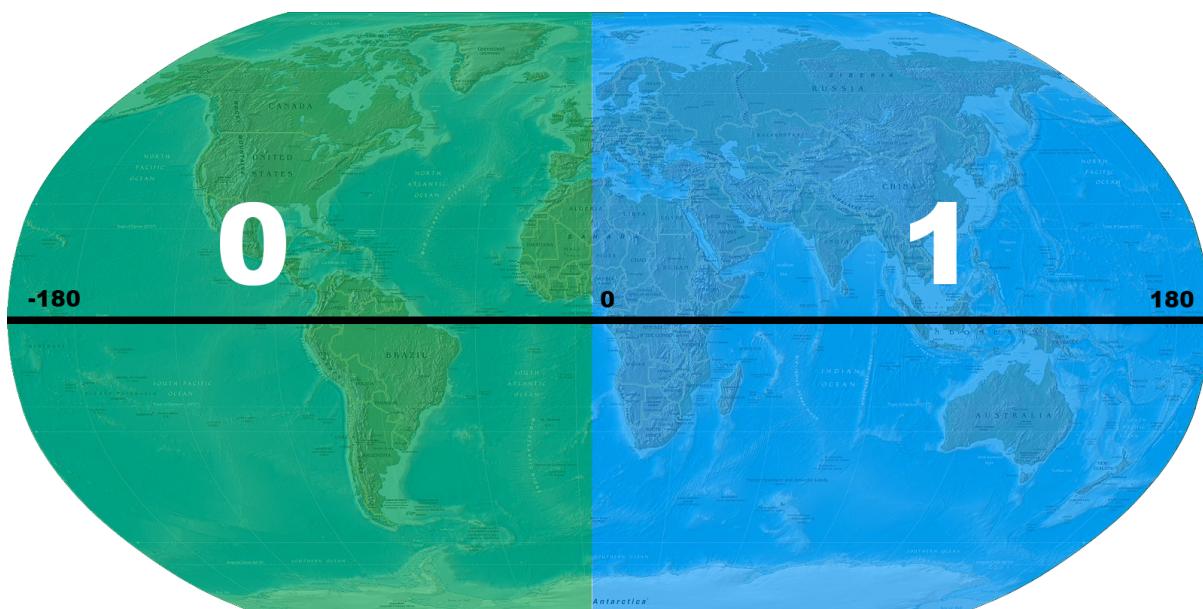
Główne przypadku użycia algorytmu to:

- Nadanie znacznikom unikalnych identyfikatorów
- Prezentacja danych znacznikowanych w bazach danych

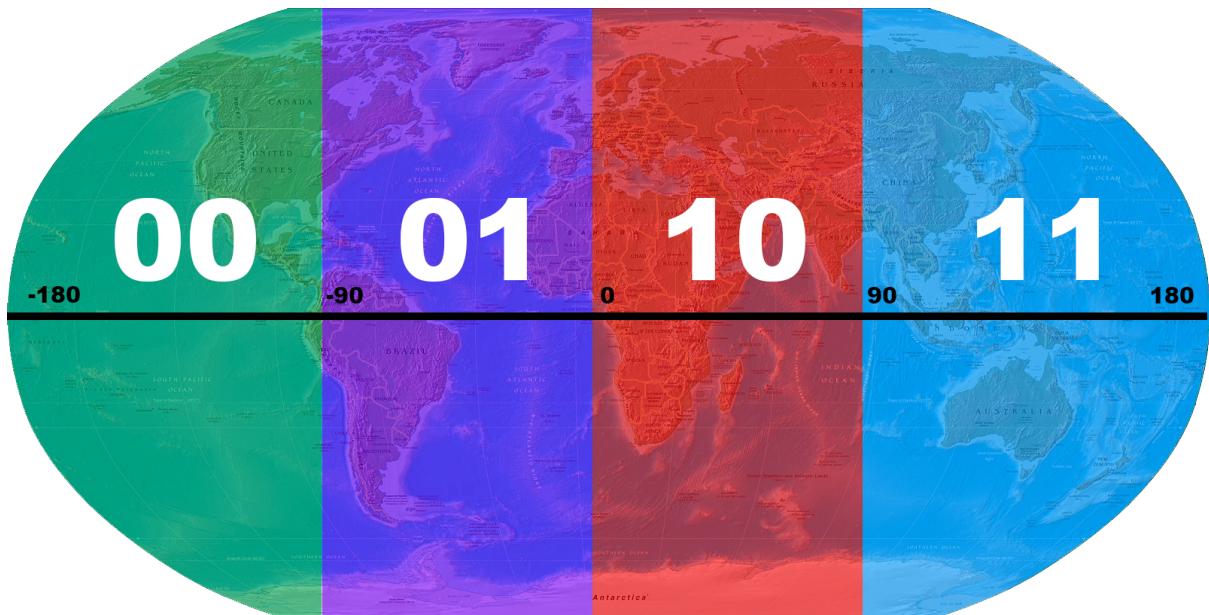
Algorytm jest bardzo prosty w idei przez co czyni go także prostym w implementacji. Aby zakodować współrzedną geograficzną najpierw trzeba określić dokładność geohash'u. Im większa dokładność tym dane reprezentowane przez hash będą przyporządkowane do mniejszych obszarów. Dla prezentacji kolejnych kroków algorytmu zostanie przeprowadzone kodowanie lokalizacji Wrocławia ($51^{\circ} 06' 36'' N$ $17^{\circ} 01' 20'' E$) przy ustalonej dokładności 4.

W pierwszym kroku mapa każda składowa lokalizacyjna (długość i szerokość geograficzna) są dzielone na n równych części. Dla przykładu Wrocławia są to 4 równe części. Każdy z tak podzielonych obszarów uzyskuje bitową reprezentację. Dla precyzji 1 będzie to jeden bit. Rys. 1.8 i Rys. 1.9 prezentują w jaki sposób algorytm przydziela bity dla poszczególnych wartości długości geograficznej. Po podziale przedziału $[-180, 180]$, gdzie przedział reprezentuje wszystkie możliwe wartości długości geograficznej otrzymujemy 2 przedziały $[-180, 0]$ i $(0, 180]$. Zatem pierwszy z nich otrzymuje 0, a kolejny 1. Przy większej dokładności zaprezentowanej na Rys. 1.9 pierwszy bit (bity są liczone od lewej) nie ulega zmianie, a dodawany jest kolejny zbudowany na tej samej zasadzie co poprzedni. Na przykładzie pierwszej połówki dzielimy przedział $[-180, 0]$ na 2 równe części $[-180, 90]$ i $(90, 0]$ i odpowiednio przypisujemy im wartości 0 i 1. Szerokość geograficzna jest kodowana w analogiczny sposób.

Dla przykładu Wrocławia i dokładności 4 bitowa reprezentacja długości to 1000, a szerokość to 1100.



Rys. 1.8 Bitowa reprezentacja długości geograficznej przy precyzji 1



Rys. 1.9 Bitowa reprezentacja długości geograficznej przy precyzyj 2

Następnym krokiem jest połączenie szerokości i długości geograficznej w jeden ciąg znaków reprezentujący bitowo ostateczny geohash. Realizuje się to przeplotem bitów kolejno długości geograficznej i szerokości. Dla Wrocławia będzie to wartość 11010 000.

Ostatnim krokiem jest zakodowanie to szyfrowaniem Base32. Polega ono na grupowaniu bitów po 5 (aby móc zapisać 32 możliwe wartości) i przypisaniu odpowiedniego znaku tym bitom. Jako, że w przykładzie mamy 8 bitów to 3 najmłodsze bity będą odrzucone. Aby zapisać 5 starszych zamieniamy je na system dziesiętny co daje wartość 26, a następnie odczytujemy wartość z Tabela 1.1. Tak otrzymana wartość jest ostatecznym GeoHashem.

Dziesiętny	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
32	0	1	2	3	4	5	6	7	8	9	b	c	d	E	f	G
Dziesiętny	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	h	j	k	m	n	p	q	r	s	t	u	v	w	X	y	Z

Tabela 1.1 Definicja znaków w transformacji na system o podstawie 32

GeoHash ma jedną bardzo użyteczną właściwość, która pozwala być podstawą do dobrego indeksowania danych geolokalizacyjnych. Każdy uzyskany GeoHash z precyją mniejszą jest początkiem GeoHashu o większej precyzyji. U (pozycja Wrocławia o precyzyji 4) jest początkiem najbardziej precyzyjnego przebiegu algorytmu (wartość u3h4g271qd89).

Właściwość ta pozwala na łatwe grupowanie znaczników geolokalizacyjnych w kubelki o różnej dokładności i w szybki sposób określić ich liczbę na podstawie prostego indeksu

tekstowego. Wraz z przybliżaniem obszaru mapy zwiększymy dokładność geohash'u co skutkuje większym filtrowaniem elementów na mapie.

1.4.5 Indeksowanie przy pomocy geohash'ów w aplikacji

Algorytm geohash'u jest wykorzystany w aplikacji do grupowania umieszczonych tam plików do kubeliów, które mają swoje odzwierciedlenie na mapie. Każdy nowo dodany plik uzyskuje geohash maksymalnej precyzji i jest zapisywany w bazie danych. Pole, które przechowuje hash jest indeksowane tekstowo, aby przyspieszyć zapytania o pliki, których hash zaczyna się od pewnego ciągu znaków, który jest jednym z najczęściej wykonywanych zapytań na bazie danych.

Wraz jak użytkownik zmienia obszar roboczy mapy poprzez powiększanie lub przesuwanie mapy generowane są geohash'e precyzji o jeden większej niż obszar wyznaczony przez mapę. Następnie odpisywane są elementy należące do tych geohash'ów przez zapytanie opisane powyżej. Pozwala to na wizualne zgrupowanie elementów. Rys. 1.10 przedstawia przykład zapytania MongoDB wykorzystywanego w aplikacji do grupowania elementów na mapie. Pierwszy argument filtruje elementy po ich geohashach przez co otrzymamy elementy widoczne na mapie, z kolei drugi argument grupuje tak otrzymane elementy w kubelki i oblicza ich liczbę.

```
1. db.media.aggregate({  
2.   _geoHash: {  
3.     '$in': [  
4.       /^6/,  
5.       /^7/,  
6.       /^k/,  
7.       /^m/,  
8.       /^q/,  
9.       /^d/,  
10.      /^e/,  
11.      /^s/,  
12.      /^t/,  
13.      /^w/,  
14.      /^f/,  
15.      /^g/,  
16.      /^u/,  
17.      /^v/,  
18.      /^y/  
19.    ]  
20.  }  
21. }, {  
22.   _id: {  
23.     $substr: ['$geoHash', 0, 1]  
24.   },  
25.   loc: { $last: '$loc' },  
26.   count: { $sum: 1 }  
27. })
```

Rys. 1.10 Przykładowe zapytanie grupujące elementy na mapie

2 Google Maps

Serwis internetowy umożliwiający wyszukiwanie obiektów, a także oglądanie map oraz cały pakiet pod aplikacji związanych bezpośrednio z mapami. Został stworzony w 2005 roku przez Google. Jest darmowy dla zastosowań niekomercyjnych.

Tematem pracy jest wykorzystanie Google Maps do prezentacji danych geoznacznikowanych więc zostanie dokładnie przeprowadzona analiza tego rozwiązania w kontekście znacznikowania.

Produkt Google jest całym pakietem aplikacji, dlatego poniżej zostanie przedstawione wszystko co wchodzi w jego skład. Opracowane na podstawie danych z [4]

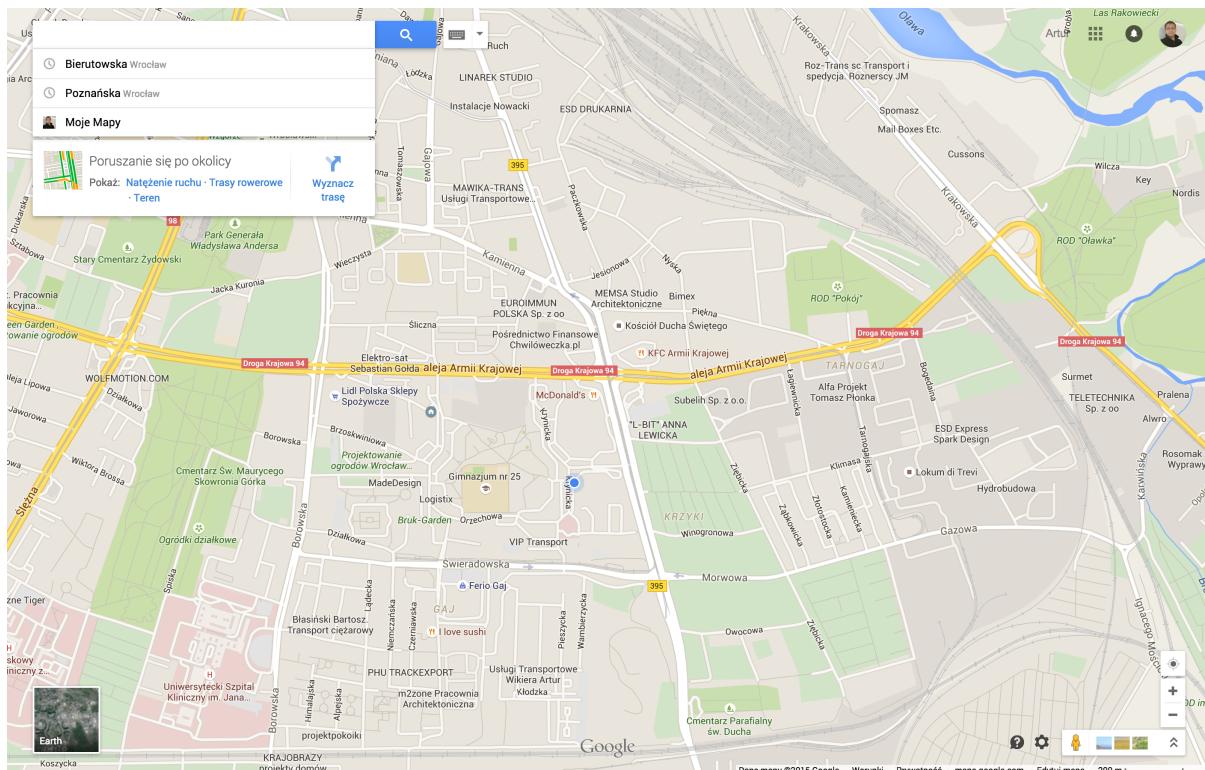
2.1 Serwis Google Maps

Główna aplikacja pokazująca pełne możliwości technologii. Wchodzi w skład aplikacji oferowanych przez Google. Korzysta ze wszystkich danych agregowanych przez wyszukiwarkę tj. wyszukiwane frazy, punkty zainteresowań, usługi lokalizacji i kalendarze. Pozwala to na personalizację aplikacji pod potrzeby użytkownika. Google przez ten zabieg bardziej uzależnia klientów od swoich rozwiązań przez integracje wszystkich oferowanych usług. Mapy dodatkowo są częścią wyszukiwarki. Wyszukując cokolwiek na <http://google.com> przeszukuje się także mapy.

Serwis pozwala na wyznaczanie tras na podstawie wielu źródeł danych. Podstawową funkcjonalnością jest po prostu wyznaczenie trasy przy pomocy zarejestrowanych ścieżek. Jednak Google jest w stanie zrobić to dla komunikacji samochodem, pieszo czy przy pomocy transportu publicznego. W Polsce dobrze są reprezentowane połączenia kolejowe, a komunikacja autobusowa nadal jest mniej dopracowana. Dla osób przyjezdnych jest to jednak wystarczająca informacja. Przy wyznaczaniu tras Google uwzględnia natężenie ruchu. Dane są dostarczane przez użytkowników serwisu przez usługi lokalizacji. Usługi te są także podstawą do wyznaczania miejsca wykonania dokumentu w aplikacji stworzonej w ramach tej pracy.

Google oferuje także Street View w ramach serwisu. Jest to forma wirtualnej wycieczki po drogach. Przy jej pomocy jest się w stanie spojrzeć na okolicę miejsca. Zdjęcia robione są przez samochody Google, które codziennie przemierzają drogi, aby pozyskiwać zdjęcia. Wyposażone są one w aparat pozwalający robić zdjęcia 360°.

Aplikacja jest ciągle doskonalona przez dodanie coraz bardziej dokładnych danych o punktach POI, natężenia ruchu, obiektów 3D i ilości personalizacji.



Rys. 2.1 Okno główne Google Maps

2.2 Mapy Google API

Google stworzyło API, umożliwiające wstawienie własnej mapy na dowolną stronę internetową. API zostało przeniesione na wiele języków, w którym zawiera się JavaScript lub można uzyskać dostęp w postaci obrazu (Google Static Maps API). Aby uzyskać dostęp trzeba wygenerować klucz, którym Google będzie autoryzowało aplikację.

Korzystanie z API jest bezpłatne do 2500 żądań na dzień dla użytkownika domowego i komercyjnego. Przekroczenie tego limitu wiąże się z opłatami, które zmieniają się w czasie.

Najnowsza wersja JavaScript API 3.26 udostępnia całą funkcjonalność oryginalnego serwisu, wliczając w to²:

- możliwość geokodowania adresów
- rysowanie polilinii
- wyznaczanie tras przejazdu wraz z listą kroków
- pełna kontrola widoku ulic (Street View)
- wsparcie dla języka KML/GeoRSS

W ramach API istnieje jeszcze kilka innych technologii:

1. Google Maps GeoCoding API

Usługa służy do zamiany adresów na współrzędne geograficzne

² Źródło: https://pl.wikipedia.org/wiki/Mapy_Google

2. Google Places API Web Service

Usługa nastawiona na podpowiadanie miejsc zainteresowań nazywanych POI (z ang. Points of Interest). Oferuje dostęp do wszystkich lokali dostępnych przez Google

3. Google Maps Elevation API

Usługa podaje wysokość n.p.m. podanych koordynatów.

4. Google Maps Roads API

Usługa oferująca dane dla dróg. Między innymi pozwala na wyznaczanie trasy przez punkty pośrednie podane w żądaniu, a także szczegóły tak wyznaczonej trasy tj. ograniczenia prędkości czy utrudnienia w ruchu.

5. Google Maps Geolocation API

Google posiada informacje o stacjach bazowych sieci komórkowych, a także o hotspotach WiFi. Posiadając takie dane jest w stanie wygenerować pozycję użytkownika szybciej i dokładniej niż GPS w miejscowościach, w których dostęp do satelit GPS jest ograniczony np. centra miast.

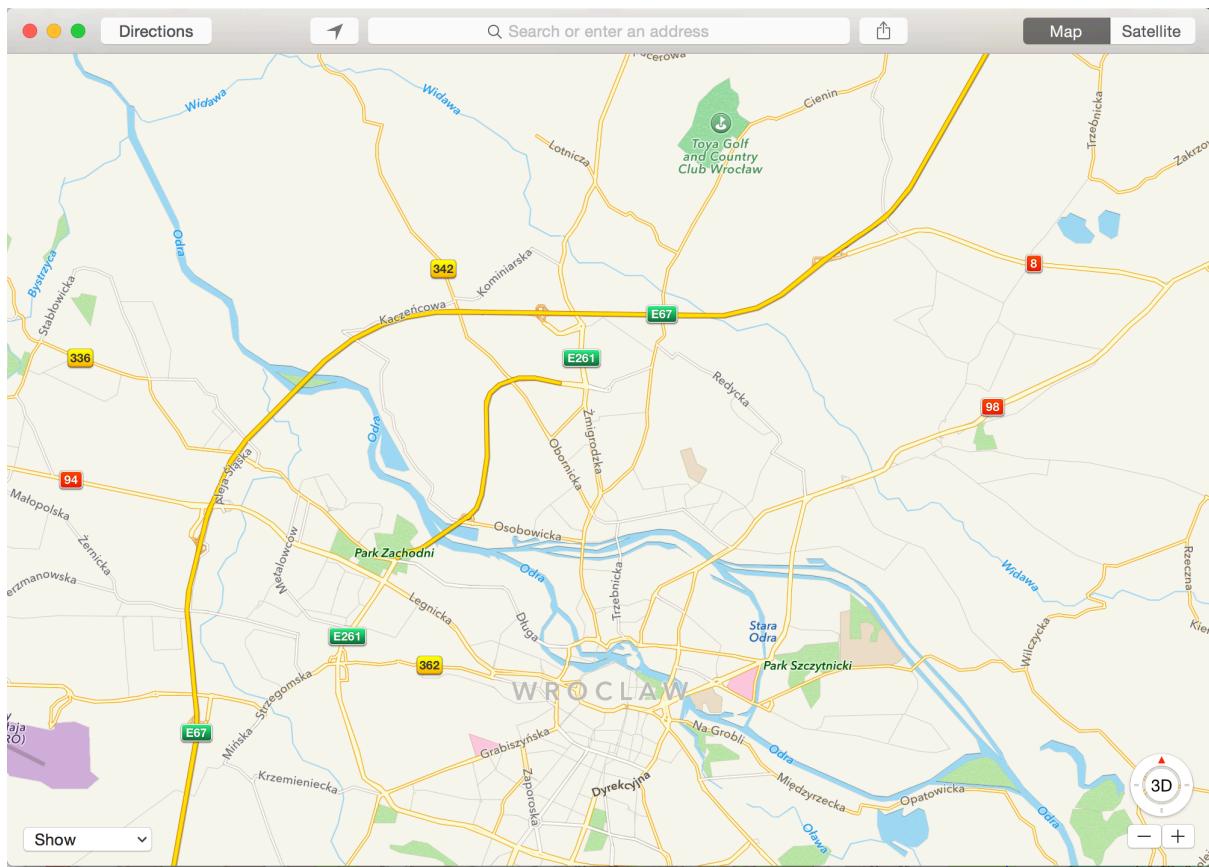
Poza wszystkimi interfejsami programistycznymi Google Maps oferuje interfejs graficzny, z którego korzystają miliony użytkowników. Wykorzystanie tej technologii pozwoli na stworzenie lepszej aplikacji zorientowanej na użytkownictwo.

2.3 Konkurencyjne rozwiązania

Poniżej pragnę przedstawić inne konkurencyjne rozwiązania dostępne na rynku. Google jest niemalże monopolistą na tym rynku, jednakże są inne systemy, które mają rozwiązania je charakteryzujące.

2.3.1 Apple Maps

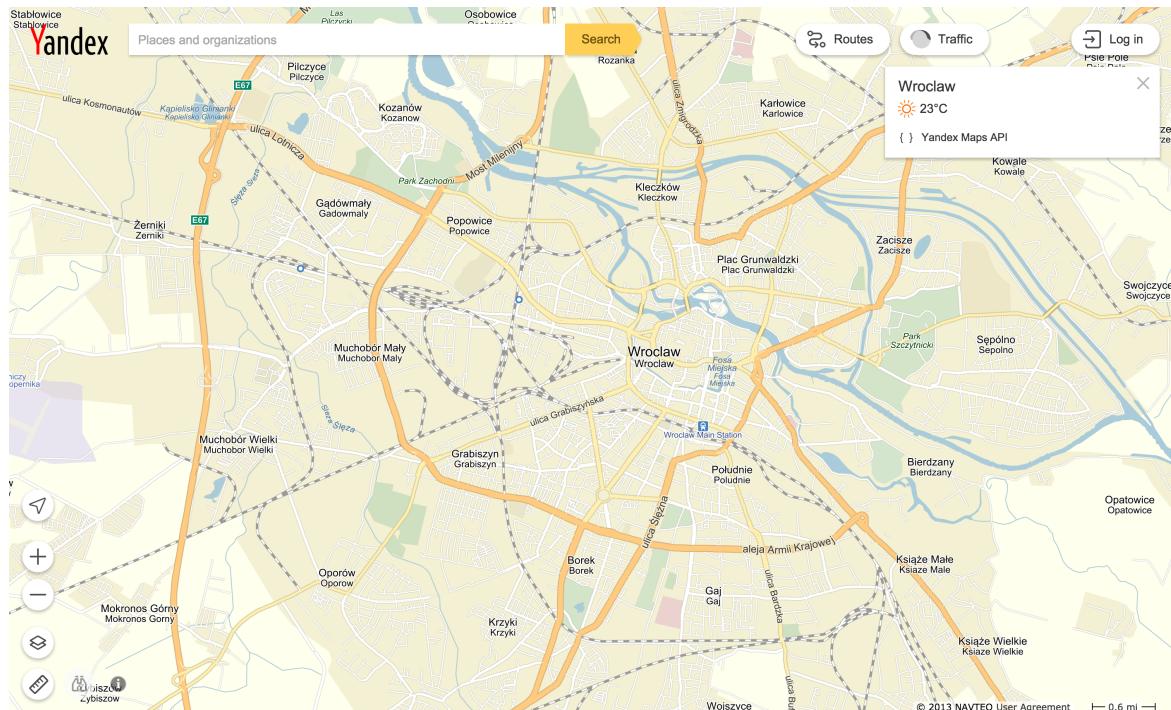
Usługa map stworzona przez Apple dostępna jedynie na OS X i iOS. Jako pierwsze posiadało innowacyjny system generowania ukształtowania terenu na podstawie danych satelitarnych. Początkowo miało wiele błędów, lecz z czasem zostały on poprawione. Aplikacja w przeciwieństwie do Google Maps ma niewiele punktów POI, a wyznaczane trasy nie uwzględniają wielu czynników.



Rys. 2.2 Okno główne aplikacji Apple Maps

2.3.2 Yandex Maps

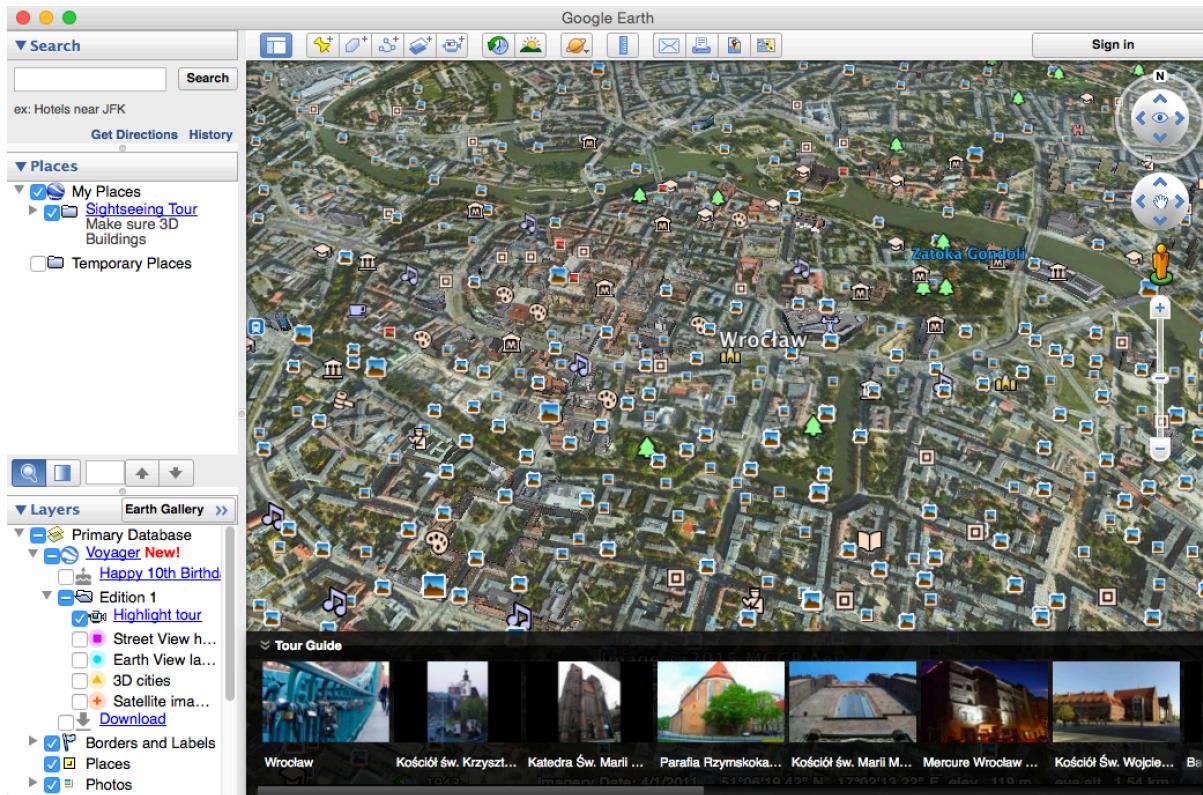
Mapy rosyjskiego lidera wyszukiwarkowego. Przede wszystkim wykorzystywane w Rosji i Chinach. W tym drugim kraju Google jest niedostępne, dlatego Yandex dominuje w krajach środkowych.



Rys. 2.3 Okno główne aplikacji Yandex Maps

2.3.3 Google Earth

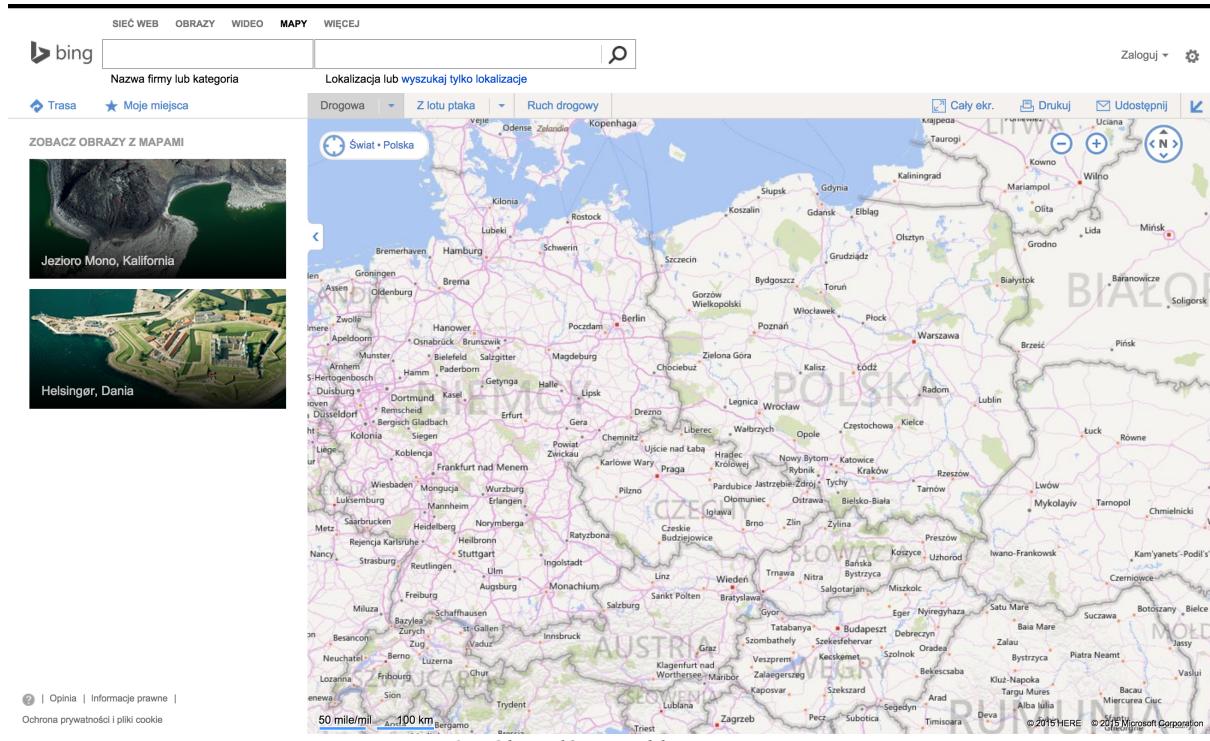
Kolejna aplikacja Google, która została zorientowana na publikowanie zdjęć przez użytkowników. Dzisiaj już jest przestarzała, ponieważ większość funkcjonalności została przeniesiona do Google Maps.



Rys. 2.4 Okno główne aplikacji Google Earth

2.3.4 Bing Maps

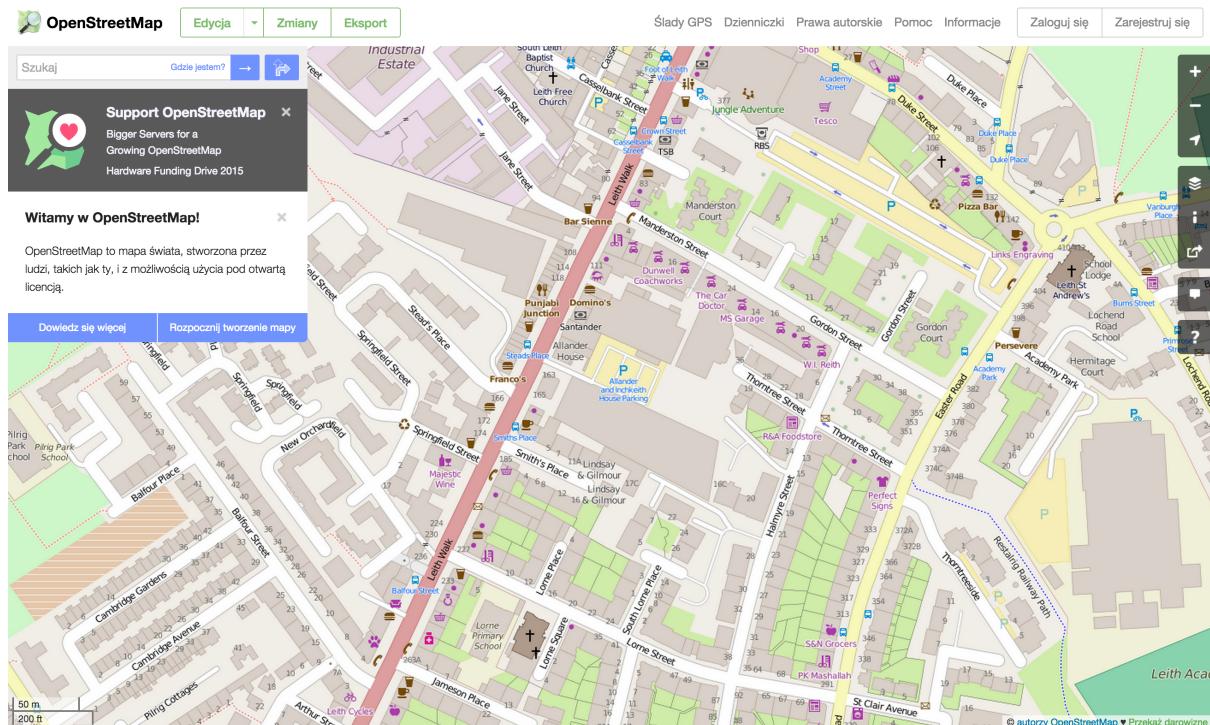
Mapy firmy Microsoft. Oferują wyznaczanie ścieżek i podstawowe informacje o punktach dostępu. Wykorzystywane na portalu Facebook do prezentacji danych, lecz poza nim niezbyt popularny. Program dostępny jest w formie strony i aplikacji dla Windows Phone.



Rys. 2.5 Okno główne aplikacji Bing Maps

2.3.5 Open Street Maps

W pełni darmowe rozwiązanie pozwalające na większość podstawowych funkcjonalności map tj. wyznaczanie trasy, wyświetlanie map, zdjęć satelitarnych. Pozwala na tworzenie aplikacji przy pomocy JavaScript API.



Rys. 2.6 Okno główne aplikacji Open Street Maps

2.4 Aplikacje wykonane przy pomocy Google Maps

2.4.1 Pokemon Go

Gra stworzona na platformy mobilne, która osiągnęła globalny sukces. Zbudowana na bazie map Google i wirtualnej rzeczywistości. Polega ona na zbieraniu stworów. Jednak aby przejść do lokalizacji trzeba wyruszyć na spacer. Stwory czekają na graczy na stacjach benzynowych, kościołach i ulicach. Wszystko przedstawione w formie wielkiej mapy, na której poza samymi stworami są także areną, na których odbywają się pojedynek i PokeStop.

Bez Google Maps i nawigacji satelitarnej gra nie odniosłaby tak wielkiego sukcesu. Dodatkowo zaktywizowała wielkie grono graczy co spowodowało do redukcji otyłości.

2.4.2 Uber

Aplikacja zrzeszająca kierowców i komunikująca ich z potencjalnymi klientami. Uber zaczynał jako startup, a teraz jest spółką posiadającą swoje samochody i kierowców na całym globie. Uber w przeciwieństwie do klasycznej taksówki wylicza szacowany koszt podróży z punktu A do B. Pozwala to klientowi na decyzję czy chce skorzystać z usług czy nie. Wyznacza on trasę przy pomocy Google Maps i także przedstawia ją na telefonach przy pomocy tego samego narzędzia.

2.4.3 JakDojade.pl

Polska aplikacja pozwalająca na wyszukiwanie połączeń komunikacją miejską. Agreguje ona rozkłady jazdy z wielu miast Polski, a liczba wspieranych miast ciągle rośnie. Swoje dane przedstawia na mapach Google na systemie Android.

3 Aplikacja

W ramach tej pracy została wykonana aplikacja, na której zostaną podane badania użyteczności. Pozwoli to sprawdzić użyteczność Google Maps API w celu przetwarzania danych geoznacznikowych.

3.1 Podstawa teoretyczna

W celu przybliżenia czytelnikowi metody wykonania aplikacji poniżej przedstawione jest kilka pojęć wymaganych do zrozumienia zaprojektowanej aplikacji.

3.1.1 Single Page Application (SPA)

Jest to koncepcja, która mówi, że strona powinna być załadowana raz, a każde przejście na kolejną stronę ma być wykonywane asynchronicznie. Wszystkie zmiany strony są pokazywane przez modyfikację drzewa DOM w dokumencie HTML. Uzasadnieniem tego typu stron są rosnące doświadczenia użytkownika przy korzystaniu z aplikacji (User Experience), a także zminimalizowanie transferu danych między przeglądarką a serwerem, przez co czas odpowiedzi serwera jest zdecydowanie szybszy, a użytkownik szybciej zobaczy efekt. Koncepcja czerpie wiele z najnowszych technologii jakimi są HTML5, CSS3, JavaScript, AJAX.

3.1.2 MVC

Złożony wzorzec architektoniczny służący do organizowania struktury aplikacji posiadających interfejsy graficzne. Wykorzystuje on 3 proste wzorce jakimi są Strategia, Obserwator, Kompozyt.

MVC zakłada podzielenie aplikacji na poniżej wymienione części składowe:

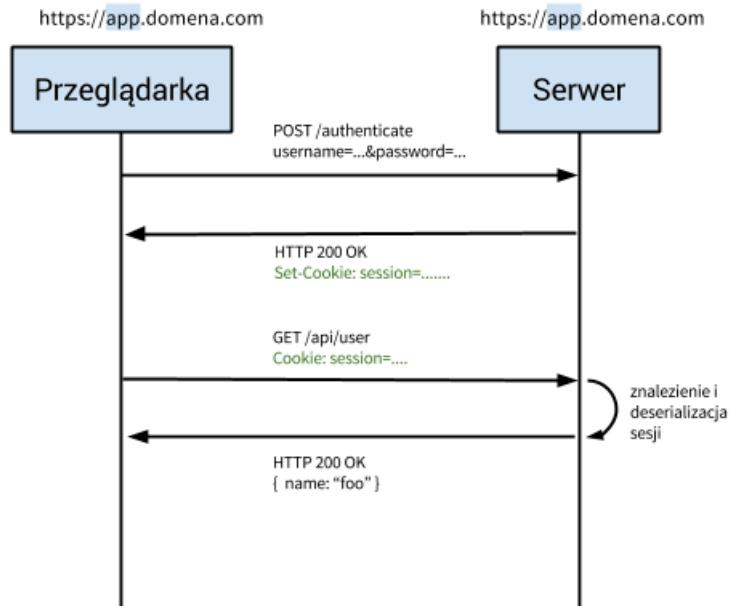
- Model - dane wymagane do utworzenia Widoku
- Widok - opisuje, jak przedstawić część modelu użytkownikowi
- Kontroler - agreguje dane z wielu modeli i przygotowuje je, aby przekazać do widoku

3.1.3 Logowanie tokenowe

Logowanie to działa na podobnej zasadzie jak logowanie przy pomocy ciasteczek, lecz znosi pewne ograniczenia jakie wcześniej wspomniany i najpopularniejszy w dzisiejszych czasach system identyfikacji sesji posiadał. Użytkownik, który chce się zalogować wpisuje swój login i hasło, na których podstawie jest generowany token (identyfikator sesji), który z kolei podajemy przy każdym wymagającym autentykacji żądaniu do serwera (poprzez dodanie w nagłówku żądania HTTP pola *Authentication: Bearer <token>*). Token jest wydawany jedynie na określony czas i służy do autentykacji tylko dla jednego użytkownika. Jedną z zalet tego logowania jest to, że możemy w aplikacjach SPA zalogować się bez odświeżenia strony i dodatkowo aplikacje klienckie możemy „hostować” na innej domenie. Nowoczesne strony wykorzystują już tego typu logowanie, idealnym przykładem jest protokół OAuth 2, który wykorzystują liczne portale:

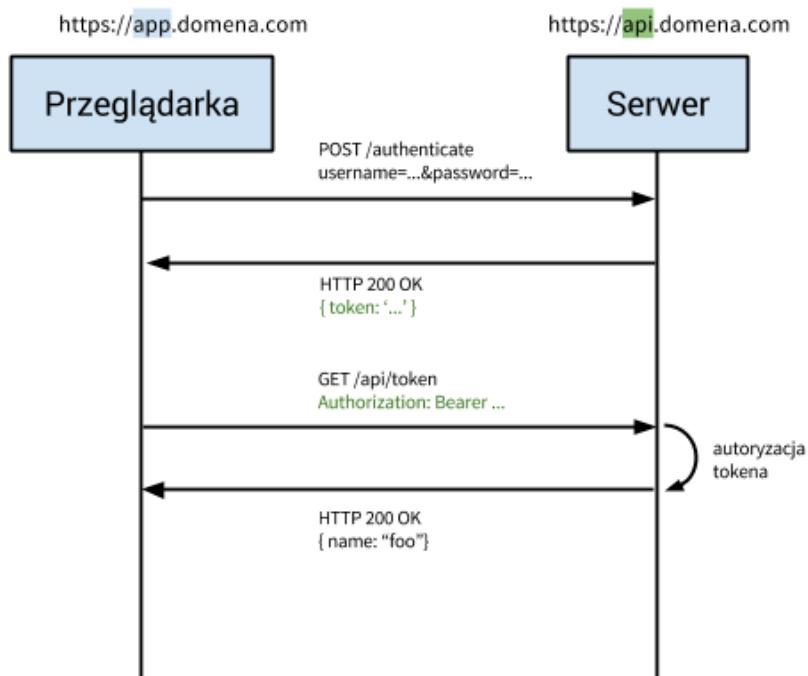
- Facebook
- Google+
- Twitter

Tradycyjna autoryzacja oparta na ciasteczkach



Rys. 3.1 Prezentacja przepływu danych przy pomocy ciasteczek

Nowoczesna autoryzacja oparta na tokenach



Rys. 3.2 Prezentacja przepływu danych z wykorzystaniem tokenu

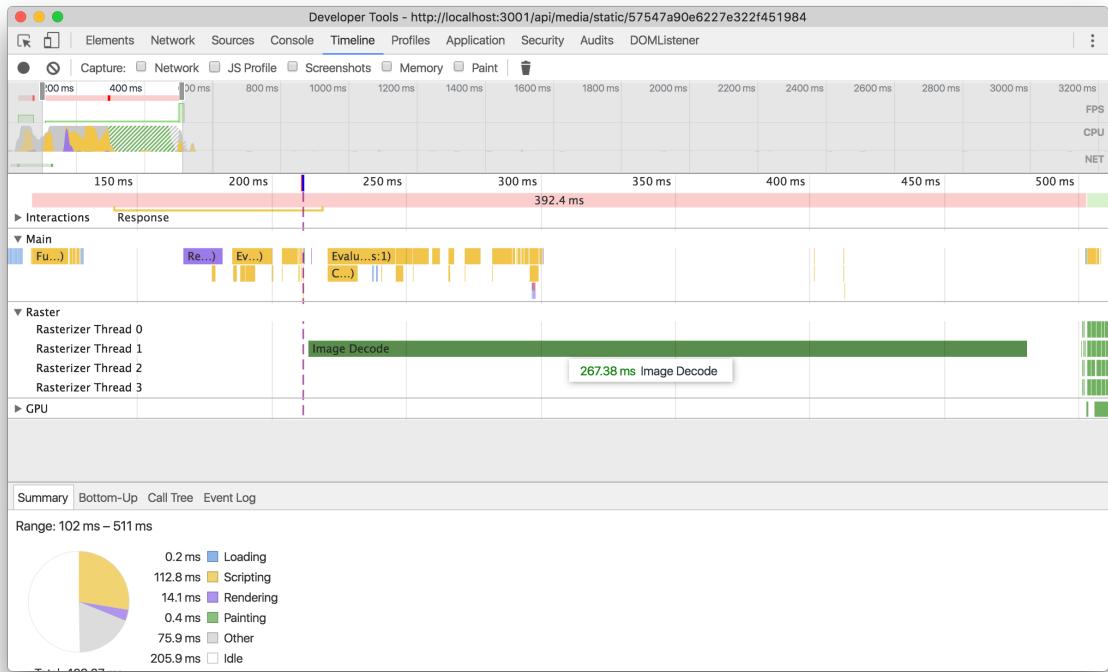
3.1.4 ImageMagick

Pakiet do obróbki grafiki rozwijany na licencji Apache License 2.0. Projekt ten jest w pełni Open Source. Udostępnia liczne funkcje obróbki obrazów jak i pozyskiwania z nich danych. Natywnie tworzone jako narzędzie linii komend, lecz posiada swoje implementacje w językach m.in. C, C++, Node.JS, Python.

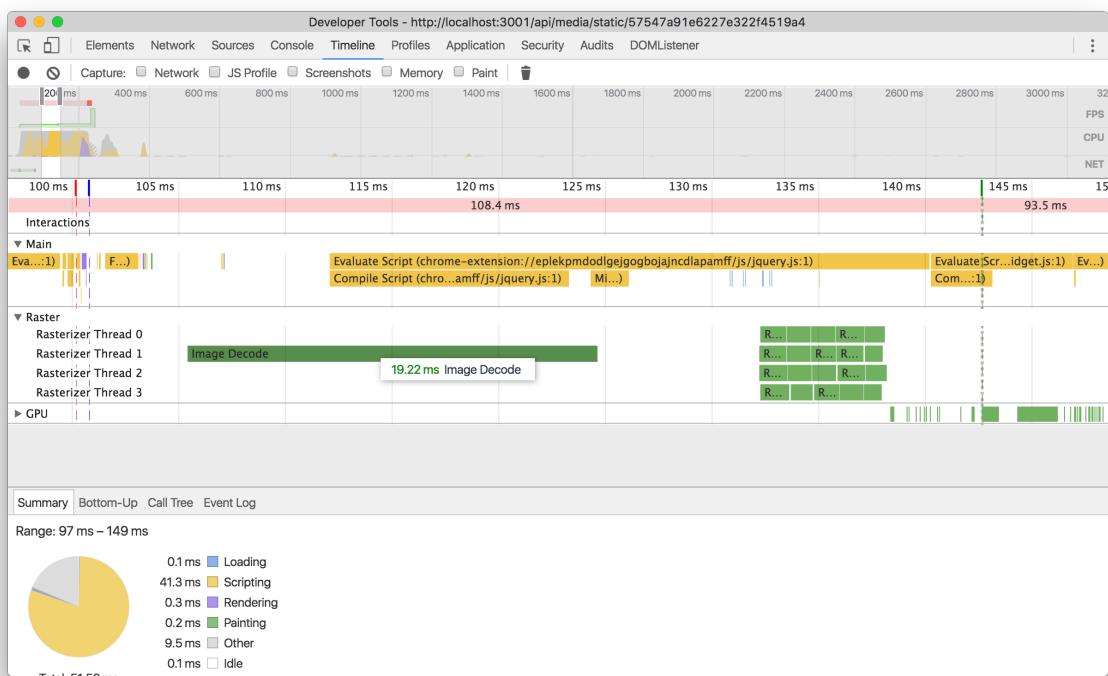
Lista funkcjonalności jest dłuża, a zawiera między innymi:

- Pobieranie właściwości zdjęcia (wymiary, rozmiar, EXIF)
- Zmiana formatu pliku
- Efekty artystyczne
- Transformacje (zmiana rozmiaru, palety barw)
- Filtry

Aplikacja wykorzystuje ten pakiet do tworzenia miniaturek dla zdjęć, aby były formatu przyjaznego stronom internetowym. Pozwala to zmniejszyć ilość transferu potrzebnego na urządzeniach mobilnych do korzystania z aplikacji, a także są szybciej interpretowane przez przeglądarkę. Rys. 3.3 i Rys. 3.4 pokazują różnicę w wydajności interpretowania obrazu o rozdzielczości połowę mniejszej niż plik oryginalny. Różnica w wydajności jest znaczna i jest rzędu 1250%. Uwzględniając, że na listingu do wyświetlenia jest przykładowo 50 obrazów, nagle niewielka różnica staje się wielkim problemem wydajnościowym.



Rys. 3.3 Zrzut z Chrome DevTools pokazujący czas potrzebny na dekodowanie obrazka JPEG w rozdzielcości 4956x3304



Rys. 3.4 Zrzut z Chrome DevTools pokazujący czas potrzebny na dekodowanie obrazka JPEG w rozdzielcości 1920x1080

3.2 Technologie

Do stworzenia aplikacji zostały wykorzystane następujące technologie:

- Google Maps API

- Node.JS
- React
- Redux
- MongoDB
- Express
- ImageMagick

Powyższe technologie zostały dobrane, tak aby przechowywanie danych geolokalizacyjnych było optymalne.

Najważniejszym technologią jest Google Maps. Zostało wykorzystane udostępnione bezpłatne API do zbudowania aplikacji, która jest jedną wielką mapą. Przy pomocy tej technologii możliwa jest efektywna prezentacja dodanych zasobów. Google Maps poza samą mapą oferuje interfejs poparty badaniami użytkownictwa co pozwoli stworzyć lepszy interfejs pozwalający na lepszą prezentację licznych danych geolokalizacyjnych.

Baza MongoDB jest bazą danych zorientowaną na dokumenty. W przeciwieństwie do standardowych baz SQL, MongoDB nie posiada tabel, również relacji. Każda kolekcja przechowuje jedynie dokumenty.

Jednym z kluczowych elementów wyboru MongoDB była możliwość indeksowania danych geolokalizacyjnych, co ułatwia przechowywanie danych na mapach. Silnik zapytań obsługuje zaawansowane metody rozproszenia danych na wielu serwerach. Pozwala to na pełne wykorzystanie środowisk chmurowych, przez utrzymywanie danych na wielu małych maszynach. Posiadając taką farmę baz danych można w prosty sposób zwiększyć wydajność całego systemu.

Język zapytań użyty w MongoDB obsługuje standardowe selekcje, projekcje czy grupowanie. Poza podstawową funkcjonalnością istnieje mechanizm map-reduce, który bardzo dobrze się sprawdza przy tworzeniu zaawansowanych zapytań. Wykorzystując tą technikę zwiększamy możliwości skalowania tak przygotowanego systemu.

Baza danych wspiera także technologię nazwaną GridFS. Sprawdza się bardzo dobrze do przechowywania wielkich plików: obrazków, filmów, prezentacji. GridFS zachowuje się jak każda kolekcja. Jedyną różnicą jest w jaki sposób jest zapisywany sam plik. Jest on dzielony na części, a następnie każda z nich jest przechowywana jako pojedynczy dokument. Jak zostało to wspomniane dokumenty mogą być rozproszone na wielu maszynach.

ImageMagick jest biblioteką Open Source wykonującą operacje na plikach graficznych. Pozwala ona na wykonywanie podstawowych operacji na plikach: powiększanie, kompresowanie, zmiana formatów. Dodatkowo najważniejszym czynnikiem wyboru jest obsługa formatu EXIF. Niestety nie można zapisywać danych EXIF do obrazu, ale umożliwiony jest odczyt wszystkich danych takich jak rozmiar pliku, a także najważniejszy parametr: lokalizacja. Jeżeli jest zapisana w pliku to traktowana jest priorytetowo, ponieważ jest dokładniejsza niż lokalizacja z Google.

Część serwerowa została całkowicie napisana w języku JavaScript. Utrzymanie tego samego języka po stronie serwera i klienta pozwala tworzyć aplikacje, które są łatwe w utrzymaniu. Jest to ważny biznesowo argument, ponieważ Ci sami programiści mogą rozwijać obie części aplikacji.

NodeJS jest to rozwiązanie serwerowe wykorzystujące silnik JavaScript V8. V8 stworzone przez Google wykorzystuje także Google Chrome. Na ten moment NodeJS jest najpopularniejszym rozwiązaniem serwerowym wspierającym JavaScript.

Cześć kliencka jest aplikacją SPA. Pozwoliło to na zredukowanie liczby przeładowań, które w dużym stopniu wpływają na doświadczenie użytkownika.

Widoki w aplikacji są tworzone przy pomocy biblioteki React. Pozwala ona podzielić aplikację na części zwane komponentami. Przykładowym komponentem może być formularz logowania. Ten także składa się z pod komponentów takich jak pole użytkownika i pole hasła. Posiadając tak przygotowaną bibliotekę komponentów jest się w stanie używać tych samych elementów w różnych częściach aplikacji. Na potrzeby aplikacji zostały zainstalowane biblioteki zbudowane na bazie React do tworzenia elementów mapy. Takim sposobem znając bibliotekę można tworzyć każdy element interfejsu niezależnie od typu: formularz, mapa, diagram, wykres.

Redux odpowiada za kontrolowanie przepływu danych w aplikacji. Uznaje się jako dopełnienie React. Redux promuje swoim podejściem programowanie funkcyjne. W przeciwieństwie do programowania obiektowego stan aplikacji jest przechowywany w jednym miejscu. Pozwala to na łatwe odpluskwianie kodu, ponieważ jest jedynie jedno miejsce, gdzie występuje potencjalny błąd. Każda operacja na magazynie Redux powinna być niezmienna co odkrywa ogromny zasób narzędzi developerskich. Wszystkie te argumenty czynią Redux idealnym rozwiązaniem dla aplikacji klasy Enterprise.

3.3 Wymagania funkcjonalne

1. Import plików z historią lokalizacji z Google z ścieżkami użytkownika
 - a. Zapis danych w bazie danych
2. Autoryzacja
 - a. Użytkownik
 - i. może zarządzać swoimi ścieżkami
 - ii. może zarządzać swoimi multimediami
 - b. Użytkownicy nie mogą podejrzeć nie swoich danych
3. Multimedia
 - a. Można wprowadzić każdy typ pliku
 - b. Jedynie pliki obrazków będą miały rozpoznawane sygnatury takie jak data utworzenia, lokalizacja GPS wykonywanego zdjęcia lub urządzenie na jakim zostało wykonane zdjęcie
 - c. Każde zdjęcie z koordynatami GPS uzupełnia ścieżkę użytkownika
 - d. Pozostałe typy plików będą wymagały podania przynajmniej daty do zlokalizowania w czasie
4. Wyszukiwanie
 - a. Będzie możliwe wyszukiwanie danych multimedialnych po geolokalizacji
 - b. Będzie dostępne wyszukiwanie po dacie
5. Przeglądanie
 - a. Dane multimedialne będą wyświetlane w formie punktów na mapie
 - b. Pliki obsługiwane przez przeglądarkę będą od razu wyświetlane jako podgląd
6. Opcjonalne
 - a. Import danych z serwisów społecznościowych

- b. Import danych z internetowych dysków takich jak Dropbox, Google Drive, Microsoft OneDrive
- c. Czytanie metadanych z plików z pakietu Office
- d. Dodanie nowej roli użytkownika: Administrator
- e. Umożliwienie importu danych w KML, GeoJSON, GPX

3.4 Wymagania niefunkcjonalne

- 1. Aplikacja ma działać w środowisku Cloud
- 2. Ma zapewniać bezpieczeństwo danych pojedynczego użytkownika
- 3. Aplikacja ma być aplikacją SPA
- 4. Aplikacja ma udostępniać API, aby można było skorzystać z dostępnej logiki przez zewnętrzne podmioty
- 5. Autoryzacja do API ma być zrealizowana przy pomocy protokołu JWT

3.5 Analiza wykonanych prac

Wszystkie wymagania funkcjonalne i niefunkcjonalne w aplikacji zostały wykonane. Istnieje szansa na dalszy rozwój aplikacji przez wykonanie prac oznaczonych jako opcjonalne. Dodatkowo pozostało wiele pracy nad użytecznością aplikacji co zostanie przedstawione w dalszych rozdziałach pracy.

3.6 Opis aplikacji

Aplikacja służy do kategoryzowania dokumentów przy pomocy danych geolokalizacyjnych.

Zdecydowanie upraszcza to zarządzanie plikami przez wyświetlanie ich na mapie Google. Wszystkie dane są przedstawione przy pomocy znaczników, które są grupowane.

W aplikacji został zaimplementowany algorytm odgadywania lokalizacji utworzenia pliku. Przy pomocy danych z *Google Historia lokalizacji* przyporządkowuje się plikom miejsce ich utworzenia. Przykładowo będąc 1-szego stycznia w Pradze telefon odnotuje pobyt w tym miejscu, a następnie Google zbierze te dane, to każdy dokument, który będzie miał datę utworzenia 1-szy stycznia zostanie automatycznie dodany do tej lokalizacji. Zostały dodane liczne optymalizacje tego procesu – kategoryzowanie po geohash'u, indeksowanie bazy danych. Wszystkie te zabiegi pozwalają, aby użytkownik otrzymywał te dane w jak najszybszy sposób. Podnosi to użyteczność aplikacji i redukuje współczynnik odrzuceń aplikacji.

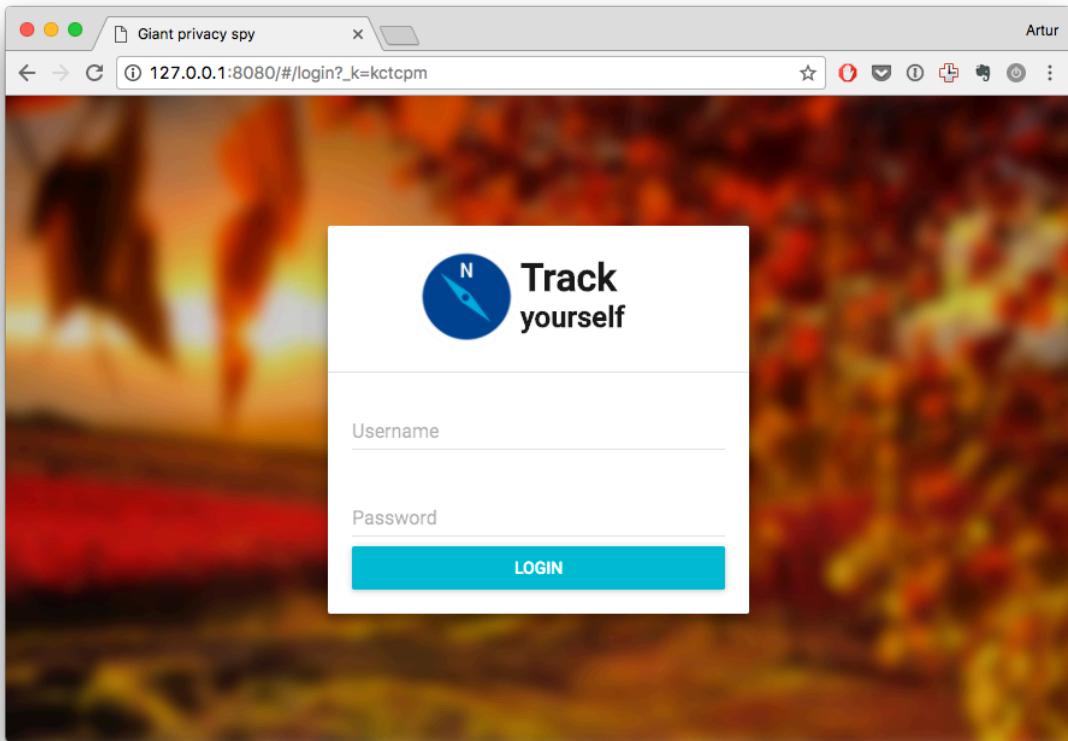
Interfejs został zbudowany z użyciem biblioteki Material UI. Pozwala ona tworzyć strony zgodne z Material Design. Cały projekt strony został stworzony tak aby przestrzegać zasad Material Design i być użytecznym także na telefonach komórkowych z systemem Android.

Operacja uzupełnienia danych z lokalizacji Google jest manualna przez dodanie pliku dostępnego pod adresem https://takeout.google.com/settings/takeout/custom/location_history. Jednym z usprawnień aplikacji w przyszłości może być uproszczenie tej czynności przez automatyczne aktualizowanie tychże danych.

Aplikacja składa się z 4 podstron.

Nawigacja odbywa się przez natywny mechanizm przeglądarki: przyciski wstecz i dalej.

3.6.1 Strona logowania



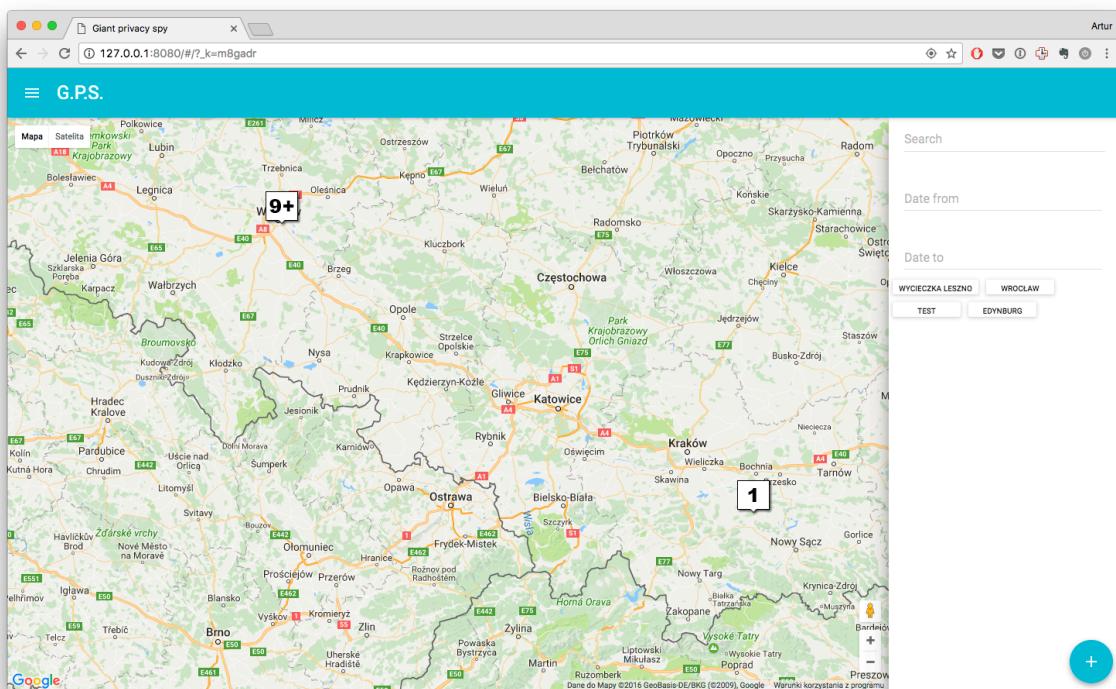
Rys. 3.5 Ekran logowania

Ekran logowania przedstawiony powyżej pozwala użytkownikowi na bezpieczne przechowywanie swoich danych.

Hasła w aplikacji przechowywane są jako sumy kontrolne, które obliczane są na podstawie hasła użytkownika i losowej soli. Algorytm użyty do wyliczenia sumy kontrolnej to *sha1*. Tak przygotowane hasło zapewnia bezpieczeństwo w przypadku kompromitacji bazy danych serwisu.

W polach tekstowych została ustawiona walidacja. Decyduje ona czy użytkownik i hasło są poprawne. Gdy użytkownik spróbuje zalogować się z niepoprawnymi danymi taka informacja zostanie także wyświetlona.

3.6.2 Ekran główny



Rys. 3.6 Ekran główny aplikacji

Główne ekran aplikacji odpowiedzialny jest za wyświetlanie dotychczas zgromadzonych dokumentów na mapie zajmującej większość ekranu. Z poziomu tego widoku użytkownik jest w stanie filtrować dokumenty, a także dodawać nowe.

Mapa wyświetla znaczniki, którym są podporządkowane numery. Określają one ilość elementów kryjących się pod daną grupą. Znowuż grupy są tworzone przy pomocy algorytmu geohashingu przedstawionego w pierwszym rozdziale. Dokładność pól przekazanych do algorytmu determinuje skala mapy jak i obszar widoczny na ekranie. Tak przygotowane informacje pozwalają zawęzić wyniki wyszukiwania co jednoznacznie przekłada się na prędkość wyszukiwania licznych dokumentów gromadzonych przez użytkowników. Po dodaniu odpowiednich uprawnień stronie, ustawiany jest środek mapy, który odpowiada aktualnej lokalizacji urządzenia użytkownika.

Prawa strona aplikacji posiada 2 funkcje.

Po pierwsze, służy jako sposób wyszukiwania dodanych dokumentów. Pierwsze pole jest przeznaczone na frazę wyszukiwania. Pozwala ona przeszukiwać zbiór elementów w sposób tolerujący drobne błędy w nazwie. Dla przykładu przy frazie *klatka* zostanie odnaleziony dokument o nazwie *kratka*. Tak przygotowana tolerancja pozwala w prosty sposób wyszukiwać dokumenty, których dokładna nazwa i/lub opis nie są znane użytkownikowi. Tak prosty zabieg w wielkim stopniu zwiększa użyteczność aplikacji.

Poniżej są 2 pola pozwalające na zawężenie zakresu datowego wyszukiwanych dokumentów. Wykorzystana została natywna kontrolka z Material design.

Ostatnim elementem wyszukiwania są etykiety. Aplikacja pozwala na kategoryzowanie elementów przy pomocy wspomnianych etykiet. Jest to bardzo prosta implementacja kategorii,

która umożliwia zapanować nad wielkim katalogiem dokumentów. Tak dla przykładu zdjęcia z wyjazdu do Edynburga można oznaczyć etykietą *Edynburg*. Następnie po wybraniu w/w etykiety w polu wyszukiwania na mapie zostaną wyświetlane wszystkie zdjęcia posiadające jedynie tą etykietę. Wybór wielu etykiet oznacza wyszukiwanie elementów, które posiadają przynajmniej jedną z wybranych etykiet.

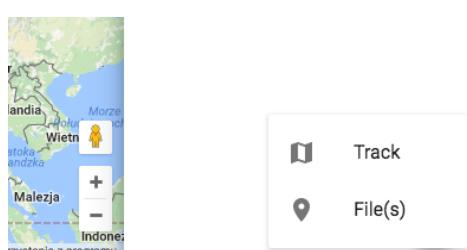
Kolejną funkcjonalnością dostępną z poziomu ekranu głównego jest możliwość dodawania dokumentów i ścieżek. Menu dodawania jest dostępne z poziomu pływającego przycisku dodawania przedstawionego na Rys. 3.7. Po naciśnięciu przycisku ukazują się 2 opcje.

Dodawanie ścieżek służy do importu historii lokalizacji użytkownika od Google. Plik ten później służy do ustalania lokalizacji dodanych dokumentów.

Kolejną opcją jest dodawanie dokumentów. Odbywa się to w bardzo prosty sposób, a następnie algorytmy serwera odczytują wszystkie metadane dostarczone przez ten plik. Dla obrazów algorytm odczytuje dane EXIF, a następnie zapisuje w bazie te dane do specjalnie przygotowanej struktury. Algorytm dodawania pliku odczytuje ścieżkę użytkownika, później przyporządkowuje po dacie utworzenia pliku lokalizację, w której mógł zostać stworzony. Geolokalizacje zapisane w obrazach są dokładniejsze i wcześniej przedstawiony krok jest dla nich pomijany.

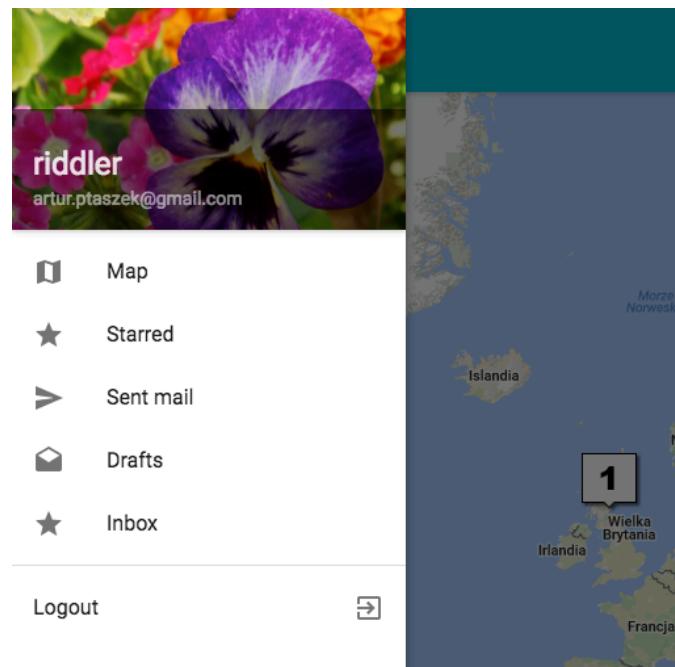


Rys. 3.7 Menu dodawania



Rys. 3.8 Rozwinięte menu dodawania

Naciśnięcie menu hamburger skutkuje otwarciem szuflady z Material Design przedstawionej na Rys. 3.9. Ukrywa ona funkcje rzadko używane, które mogłyby niepotrzebnie skupiać uwagę użytkownika. Znajduje się tam nazwa użytkownika aktualnie zalogowanego użytkownika, link do ekranu głównego, a także link wylogowujący użytkownika. W przyszłości można dodać więcej opcji użytkownika, a także linki do innych podstron.



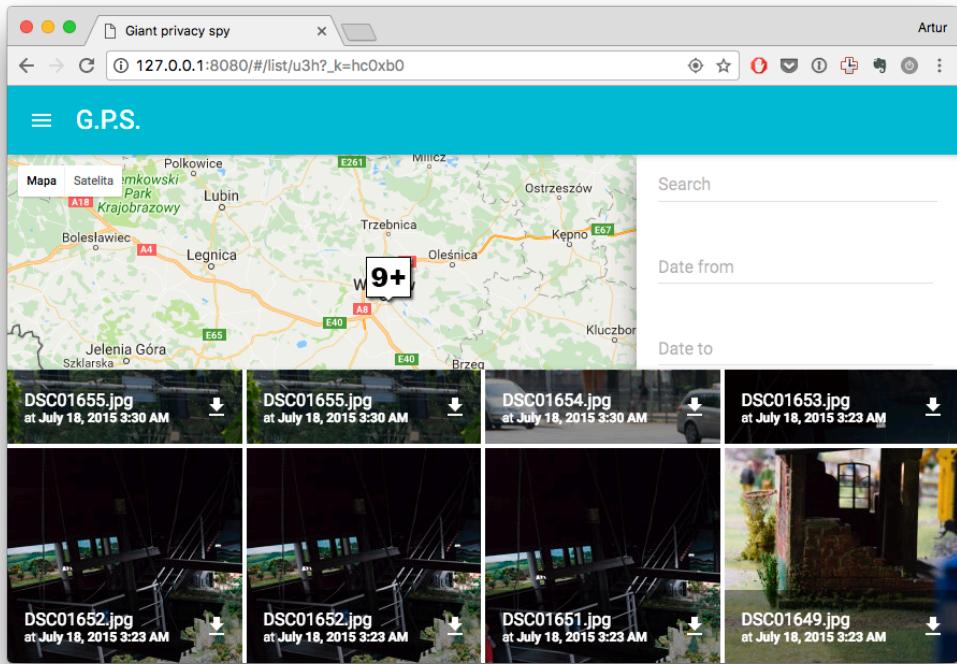
Rys. 3.9 Szuflada aplikacji

3.6.3 Widok listy

Naciśnięcie etykiety skutkuje otwarciem listy plików dostępnych w danej lokalizacji, a także spełniających wszystkie kryteria zadane w wyszukiwarce. Rys. 3.10 przedstawia widok plików graficznych dostępnych pod etykietą w okolicach Wrocławia.

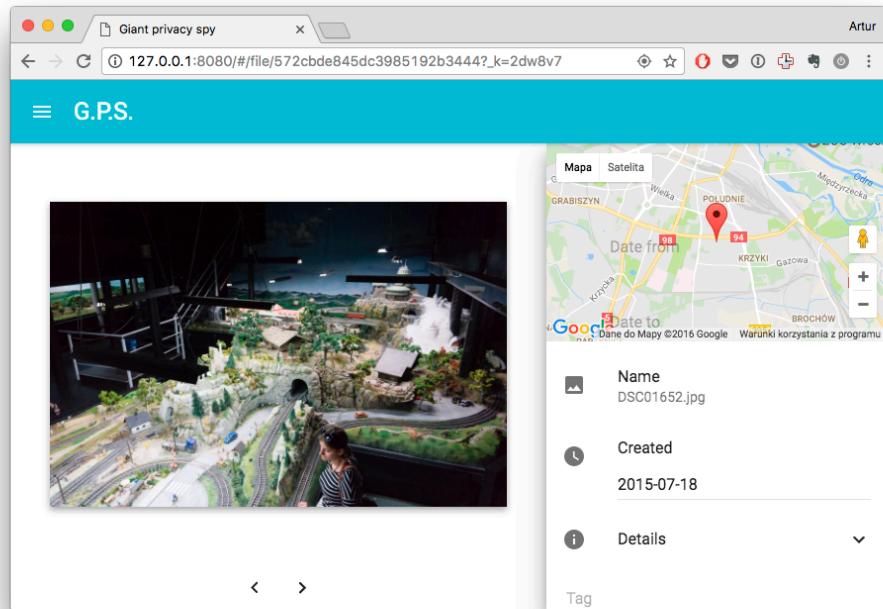
Lista wyświetla znane pliki w postaci miniatur. Rozpoznawanymi plikami są m. in. pliki video i obrazy. Pozostałe są wyświetlane jako pusta karta z tytułem. Z tego poziomu jest się w stanie również pobrać plik na dysk.

Kliknięcie karty kieruje użytkownika do widoku szczegółów dokumentu.



Rys. 3.10 Widok listy

3.6.4 Widok szczegółów dokumentu



Rys. 3.11 Widok szczegółów

Widok ten pozwala na podgląd wszystkich danych pobranych nt. danego dokumentu, a również ich edycję.

Główna część jest to pogląd pliku. Obsługiwane są obrazy i wideo, dla pozostałych formatów wyświetlna jest zaślepka. Poniżej podglądu znajdują się przyciski przewijania. Respektują one kryteria wyszukiwania i wybraną grupę na mapie.

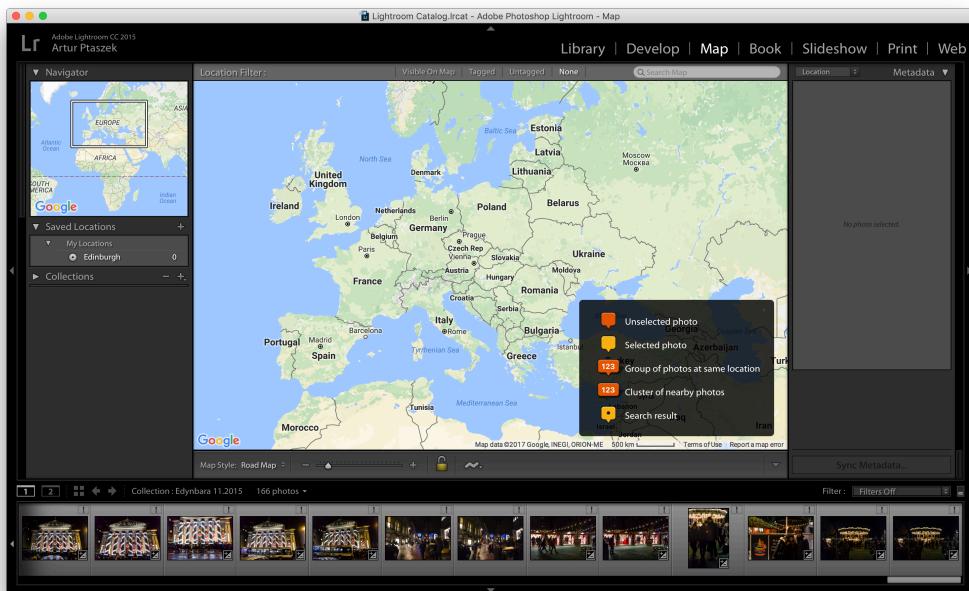
Prawa strona zawiera wszystkie informacje o aktualnie wyświetlonym pliku. Lokalizacja jest przedstawiona na mini-mapie jest przy pomocy domyślnego znacznika Google Maps. Przesunięcie go skutkuje zaktualizowaniem pozycji danego pliku w bazie danych. Poniżej znajduje się nazwa i data utworzenia. Zmiana tej drugiej właściwości uruchamia algorytm ustalania pozycji pliku. Wykonywanie takich kroków w tle znacznie wpływa na użyteczność aplikacji. Na samym dole prawego panelu znajduje się pole do wprowadzania etykiet. Funkcjonalność ta korzysta z mechanizmu podpowiadania już istniejących etykiet, co w przypadku tworzenia serii dokumentów zmniejsza szansę na pomyłki przy wprowadzaniu etykiet. Kliknięcie dodanej etykiety usuwa ją. Możliwość tą potwierdza kursor myszki.

3.7 Porównania podobnych systemów dostępnych na rynku

Rynek jest obfit w rozwiązania oferujące kategoryzowanie zdjęć. Wiele z nich oferuje o wiele większe możliwości kategoryzowania, lecz nie posiadają wszystkich opcji dostępnych w aplikacji stworzonej w ramach tej pracy.

3.7.1 Adobe Lightroom

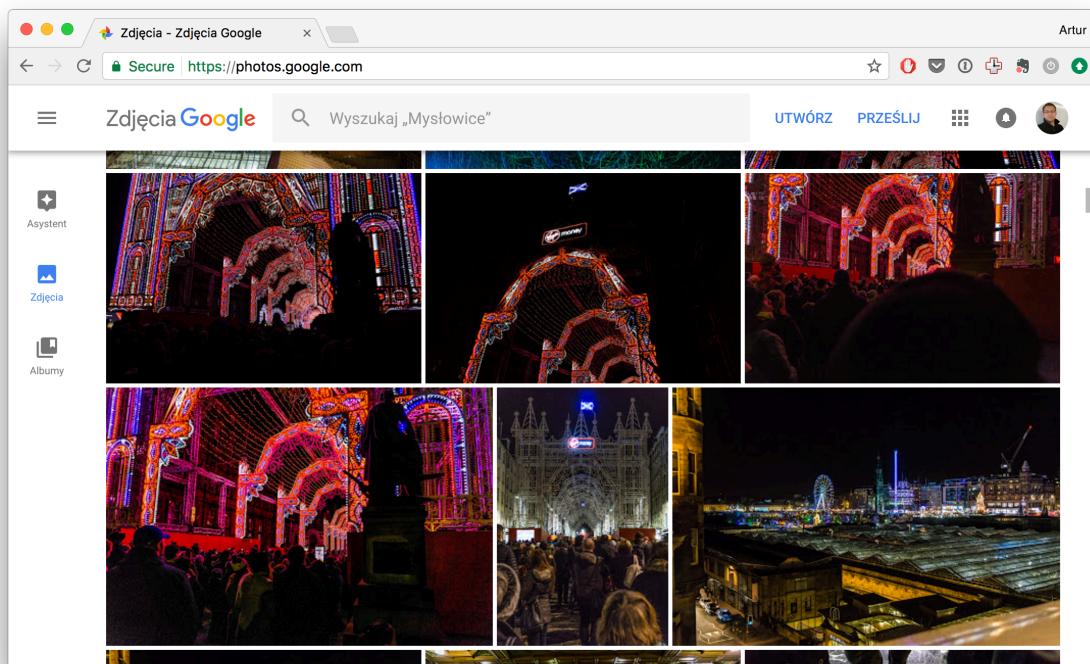
Aplikacja stworzona przez firmę Adobe skierowana do profesjonalistów. Zawiera wiele opcji organizowania i kategoryzowania zdjęć, a także opcje edycyjne. Najważniejszym elementem w kontekście porównywania rozwiązań jest mapa, na której przedstawiane są zdjęcia na podstawie danych EXIF. Pomimo ilości funkcji i obsługiwanych formatów program nie pozwala na wprowadzenie historycznych danych lokalizacji przez co nie jest w stanie domyślić się, gdzie zostały wykonane zdjęcia. Program nie obsługuje plików innych niż obrazki, co posiada stworzona aplikacja. Jest to rozwiązanie płatne, które kosztuje prawie 13 euro miesięcznie w pakiecie z aplikacją Adobe Photoshop. Niestety nie jest możliwe kupno tych aplikacji oddziennie.



Rys. 3.12 Moduł mapy programu Adobe Lightroom

3.7.2 Zdjęcia Google

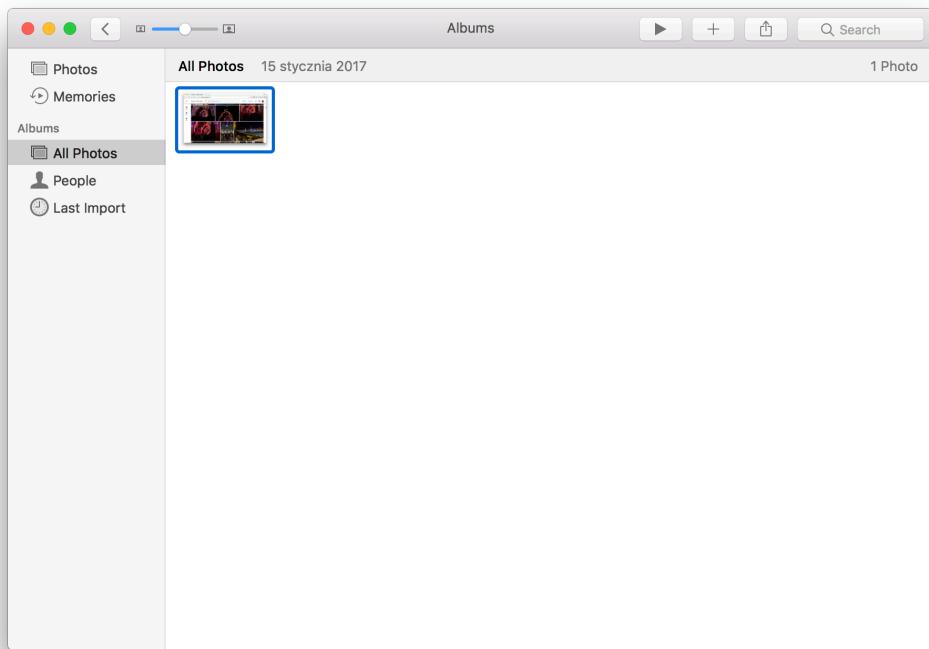
Aplikacja internetowa stworzona przez potentanta wyszukiwarkowego. Stosuje ona zaawansowane algorytmy uczenia maszynowego do uzyskania kategorii zdjęć czy lokalizacji ich wykonania. Bazuje ona na tych samych danych, które wykorzystywane są w programie stworzonym w ramach tej pracy. Dodatkowo posiada ona wiele innych informacji o użytkowniku co pozwala jej na automatyczne kategoryzowanie albumów względem twarzy i lokalizacji. Google analizuje fotografie i pozwala on wychwycić zabytki na fotografiach. Wbudowana wyszukiwarka pozwala na wyszukiwania bezpośrednio lokalizacji – jest to kolejna zaleta programu. Jednak pomimo takich wielkich możliwości wszystko to sprowadza się do zdjęć. Program nie potrafi przechowywać innych plików, co jest przewagą stworzonego programu. Publikując zdjęcia musimy się liczyć z utratą praw do nich. Wysyłając pliki daje się Google więcej informacji o ich użytkownikach, co dla niektórych może być wystarczającym argumentem przeciwko temu rozwiązaniu.



Rys. 3.13 Aplikacja Zdjęcia Google

3.7.3 Zdjęcia Apple

Aplikacja wbudowana dla urządzeń wyprodukowanych przez Apple. Dostępna dla platform desktop, a także dla urządzeń mobilnych. Pozwala na przechowywanie zdjęć w chmurze. Posiada funkcjonalność kategoryzowania zdjęć w albumy i rozpoznawanie twarzy na fotografiach. Moduł map został także wbudowany w aplikację, który pozwala wyświetlić geolokalizowane zdjęcia, jednakże nie analizuje ścieżek użytkownika, przez co nie dodaje sama takich informacji. Jak każdy zaprezentowany program nie pozwala na magazynowanie innych danych niż fotografie.



Rys. 3.14 Widok główny aplikacji Zdjęcia Apple

4 Analiza użyteczności

Poniżej szczegółowo zostaną przeprowadzone analizy poszczególnych podstron w aplikacji. Metodą analizy są heurystyki zdefiniowane przez Jakoba Nielsena. Na sam koniec rozdziału będą przedstawione wnioski z analizy [5].

4.1 Podstawa teoretyczna

W celu przybliżeniu czytelnikowi zagadnienia użyteczności zostaną przedstawione podstawowe pojęcia związane z użytecznością.

4.1.1 Użyteczność aplikacji internetowych

Użyteczność jest to miara jakości, która determinuje prostotę interfejsu dla użytkownika.

Użyteczność składa się 5 komponentów:

- Nauczalność – określa z jaką prostotą użytkownik jest podjąć pierwsze kroki w aplikacji
- Wydajność – z jaką szybkością pracują użytkownicy, którzy poznali aplikację
- Zapamiętywalność – w jak szybkim stopniu użytkownik aplikacji jest w stanie przypomnieć sobie funkcje aplikacji po przerwie w jej korzystaniu
- Błędy – jak często popełniane są błędy w rozumieniu aplikacji i jak szybko użytkownik przestaje je wykonywać
- Satysfakcja – jak produkt wydaje się być przyjemny w korzystaniu

Wszystkie powyżej podane komponenty zostały zdefiniowane przez Jakoba Nielsena³.

Użyteczność odgrywa kluczową rolę podczas tworzenia oprogramowania. Decyduje ona o ogólnym odbiorze aplikacji, a także pozwala zachęcić użytkownika do dalszego korzystania z aplikacji. Często aplikacje są wybierane pod kątem użyteczności, ponieważ pozwala to zredukować późniejsze koszty szkoleń z obsługi systemu.

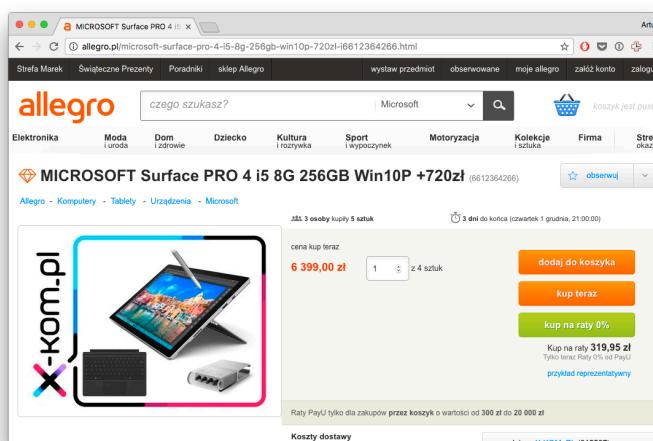
Ostatecznie w środowiskach e-commerce to użyteczność decyduje o sprzedaży. W aplikacjach tego typu bardzo popularne są zabiegi, które ułatwiają klientowi odnalezienie produktu, którego poszukuje, chociażby przez rozbudowane systemy rekommendacji. Proponuję one nie tylko produkty podobne do tych, których poszukujemy, ale także produkty uzupełniające zakup np. przewody, środki czyszczące, akcesoria.

Jak klient już dotrze do strony produktu zastaje on starannie dobraną przestrzeń tak aby zakup był jak najłatwiejszy. Można zauważyc pewne prawidłowości, z których korzystają zagraniczne i Polskie serwisy.

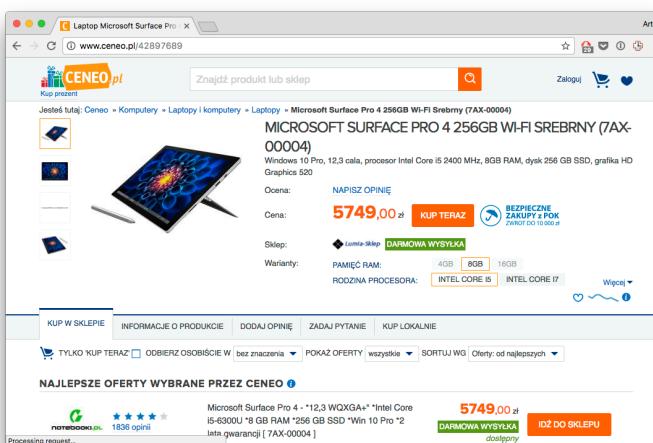
³ (Nielsen, Nielsen Norman Group, 2012)



Rys. 4.1 Przykład strony produktu serwisu Amazon



Rys. 4.2 Przykład strony produktu serwisu Allegro



Rys. 4.3 Przykład strony produktu serwisu Ceneo

Przedstawione powyżej zrzuty ekranu bardzo dobrze pokazują jak sklepy internetowe są projektowane. Najbliżej góry strony możemy wyszczególnić:

- Duże zdjęcia produktu

- Nazwa produktu
- Cena produktu na czerwono
- Pomarańczowy przycisk zakupu

Nie bez powodu zostały podane kolory strategicznych elementów takich jak cena i przycisk zakupu. Mają one wielki wpływ na decyzję użytkownika o zakupie. Istnieje wiele badań, które potwierdzają, że w e-commerce najlepiej sprawdza się kolor czerwony i zielony. W wymienionych czerwony został użyty do wyświetlenia strony produktu, a jego odcień (pomarańczowy) został wykorzystany do przycisku zakupu.

Google także prowadzi badania nt. kolorów. Przeprowadzał on badania A/B, w których użytkownikom serwisu wyświetlał różne odcienie niebieskiego w linkach. Z pozoru bardzo nieważna rzecz, lecz dla wyszukiwarki tej skali jest to bardzo ważne.

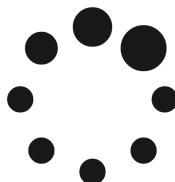
Użyteczność jest badana w wielu aplikacjach i jest ona kluczem do zatrzymania użytkownika przy swoim produkcie.

Jest wiele technik badania użyteczności na stronach internetowych.

4.1.2 Heurystyki

Jakob Nielsen na swoim blogu spisał zbiór 10 reguł pomagających z ewaluować użyteczność aplikacji⁴.

1. Aplikacja powinna pokazywać swój status
Najlepszym przykładem ukazywania statusu jest wskaźnik ładowania przedstawiony niżej.



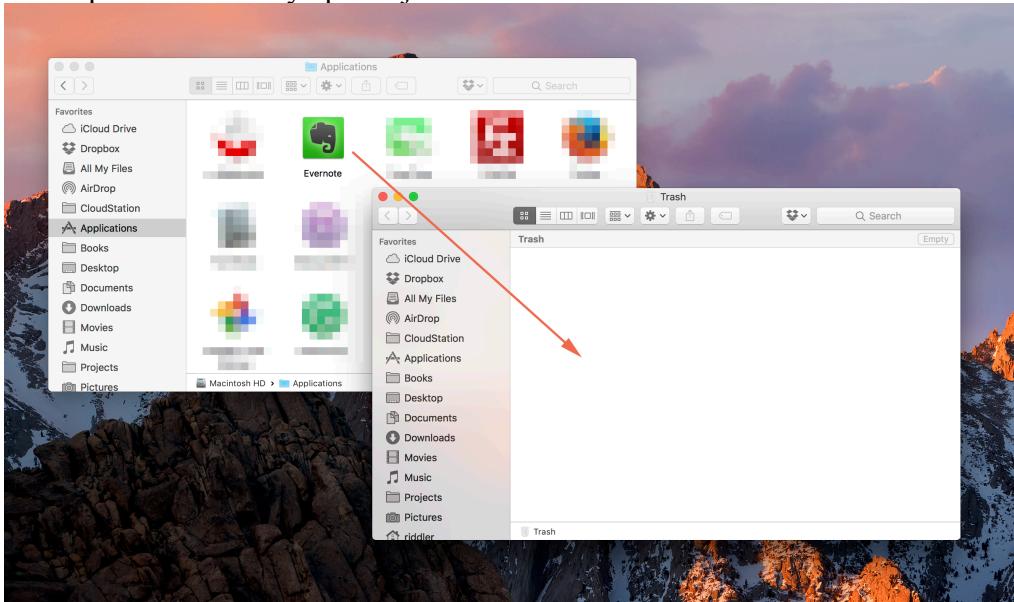
Rys. 4.4 Wskaźnik ładowania

Zawsze, gdy wykonujemy jakiś proces trwający dłuższą ilość czasu pokazuje ikonkę, która pokazuje, że wykonuje się pewien proces w tle. Pozwala to użytkownikowi zrozumieć co w danym momencie dzieje się z aplikacją. Nieposiadanie wskaźnika statusu aplikacji powoduje u użytkowników frustrację, która kończy się na wielokrotnym kliknięciu w przycisk akcji, co powoduje dodatkowo opóźnienie całego procesu. Użytkownik także może stwierdzić, iż program się zawiesił i będzie próbował bez skutku restartować aplikację. Jest to jedna z ważniejszych heurystyk i zaraz najłatwiejsza do naprawienia.

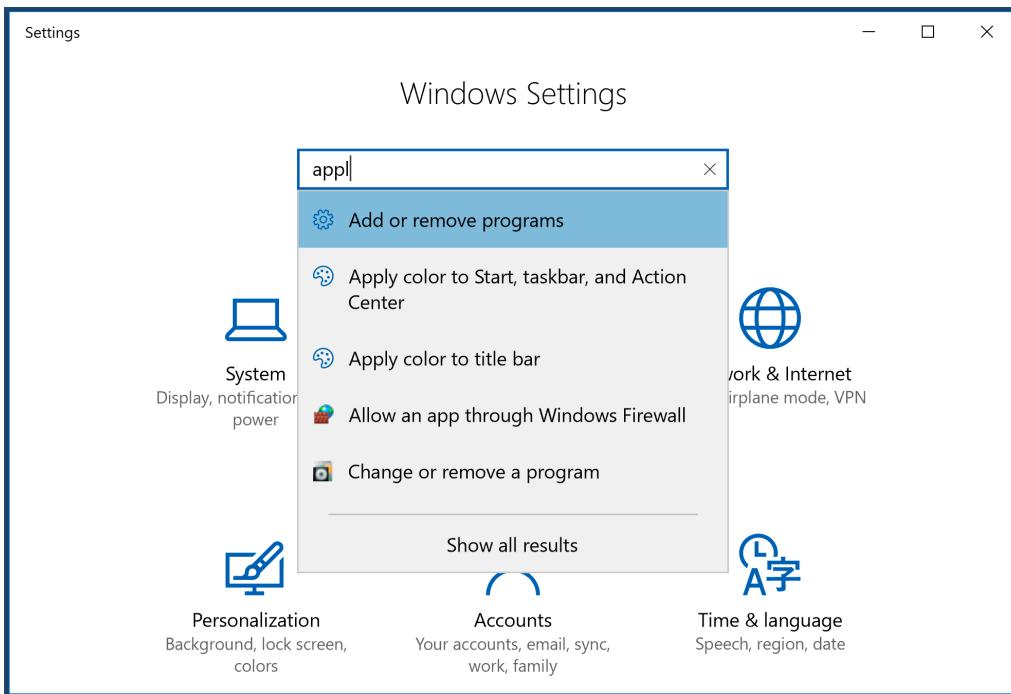
2. Dopasuj system do świata rzeczywistego
Powinno projektować się aplikacje tak aby miały jak najwięcej analogii do świata rzeczywistego. System macOS posiada idealny przykład projektowania aplikacji z zastosowaniem się do tej reguły. Aby na tym systemie usunąć

⁴ (Nielsen, 10 Usability Heuristics for User Interface Design, 1995)

aplikację trzeba ją dosłownie przenieść do kosza. Jak w życiu rzeczywistym, jeżeli chcemy się pozbyć rzeczy wrzucamy je do kosza tak interfejs pozwala dokładnie na to samo. Kontrprzykładem jest rozwiązanie problemu przez system Windows. Istnieje specjalna aplikacja z ustawieniami, gdzie jesteśmy w stanie w sekcji *Dodaj/usuń programy* usunąć program. Szczególnie na początku istnienia Windows'a użytkownicy zamiast przechodzić do w/w programu przenosili skróty aplikacji do kosza.



Rys. 4.5 Przykład usuwania aplikacji na systemie macOS



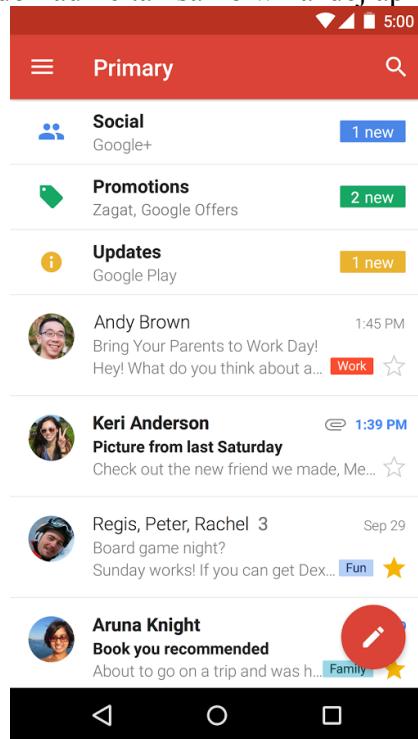
Rys. 4.6 Przykład usuwania aplikacji na systemie Windows Daj użytkownikowi pełną kontrolę

Każda operacja wykonana w aplikacji powinna być odwracalna. Gdy użytkownik zawędruje do części aplikacji, której nie zna powinien w prosty sposób z niej wyjść przez przycisk wstecz lub inne tego typu rozwiązanie. W edytورach tekstu każda operacja powinna zostać rejestrowana i pozwalać na wracanie w historii (funkcje cofnij, powtóż). Dobrym przykładem jest edytor

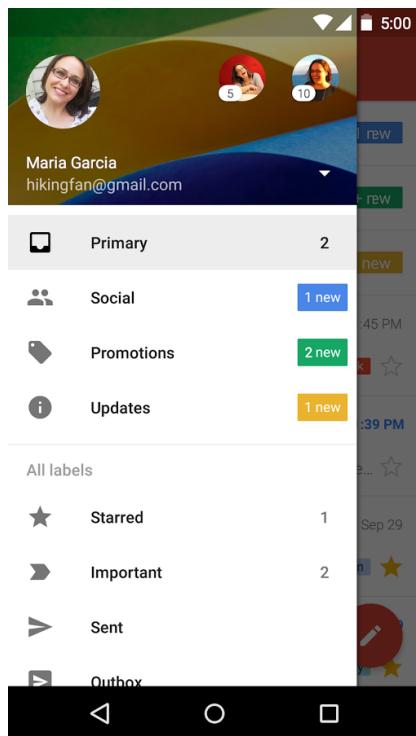
Word, który pozwala każdą zmianę wykonaną w dokumencie cofnąć niezależnie jaka to była operacja.

3. Trzymaj się standardów i dbaj o konsystencję

Gdy projektujemy aplikację powinniśmy się trzymać dotychczas wypracowanych standardów. Przykładowo pisząc aplikacje pod system Android stosowany jest Material Design. Strona <https://material.google.com/> opisuje położenia elementów interfejsu, a także jak powinna się zachowywać sama aplikacja. Użytkownik widząc dobrze zaprojektowaną aplikację od samego początku wie jak po niej nawigować, ponieważ wszystkie elementy zachowują się i wyglądają dokładnie tak samo w każdej aplikacji tego systemu.



Rys. 4.7 Widok aplikacji GMail na systemie Android



Rys. 4.8 Widok na "Navigation Drawer" z Material Design

Na powyższych zrzutach widać zastosowane komponenty zdefiniowane przez Material Design.

4. Unikaj błędów

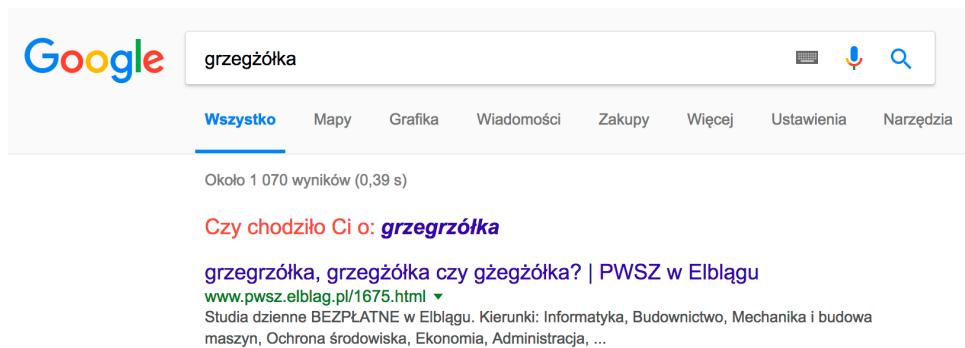
Błędy skutecznie odstraszają użytkownika od korzystania z aplikacji. Nie tylko te wyświetlane przy pomocy komunikatu błędu, ale także takie, które po prostu nie umożliwiają użytkownikowi podjąć akcji. Przykładowo, gdy nie możemy kupić produktu, który jest niedostępny na magazynie a w aplikacji wyświetla się jako możliwy do kupienia.

5. Pozwalaj rozpoznawać zamiast zmuszać do zapamiętywania

Podczas gdy jest w aplikacji proces składający się z wielu kroków powinniśmy dbać o to, aby użytkownik podsumowanie wypełnionych danych miał w zasięgu wzorku. Użytkownicy z natury są leniwi i nie powinno się tego zmieniać. To aplikacja powinna się dostosować do użytkownika, lecz nigdy w drugą stronę.

6. Zapewnij elastyczność i efektywność

Wielokrotne powtarzanie akcji jest niezmiernie irytujące. Tak samo obowiązek pamiętania możliwych opcji w polu do wprowadzania danych. W zamian tego powinno się projektować systemy pamiętające akcje użytkownika poprzez wypełnianie pól wcześniej zdefiniowanych przez użytkownika takich jak np. adres wysyłki czy frazy wyszukiwania. System także powinien korygować wszystkie pomyłki użytkownika tak jak np. robi to Google na zrzucie poError! Reference source not found..



Rys. 4.9 Korekta zastosowana przez wyszukiwarkę Google

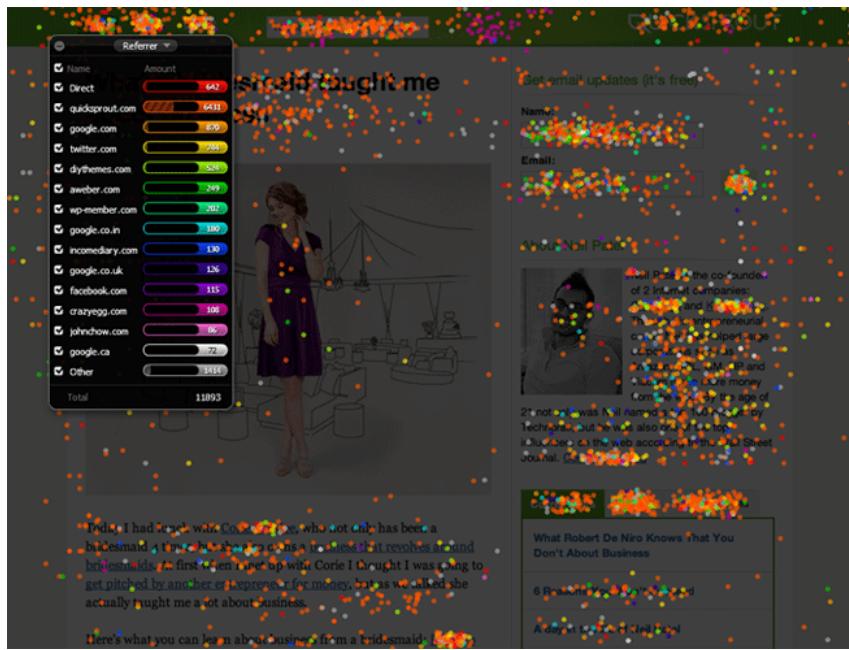
7. Estetyka oraz umiar
Mniej znaczy więcej. Przy projektowaniu warto zwrócić uwagę na ilość informacji wyświetlnych na ekranie. Podczas procesu zamówienia produktu nie jest ważny szczegółowy opis produktu, lecz jego cena i nazwa. Należy minimalizować ilość wiedzy jaką użytkownik musi posiadać przy wykonywanej operacji. Reguła ta nie tylko odnosi się do przeładowania ilością elementów na stronie, ale też do projektu strony. Inwencja twórcza grafika jest ważna, lecz na to także trzeba uważać. Minimalizm jest postrzegany za bardziej atrakcyjny i pozbawiony zbędnego szumu informacyjnego.
8. Skuteczna obsługa błędów
Komunikaty o błędach powinny w jasny sposób mówić o przyczynie jego wystąpienia. Kody błędów nic nie mówią i potrzeba zaglądania do instrukcji jest uciążliwa. Uzyskanie informacji zwrotnej jest w przypadku błędów w implementacji jest bardzo wartościowa. Każdy system powinien udostępniać miejsce do zgłoszenia takiego typu błędów, by później go wyeliminować.
9. Pomoc i dokumentacja
Nie wszystkie elementy interfejsu są łatwe do zrozumienia w pierwszym kontakcie ze stroną. Aplikacja powinna udostępniać dokumentację elementów, a nawet oferować wsparcie przez e-mail. Informacje te powinny być łatwo dostępne, przeszukiwalne i oferować listę kroków potrzebą do wykonania akcji. Warto też pamiętać, aby nie była zbyt obszerna dla użytkownika.

4.1.3 Śledzenie kliknięć

Technika ta umożliwia mierzenie zachowań użytkownika na już istniejącej stronie. Dzięki narzędziom takim jak CrazyEgg⁵ możliwa jest obserwacja kliknięć na przygotowanej przez oprogramowanie mapie termicznej. Przedstawia ona dane wielu sesji użytkowników z podziałem na podstrony i rozdzielcości. Analiza tak przygotowanej mapy polega na odnalezieniu elementów interfejsu, które są rzadko używane lub niezrozumiałe. Miejsca „zimne” to miejsca, w których użytkownicy rzadko decydują się nawigować przez kliknięcie lub przewijanie strony. Lokalizacja tych rzadko odwiedzanych elementów ułatwia usunięcie zbytecznych elementów. Dzięki tej technice można zidentyfikować jeszcze jedną klasę problemu, czyli próba nawigacji na elementach, które nie oferują żadnej akcji. Dla przykładu, gdy logo w lewym górnym rogu nie jest aktywne, a jest częstym elementem klikanym przez użytkowników powinno posiadać akcję powrotu do strony głównej. Metoda jest bardzo tania w implementacji i analizie, ale jedynie dla stron, które posiadają wystarczający ruch.

⁵ <https://www.crazyegg.com/>

Środowisko e-commerce w bardzo prosty i relatywnie niedrogi sposób najwięcej czerpie z tej techniki.



Rys. 4.10 Przykładowa mapa termiczna CrazyEgg

4.1.4 Testy A/B

Testy A/B tak samo jak poprzednia metoda są do zastosowania jedynie na już istniejącej aplikacji. Polega na stworzeniu nowej wersji funkcjonalności, a następnie mierzeniu konwersji. Konwersja w tym przypadku jest definiowana przez cel jaki został wyznaczony przez biznes dla tego elementu interfejsu. Celem tym może być zwiększenie sprzedaży w przypadku sklepów internetowych lub zwiększoną ilość kliknięć w reklamę. Przy badaniu bierze udział mała grupa aktualnych użytkowników aplikacji. Grupa badawcza jednak nie może być za mała. Jest to zazwyczaj 10% użytkowników serwisu. Ważnym elementem metody jest zdolność badania konwersji. Warto skorzystać z gotowych narzędzi monitorujących poczynania tj. Google Analytics [6]. Następnie na podstawie zgromadzonych danych można porównać 2 wersje. Wybierana jest, ta która najlepiej wpłynęła na konwersje. Trzeba jednak pamiętać o jeszcze jednym ważnym elemencie. W trakcie jednego badania nie powinno się przeprowadzać innego, ponieważ znacznie może to zakłócić jednoznaczne stwierdzenie co bezpośrednio wpłynęło na zrealizowanie bądź nie postawionego celu.

4.1.5 Zdalne badania z użytkownikami

Badania przeprowadzane drogą internetową. Polegają na obserwacji zachowań, które pozwalają później sprawdzić w jaki sposób zostały wykonane postawione przed uczestnikami zadania. Głównymi zaletami tej metody jest niski koszt i krótki czas potrzebny na uzyskanie wyników. Serwisy tj. <https://www.usertesting.com/> oferują platformę do przeprowadzania tego typu testów. Pracownicy firmy analizują postawione przed nimi strony i otrzymuje się natychmiastową informację zwrotną.

4.1.6 Metoda śledzenia oka

Badanie polega na śledzeniu wzroku osoby badanej. Wykonuje się to przy pomocy kamery lub specjalnego urządzenia nazywającego się eyetracker. Następnie na podstawie tego

generuje się mapy termiczne, które pozwalają stwierdzić w jaki sposób badany analizował stronę. Grupa badawcza powinna się składać z minimum 7 osób, żeby dostać zadowalające wyniki. Firmy takie jak <http://www.tobii.com/> zajmują się dostarczeniem oprogramowania i sprzętu do przeprowadzania takich badań.

4.2 Badanie przy pomocy heurystyk

4.2.1 Strona logowania

Pierwsza z heurystyk traktuje o powiadamiania użytkownika o statusie aplikacji. Na tym ekranie jest wiele przypadków spełnienia tej reguły. Sztandarowym przykładem jest ikona ładowania. Na Rys. 4.11 jest pokazane wykorzystanie tego elementu interfejsu. Użytkownik przez zastosowanie tego komponentu jest poinformowany, że musi chwilę poczekać, a system za chwilę odpowie. Gdyby zabrakło wskaźnika użytkownik mógłby pomyśleć, iż system się zawiesił i próbował zalogować się jeszcze raz. Skutkowałoby to jedynie dłuższym czasem oczekiwania na odpowiedź.

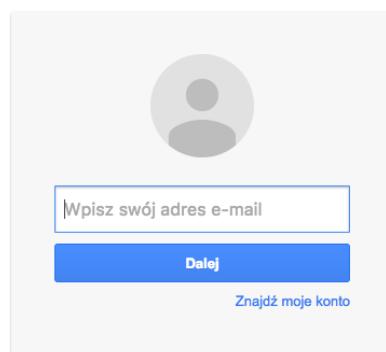


Rys. 4.11 Wskaźnik ładowania na stronie logowania

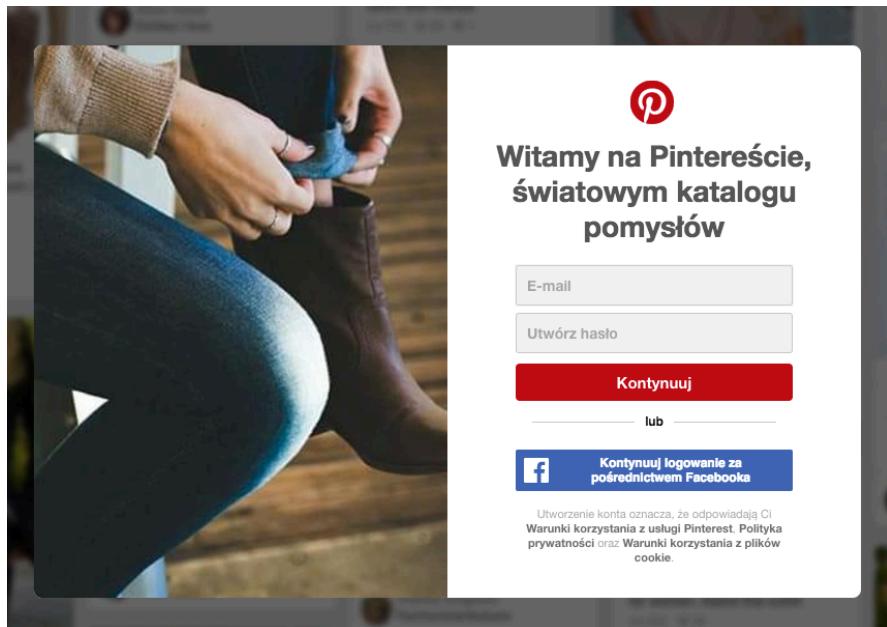
Kolejna z reguł mówi o dopasowaniu interfejsu do świata rzeczywistego. Jest to jedna z cięższych do analizy reguł. Przypadek strony logowania posiada za mało akcji, aby można było mówić o analogiach do świata rzeczywistego. Reguła w tym przypadku nie obowiązuje.

Reguła nr 3 opowiada o pełnej kontroli. Strona logowania oddaje użytkownikowi kontrolę. Każda operacja niepoprawnego wpisania danych jest odwracalna, a poprawne zalogowanie jest także odwracalne przez przycisk wylogowywania, co zostanie pokazane przy analizie głównej strony aplikacji.

Reguła nr 4 opisuje problem standaryzacji interfejsu. Cała aplikacja została wykonana przy pomocy Material UI przez co jeden z standardów jest spełniony. Kolejnym wykorzystanym standardem jest układ samego formularza. Poniższe zrzuty przedstawiają podejścia do logowania użytkowników. Można zauważyć zależność – pierwsze nazwa użytkownika, a następnie hasło. Dół formularza zajmuje dobrze oznaczony przycisk akcji.



Rys. 4.12 Strona logowania Google



Rys. 4.13 Ekran logowania Pinterest

Reguła piątej poświęcone jest unikanie błędów. Ciężko doszukać się sytuacji, w której wystąpi błąd logowania, ale gdy serwer nie odpowiada nie została dodana obsługa poprzez komunikat. Jest to bardzo rzadki przypadek, który wystąpi jedynie w bardzo specyficznych warunkach. Reguła odnosi się też do powstrzymywania użytkownika przed popełnianiem błędów. Na te potrzeby została stworzona walidacja formularza przedstawiona poniżej. Przestrzega to odbiorcę przed popełnianiem błędów w przypadku uzupełniania pól logowania.

Rys. 4.14 Walidacja pola użytkownika

Rys. 4.15 Walidacja hasła

Reguła nr 6 nie ma zastosowania do tego ekranu, ponieważ nie mamy kroków i użytkownik nic nie musi zapamiętywać.

Reguła nr 7 wymaga, aby witryna była elastyczna. Ekran logowania jest to funkcjonalność, która ma jedno zadanie – zautoryzować użytkownika. Przykład idealnie pokazany na Rys. 4.13 zapewnia ją w przeciwieństwie do zaimplementowanej w aplikacji wersji. W serwisie Pinterest użytkownik ma do wyboru kilka sposobów autoryzacji: Facebook, logowanie email. Także w przypadku zapomnienia hasła istnieje możliwość przypomnienia hasła co jest niemożliwe w przypadku zaimplementowanego ekranu.

Estetyka poruszona przez regułę nr 8 jest zastosowana na tej podstronie co czyni ją atrakcyjną dla użytkownika. Wielkie rozmyte zdjęcie w tle nie powoduje skupienia uwagi, a drobne kreski pól tekstowych nie zaciemniają funkcjonalności jaką chce przekazać interfejs. Najważniejszą funkcja, czyli przycisk logowania został podkreślony kolorem niebieskim.

Podstrona posiada zaimplementowaną obsługę błędów użytkownika. Idealnym przykładem jest wprowadzenie błędnych danych logowania. Skutkuje to wyświetleniem komunikatu błędu przedstawionego na Rys. 4.13.

The screenshot shows a login interface with a light gray background. At the top, there is a 'Username' field containing 'testssss'. Below it is a 'Password' field with several dots. At the bottom of the form, a red error message reads 'Wrong username or password'. A large blue button labeled 'LOGIN' is centered at the bottom.

Rys. 4.16 Komunikat błędu podczas niepoprawnych danych logowania

Reguła nr 10 jest niespełniona, ponieważ żadna część aplikacji nie została udokumentowana. Jest to jedna z rzeczy, którą warto poprawić w następnych iteracjach. Punkt ten nie będzie już poruszany w dalszych partiach pracy.

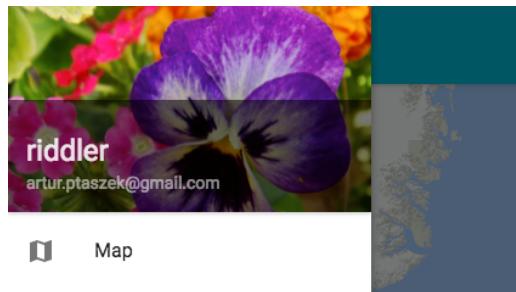
4.2.2 Strona główna

Pierwsza reguła opowiada o statusie aplikacji, w przypadku strony głównej statusem jest sama mapa. Dokładnie pokazuje w jakiej lokalizacji się znajduje użytkownik, a także jakie pliki są dodane do serwisu. Dodanie jakiegokolwiek pliku skutkuje wyświetleniem wskaźnika ładowania przedstawionego poniżej. Wszystkie filtry wybrane po prawej stronie aplikacji mają jasny przekaz, kiedy zostały wybrane. Doszukać także można złamania tej reguły podczas zmiany obszaru mapy aplikacja nie informuje użytkownika nt. ładowania, jednak jest to bardzo proste do naprawienia. Kolejnym z naruszeń jest brak jasnego przekazu, gdzie został dodany plik na mapie. Powinno być to jasno zasygnalizowane przez drobny ruch etykiety na mapie. Zdecydowanie pomogłoby to użytkownikowi na zrozumienie co się właśnie wydarzyło.



Rys. 4.17 Wskaźnik ładowania podczas dodawania plików

Menu hamburger otwiera półkę zawierającą informacje o aktualnie zalogowanym użytkowniku. Także jest to forma pokazywania statusu aplikacji.



Rys. 4.18 Informacje o użytkowniku na półce

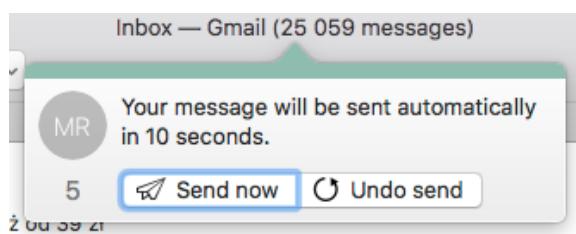
Zgodnie z drugą regułą zostało zaprojektowanych wiele elementów interfejsu. Między innymi przycisk dodawania plików reprezentuje jasny przekaz przez zastosowanie znaku „+”. Użytkownik naturalnie sięga w prawy dolny róg, aby nie wiedząc jeszcze co, ale coś dodać. Następnie ukazuje mu się menu podręczne, w którym już teraz wie co dokładnie może. Etykiety na mapie umieszczone w miejscach wykonania prezentują idealną prezentację dokumentów. Grupowanie tych elementów w kubelki pozwala w prosty i intuicyjny sposób kategoryzować elementy. Przykładowo filtrowanie plików z wyjazdu do Leszna odbywa się przez odpowiednie powiększenie mapy, a następnie nawigowanie do miejsca.



Rys. 4.19 Przykład zgrupowanej etykiet

Dodanie wyszukiwania po lokalizacji byłoby także wskazane co można w prosty sposób dodać w przyszłości. Pozwoliłoby by to na większą synergię z rzeczywistością.

Swoboda zdefiniowana przez regułę nr 3 jest częściowo zastosowana. Z poziomu strony głównej może on filtrować, a także wycofać każdy filtr. Półka zawiera przycisk wylogowania, co pozwala przełączyć użytkownika. Jednak przycisk dodawania nie pozwala na akcję wycofania operacji. Podczas dodania ścieżki użytkownik nie jest w stanie jej usunąć. Popełniając błąd przez import nie swojej ścieżki system będzie podpowiadał lokalizacje, które są nieprawidłowe, co jest fatalne w skutkach. Dodanie plików powinno także zawierać opcję na proste cofnięcie m.in. ze względu na możliwość importu wielu dokumentów. Jedyną opcją jest odnalezienie każdego pojedynczego pliku i jego usunięcie, co jest niezwykle trudne przy 10 plikach, a przy 100 niemalże niemożliwe. Przykładem dobrego zastosowania tego mechanizmu jest przy próbie wysłania w programie MailButler.



Rys. 4.20 Możliwość wycofania operacji wysłania email

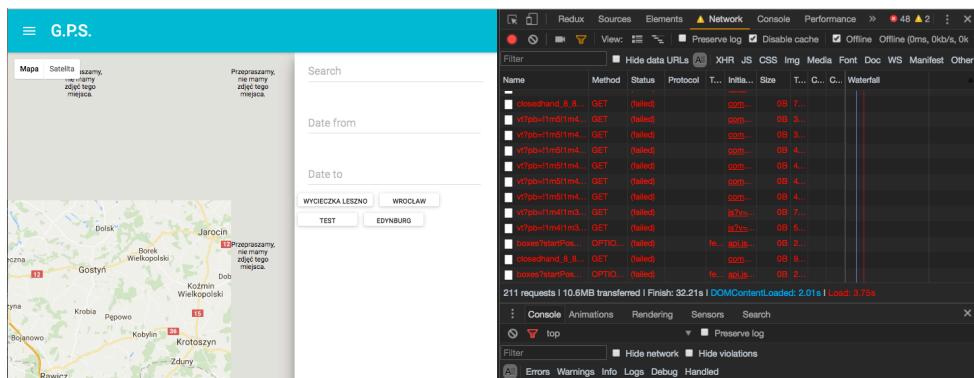
Spójność strony głównej jest to jej wielka zaleta. Począwszy od skorzystania z dobrze znanej i już ustandaryzowanej mapy Google, po wykorzystanie reguł dostępności dla aplikacji internetowych. Każdy element interfejsu jest dostępny przy pomocy klawiatury. Klawiszem tabulacji można nawigować przez kontrolki dostępne w zasięgu widoku. Sygnalizowane jest to przez efekt plusku zdefiniowany w Material Design.



Rys. 4.21 Efekt plusku

Utartym standardem jest umieszczanie najważniejszych obiektów na stronie bliżej lewej strony. Mapa jako najważniejszy element znalazła się po lewej stronie, a filtry, które przewidziane są do doraźnego użytku po stronie prawej [7]. Tak samo jak przy stronie logowania zostało zastosowane Material Design, z oczywistych względów nie będzie już poruszane w dalszej części pracy.

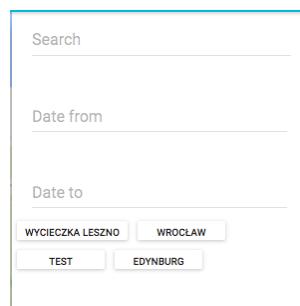
Wiele miejsc tego interfejsu łamie regułę nr 5. Począwszy od braku możliwości cofnięcia dodania wielu plików lub ścieżek, po brak miejsca przeznaczonego na ich wyświetlanie. Niepoprawna odpowiedź serwer lub brak połączenia skutkuje dziwnymi zachowaniami aplikacji co powoduje u użytkownika uczucie konsternacji. Brak odpowiedzi nagle przestaje wyświetlać elementy na mapie nie informując jaka jest istota problemu. Użytkownik w akcji desperacji zaczyna kliknąć różne przyciski, co nie poprawia zaistniałej sytuacji. Wykorzystanie Google Maps w drobnym stopniu naprawiło sytuację przez wyświetlenie komunikatu „Przepraszamy nie mamy zdjęć tego miejsca”. Nie jest to do końca prawda, ale jest to zdecydowanie lepsze niż brak jakiekolwiek informacji zwrotnej. Zdecydowanie warto poprawić ten aspekt.



Rys. 4.22 Stan aplikacji przy braku połączenia internetowego

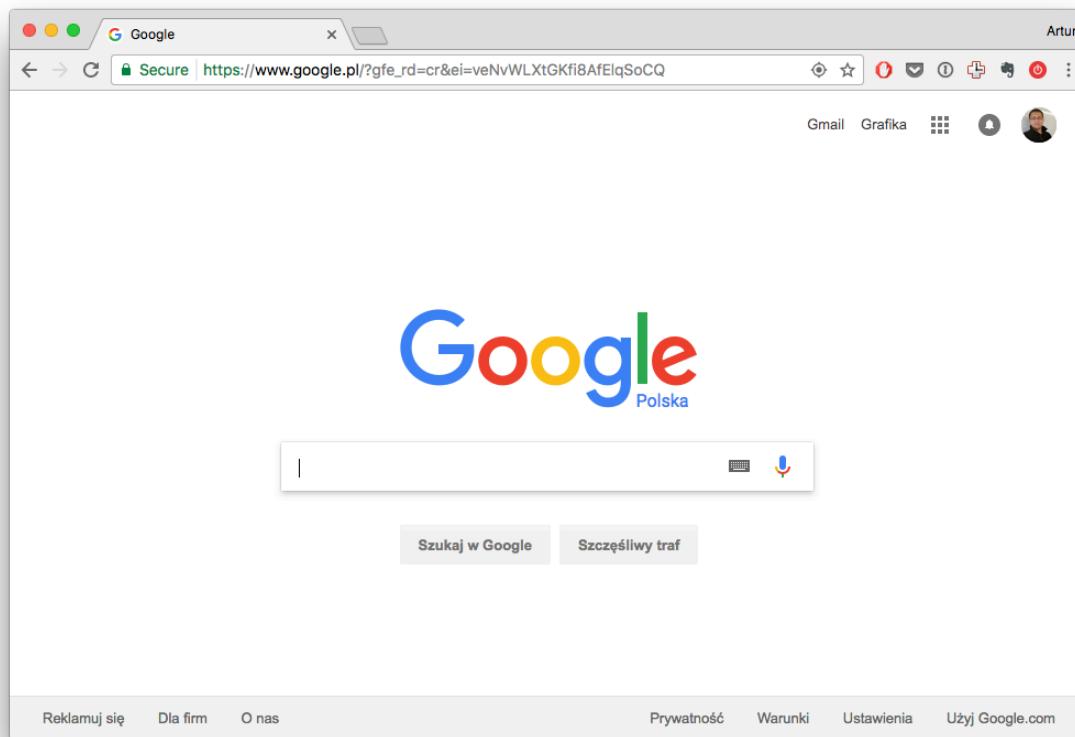
Użytkownik przy tak minimalistycznym interfejsie nie musi pamiętać o żadnej wybranej opcji w aplikacji. Wszystkie komponenty są w zasięgu wzroku co pozwala w każdej chwili na zmianę filtra lub zmianę oglądanej lokalizacji. Powoduje to spełnienie reguły nr 6.

Elastyczność została zapewniona przez różne typy filtrów operujących na plikach. Jest to wystarczająca liczba biorąc uwagę o zakres funkcjonalności całej aplikacji. Oczywiście można dodać kategoryzowanie w formie galerii, ale jest to możliwość usprawnienia kolejnych wersji aplikacji.



Rys. 4.23 Filtry aplikacji

Interfejs całej aplikacji jest bardzo minimalistyczny między innymi ze względu na niewielką ilość funkcjonalności, które użytkownik widzi na pierwszy rzut oka. Duża część kodu domyśla się czego użytkownik oczekwał. Przykładem jest sposób dodawania plików. Poprzez odgadywanie ścieżki z metadanych i danych lokalizacji nie musi on myśleć, żeby dodać za każdy razem lokalizację dokumentu. Innym przykładem serwisu, który oferuje bardzo prosty interfejs, za którym stoi bardzo dużo funkcjonalności jest wyszukiwarka Google. Jedno pole tekstowe zawiera ogromną ilość mechanizmów podłączonych do niej m. in. Adwords, wyszukiwanie po ogromnej bazie danych, personalizacje użytkowników, liczne integracje z mapami, obrazami.

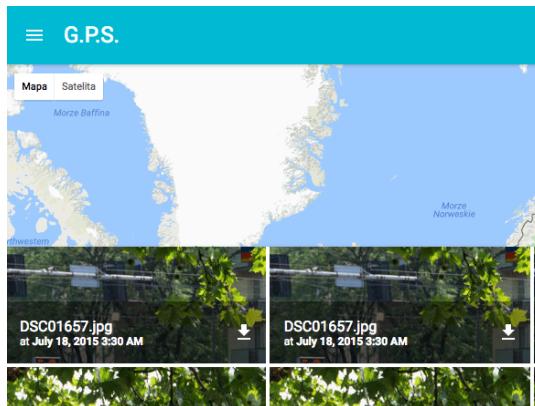


Rys. 4.24 Wyszukiwarka Google

Wspomniane zostało to po części w punkcie o unikaniu błędów, ale podstrona nie posiada mechanizmów do wyświetlania błędów. Powoduje to, że niniejsza reguła nie jest spełniona.

4.2.3 Widok listy

Podstrona ta idealnie pokazuje status systemu. Zastosowanie dodatkowej warstwy na mapie pozwala w prosty sposób użytkownikowi zrozumieć, dlaczego znalazł się na tej podstronie. Gdyby została zastosowana osobna podstrona, która byłaby umieszczona zamiast mapy, użytkownik mógłby poczuć się zagubiony. Powstała także dziedzina badająca to zjawisko. Psychologia poznawcza zajmuje się procesami uczenia się, postrzegania. Powstało wiele badań na ten temat, a badacze użyteczności bazują na ich wynikach.

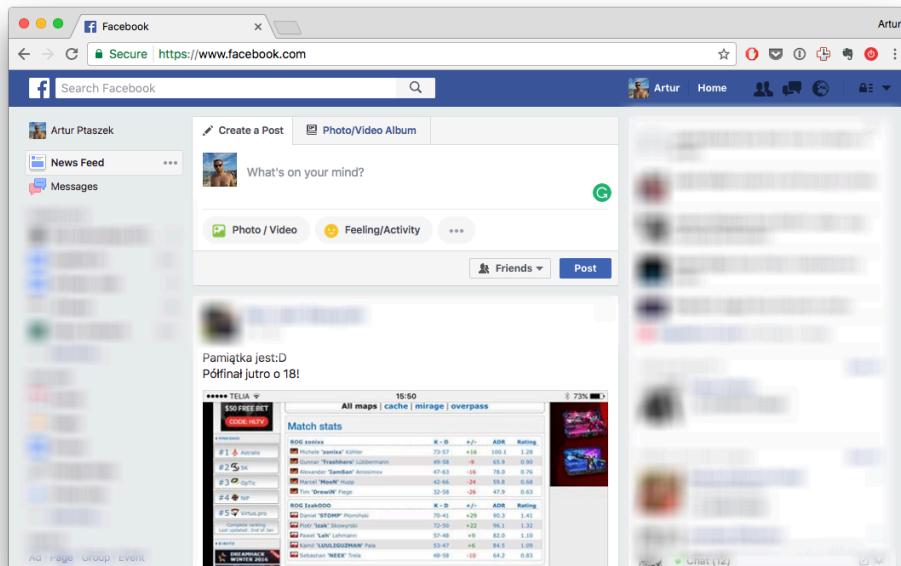


Rys. 4.25 Warstwa widoku listy

Duże kafelki przedstawiające zdjęcia są proste do zrozumienia dla użytkownika. Przejście do kolejnego widoku też nie powinno stanowić problemu dla nawet niedzielnego użytkownika aplikacji internetowych. Podstrona nie korzysta z ciężkich pojęć m. in. ze względu na nikłą ilość tekstu. Jednoznacznie pozwala to na spełnienie kolejnej reguły.

Użytkownik, który znalazł się w tym widoku w każdym momencie może wrócić. Pomyłkowe kliknięcie nie tego znacznika jest bardzo proste do wykonania – wystarczy kliknąć przycisk wstecz w przeglądarce, a znajdzie się w poprzednim widoku. Dodatkowo może pobrać zdjęcie co jest pewnym oznakiem swobody interfejsu. Kliknięcie w kartę powoduje przejście do kolejnego widoku.

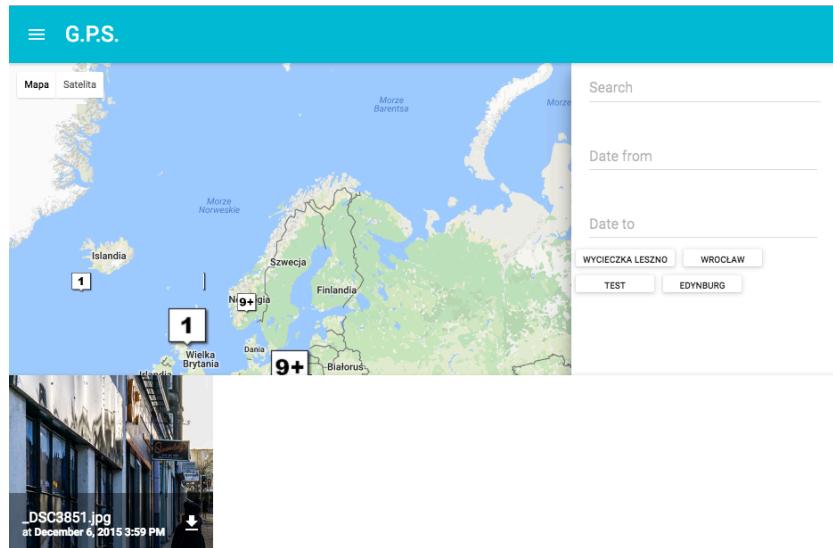
Wspomniany wcześniej wstecz przeglądarkowy to jedna z cech wyróżniających podaną podstronę. Jest to bardzo oczywisty standard w przypadku wielostronnych aplikacji internetowych, jednakże w przypadku SPA obsługa historii przeglądarki jest to dodatkowa funkcjonalność. Został zastosowany nieskończony suwak, który pozwala na doładowywanie kolejnych stron dokumentów. Rozwiążanie to spopularyzowała firma Facebook, która zastosowała na swojej ścianie.



Rys. 4.26 Ściana portalu Facebook

Jest jeden przypadek, w którym użytkownik może doprowadzić do błędu. Otwierając tą podstronę na wielu urządzeniach równocześnie, a następnie usuwając jeden z dokumentów doprowadza do przypadku wyświetlania nieistniejących elementów. Jednak ponowne przejście do tej samej listy odświeża ją co skutkuje usunięciem nieprawidłowych elementów. Powinien zostać wyświetlony komunikat błędu, co się nie dzieje. Więcej na ten temat będzie opisane trochę później.

Użytkownik w zasięgu wzroku posiada cały kontekst. Etykieta, filtry są widoczne na ekranie nawet przy wyświetleniu listy. Jednoznacznie spełnia to kolejną regułę.



Rys. 4.27 Kontekst użytkownika na widoku listy

Widok jest bardzo, ale można doszukać się naruszenia reguły nr 7. Wielki katalog zdjęć powinien być możliwy do posortowania. Jest to bardzo pożądana przez użytkowników funkcjonalność. Strona jest domyślnie sortowana malejąco wg. daty. Umożliwienie tej akcji jest bardzo proste do wykonania i jest bardzo dobrym usprawnieniem przyszłych iteracji projektu.

Przepiękne karty Material Design idealnie sprawdzają się do wyświetlania zdjęć i filmów. Zapewniają minimum funkcjonalności, której użytkownik oczekuje od takiego widoku. Zachowanie minimalizmu spełnia regułę nr 8.



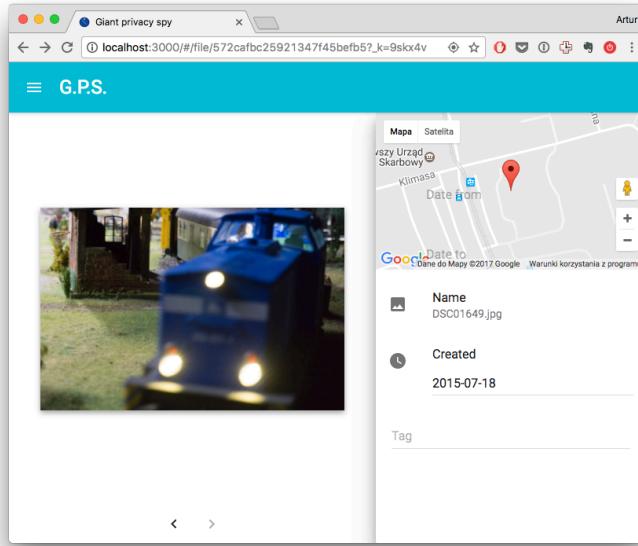
Rys. 4.28 Przykład kart z Material Design

Zostało to wcześniej wspomniane – podstrona nie posiada obsługi błędów. Jednak błędy, które mogą wystąpić są niezwykle rzadkie i jedynie brak połączenia lub sytuacja z obsługą wielu

instancji aplikacji mogą doprowadzić do błędu, który nie będzie zrozumiany przez użytkownika.

4.2.4 Widok szczegółów

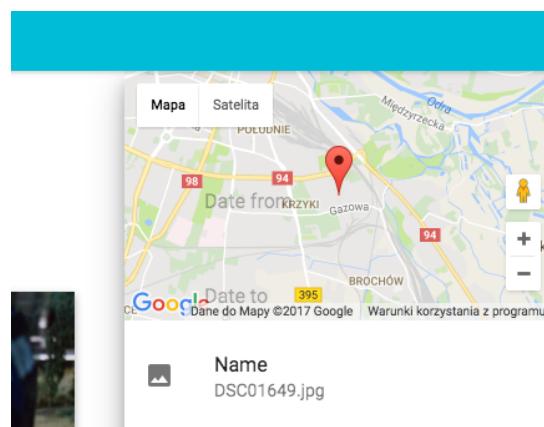
Wyświetlanie statusu na tej podstronie można uznać wyszarzanie strzałek nawigujących po dokumentach. Użytkownik nie jest w stanie przejść do elementu spoza listy, przekłada się to jasną informację dla niego – dalej nic nie ma.



Rys. 4.29 Przykład wyszarzonej strzałki w widoku szczegółów

Podejście to spełnia regułę o wyświetlaniu statusu.

Zgodność z rzeczywistością została zachowana. Informacja o lokalizacji stworzenia pliku jest zaprezentowana na minimapie przedstawiającej lokalizacje stworzenia dokumentu. Chcąc zmienić tą lokalizację po prostu trzeba tą pinezkę przesunąć. System automatycznie zapisuje tą lokalizację, co także jest naturalne dla użytkownika. Przycisk zapisz byłby dodatkową akcją, którą musiałby wykonać – w tym przypadku zbyteczną. Zmiana innych pól odbywa się w podobny sposób.



Rys. 4.30 Minimapą widoku szczegółów

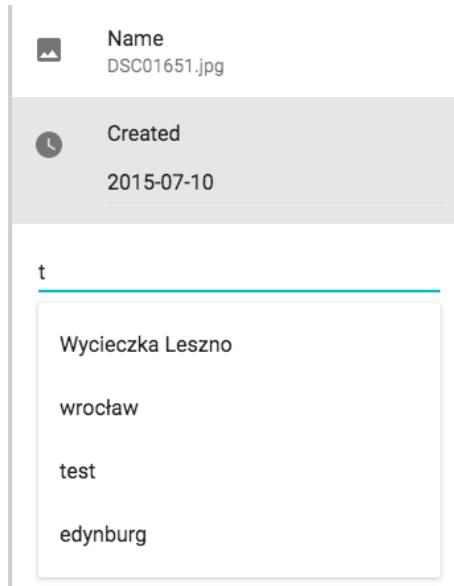
Zmiany wykonane w polach tekstowych są bardzo proste do wycofania przez natywny mechanizm cofnij/powtórz. Inaczej jest to rozwiążane w mapie, gdzie nie jest możliwe łatwe

wycofanie akcji zmiany lokalizacji. Stworzenie takiego mechanizmu nie jest trudne, a byłoby miłym dodatkiem zwiększającym użyteczność całej strony. Dodane etykiety są także proste do usunięcia, przez kliknięcie na samą etykietę. Dla kontrastu akcja usunięcia pliku nie pozwala go przywrócić. Jest to jednak bardzo skrajny przypadek, który nie musi być obsłużony. Warto by było stworzyć komunikat o treści: „Proces usunięcia jest nieodwracalny. Czy jesteś pewien?”. Pozwoli to na uniknięcie błędu, które zostanie opisane podczas analizy reguły traktującej o błędach.

Najważniejszym elementem jest sam dokument. Został on wyeksponowany po lewej stronie i okupuje jak największą przestrzeń. Mniej ważne elementy zaprezentowane zostały po prawej stronie. Standard ten był już poruszany i jest stosowany w całej aplikacji.

Podczas akcji wykonywanych na stronie bardzo ciężko doprowadzić do jakiegokolwiek błędu. Pole edycji daty nie pozwala na wprowadzenie błędnej, a lokalizacja nie jest możliwa do ręcznego wprowadzenia przy pomocy koordynatów. Projekt interfejsu nie zezwala na niepoprawne wprowadzenie jakichkolwiek danych, które wprowadziłyby cały system w niepoprawny stan. Wielokrotnie wspomniany brak Internetu może powodować brak zapisu danych, lecz spowoduje to jedynie nie zapisanie ich. Nie jest to wielki problem, a powiadomianie użytkownika o niepoprawnym żądaniu mogłoby wprawić go w zakłopotanie.

Użytkownik nie musi pamiętać wprowadzonych wcześniej przez siebie etykiet, ponieważ specjalnie dla tej sytuacji został stworzony interfejs. Poniższy zrzut ekranu przedstawia jeden z standardów auto podpowiadania. Stosuje on tą samą technikę wyszukiwania co miało miejsce w przypadku strony głównej. Jednoznacznie wpływa to na użyteczność i pozwala na zaakceptowanie kolejnej reguły Nielsena jako spełnionej.



Rys. 4.31 Interfejs podpowiadania etykiet

Ciążko mówić o elastyczności w zakresie widoku szczegółów, gdy interfejs ma jedno przeznaczenie. Na pewno miłym dodatkiem zwiększającym elastyczność byłoby dodanie możliwości definiowania własnych pól opisujących dokumenty, ale poświęcając prostotę interfejsu. Takie pola musiałyby mieć osobną definicję nazwy, ikony, a także zagłębiania. Koszt takie jest nie do przyjęcia dla aplikacji skupionej na użytkownika. Zwiększenie elastyczności w tym szczególnym przypadku jest nieodpowiednie.

Aplikacja została zaprojektowana tak aby była minimalistyczna. Nie tylko ze względu na oferowane funkcjonalności, lecz sam interfejs. Zastosowanie skomplikowanych algorytmów pozwoliło zmniejszyć ilość kontrolek co bezpośrednio przekłada się na regułę nr 8. Nie inaczej jest w przypadku widoku szczegółów. Ograniczona ilość informacji pozwala skupić uwagę użytkownika na najważniejszym dla tego widoku – dokumencie.

Wspomniane zostało to przy regule piątej, lecz pasuje także do kolejnej poświęconej błędom. System w tym widoku bardzo ciężko wprowadzić w stan złego działania. Błąd z powodu braku połączenia internetowego nie skutkuje dziwnym zachowaniem. Dodanie komunikatu w takim przypadku mogłoby zniechęcać użytkownika, dlatego zostało to świadomie pominięte. Istnieje jednak rozwiązanie nie wyświetlające błędu, ale wyświetlające informacje o braku Internetu. Każda akcja byłaby wtedy kolejkowana i gdy użytkownik znajdzie się w zasięgu połączenia żądania zostałyby ponowione. Informowanie jest lepiej odbierane niż czerwony ekran błędu. Istnieje wiele badań na ten temat.

4.3 Badania według scenariusza

4.3.1 Scenariusze

Badania zostały przeprowadzone wg. określonych scenariuszy. Badani zostali postawieni przed następującymi zadaniami:

1. Zarejestrować konto w aplikacji
2. Wylogować się
3. Zalogować się
4. Dodać ścieżkę do aplikacji
5. Dodać dokument
6. Znaleźć dodany dokument
7. Pogrupować zdjęcia w album
8. Usunąć dokument
9. Pobrać dokument
10. Zmodyfikować właściwości dokumentu nazwę i lokalizację

Każdy scenariusz zostanie opisany przez dwie osoby – badanego i badającego. Badany będzie omawiał wykonywane przez niego akcje i czego poszukuje, a badający opisze błędy wykonywane przez badanego błędy. Pod koniec każdego scenariusza zostanie podsumowane wykonane badanie i jakie wartości niosą one dla aplikacji.

Badanie zostało przeprowadzone na grupie 3 osób z różnych grup społecznych, aby badanie było bardziej wiarygodne. Badanie zostało przeprowadzone na komputerze w przeglądarce Chrome. Początkowo każdej osobie zostało przedstawiony cel badania, przeznaczenie aplikacji i długość badania, która miała nie przekraczać 1h 30m.

4.3.2 Badana nr 1

Studentka lat 25. Osoba obeznana z technologią i Internetem. Wcześniej nie korzystała z aplikacji oferującej podobne funkcjonalności. Dodatkowo nie posiada problemu ze zrozumieniem języka angielskiego, który jest podstawowym językiem aplikacji.

4.3.2.1 Rejestracja konta

Po przejściu na stronę logowania osoba bez problemu odnalazła funkcjonalność rejestracji. Bardzo mile została zaskoczona, że od razu po rejestracji ukazał się ekran główny

aplikacji i mogła z niego korzystać bez potwierdzania za pomocą maila. Wrażenia z tej funkcjonalności były bardzo wysokie. Interfejs został opisany jako bardzo prosty i przejrzysty, a uroku dodawało wielkie zdobycie tła, które umilało odczucie w aplikacji.

4.3.2.2 Wylogowanie się

Czynność ta zajęła badanej trochę więcej czasu. Poszukiwała funkcjonalności pierwsze w prawym górnym rogu: „Zawsze ta opcja tam jest”. Po chwili jednak ujrzała menu hamburger, które rozsuwało półkę i odkryła, że tam jest przycisk wylogowywania.

4.3.2.3 Logowanie do aplikacji

Ekran logowania jest widoczny jako pierwszy, więc odnalezienie tej funkcjonalności nie sprawiało problemów. Badana nie skorzystała z opcji zaloguj przy pomocy myszy, lecz przycisk „enter” na klawiaturze. Aplikacja była zaprojektowana tak, aby obsługiwać taki sposób wysyłki formularza i nie było najmniejszego problemu z wykonaniem tego zadania.

4.3.2.4 Dodanie ścieżki

Badana bardzo szybko odnalazła przycisk dodawania, gdzie dostępna jest opcja dodawania ścieżki, ale nie wiedziała co za plik ma tam umieścić. Całe badanie mogłoby się zakończyć na tym punkcie. Badający podał odpowiedź, która wskazywała na usługi lokalizacji Google. Osoba starała się wyszukać w wyszukiwarce więcej informacji na ten temat, jednak i tak bezskutecznie, ponieważ nie ma włączonych opcji lokalizacji na telefonie. Ostatecznie badana nie była w stanie wykonać tego zadania

4.3.2.5 Dodanie dokumentu

Zadanie zostało wykonane błyskawicznie, ponieważ opcja była widoczna już podczas poprzedniego scenariusza. Wybrała pierwszy plik graficzny, a następnie poczekała na wysłanie pliku graficznego. Zauważyła wskaźnik ładowania i cierpliwie czekała na ładowanie. Badana zauważyła, że miłym dodatkiem byłby wskaźnik postępu, ponieważ nie wiedziała, ile jeszcze potrwa dodawanie. Po chwili ukazał się komunikat o poprawnym dodaniu pliku. Osoba podkreśliła, że bardzo podoba jej się sposób organizacji tego menu, ponieważ zna to zachowanie z systemu Android i było to dla niej bardzo naturalne, gdzie poszukiwać tej opcji.

4.3.2.6 Odnalezienie dodanego dokumentu

Badana zaraz po dodaniu zauważyła animację etykiety na mapie i odruchowo na nią kliknęła. Dodała, że ruch etykiety pozwolił jej w prostszy sposób zauważyc co się stało. Jednoznacznie zadanie zakończyło się.

4.3.2.7 Grupowanie w albumy

Początkowo nie było to oczywiste dla badanej, lecz po 5 minutach zdecydowała się dodać etykiety do poprzednio dodanych dokumentów. Badana stwierdziła, że powinno się to nazywać bezpośrednio albumami, a etykiety to jest dodatek do takiej kategoryzacji. Osoba także stwierdziła, że dodanie kilku zdjęć do albumu jest uciążliwe i powinny być dostępne opcje zmiany na wielu elementach.

4.3.2.8 Usunięcie dokumentu

Zadanie to nie stanowiło dla badanej osoby problemu. Od razu przeszła do podglądu pliku, a następnie wyszukała ikony kosza. Przypadkowo kliknęła 2 razy ikonę kosza, co powodowało usunięcie zdjęcia, którego nie chciała usunąć. Jest to na pewno pole do usprawnień.

4.3.2.9 Pobranie dokumentu

Podobnie jak w przypadku poprzednim badana wiedziała, gdzie skierować swoją uwagę, ponieważ już wcześniej widziała tą opcję. Ikona pobierania była identyczna, do tej z systemu Android stosowanej w przypadku pobierania.

4.3.2.10 Zmiana właściwości dokumentu: nazwy i lokalizacji

Badana od razu skierowała swoją uwagę na stronę szczegółów dokumentu. Następnie przez długi czas poszukiwała przycisku edytuj, którego nie potrafiła odnaleźć. Następnie kliknęła w nazwę dokumentu, a następnie zauważała migający kursor sugerujący edycję tekstu. Od razu zrozumiała, że tutaj może wyedytować właściwości bez potrzeby przechodzenia w osobny tryb. Dodatkowo osoba zgodnie z zamierzeniem zrozumiała, iż nie jest potrzebny zapis tych właściwości, ponieważ wykonuje się to w tle. Zadanie edycji lokalizacji także nie stanowiło problemów, ponieważ badana od razu przesunęła pinezkę. Dodała, że warto dodać wyszukiwarkę lokalizacji przy jej edycji.

4.3.3 Badana nr 2

Uczennica lat 15. Zna język angielski, a także większość swojego czasu spędza przy komputerze. Codziennie korzysta z systemu Android. Obeznana z nowymi technologiami, co czyni ją idealną osobą do badań.

4.3.3.1 Rejestracja konta

Bez problemu odnalazła opcję, a następnie zarejestrowała się przy pomocy znalezionej formularza także została miło zaskoczona natychmiastowym przejściem do aplikacji bez potrzeby potwierdzenia mailowego.

4.3.3.2 Wylogowanie się

Zadanie nie sprawiło jej najmniejszych problemów. Doszła do wniosku, że jak wygląda to jak interfejs aplikacji na systemie Android to musi być to rozwiązane podobnie do innych aplikacji zbudowanych na tym samym interfejsie. Odnalazła hamburger menu i tam już czekała na nią opcja wylogowywania.

4.3.3.3 Logowanie do aplikacji

Kolejne zadanie, które nie sprawiło badanej problemów. Wpisała podane dane formularza, a następnie lewym przyciskiem wybrała opcję zaloguj.

4.3.3.4 Dodanie ścieżki

Podobnie jak w przypadku poprzedniej badanej, osoba nie była w stanie odnaleźć formatu, w którym powinna przesłać do aplikacji. W przeciwnieństwie jednak do poprzedniej badanej po podaniu podpowiedzi badana zaczęła szukać informacji o lokalizacji Google i odnalazła, jak wyeksportować te dane do pliku. Następnie badana zadała pytanie: „Czy za każdym razem będę musiała to dodawać?”. Sugeruje to, aby w przyszłości zautomatyzować proces importu danych lokalizacyjnych z Google. Cały scenariusz trwał około 25 minut, na co składało się oczekiwanie na wyeksportowane dane, a także na poprawny import. Trwało to zdecydowanie za długo.

4.3.3.5 Dodanie dokumentu

Podobnie jak w poprzednim przypadku zadanie wykonane zostało błyskawicznie. Badana przeszła do płynącego przycisku akcji, a następnie wybrała dodanie pliku. Żadna z akcji nie była dla niej niezrozumiałą.

4.3.3.6 Odnalezienie dodanego dokumentu

Ten przypadek nie zostaje odosobniony. Badana nie miała najmniejszego problemu z odnalezieniem dokumentu. Animacja pinezki pozwoliła odnaleźć jej plik, który właśnie dodała.

4.3.3.7 Grupowanie w albumy

Osoba po chwili zauważała, że w widoku szczegółów jest słowo kluczowe „tag”. Postanowiła przy pomocy tego narzędzia zorganizować swój album zdjęć. Bardzo przypomina jej to organizację zdjęć na portalu, z którego korzysta na co dzień: Instagram. Podejście jest bardzo podobne i dla osób młodych bardzo intuicyjne.

4.3.3.8 Usunięcie dokumentu

Tak samo jak w przypadku poprzedniej osoby znalezienie tej opcji było bardzo proste. Zadanie zostało wykonane błyskawicznie.

4.3.3.9 Pobranie dokumentu

Funkcjonalność pobierania dokumentu jest umiejscowiona zaraz obok funkcjonalności usuwania. Ikona pobierania jest bardzo intuicyjna, a także znana szczególnie przez użytkowników systemu Android. Zadanie dla badanej nie stanowiło najmniejszego problemu.

4.3.3.10 Zmiana właściwości dokumentu: nazwy i lokalizacji

Podobnie jak w przypadku poprzedniej badanej, zadanie to sprawiło odrobinę problemu. Jednak badana poradziła sobie szybciej.

4.3.4 Badany nr 3

Mężczyzna lat 48. Korzysta z systemu Android i zna język angielski. Nie posiada doświadczenia z podobnymi aplikacjami.

4.3.4.1 Rejestracja konta

Wszyscy badani razem z osobą nr 3 poradzili sobie wyśmienicie z rejestracją w aplikacji. Badany po prostu przeszedł do wymaganej opcji.

4.3.4.2 Wylogowanie się

Zadanie to osobie sprawiło bardzo dużo problemów. Badany nie mógł odnaleźć opcji przez 5 minut. Szukał jej przede wszystkim w prawym górnym rogu, co zwyczajowo jest umieszczone na większości portali internetowych.

4.3.4.3 Logowanie do aplikacji

Zadanie okazało się prostolinijskie. Badany się po prostu zalogował danymi podanymi przy rejestracji.

4.3.4.4 Dodanie ścieżki

Badanemu nawet po podaniu podpowiedzi nie udało się dodać ścieżki do aplikacji. Odnalazł on miejsce, w którym kryło się dodawanie, lecz ciężko było mu znaleźć co ma tam dodać.

4.3.4.5 Dodanie dokumentu

Zadanie wykonane bezbłędnie. Badany wypróbował, czy można dodać kilka zdjęć na raz – z powodzeniem.

4.3.4.6 Odnalezienie dodanego dokumentu

Badany jako pierwszy posiadał problemy z odnalezieniem dodanego pliku. Okazało się, że nie zauważył animacji, która była kluczowa dla poprzednich uczestniczek. Jednakże po spojrzeniu na mapę badany zauważył, że pokazała się pinezka z ilością zdjęć, które poprzednio dodał.

4.3.4.7 Grupowanie w albumy

Uczestnik badania nie poradził sobie z tym zadaniem. Nie potrafił zrozumieć funkcjonalności etykietowania, która miała do tego służyć. Poszukiwał słowa kluczowego album lub kolekcja.

4.3.4.8 Usunięcie dokumentu

Badany próbował początkowo zaznaczyć etykietę i nacisnąć przycisk „delete” na klawiaturze, jednakże bez powodzenia. Po ujrzeniu widoku listy starał się także zaznaczyć elementy i tam nacisnąć ten sam klawisz, także bez powodzenia. Przejście do widoku szczegółów pozwoliło mu dogłębnym przejrzeniu dostępnych opcji – ostateczne usunięcie pliku. Cała operacja trwała około 6 minut.

4.3.4.9 Pobranie dokumentu

Kroki jakie badany poczynił w przypadku poprzednim pozwoliły mu na bardzo szybkie odnalezienie funkcjonalności, o której mowa w tym scenariuszu.

4.3.4.10 Zmiana właściwości dokumentu: nazwy i lokalizacji

Zmiana lokalizacji nie była zrozumiała dla badanego, ponieważ nie odkrył w jaki sposób ma ją zmienić. Także bardzo dużo problemów miał z edycją nazwy. Poszukiwał ikony kredki, która oznacza edycję jednak po bardzo długim poszukiwaniu tego elementu postanowił kliknąć w nazwę. Jego oczom ukazał się kurSOR edycji tekstu. Dalsze kroki były dla niego zrozumiałe.

5 Ocena metod

5.1 Heurystyki

Heurystyki Jakoba Nielsena są niezwykle potężnym narzędziem. Analizując 10 punktów jest się w stanie bardzo prosty sposób usprawnić aplikacje. Nie jest to jednak narzędzie bez wad, co zostanie przedstawione poniżej.

5.1.1 Usystematyzowanie

Metoda ta w bardzo prosty sposób określa jakie kroki podjąć, żeby przeanalizować aplikację. Dzieląc aplikację na ekrany, a następnie je analizując wg. podanych reguł jest się w stanie ocenić poziom użyteczności w aplikacji. Nie trzeba przeprowadzać szczegółowych badań w Google Analytics czy przy pomocy grup fokusowych. Rezultaty są widoczne od razu i jest określony sposób w jaki powinno się postępować. Korzystając z tego sposobu także łatwiej projektować sam interfejs.

5.1.2 Niski koszt

Rozpoczęcie prac nad analizą przy pomocy heurystyk jest bardzo tanie. Nie trzeba korzystać z drogich narzędzi jak w przypadku metody śledzenia gałek ocznych czy Google Analytics, który do wydajnej obsługi wymaga wyspecjalizowanej osoby i modyfikacji w badanym obiekcie. Koszt jaki się ponosi jest to jedynie czas pracy specjalisty od użyteczności.

5.1.3 Natychmiastowe wyniki

Wyniki są widziane od razu po samej analizie. Czas potrzebny na otrzymanie informacji zwrotnej przy użyciu innych metod jest relatywnie wysoki. Przeprowadzenie badań w grupach fokusowych trwa długie dni przygotowań, a następnie samych badań. Później przychodzi czas na analizę, która także trwa dużą ilość czasu. Badanie przy pomocy Google Analytics wymaga większej próbki danych, która w przypadku mniejszych serwisów jest ciężka do kolekcji.

5.1.4 Odpowiednia dla każdego projektu

Konkurencyjne metody zawodzą w przypadku systemów, które nie są upublicznione lub są to aplikacje wewnętrzne. Przypadek ten jest bardzo ciężki do badania przy pomocy zewnętrznych narzędzi tj. Google Analytics, Hotjar czy CrazyEgg. Często też polityka firmy nie zezwala na przeprowadzenie badań w grupach fokusowych. Jednak heurystyki w takich środowiskach sprawdzają się idealnie.

5.1.5 Dobra metoda startowa

Zostało wspomniane już, że inne metody badań są bardzo kosztowne co niesie ze sobą wiele wad. Audyt systemu pod względem użyteczności warto rozpocząć od heurystyk, aby sprawdzić w jakiej kondycji jest cały system. Jednak to dopiero początek. E-commerce wymaga metod o optymalizacji o wiele większej niż to co oferują heurystyki. Badane są kolory elementów, ich położenie. Pozwala to osiągnąć zdecydowanie lepsze wyniki konwersji niż może zaoferować heurystyka.

5.1.6 Zasady są bardzo ogólne

Wielkim minusem metody jest przede wszystkim bardzo ogólne podejście do tematu. Bardzo dużą rolę odgrywa rola audytora i jego doświadczenie w branży użyteczności. Dodatkowo liczy się jego specjalizacja. Inaczej podchodzi się do systemów e-commerce, a całkiem innym podejściem trzeba się kierować w systemach ERP. Heurystyki nie podają w

jaki sposób rozwiązać bardzo złożone problemy np. w rozbudowanych formularzach. Bez dodatkowych badań nie jest się w stanie stworzyć bardzo dobrego interfejsu.

5.2 Badanie wg. scenariusza

5.2.1 Wizja osób korzystających z aplikacji

Badając tą metodą ma się do czynienia z realnymi użytkownikami systemu, co bardziej pozwala dostosować aplikację do nich. Dodatkowo po określeniu grupy docelowej, można powtarzać scenariusze wielokrotnie, aby sprawdzić poprawność rozwiązania na użytkownikach rozwijanego serwisu.

5.2.2 Wymagane przygotowania i znalezienie osób chętnych

Aby przeprowadzić badania trzeba przygotować scenariusze i aplikację do potencjalnych testów. Dodatkowym wyzwaniem może być znalezienie osób do badania, ponieważ trzeba odnaleźć osoby, które wpasują się w grupę docelową aplikacji. Mając tak wiele reguł do spełnienia, może się okazać, że koszt takiego przedsięwzięcia jest dosyć wysoki.

5.2.3 Polityka firmy może nie pozwalać na wykonanie takich badań

Firmy, które pracują dla rządu lub firm, których polityka nie pozwala na przeprowadzenie takich badań nie mogą skorzystać z dobrodziejstw tej metody. Także czasami umowy jakie trzeba podpisać, aby móc przystąpić do takiego badania mogą odstraszyć potencjalnych badanych.

5.2.4 Problem ze znalezieniem badanych

Przypadek, w którym projektujemy aplikację dla użytkowników przykładowo naukowych jest ciężkim do optymalizacji pod względem badań. Terminologia, której używają środowiska do wykonywania swojej pracy jest za ciężka dla przeciętnego człowieka. Problemem tutaj jest zbyt wąska grupa docelowa, która nie pozwala na odnalezienie ludzi do badania użyteczności.

6 Wnioski

Heurystyki pozwoliły znaleźć wiele kardynalnych błędów interfejsu. Wiele z nich są to bardzo proste do wykonania poprawki, które nie niosą za sobą przebudowania całych widoków. Jednym z nich jest dodanie instrukcji, która pomoże z trudniejszymi elementami interfejsu. Wielokrotnie została poruszona obsługa braku połączenia internetowego. Technologia pozwala w łatwy sposób udostępnić aplikację nawet będąc offline. Aplikacja jest dostępna na telefony, co często wiąże się z utraconym połączeniem lub bardzo słabą jego jakością. Wiele miejsc nie posiadało obsługi błędów, co definitywnie powinno zostać poprawione.

Jednak podczas normalnego użytkowania większość z powyższych błędów nie wystąpi. Pomimo tego użyteczność aplikacji jest daleka od ideału. Metoda ta nie daje tak wymiernych korzyści jak inne przedstawione metody. Kolejnym etapem rozwoju oprogramowania powinno zdefiniować się grupę docelową aplikacji, a następnie przeprowadzać na niej liczne badania na aplikacji. Podniosłoby to użyteczność w wielkim stopniu.

Scenariusze przedstawiły zgoła inne wyniki. Problemy, które zostały wyspecyfikowane podczas heurystyk nie wystąpiły przy badaniach na użytkownikach. Dla kontrastu ukazało się także wiele błędów, które nie zostały odnalezione podczas heurystyk np. dodawanie ścieżki. Użytkownicy myślą zupełnie inaczej niż twórca aplikacji i zdecydowanie podnosi to obiektywność testów. Dodatkowo warto zdefiniować grupę docelową systemu. Badania zostały przeprowadzone na ludziach z różnych grup wiekowych co pokazało różnice w ich toku rozumowania.

Obie metody pozwoliły na odnalezienie problemów w aplikacji. Patrząc w przyszłość warto by było poprawić najpierw wszystkie krytyczne błędy użyteczności tj. dodawanie ścieżek czy edycja właściwości zanim zacznie się dodawać nowe. Także użyteczność jest bardzo ważnym elementem rozwoju oprogramowania, które pozwoli na uzyskanie lepszego odbioru. Użyteczność ostatecznie także przekłada się na sukces biznesowy przedsięwzięcia, ponieważ przyjemna w użytkowaniu aplikacja zostanie wybrana spośród innych konkurencyjnych rozwiązań.

Dodatkowo decyzja o wykorzystaniu map do prezentacji grup dokumentów była kluczowym elementem do podniesienia użyteczności aplikacji. Badani nie mieli problemów ze zrozumieniem charakterystyki aplikacji ani z nawigacją po samej mapie. Google Maps dla każdego z badanych był znaną technologią, a wykorzystanie jej dodatkowo rozjaśniało kwestie grupowania elementów. Jednoznacznie można stwierdzić, że przy pomocy map i historii lokalizacji można tworzyć systemy przyjazne użytkownikowi.

7 Spis tabel

Tabela 1.1 Definicja znaków w transformacji na system o podstawie 32 21

8 Spis rysunków

Rys. 1.1 Przykładowy plik GeoJSON.....	13
Rys. 1.2 Przykładowy plik KML	14
Rys. 1.3 Przykładowy plik GPX	14
Rys. 1.4 Przykładowe dane EXIF	16
Rys. 1.5 Dworzec główny przy mniejszym powiększeniu	17
Rys. 1.6 Dworzec główny przy większym powiększeniu.....	17
Rys. 1.7 Indeks książki "Nauka masażu w Weekend" Nitya Lacroix	18
Rys. 1.8 Bitowa reprezentacja długości geograficznej przy precyzji 1	20
Rys. 1.9 Bitowa reprezentacja długości geograficznej przy precyzji 2	21
Rys. 1.10 Przykładowe zapytanie grupujące elementy na mapie	22
Rys. 2.1 Okno główne Google Maps	24
Rys. 2.2 Okno główne aplikacji Apple Maps	26
Rys. 2.3 Okno główne aplikacji Yandex Maps.....	26
Rys. 2.4 Okno główne aplikacji Google Earth	27
Rys. 2.5 Okno główne aplikacji Bing Maps	28
Rys. 2.6 Okno główne aplikacji Open Street Maps	28
Rys. 3.1 Prezentacja przepływu danych przy pomocy ciasteczek	31
Rys. 3.2 Prezentacja przepływu danych z wykorzystaniem tokenu	32
Rys. 3.3 Zrzut z Chrome DevTools pokazujący czas potrzebny na dekodowanie obrazka JPEG w rozdzielczości 4956x3304.....	33
Rys. 3.4 Zrzut z Chrome DevTools pokazujący czas potrzebny na dekodowanie obrazka JPEG w rozdzielczości 1920x1080.....	33
Rys. 3.5 Ekran logowania	37
Rys. 3.6 Ekran główny aplikacji	38
Rys. 3.7 Menu dodawania.....	39
Rys. 3.8 Rozwinięte menu dodawania	39
Rys. 3.9 Szuflada aplikacji.....	40
Rys. 3.10 Widok listy	41
Rys. 3.11 Widok szczegółów	41
Rys. 3.12 Moduł mapy programu Adobe Lightroom	42
Rys. 3.13 Aplikacja Zdjęcia Google	43
Rys. 3.14 Widok główny aplikacji Zdjęcia Apple	44
Rys. 4.1 Przykład strony produktu serwisu Amazon.....	46
Rys. 4.2 Przykład strony produktu serwisu Allegro	46
Rys. 4.3 Przykład strony produktu serwisu Ceneo	46
Rys. 4.4 Wskaźnik ładowania.....	47
Rys. 4.5 Przykład usuwania aplikacji na systemie macOS.....	48
Rys. 4.6 Przykład usuwania aplikacji na systemie WindowsDaj użytkownikowi pełną kontrolę	48
Rys. 4.7 Widok aplikacji GMail na systemie Android	49
Rys. 4.8 Widok na "Navigation Drawer" z Material Design	50
Rys. 4.9 Korekta zastosowana przez wyszukiwarkę Google.....	51
Rys. 4.10 Przykładowa mapa termiczna CrazyEgg	52
Rys. 4.11 Wskaźnik ładowania na stronie logowania.....	53
Rys. 4.12 Strona logowania Google	53
Rys. 4.13 Ekran logowania Pinterest	54
Rys. 4.14 Walidacja pola użytkownika.....	54
Rys. 4.15 Walidacja hasła.....	54

Rys. 4.16 Komunikat błędu podczas niepoprawnych danych logowania.....	55
Rys. 4.17 Wskaźnik lądowania podczas dodawania plików.....	55
Rys. 4.18 Informacje o użytkowniku na półce	55
Rys. 4.19 Przykład zgrupowanej etykiety.....	56
Rys. 4.20 Możliwość wycofania operacji wysłania email.....	56
Rys. 4.21 Efekt plusku.....	57
Rys. 4.22 Stan aplikacja przy braku połączenia internetowego.....	57
Rys. 4.23 Filtry aplikacji.....	57
Rys. 4.24 Wyszukiwarka Google.....	58
Rys. 4.25 Warstwa widoku listy	59
Rys. 4.26 Ściana portalu Facebook.....	59
Rys. 4.27 Kontekst użytkownika na widoku listy.....	60
Rys. 4.28 Przykład kart z Material Design	60
Rys. 4.29 Przykład wyszarzonej strzałki w widoku szczegółów	61
Rys. 4.30 Minimapą widoku szczegółów	61
Rys. 4.31 Interfejs podpowiadania etykiet.....	62

9 Bibliografia

- [1] Wikimedia Foundation Inc., „Global Positioning System,” [Online]. Available: https://en.wikipedia.org/wiki/Global_Positioning_System.
- [2] SAE International, „Automated driving,” [Online]. Available: https://www.sae.org/misc/pdfs/automated_driving.pdf.
- [3] Wikimedia Foundation Inc., „Autonomous car,” [Online]. Available: https://en.wikipedia.org/wiki/Autonomous_car.
- [4] „Wikipedia,” 17 Styczeń 2017. [Online]. Available: https://en.wikipedia.org/wiki/Google_Maps.
- [5] M. Traczyk, „Uxeria,” 30 Lipiec 2016. [Online]. Available: <http://blog.uxeria.com/metody-i-rodzaje-badan-uzytecznosci/>.
- [6] M. Beasley, UX i analiza ruchu w sieci. Praktyczny poradnik, Gliwice: Helion, 2014.
- [7] S. Krug, Nie każ mi myśleć! O życiowym podejściu do funkcjonalności stron internetowych., 3 red., Gliwice: Helion, 2014.
- [8] J. Nielsen, „Nielsen Norman Group,” 4 Styczeń 2012. [Online]. Available: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>.
- [9] J. Nielsen, „10 Usability Heuristics for User Interface Design,” 1 Styczeń 1995. [Online]. Available: <https://www.nngroup.com/articles/ten-usability-heuristics/>.