



Politechnika Wrocławska

Wydział Informatyki i Zarządzania

Kierunek studiów: Informatyka

Specjalność: Internetowe Technologie Informacyjne

Praca dyplomowa - magisterska

**PROJEKTOWANIE I IMPLEMENTACJA
APLIKACJI MOBILNYCH W
ZASTOSOWANIACH BIZNESOWYCH**

Zygmunt Horodyski

Słowa kluczowe:

Aplikacja mobilna, projektowanie, Implementacja

Inżynieria oprogramowania, Android OS

Krótkie streszczenie:

Poniższa praca porusza temat procesu projektowania i implementacji mobilnych aplikacji biznesowych. Efektem końcowym pracy jest zbór reguł i zaleceń pomocnych w tworzeniu oprogramowania na urządzenia przenośne.

opiekun pracy
dyplomowej	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 Instytutu, w którym
student wykonywał pracę

Wrocław 2014

Spis treści

Wstęp.....	4
1. Zalety mobilności w aplikacjach na urządzenia przenośne	5
1.1. Przechwytywanie danych w czasie rzeczywistym	5
1.2. Praca offline.....	6
1.3. Skrócony czas reakcji	6
1.4. Szybki i łatwy do użycia.....	6
2. Zdefiniowanie mobilnych aplikacji biznesowych.....	7
3. Wybór systemu operacyjnego dla mobilnej aplikacji biznesowej	8
3.1. Dostępność.....	8
3.2. Koszt aplikacji	9
3.2.1. Koszt licencji do tworzenia aplikacji	10
3.2.2. Koszt zakupu oprogramowania.....	10
3.2.3. Koszt produkcji	10
3.2.4. Koszt wdrożenia i utrzymania.....	11
3.2.5. Zestawienie kosztów	11
4. Wyzwania w projektowaniu mobilnych aplikacji biznesowych	12
4.1. Projektowanie w kontekście użyteczności	12
4.2. Projektowanie w kontekście prezentowanej treści	13
5. Przegląd współczesnych mobilnych aplikacji biznesowych.....	15
5.1. Komunikaty	15
5.2. Logowanie	19
5.3. Ładowanie treści.....	22
5.4. Menu.....	25
5.5. Nawigacja	29
5.6. Reklamy.....	31
5.7. Rejestracja	37
5.8. Social media	40
5.9. Splashscreen	42
5.10. Ustawienia.....	44
5.11. Wielkość przycisków	46
5.12. Wprowadzanie danych.....	49

6.	Analiza przeglądu mobilnych aplikacji biznesowych	50
6.1.	Komunikaty	50
6.2.	Logowanie	51
6.3.	Ładowanie treści.....	51
6.4.	Menu.....	52
6.5.	Nawigacja	53
6.6.	Reklamy.....	54
6.7.	Rejestracja	54
6.8.	Social media	55
6.9.	Splashscreen	56
6.10.	Ustawienia.....	56
6.11.	Wprowadzanie danych.....	57
6.12.	Podsumowanie postawionych tez	58
7.	Projektowanie mobilnej aplikacji biznesowej	59
7.1.	Badanie użyteczności wybranych narzędzi do prototypowania	59
7.1.1.	FluidUI	59
7.1.2.	NinjaMock.....	60
7.1.3.	Indigo Studio	61
7.1.4.	Podsumowanie badania	62
7.2.	Zdefiniowanie projektowanej aplikacji	63
7.3.	Projektowanie aplikacji w oparciu o postawione tezy.....	64
8.	Implementacja mobilnej aplikacji biznesowej	72
8.1.	Badanie frameworków implementowania bazy danych	72
8.1.1.	Natywny SQLite	72
8.1.2.	OMRLite	74
8.1.3.	GreenDAO	76
8.1.4.	Podsumowanie badania	76
8.2.	Badanie sposobów budowania produktu	77
8.2.1.	Ręczne budowanie produktu	77
8.2.2.	Ant	78
8.2.3.	Maven.....	78
8.2.4.	Gradle	79
8.2.5.	Podsumowanie badania	80

9.	Prezentacja i porównanie utworzonej aplikacji serwisu napwr.pl	81
10.	Podsumowanie	86
11.	Źródła	88
12.	Wykorzystane aplikacje mobilne	91

Wstęp

Poniższa praca poświęcona będzie analizie problemów projektowania i implementacji mobilnych aplikacji biznesowych.

Na potrzeby pracy sformułowano definicję mobilnych aplikacji biznesowych. Zgodnie z tą definicją jest to oprogramowanie eksploatowane na urządzeniach mobilnych, umożliwiające wykonywanie różnych funkcji biznesowych, które przynoszą bezpośrednią lub pośrednią korzyść finansową klientowi. Są to np. takie aplikacje jak Allegro - umożliwiające kupno towaru, iPKO - do zarządzania swoim kontem bankowym, czy też Onet News, pozwalające na szybkie reagowanie dzięki uzyskanym informacjom.

W pracy zdefiniowane zostaną reguły i zalecenia odnoszące się do projektowania i implementowania mobilnych aplikacji biznesowych. Ponadto celem pracy jest omówienie problemów związanych z projektowaniem oprogramowania mobilnego oraz zbadanie relacji pomiędzy systemem operacyjnym, pod kontrolą którego będzie działała aplikacja, a finalną postacią tej mobilnej aplikacji biznesowej.

Aby osiągnąć te cele, zostanie przeprowadzony przegląd mobilnych aplikacji biznesowych oraz ich analiza, która pozwoli wyznaczyć aktualne trendy i najlepsze rozwiązania z dziedziny projektowania biznesowych aplikacji mobilnych. Zostaną także przeprowadzone badania oceniające narzędzia do prototypowania, zarządzania bazami danych oraz budowania produktu.

Ostatnim etapem pracy będzie badanie poprawności i przydatności zaproponowanych reguł i zaleceń. W oparciu o wspomniane reguły i zalecenia, skonstruowana zostanie mobilna aplikacja biznesowa, realizująca funkcje zdefiniowane i zaimplementowane w wykonanej wcześniej aplikacji, traktowanej jako punkt odniesienia w przeprowadzonych w pracy badaniach. „Wcześniejsza” aplikacja została skonstruowana bez zamierzonego stosowania zdefiniowanych w pracy reguł i zaleceń. Porównanie istotnych dla użytkownika cech tych dwóch aplikacji stanowi podstawę badań użyteczności opracowanego zbioru reguł i zaleceń i pozwoli na wyciągnięcie wniosków.

1. Zalety mobilności w aplikacjach na urządzenia przenośne

Aplikacja mobilna jest to kod zaprojektowany i utworzony z przeznaczeniem na urządzenia przenośne, takie jak smartfony i tablety. Ich zadaniem jest polepszenie cech przenośnych urządzenia poprzez dostarczanie dodatkowych funkcjonalności, które zwiększą przydatność urządzenia. [2]

Aplikacje mobilne różnią się od zintegrowanych aplikacji, które znajdują się na komputerach, tym, że każda aplikacja mobilna dostarcza limitowaną i izolowaną funkcjonalność, taką jak np. gra, kalkulator czy przeglądarka internetowa. [3]

Jednym z pierwszych etapów planowania wdrożenia mobilnej aplikacji jest rozważenie możliwych korzyści wynikających z użycia rozwiązań mobilnych. Bardzo ważnym jest na tym etapie zdefiniowanie jaką wartość dodaną wynikać będzie z faktu stworzenia mobilnej aplikacji. Będzie to stanowiło przedmiot omawiany w tym rozdziale.

1.1. Przechwytywanie danych w czasie rzeczywistym

Przetwarzanie danych jest bardzo powolne jeśli zachodzi potrzeba ręcznego gromadzenia informacji i wprowadzenia ich do systemu desktopowego. Użycie mobilnej aplikacji dostarcza możliwość przechwytywania potrzebnych treści w czasie rzeczywistym. Mogą to być teksty, numery, zdjęcia, lokalizacje GPS, podpisy i wiele innych. [30]



Przykład skanowania towarów w celu znalezienia konkurencyjnych ofert [32]

Przykładem dobrze wyjaśniającym ten aspekt są skanery kodów kreskowych w celu sprawdzenia ceny danego produktu w innych sklepach, przykładem takiego rozwiązania na polskim rynku jest mobilna aplikacja serwisu Ceneo. [33]

Osoby handlujące na giełdzie poprzez odpowiednie oprogramowanie na urządzenia przenośne zyskują możliwość szybkiego reagowania na zmieniające się kursy walut i posiadanych akcji. Dzięki czemu mogą wyprzedzić konkurencję co może się okazać kluczowe w odniesieniu sukcesu.

1.2. Praca offline

Ogromną zaletą jest możliwość dostępu w każdym miejscu bez dostępu do połączenia internetowego. Mobilna aplikacja może automatycznie synchronizować dane w tle i zapisywać je. Pozwala to na korzystanie z zalet mobilnych rozwiązań bez aktywnego połączenia z siecią. [30]

1.3. Skrócony czas reakcji

Rozwiążanie problemu klienta może zajść sporą ilość czasu jeśli ma miejsce brak łatwej komunikacji, może to spowodować spore dodatkowe koszty i opóźnienia. Przepływ danych jaki oferują rozwiązania mobilne umożliwia szybkie i profesjonalne rozwiązanie problemu. [31]



Aplikacja TeamViewer z zdalnym dostępem do komputera [34]

W przypadku obsługi informatycznej firmy rozwiązaniem problemu pomocy klientowi może być aplikacja TeamViewer pozwalająca na szybki i łatwy zdalny dostęp do problematycznego komputera.

1.4. Szybki i łatwy do użycia

Mobilne aplikacje są proste do użycia i bardzo przenośne. Wykluczają one potrzebę noszenia większych urządzeń takich jak laptopy lub notesy z długopisem. Wszelkie potrzebne informacje mogą być za pośrednictwem mobilnego oprogramowania szybko dostępne z dowolnego miejsca za pomocą zaledwie kliku kliknięć. [30]

2. Zdefiniowanie mobilnych aplikacji biznesowych

Istnieje wiele definicji i sposobów rozumienia terminu aplikacja biznesowa. Wiele dobrze znanych organizacji i korporacji posiada własne definicje aplikacji biznesowej. Dla przykładu ISO definiuje ten termin, jako generalne pojęcie wymiany informacji w przedsiębiorstwie oraz pomiędzy przedsiębiorstwem i klientem. GIIC to, jako aktywność ekonomiczną z wykorzystaniem elektronicznych środków komunikacji, za pośrednictwem której ludzie mogą między innymi kupić produkt. Intel definiuje aplikacje biznesowe, jako składową elektronicznego marketu, elektronicznej wymiany i usług. IBM, jako składową technologii informacyjnej, weba i biznesu. [4]

W toku funkcjonowania finansowego każda operacja gospodarcza w ostatecznym rachunku ekonomicznym jest wydatkowaniem lub otrzymaniem środków pieniężnych lub ich odpowiedników. Metody uzyskania przepływów mogą być bezpośrednie i pośrednie. W metodzie bezpośrednią uwzględnia się wszystkie operacje przynoszące dochód netto. Natomiast metoda pośrednia uwzględnia jedynie uogólniające wielkości zysku, takie różnica wynikająca z kupna tańszego towaru lub zysk z szybszego wykonania usługi. [35]

Aspekt biznesowy w kontekście oprogramowania mobilnego odzwierciedla podobna definicja, z tą różnicą, że aplikacje są ograniczone do programów operujących jedynie na urządzeniach przenośnych. [5]

Zestawienie powyższych faktów sprawi, że na potrzeby tej pracy przyjęta zostanie następująca definicja.

Mobilna aplikacja biznesowa jest to oprogramowanie przeznaczone na urządzenia przenośne umożliwiające wykonywanie różnych funkcji biznesowych, które przynoszą bezpośrednią lub pośrednią korzyść finansową klientowi.

Przykładami biznesowych aplikacji mobilnych są sklepy, które w oczywisty sposób umożliwiają kupno towarów. Wszelkie aplikacje umożliwiające skanowanie kodów kreskowych i porównanie cen i w efekcie dające możliwość tańszego kupna wybranego przedmiotu.

Mniej oczywistymi przykładami są aplikacje umożliwiające łatwe zarządzanie telefonem co powoduje niższe koszty obsługi informatycznej, serwisy informacyjne, które stanowią cenne źródło informacji niekiedy pozwalające podjąć prawidłowe decyzje, które mogą już mieć bezpośrednie przełożenie na zyski, także jest to aplikacja umożliwiająca nawigację dzięki której można ominąć korki, a tym samym zaoszczędzić na paliwie.

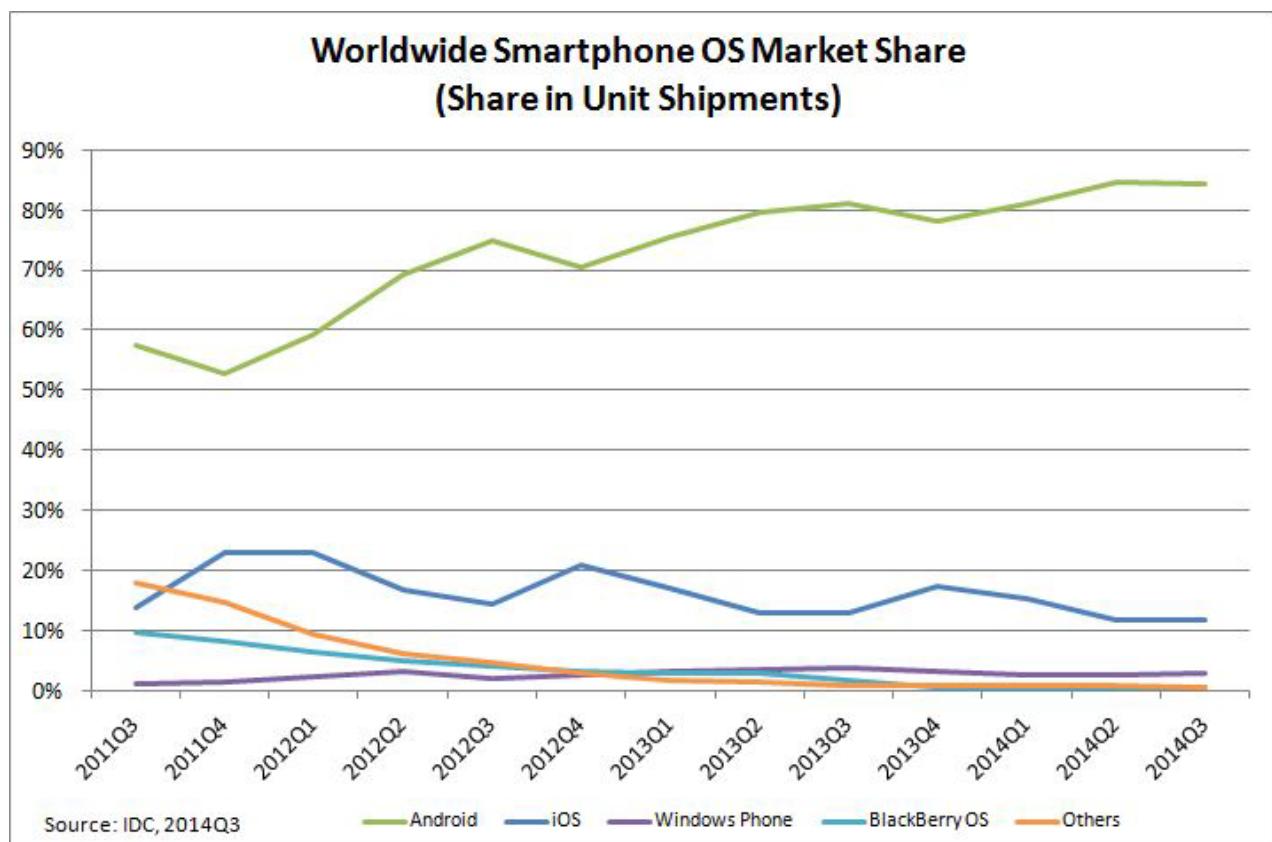
3. Wybór systemu operacyjnego dla mobilnej aplikacji biznesowej

Podstawą każdej aplikacji mobilnej jest system operacyjny w obrębie, którego ona działa, wybór platformy ma kluczowe znaczenie i źle podjęta decyzja może stanowić o porażce tworzonego oprogramowania. Na selekcję platformy składa się wiele czynników takich jak dostępność oraz koszt produkcji i utrzymania, elementy te zostaną opisane w tym fragmencie pracy magisterskiej.

3.1. Dostępność

Dostępność danego systemu operacyjnego na rynku jest jednym z kryteriów wyboru podstaw aplikacji, jeśli aplikacja jest kierowana do szerszej grupy osób oczywistym jest, że zwiększenie liczby potencjalnych klientów poprzez wybranie bardziej dostępnej platformy może znaczaco poprawić płynność finansową aplikacji.

Na rynku dostępnych jest wiele systemów operacyjnych dla urządzeń przenośnych, ale kluczowymi graczami na rynku są Android, iOS, Windows Phone oraz BlackBerry OS. Biorąc pod uwagę całkowity udział w rynku Android jest zdecydowanym liderem posiadając 84,4%, co stanowi zdecydowaną większość urządzeń.[1]



Calkowity podzial rynku smartfonów 3Q2014 [1]

Jednakże badania te pokazują wszystkie rynki razem, bez podziału na poszczególne części świata. Z pomocą przychodzi badanie grupy KANTAL WORLD PANEL, które publikuje podział rynku smartfonów na 12 państw z różnych regionów świata dobrze oddający istniejący podział. [38]

	Android	BlackBerry	iOS	Windows	Inne
Niemcy	82.3%	0.3%	8.9%	6.9%	1.5%
Włochy	73%	0.7%	10.6%	14.4%	1.3%
Francja	76.2%	0.6%	12.8%	9.9%	0.4%
Hiszpania	88.1%	0%	7.5%	4.2%	0.2%
Wielka Brytania	61.1%	0.9%	27.1%	10.3%	0.6%
USA	63.6%	1.2%	30.1%	4.7%	0.4%
Meksyk	84.4%	4%	2.5%	6.4%	2.6%
Brazylia	89.1%	0.8%	5.2%	3.8%	1.1%
Argentyna	86%	3.6%	0.2%	7.9%	2.3%
Chiny	83.9%	0%	14.2%	0.6%	1.3%
Japonia	70.2%	0.5%	27.1%	0.9%	1.4%
Australia	64.3%	0.8%	29.3%	5.1%	0.4%

Dane udziału w rynku smartfonów w wybranych państwach [38]

Jak można odczytać z danych globalnego raportu i w wybranych państwach liderem jest system operacyjny Android, lecz w USA, Wielkiej Brytanii, Japonii, Australii i Chinach kluczowym graczem jest także iOS, co trzeba wziąć pod uwagę planując wydanie aplikacji w tych państwach.

3.2. Koszt aplikacji

Aby aplikacja odniosła sukces ważnym jest poznanie wysokości kosztówtworzenia i utrzymania aplikacji, dobrze zaplanowane wydatki mogą w znaczącym stopniu usprawnić płynność aplikacji. Jest to o tyle ważne, gdyż blisko 60% aplikacji na iOS nie zwraca się, [39] a nieznacznie lepiej sytuacja wygląda na Androida. [40]

Na koszty podczas czasu życia aplikacji składają się wydatki na zakup oprogramowania i licencji do tworzenia aplikacji, opłacenie osób tworzących aplikację, koszty wdrożenia i utrzymania oraz wydatki na promocję.

3.2.1. Koszt licencji do tworzenia aplikacji

Niemalże każdy z mobilnych systemów operacyjnych udostępnia możliwość darmowej produkcji i testowania oprogramowania na urządzeniach rzeczywistych. Wyjątkami są tutaj iOS oraz Windows.

iOS za darmo umożliwia jedynie produkcję z użyciem symulatorów, natomiast za uruchamianie aplikacji na urządzeniach mobilnych wymagana jest licencja kosztująca \$99 i wymagająca corocznego odnowienia. [42]

Natomiast Windows dostarcza dwa typy licencji, dla osób indywidualnych w cenie \$19/rok oraz dla firm w cenie \$99/rok, warto jednak zaznaczyć, że cena dla osób indywidualnych może się różnić w zależności od państwa. [43]

3.2.2. Koszt zakupu oprogramowania

Dodatkowym kosztem który jest generowany podczas tworzenia aplikacji są koszty zakupu licencji na oprogramowanie. We wszystkich platformach SDK jest darmowe. Natomiast dodatkowe koszty mogą wygenerować dodatkowe frameworki pomocne podczas tworzenia aplikacji, lecz obciążenie wynikające z nich jest trudne do oszacowania, gdyż potrzeba użycia płatnych frameworków mocno zależy od kontekstu aplikacji i często nie jest wymagane aby aplikacja powstała.

3.2.3. Koszt produkcji

Koszt produkcji aplikacji jest bardzo trudną rzeczą do przewidzenia. Wydatki z tym związane zależą od poziomu skomplikowania oprogramowania i jego zależności. Niemniej możliwym jest podać zakres wydatków w zależności od poziomu skomplikowania.

Prosta nieskomplikowana aplikacja na iOS pochłonie od \$1,500 do \$4,000. Natomiast jeśli aplikacja jest bardziej złożona oraz przechowywane dane znajdują się na serwerze koszt rośnie do \$8,000-\$50,000. Dla wymagających aplikacji dla przedsiębiorstw bardzo trudno sprecyzować i mówić o koszcie wynieścącym \$50,000, \$150,000 a nawet więcej. [45]

Na platformie Android koszt prostej aplikacji wynosi \$2,000-\$10,000, nieco bardziej złożona aplikacja już wydatek rzędu \$10,000-\$40,000, natomiast bardzo skomplikowana aplikacja to \$40,000-\$80,000. [46]

Niestety ze względu na mocno ograniczony rynek developerów platform Windows OS oraz BlackBerry koszt ten jest bardzo trudny do oszacowania.

3.2.4. Koszt wdrożenia i utrzymania

Aby móc oficjalnie wdrożyć aplikację potrzebna jest rejestracja w oficjalnym markecie i zakup licencji pozwalającej tego dokonać. Po dokonaniu opłaty możliwe jest publikowanie swojej aplikacji, z zaznaczeniem, że niektóre platformy jak iOS i BlackBerry posiadają proces akceptacji oprogramowania przed oficjalnym umieszczeniem aplikacji, co może znaczco wydłużyć okres wdrażania.

Tylko BlackBerry umożliwia dzięki swoim programom wspierania rozwoju platformy w pełni darmową publikację i brak kosztów dodatkowych. [44]

Jednym z tańszych platform jest Android, licencja kosztuje jedynie \$25 i nie wymaga odnawiania. Raz zakupiona nie wygasa, co po opublikowaniu praktycznie zmniejsza koszt utrzymania do zera. [41]

Znacznie droższym rozwiązaniem jest publikowanie treści na Windows OS, które wymaga aktywnego konta i corocznej odnowy. Podobnie jak w przypadku licencji do tworzenia aplikacji dla osób indywidualnych w cenie \$19/rok oraz dla firm w cenie \$99/rok. [43]

Apple stosuje podobną strategię co Windows i wymaga aktywnego konta i corocznej opłaty w wysokości \$99. [42]

Dodatkowymi kosztami niezależnymi od platformy są między innymi utrzymanie serwerów (\$100-\$200/miesiąc) i integracja z Social Media (\$500-\$1,500) oraz inne wynikające ze specyfiki tworzonej aplikacji.

3.2.5. Zestawienie kosztów

Porównując koszty wytwarzania oprogramowania na różnych platformach zauważać można, że wydatki związane z licencją, opublikowaniem i oprogramowaniem są marginalne i nie mają większego znaczenia.

Największym wydatkiem jest natomiast czas programisty, testerów, wdrożeniowców i grafików potrzebny do rozwoju oprogramowania. W tym zestawieniu dla prostych aplikacji okazuje się iOS o wiele tańszy aniżeli Android, natomiast wraz ze wzrostem skomplikowania aplikacji stworzenie aplikacji na Androida okazuje się tańszym rozwiązaniem.

4. Wyzwania w projektowaniu mobilnych aplikacji biznesowych

Projektowanie aplikacji mobilnej jest częścią procesu wytwarzania oprogramowania. Jednym z głównych problemów projektowania aplikacji mobilnych, w porównaniu do aplikacji desktopowych, jest stworzenie dobrze wyglądającego i użytkelnego interfejsu przy zapewnieniu pełnej funkcjonalności. Urządzenia przenośne mają z reguły mniejsze rozdzielczości i mniejszą moc obliczeniową oraz bardzo dużą ilość różnych urządzeń, które z kolei mają jeszcze inne rozdzielczości i możliwości funkcjonalne. Różnią się także oferowaną treścią, treść przeznaczona na urządzenia przenośne jest o wiele mniej przewidywalna w porównaniu do platform stacjonarnych. [6]

Przedstawione problemy świadczą o tym, że mobilne aplikacje są i muszą być projektowane w odmienny sposób od aplikacji desktopowych. Następujące rozdziały prezentować będą różne aspekty projektowania w odniesieniu do oprogramowania na urządzenia przenośne.

4.1. Projektowanie w kontekście użyteczności

Użyteczność aplikacji mobilnej może zależeć od obecnej sytuacji oraz różnych potrzeb użytkownika. Model kontekstu aplikacji mobilnej może być przedstawiony przy użyciu trzech kluczowych zmiennych - środowiska, uczestników i aktywności. Zmienna środowiska opisuje właściwości obiektu w fizycznym środowisku, zmienna uczestników opisuje zaś status użytkownika oraz pozostałych uczestników znajdujących się w danym środowisku. Zmienna aktywności opisuje użytkownika, uczestnika i aktywności środowiskowe. Ponadto model uwzględnia interakcje między zmiennymi oraz wpływ czasu na obecny stan. [6]

Kategoria	Przykładowe charakterystyki
Środowisko	Fizyczne własności Naświetlenie i poziom hałasu Dostępność i jakość (urządzeń mobilnych)
Uczestnicy	Cechy osobowe (np. wiek, płeć) Stan psychiczny Stan zdrowia Oczekiwania
Aktywności	Zadania i cele (uczestników) Zdarzenia w środowisku (np. pogoda)

Przykładowe charakterystyki dla zmiennych modelu kontekstu [6]

W powyższej tabeli przedstawione są przykładowe charakterystyki dla kategorii z modelu kontekstu dla aplikacji mobilnych. Każda poprawnie zaprojektowana aplikacja powinna uwzględniać te kategorie w projektowaniu aplikacji, aby lepiej być dostosowaną do wymagań klienta. [6]

4.2. Projektowanie w kontekście prezentowanej treści

Kluczowym ograniczeniem dla aplikacji jest dostępna przestrzeń w której użytkownik może operować, umieszczenie między innymi wymaganej treści i przycisków często uniemożliwia wykorzystanie przestrzeni na komentarze wyjaśniające działanie poszczególnych akcji. Z tego powodu kluczowym dla biznesowej aplikacji mobilnej jest krótkie i precyzyjne nazewnictwo akcji nie pozostawiającej możliwości zaistnienia nieporozumienia.

Bardzo dobrym przykładem oprogramowania, które nie może pozwolić na zaistnienie dwuznaczności i nadinterpretacji są mobilne aplikacje bankowe. [36] Z każdą pomyłką wiąże się odpowiedzialność prawną i możliwy spory koszt naprawy zaistniałego błędu.



Przykładowa mobilna aplikacja bankowa [37]

W bankowości internetowej na stronie www, zobaczyć można liczne podpowiedzi i komunikaty związane z aktualnie dokonywaną czynnością, natomiast ze względu na brak odpowiednio dużej przestrzeni i potrzebę zachowania estetyki aplikacji, informacje przesyłane do klienta są mocno ograniczone, jednocześnie każdy opis jasno i klarownie tłumaczy możliwą do wykonania akcję.

O ile w przypadku aplikacji bankowych, precyzyjne nazewnictwo, jest rzeczą oczywistą, o tyle jest to rzecz często bagatelizowana w pozostałych typach aplikacji, co mocno wpływa na pogorszenie stosunków klient – aplikacja.

5. Przegląd współczesnych mobilnych aplikacji biznesowych

Ten rozdział poświęcony będzie przeglądowi współczesnych mobilnych aplikacji biznesowych pod względami projektowymi, między innymi takimi jak ekran startowy, czy ekran logowania. Przegląd będzie w sobie zawierał zestawienie używania danego komponentu w badanych aplikacjach oraz wyjaśnienia z tym związane.

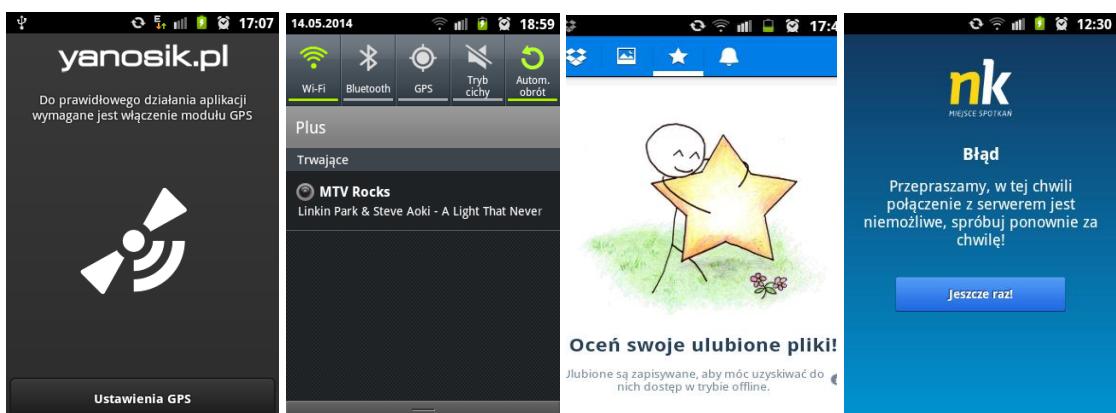
Prezentowane tutaj aplikacje zostały w pełni wymienione w rozdziale „Wykorzystane aplikacje mobilne”. Wybrana grupa jest reprezentatywną grupą 210 aplikacji, pochodząca zarówno od osób prywatnych jak i małych i dużych przedsiębiorstw. Wykorzystane aplikacje mobilne zostały wybrane z między innymi takich grup jak transport, multimedia, podróże oraz edukacja.

Wyniki przeglądu będą przedstawione w formie wykresów dla wszystkich przebadanych aplikacji oraz dla ograniczonego zakresu do tych, które posiadają powyżej pięciu milionów instalacji.

Wszystkie akcje i interakcje potrzebne użytkownikowi do osiągnięcia jego celu można przetłumaczyć na funkcjonalności, które aplikacja powinna posiadać. Z każdą dodaną funkcjonalnością aplikacja zbliża się do osiągnięcia celu klienta, jednak także staje się coraz bardziej złożona [23], takie elementy projektu jak komunikaty i rodzaj menu pomaga dobrze zorganizować te funkcjonalności, dlatego ważnym jest zwrócenie na te elementy uwagi.

5.1. Komunikaty

Komunikaty są elementem aplikacji mobilnej informującej nas o takich aktywnościach jak możliwe i wykonane czynności, braku aktywacji wymaganych usług, czy też stanowiące element informacyjny.



Przykładowe komunikaty

pl.neptis.yanosik.mobi.android.
mobi.android

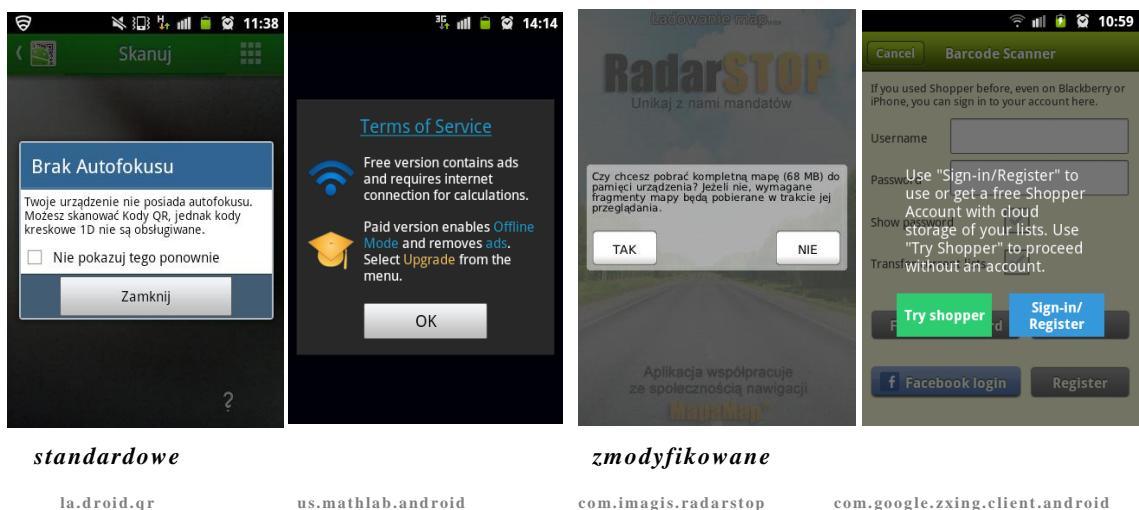
pl.gadugadu.openfm

com.dropbox.android

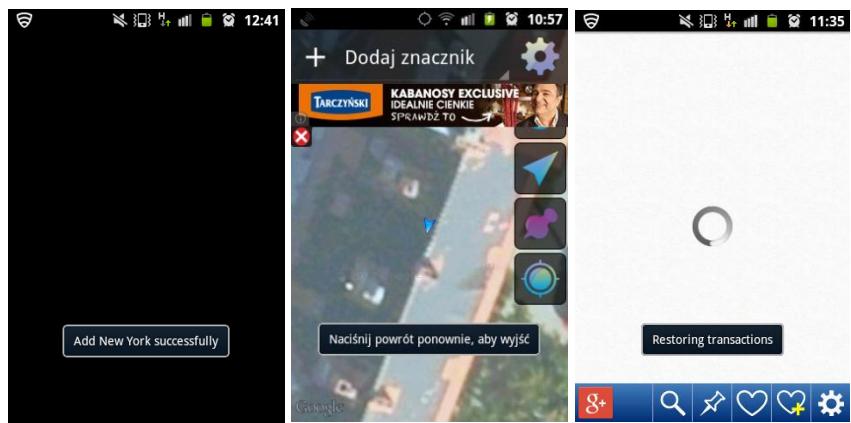
pl.nk.m.android

Przykładem komunikatów są okienka pop-up, jest to jedna z funkcji wyświetlająca nowe okno na żądanie albo automatycznie. Przykładami są zapytania potwierdzające zamknięcie aplikacji, chcąc pobrania danych albo poinformowania o zaistniałej sytuacji jak brak autofocusu. Platformy mobilne umożliwiają wykorzystanie istniejących szablonów, są one zazwyczaj przystosowane do większości obsługiwanych urządzeń, jednakże nie pozwalają na wiele modyfikacji, przez co niekoniecznie można je dostosować do projektu wyglądu reszty aplikacji. Alternatywnym rozwiążaniem jest stworzenie własnych widoków, zaletą tego rozwiązania jest większa możliwość dostosowania wyglądu okna do potrzeb aplikacji, jednakże wymaga przetestowania na większej liczbie urządzeń. Poniżej przedstawiony jest przykład szablonowego okna pop-up (po lewej stronie) oraz trzech własnoręcznie zaprojektowanych. Drugie okno pop-up jest czytelne i dobrze zaprojektowane, trzecie zawiera za dużo, za mało testu, natomiast w czwartym komunikacie przez niedostateczne przetestowanie zabrakło tła komunikatu, co negatywnie wpływa na jego odbiór.

Okna pop-up



Kolejnym typem komunikatów są wiadomości toast. Jest to prosty rodzaj okien pop-up zaprojektowany do szybkiego informowania bez przerywania działania aplikacji. Służą do szybkiego informowania o zmianie stanu aplikacji, takim jak pomyślne udane dodanie rekordu do bazy danych. Poniżej zaprezentowany jest przykład komunikatów toast.



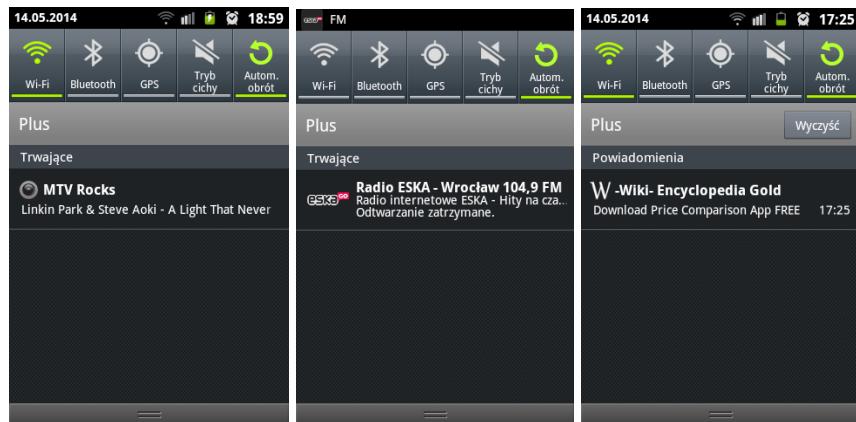
Komunikaty toast

yong.tools.life.weather

com.arume.whereismycar

com.audioguidia.myweather

Dodatkowym typem stosowanych komunikatów są powiadomienia (ang. notifications). Głównymi elementami są tytuł i wiadomość, ale mogą także zawierać obrazki oraz przyciski. Działają one poza aplikacją i pozwalają na wyświetlanie informacji o aktualnym stanie aplikacji. Dobrym przykładem zastosowania jest informowanie o otrzymaniu nowego komunikatu w serwisie społecznościowym, czy aktualnie odtwarzanym utworze w radiu. Ponadto, ponieważ powiadomienia mogą zawierać dowolny tekst możliwe jest wyświetlanie reklam, jednak nie jest to oczekiwane ze strony klienta aplikacji.



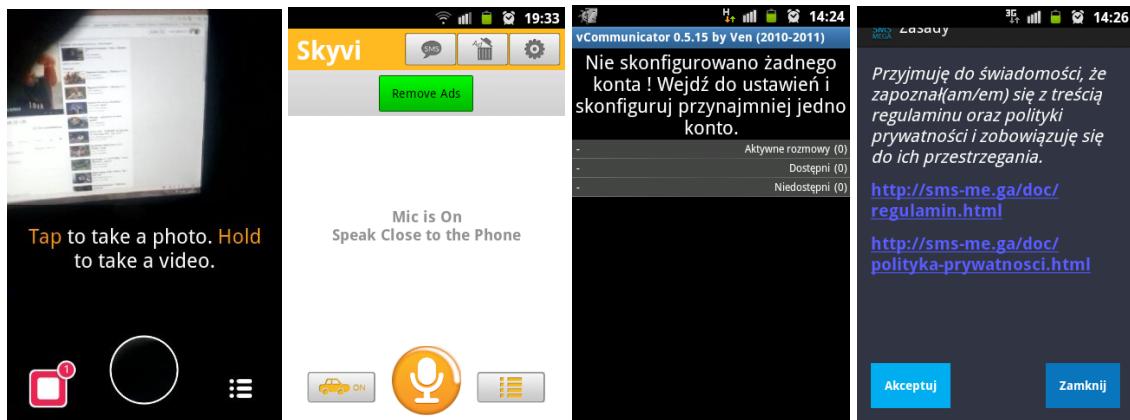
Powiadomienia

pl.gadugadu.openfm

smi.android.eska

uk.co.appsunlimited.wikiapp

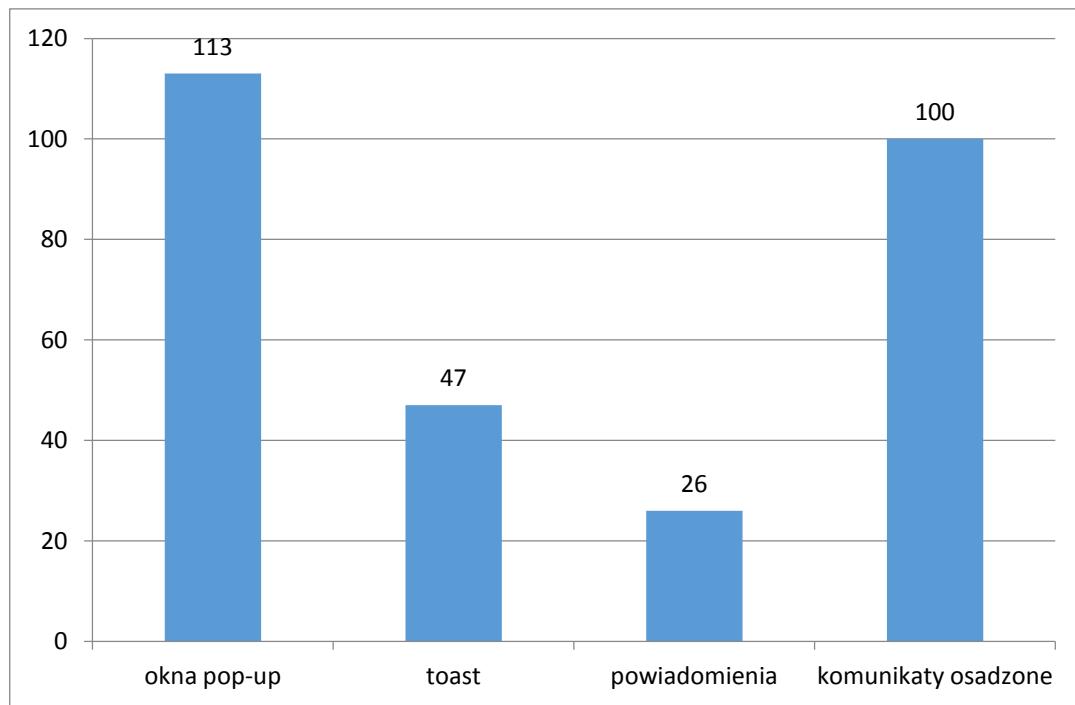
Komunikaty mogą być także osadzone w aplikacji. W przypadku takiego rozwiązania integralną część wyświetlonego widoku i ich wygląd w pełni zależy od osoby projektującej dany system. Komunikaty tego typu powinny być wyróżniające się od reszty zawartości oraz zachęcać do ich przeczytania, nie powinny też być zbyt długie.



Komunikatów osadzone

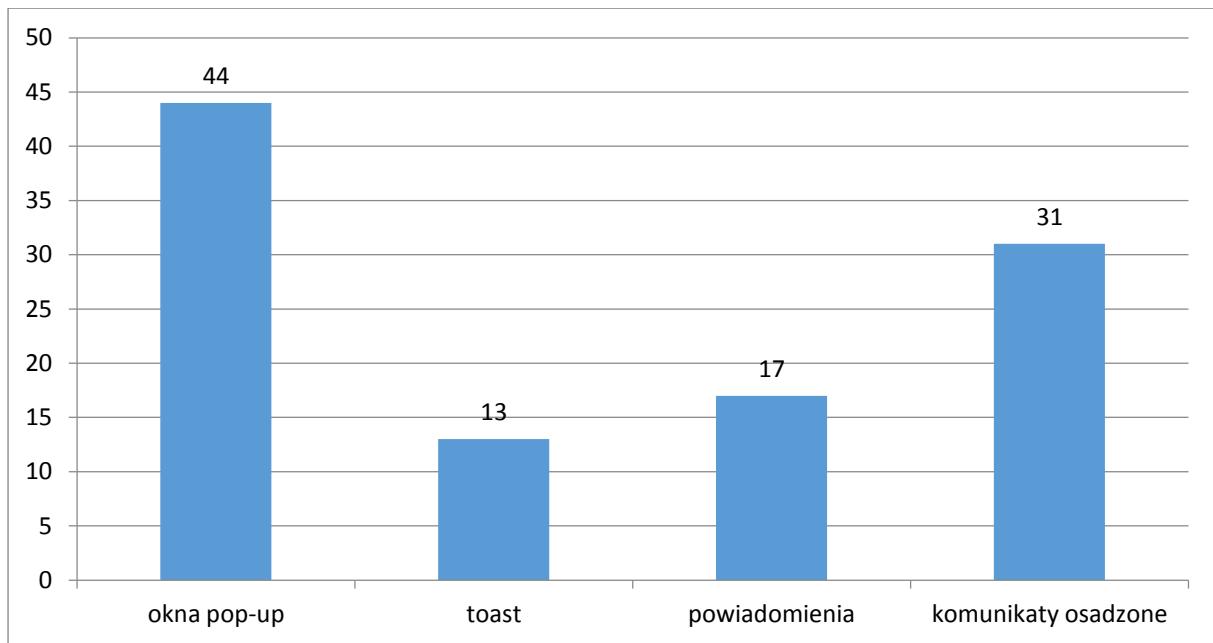
com.snapchat.android com.bluetornadosf.smartyrant pl.komur.android.vencomm
nicator ga.smsme.smsmegauuniversal

Z przeprowadzonego przeglądu mobilnych aplikacji biznesowych wynika, że najczęstszą formą stosowana komunikatów są pop-up (113 aplikacji), a najrzadziej wykorzystywane są powiadomienia.



Wykres wykorzystania rodzaju komunikatów

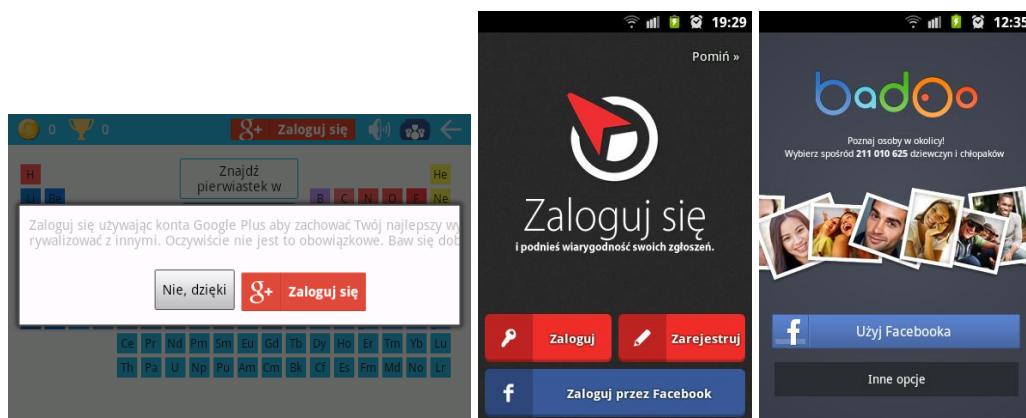
Podobnie sytuacja wygląda dla aplikacji posiadających 5mln instalacji i więcej. Okna pop-up i osadzone są najczęstszym rodzajem stosowanych komunikatów. Wartym zaznaczenia jest fakt, że 17 z 26 aplikacji, które korzystały z powiadomień były aplikacjami posiadającymi, co najmniej 5mln instalacji.



Wykres wykorzystania rodzaju komunikatów dla aplikacji powyżej 5mln instalacji

5.2. Logowanie

Logowanie jest procesem uwierzytelniania i autoryzacji użytkownika. Element logowania w systemach mobilnych nie jest najlepiej postrzeganą czynnością, wymusza on często na użytkowniku wprowadzanie dużej ilości znaków oraz połączenia z Internetem, aby w pełni funkcjonować, nawet, jeśli takie połączenie nie jest wymagane do działania aplikacji. Można wyróżnić kilka głównych sposobów na realizację tej funkcjonalności. Pierwszym z nich jest użycie serwisów społecznościowych takich jak Facebook i Google+. W przypadku wyboru takiego rozwiązania logowanie ogranicza się do przyciśnięcia jednego przycisku, jednakże wymaga to zainstalowanej aplikacji danego serwisu i bycie w nim zalogowanym, na przykład w aplikacji Facebook.



Logowanie z użyciem portali społecznościowych

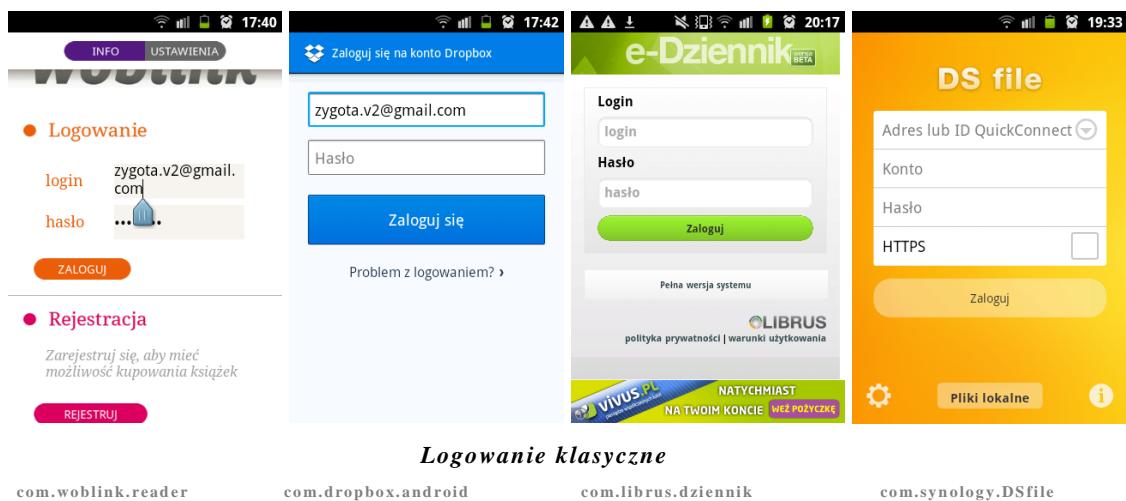
com.bk.advance.chemik

pl.neptis.yanosik.mobi.android

.mobi.android

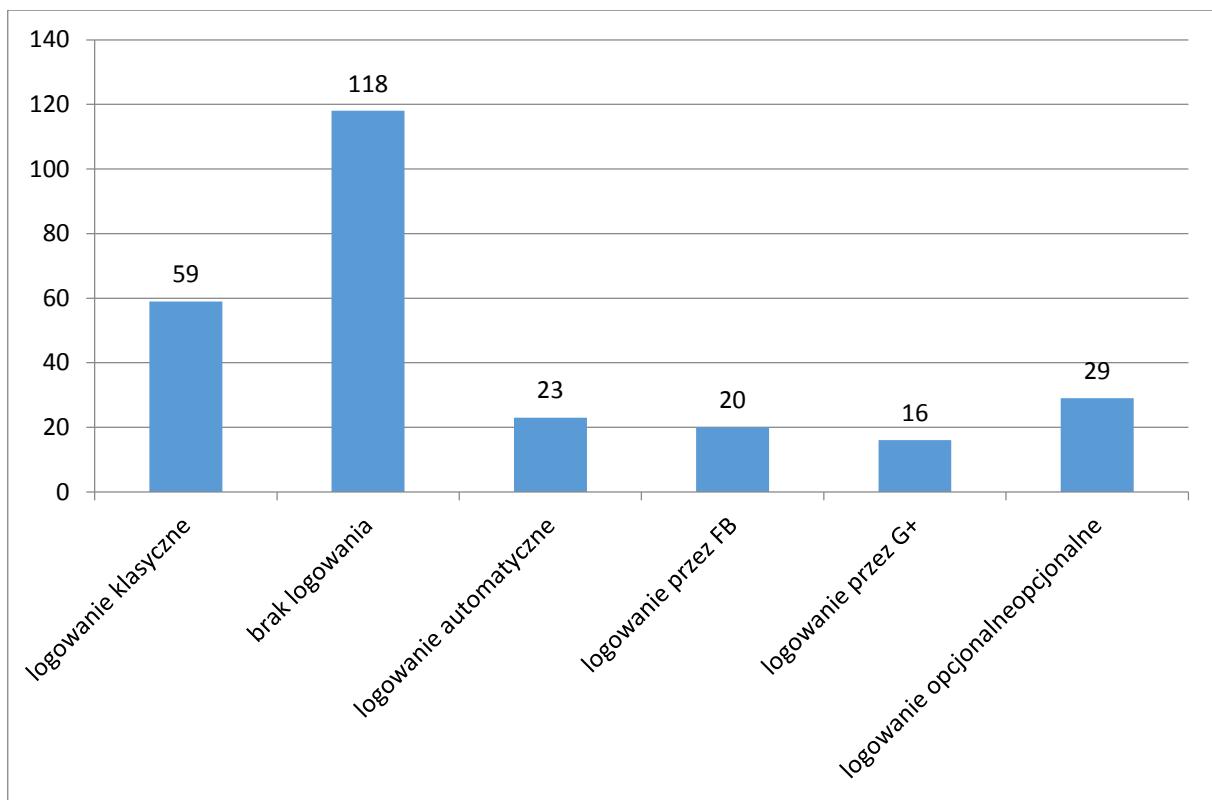
com.badoo.mobile

Drugim możliwym rozwiązaniem, bardzo często stosowanym, jest klasyczne wprowadzenie nazwy użytkownika i hasła. W przypadku aplikacji niezapamiętujących użytkownika i wymagających każdorazowego logowania przy uruchomieniu może źle wpłynąć na popularność aplikacji. Wyjątkiem są aplikacje wymagające zwiększonego poziomu bezpieczeństwa takie jak aplikacje bankowe.



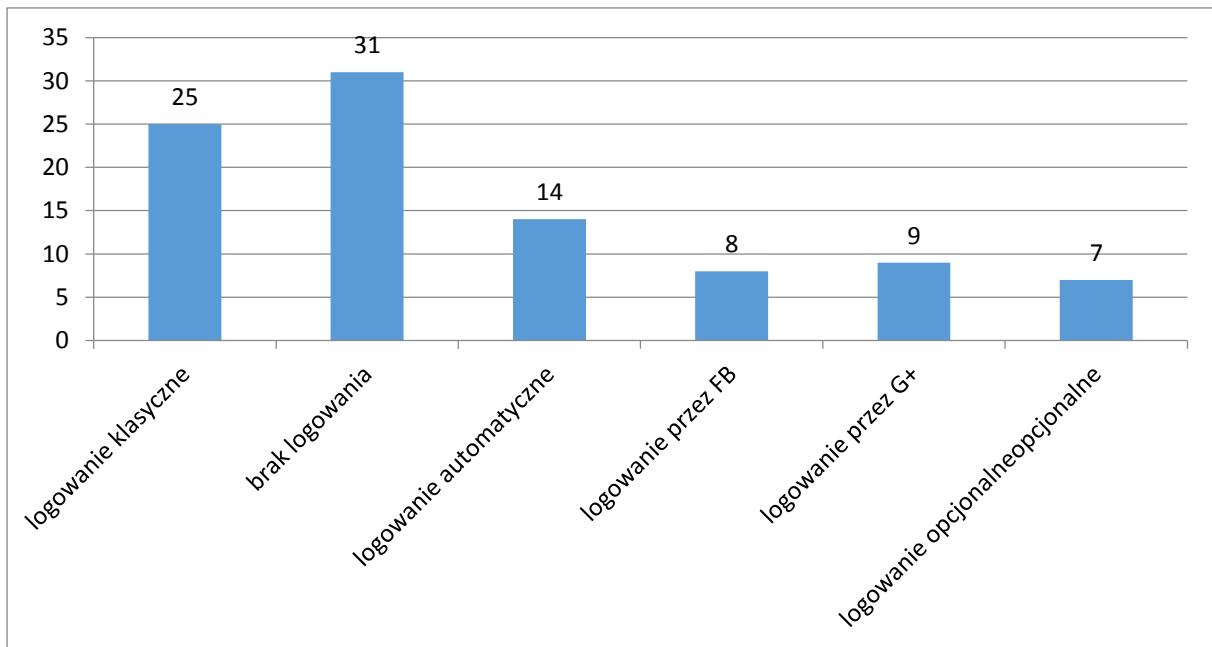
Alternatywnym i mało stosowanym przykładem logowania jest wykorzystanie danych użytkownika dostępnych z poziomu aplikacji, w tym adresu e-mail oraz numeru telefonu. W przypadku takiego rozwiązania użytkownik jest rozpoznawany na podstawie danych zapisanych w telefonie i logowany automatycznie. Rozwiązanie to jest spotykane w przypadku gier on-line z dostępem jedynie przez smartfony, gdzie istnieje potrzeba identyfikacji gracza i samo posiadanie wybranego telefonu jest wystarczające.

Z przeprowadzonego przeglądu wynika, że zaledwie 44% aplikacji wymaga autoryzacji, niewiele, ponieważ tylko 20 aplikacji pozwalało poza klasycznym logowaniem na uwierzytelnianie za pośrednictwem aplikacji Facebook, przybliżona ilość za pośrednictwem Google+. Wartym zaznaczenia jest fakt, że tylko 32% pozwoliło na korzystanie bez finalizacji tego procesu. W poniższym wykresie w przypadku, gdy aplikacja korzystała z logowania za pośrednictwem aplikacji Facebook i klasycznego - była wliczona do obu sekcji.



Wykres wykorzystania rodzaju logowania

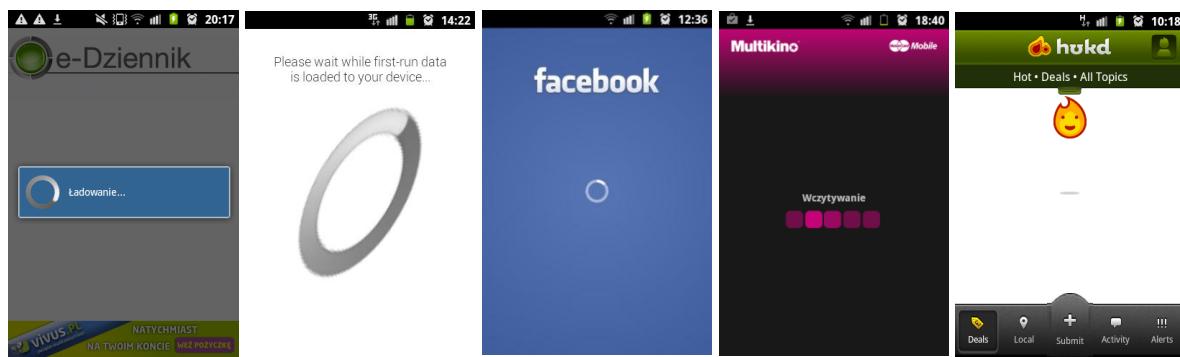
W przypadku aplikacji z dużą ilością instalacji można zaobserwować różnicę w zwiększym wykorzystaniu logowania za pomocą mediów społecznościach.



Wykres wykorzystania rodzaju logowania dla aplikacji powyżej 5mln instalacji

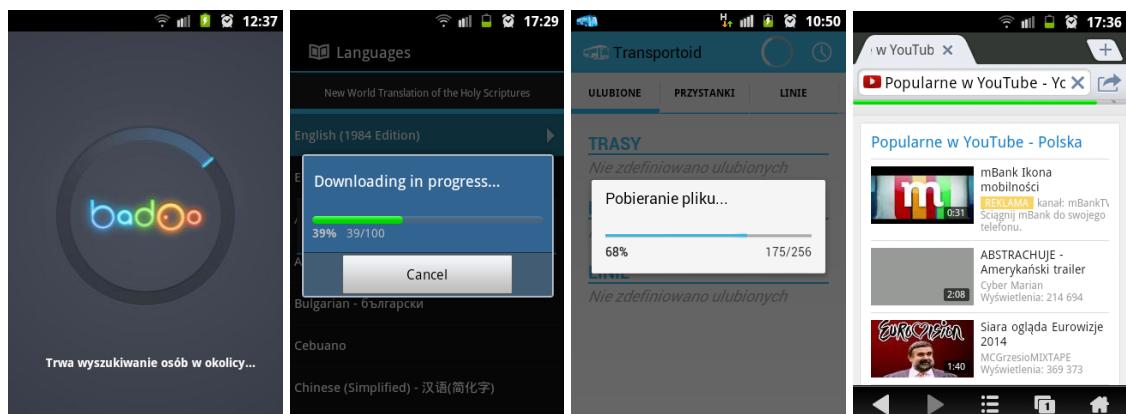
5.3. Ładowanie treści

Spora część aplikacji wymaga aktywnego połączenia z Internetem oraz systemu pobrania dodatkowych informacji zmiennych w czasie, których dostarczenie wraz z instalacją aplikacji nie jest możliwe lub niepraktyczne. Ponadto w celu obniżenia wagi instalatora część danych pobierana jest dopiero przy pierwszym uruchomieniu, dla przykładu nawigacja po uruchomieniu wysunie zapytanie o wybór map, zamiast załączać dużą ilość niepotrzebnych informacji do instalatora. Informacja o lądowaniu treści może się odbywać za pomocą komunikatów, lub też może być osadzona bezpośrednio w aplikacji.



Ladowania treści

W większości przypadków czynność ta jest obrazowana poprzez prosty komunikat oraz animację ładowania, takie rozwiązanie nie dostarcza jednak informacji o pozostałym czasie do ukończenia procesu. Jednak przy wydłużającym się czasie, w skrajnych przypadkach, użytkownik może wyłączyć aplikacje na chwilę przed załadowaniem się w pełni informacji, dlatego też stosowane są komunikaty z informacją o aktualnym stanie ładowania z paskiem postępu albo zmieniające się komunikaty.



Komunikaty z informacją o postępie

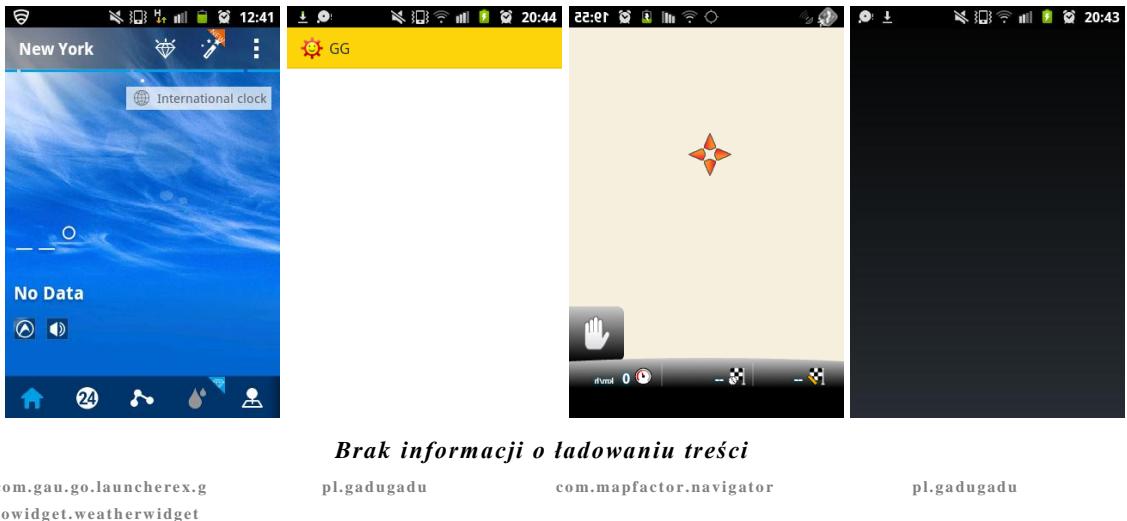
com.badoo.mobile

kst NWTBibleLite

com.transportoid

com.opera.mini.android

Mimo, że oczywistym jest potrzeba informowania o ładowaniu treści, wiele aplikacji nie powiadamia o tym fakcie użytkownika. Aplikacja w takim przypadku sprawia wrażenie zaprzestania działania, z omówionych możliwości jest to zdecydowanie najgorsze podejście.

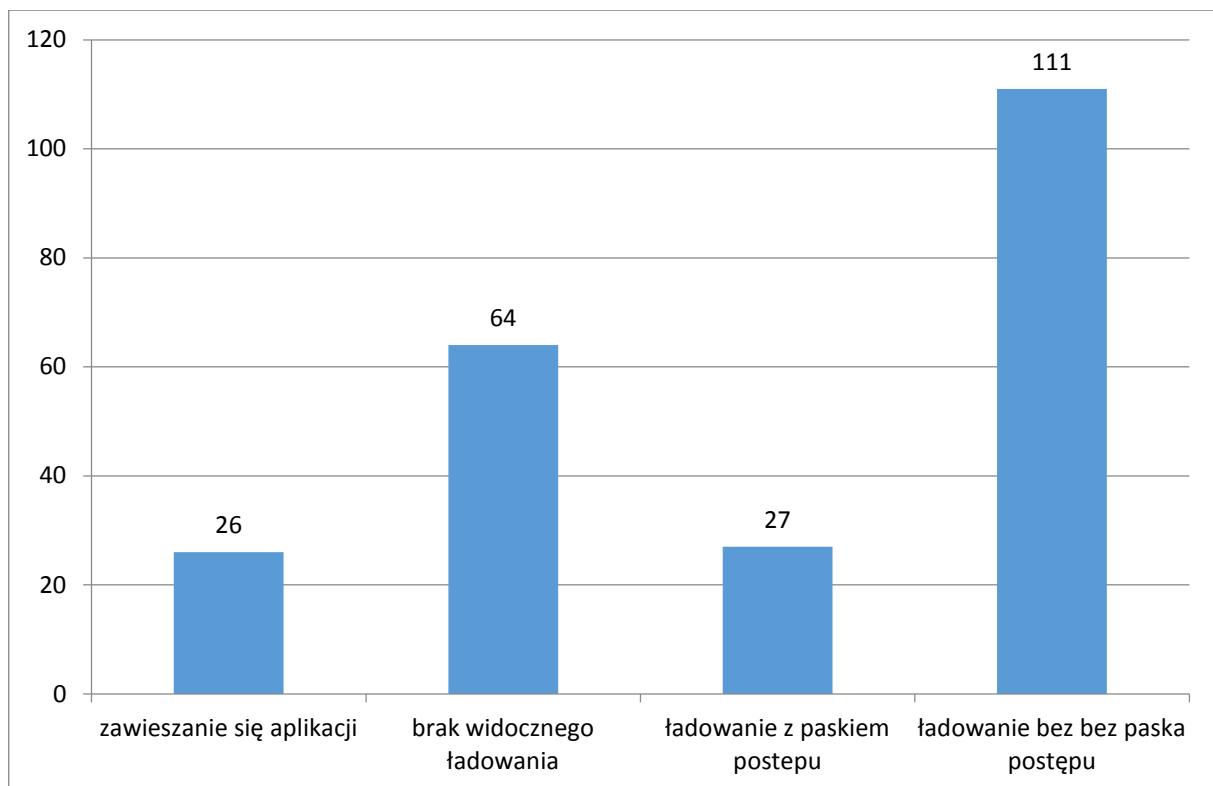


Alternatywnym sposobem, który także nie informuje użytkownika o pobieraniu, jest pobieranie w tle bez blokowania aplikacji. Pobieranie części danych następuje przy ładowaniu się aplikacji podczas Splashscreena albo na bieżąco podczas działania aplikacji. W przypadku takiego rozwiązania, dane są przygotowywane użytkownikowi wyprzedzając jego akcje, często jest to spotykane w przypadku serwisów informacyjnych. Pozwala to na czytanie artykułów już załadowanych i przeczytanie nowszych w następnej kolejności.

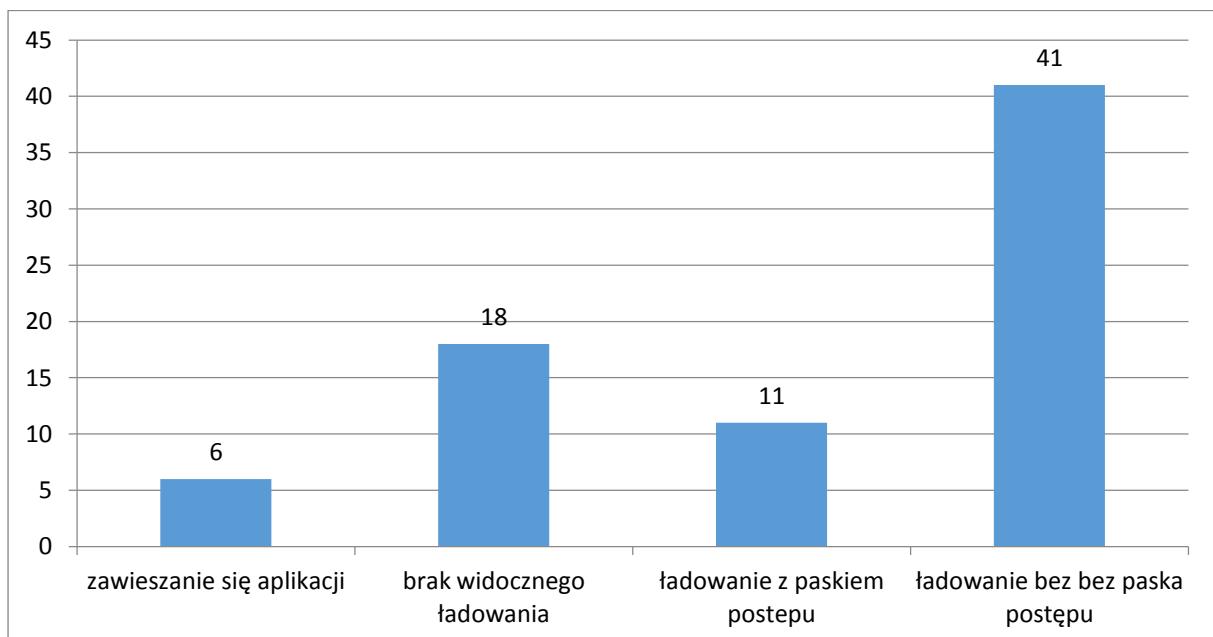
Przeprowadzony przegląd pokazał, że prawie każda aplikacja korzysta z mechanizmu ładowania danych. Zdecydowana większość badanych programów używa animacji ładowania, ale bez żadnego informowania o postępie. Wartym zaznaczenia zaznaczyć, że aż 26 widocznie zawiesiło działanie na wskutek pobierania treści podczas aktywnego ekranu.

Dla aplikacji posiadających 5 mln instalacji, aż 6 aplikacji widocznie się zawiesiło, jednak były to jedynie sytuacje wynikające z przesyłania bardzo dużej ilości danych, tak jak na przykład aplikacja Facebook, w żadnym przypadku błędного sposobu przesyłania danych.

Poniższy wykres nie bierze pod uwagę ładowania wynikającego z czynności rejestracji i logowania, a także brak tej funkcji oraz pobieranie w tle zostały zawarte razem w kategorii „brak widocznego ładowania”.



Wykres występujących rodzajów ładowania treści

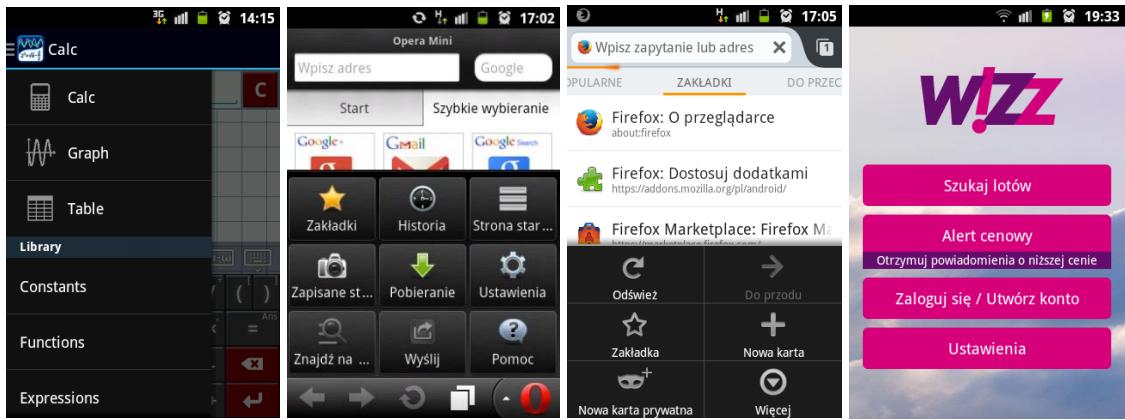


Wykres występujących rodzajów ładowania treści dla aplikacji powyżej 5mln instalacji

5.4. Menu

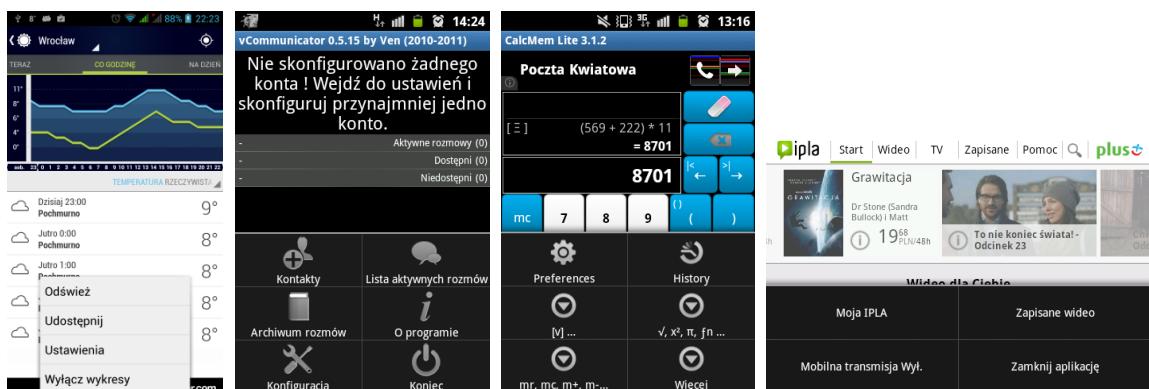
Nieodzownym elementem niemalże każdej aplikacji jest menu. Jego rodzajów jest wiele, począwszy od menu złożonego z kafelków charakterystycznych dla Windows Phone poprzez klasyczną listę opcji do rozwijanego menu często z lewej krawędzi aplikacji.

Dobrze zaprojektowana aplikacja powinna posiadać nawigację, która jest dla klienta niewidzialna, taką, na którą nawet nie musi zwracać uwagi. Z tego powodu projektowanie menu w aplikacjach jest kluczowym elementem procesu powstawania aplikacji. [22]



Przykłady rodzajów menu

System Operacyjny Android wspiera natywnie korzystanie wbudowanego menu po przyciśnięciu klawisza fizycznego menu. Po przyciśnięciu klawisza menu, możliwe opcje do wyboru pojawią się na spodzie aplikacji przykrywając prezentowaną treść. Dostępne możliwości personalizacji tego rodzaju menu polegają na zmianie nazwy pozycji oraz opcjonalnym nadaniu ikony akcji. Ponadto akcja przycisku menu może być nadpisana i powodować otwieranie się bądź zamykania innego rodzaju menu.



Natywne menu Androida

com.accuweather.android

pl.komur.android.vencom

municator

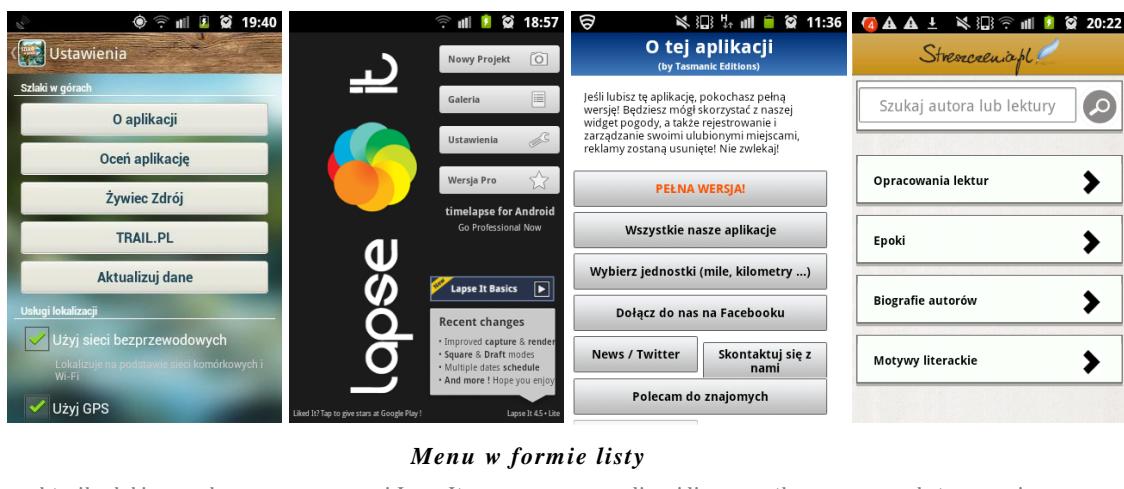
com.j2eeknowledge.com

pl.redefine.ipla

Innym często spotykanym rodzajem jest menu kafelkowe, które jest najbardziej znane z eksploratora plików w systemach operacyjnych. Najczęściej spotykane są tego typu menu w formie znanej z pierwszych telefonów z kolorowymi wyświetlaczami oraz w stylu ikonek z pulpitu systemów desktopowych, jednakże adaptowana jest także koncepcja nowoczesnych kafelków z mobilnego systemu operacyjnego Windows Phone 8.



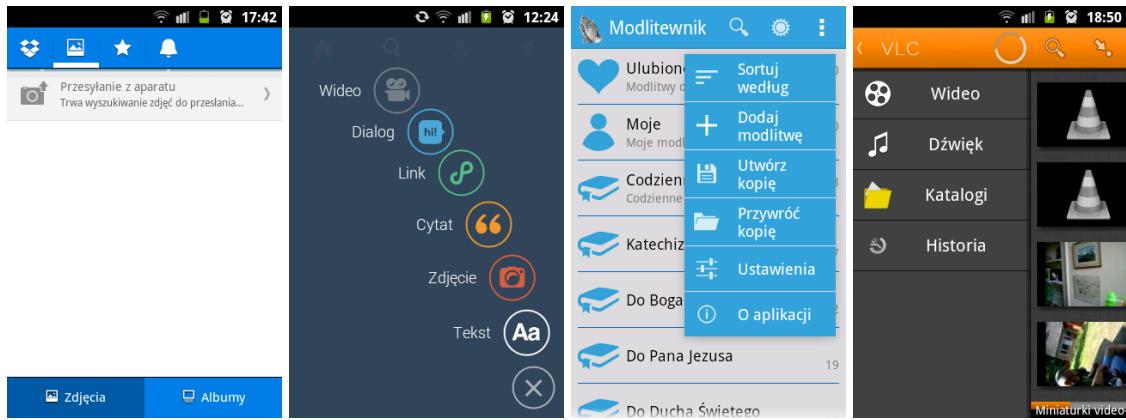
Następnym w kolejności jest menu w formie listy, jest często spotykane i charakteryzuje się często przewijaną listą opcji z elementami w formie przycisków zajmujących sporą długość ekranu. Tego rodzaj menu znany jest ze starych czarnobiałych telefonów komórkowych.



Nowoczesną formą menu spotykaną w smartfonach są rozsuwane menu najczęściej z wirtualnego przycisku menu umiejscowionego w lewym górnym rogu lub z lewej krawędzi aplikacji. Powstanie takich menu umożliwiła spora moc obliczeniowa smartfonów niejednokrotnie przewyższająca moc obliczeniową komputerów sprzed kilku lat, która pozwala na zaistnienie

animacji i położenia menu w dowolnej przestrzeni aplikacji. Przy użyciu menu wysuwanego z krawędzi aplikacji często cała aplikacja jest przesuwana adekwatnie do szerokości menu.

Spotykanym rozwiązańiem jest także Action Bar, posiada on formę paska zakładek w górnej i/lub dolnej części aplikacji pozwalając poprzez kliknięcie na ikonkę na zmianę aktywności.



Nowoczesne menu

com.dropbox.android

com.tumblr

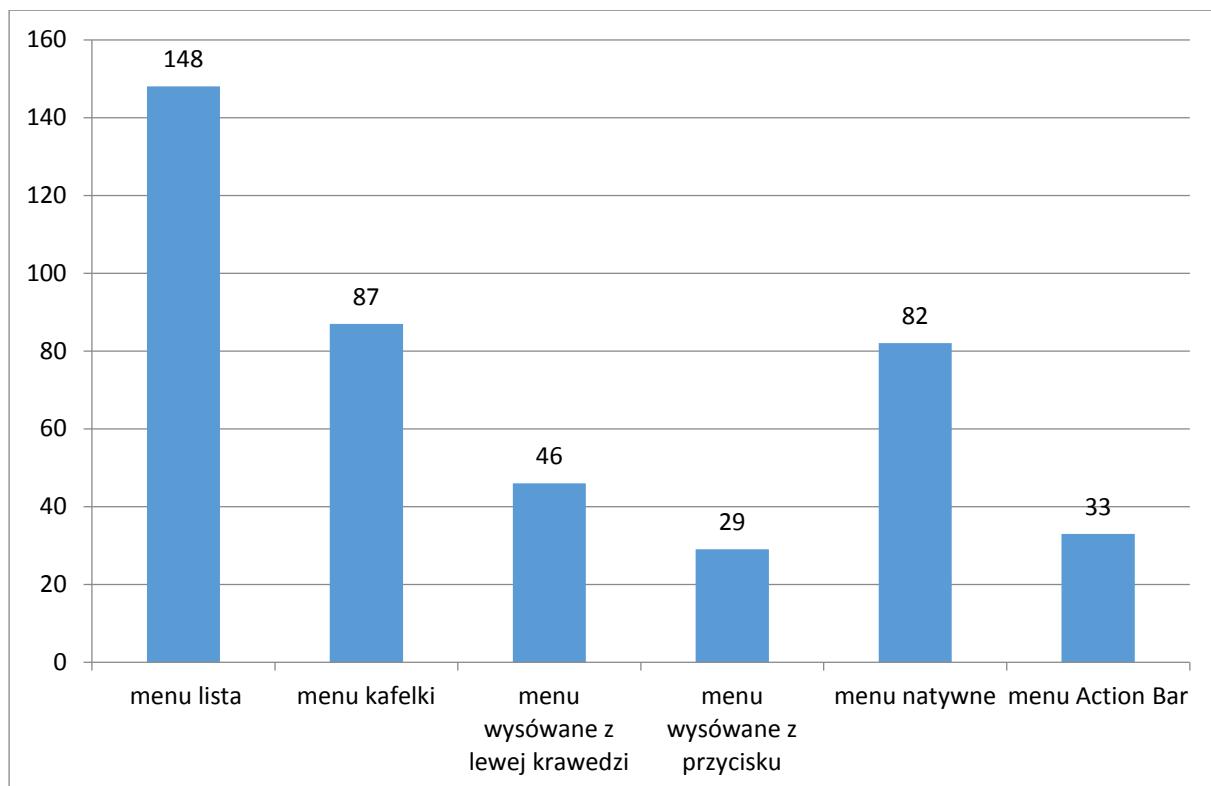
pl.mb.modlitewnik

org.videolan.vlc.betav7neon

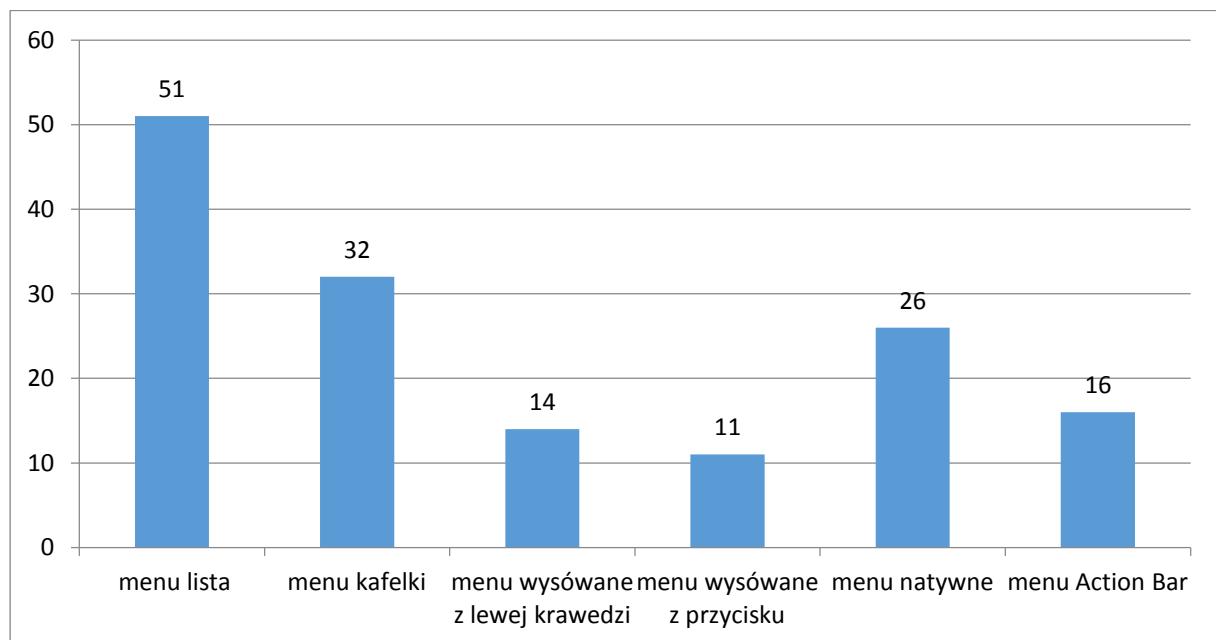
Częstym błędem jest łączenie ze sobą wielu rodzajów menu dostępnych z poziomu jednego widoku. Dla przykładu połączenie menu wysuwanego z lewej krawędzi aplikacji oraz z inną treścią z przycisku umiejscowionego w prawym górnym rogu. Błądem jest także niewykorzystywanie fizycznego przycisku menu do otwierania bądź zamknięcia dostępnego menu w przypadku rezygnacji z menu natywnego.

Z przeprowadzonego przeglądu mobilnych aplikacji biznesowych wynika, że najbardziej popularnym sposobem prezentacji menu są klasyczne rodzaje menu - lista i kafelki, a także wykorzystujące natywne menu. Nowocześniejsze rodzaje menu jak widać są wykorzystywane, lecz nadal nie jest to nadal połowa którykolwiek z liderów rodzajów menu. Podobny trend można zaobserwować w przypadku aplikacji powyżej 5 mln instalacji.

W tym badaniu nie zostały wzięte pod uwagę rodzaje menu stosowanego przy wyborze logowania i rejestracji.



Wykres występujących rodzajów menu

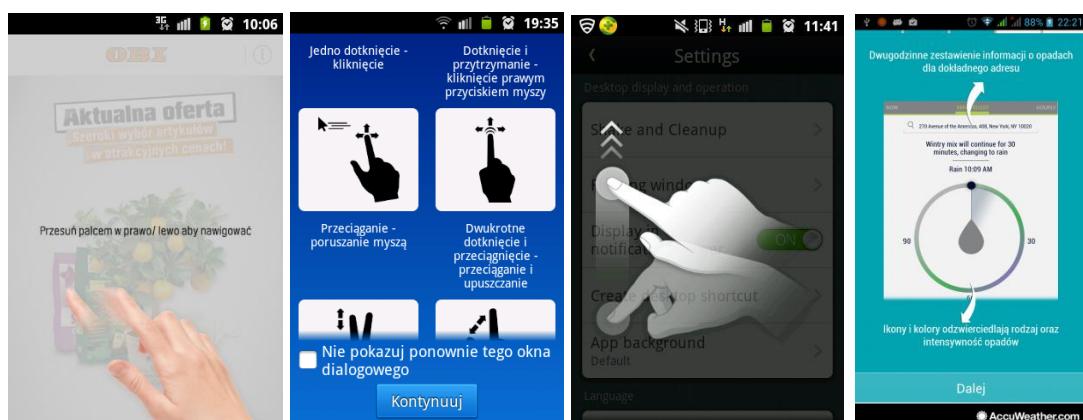


Wykres występujących rodzajów menu dla aplikacji powyżej 5mln instalacji

5.5. Nawigacja

Sporą część nawigacji, która odbywa się poprzez menu, została opisana w poprzednim podrozdziale, lecz smartfony są urządzeniami dotykowymi i posiadają wiele możliwych do wykonania akcji przy użyciu jednego bądź większej ilości palców. Przykładami takich akcji są kliknięcie, podwójne kliknięcie, przeciągnięcie, przesunięcie, przyciśnięcie i przytrzymanie oraz wiele innych. Dostępnych akcji jest bardzo dużo i często są do siebie bardzo podobne, niemniej jednak mają one spore zastosowanie w aplikacjach mobilnych, dla przykładu w iOS 4.3 odpowiednia akcja z użyciem pięciu palców zamykała aktywną aplikację.

Ponieważ dostępnych możliwych akcji jest bardzo dużo, w przypadku wykorzystania niestandardowej nawigacji dobrą praktyką jest informowanie użytkownika o istniejącym sposobie przemieszczania się w aplikacji. Można to osiągnąć przez proste samouczki wyświetlane przy pierwszym uruchomieniu aplikacji.



Przykłady samouczków do nawigacji

com.biuropodrozyreklam
y.yobi

com.teamviewer.teamviewer.m
arket.mobile

com.qihoo.msafe

com.accuweather.android

Dodatkowo w przypadku nawigacji w postaci przesuwania ekranów lub przechodzenia po długiej liście, ważnym jest informowanie w czytelny sposób o aktualnym położeniu widocznego ekranu w kontekście reszty dostępnej treści. Na pierwszym z poniższych przykładów możemy zobaczyć kropki w dolnej części ekranu, w przypadku takiego rozwiązania aktywny ekran odróżnia się od pozostałych informując o położeniu. W drugi przykładzie ta koncepcja została rozwinięta i dwie pierwsze ikonki informują o ustawieniach oraz geolokalizacji natomiast cztery pozostałe traktują o pogodzie, możemy zobaczyć, że aktywnym ekranem jest ekran czwarty. W ostatnim przykładzie zamiast scroll bara prawej stronie listy, zastosowano miarkę odpowiadającą ilości punktów potrzebnych do zamówienia danej nagrody, w prosty sposób możemy przesunąć się do interesującego nas zakresu punktów.



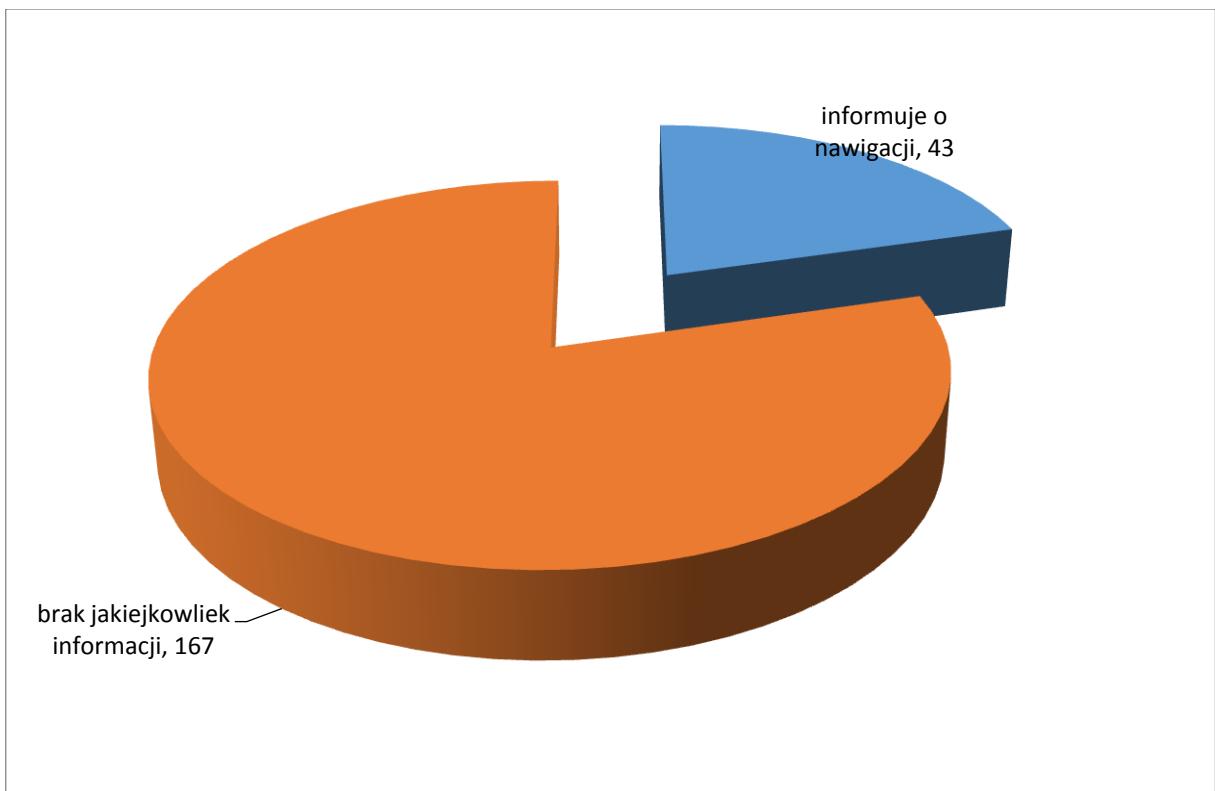
Przykłady informowania o nawigacji

pl.tvn.meteo

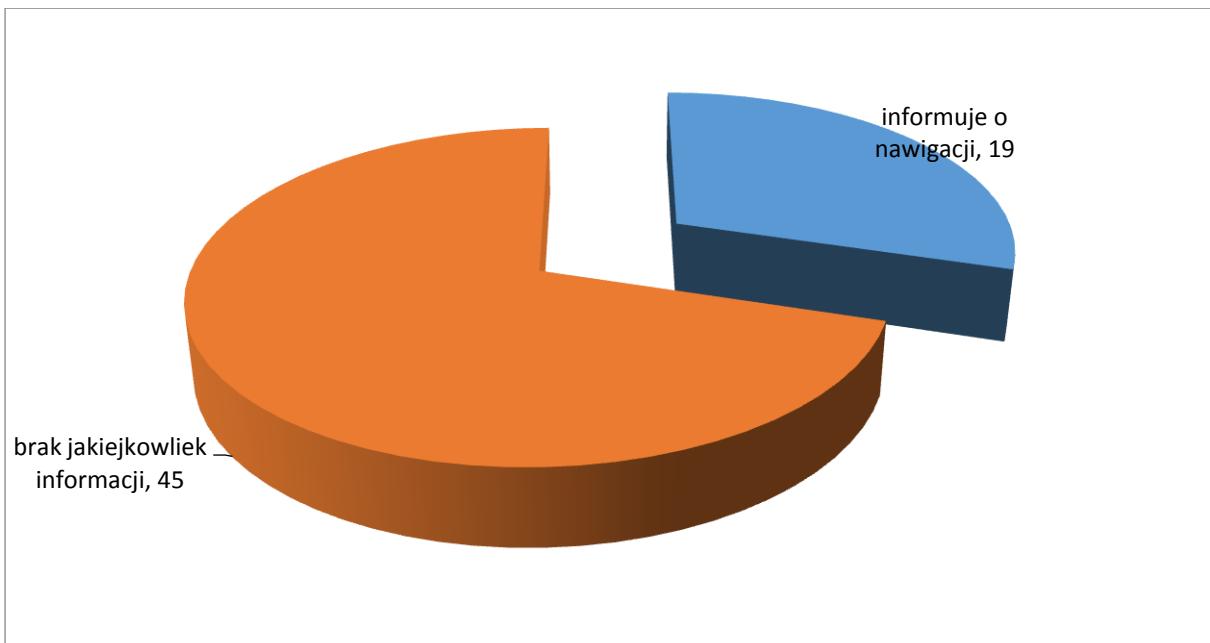
com.macropinch.swan

pl.seriousimpact.loots

Poniżej przedstawiono dane dotyczące informowania o istniejącej nawigacji w formie samouczka, lub też informacji o aktualnym położeniu. Bardzo mało, bo zaledwie 20% informuje użytkownika o dostępnej nawigacji. Podobny trend można zaobserwować w przypadku aplikacji powyżej 5 mln instalacji. W badaniu nie wzięto pod uwagę elementów Action Bar informujących o położeniu uznanych ze względu na pełniące funkcje, jako menu.



Wykres informowania o dostępnej nawigacji



Wykres informowania o dostępnej nawigacji dla aplikacji powyżej 5mln instalacji

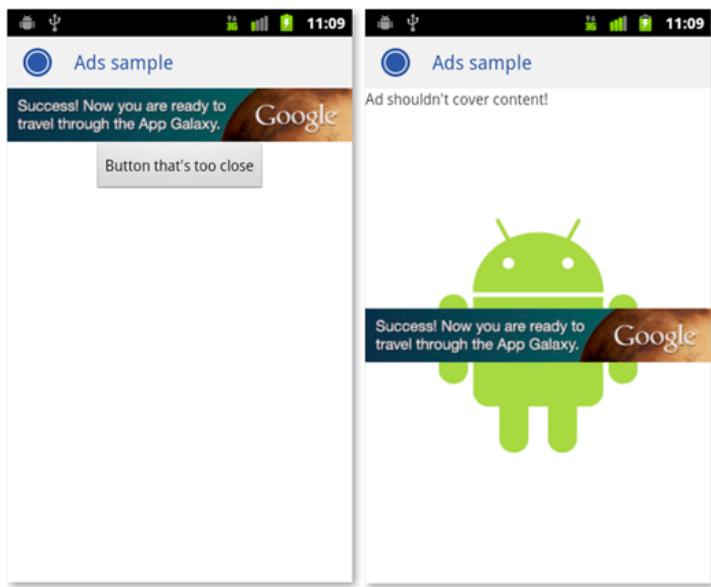
5.6. Reklamy

Reklamy są jednym ze sposobów monetaryzacji aplikacji. Dostarczają one bezpośrednio fundusze z kliknięć i/lub wyświetleń osadzonych reklam. Dobrze użyte reklamy mogą przynieść w przypadku popularnej aplikacji znaczące fundusze, ponadto wiele aplikacji oferuje za drobną opłatą usunięcie reklam.

Reklamy są jednak dobrą formą monetaryzacji w przypadku co najmniej 100 tysięcy aktywnych użytkowników, często lepszym rozwiązaniem są płatne aplikacje oraz płatności wewnętrz aplikacji. [25]

Jednakże źle umieszczone reklamy mogą pogorszyć wrażenie użytkownika z aplikacji i spowodować znaczący spadek popularności aplikacji. Aby temu przeciwdziałać Google w materiałach treningowych zmieścił wskazówki dobrego osadzania reklam. [15]

Jedną z głównych wskazówek jest, aby reklama nie pokrywała żadnej treści, także elementy klikalne nie powinny być zbyt blisko reklam, aby uniknąć nieumyślnego klikania na reklamy.



Wizualizacje wskazówek zamieszczonych w materiałach treningowych Google [15]

Zachowanie odpowiedniego balansu w przypadku aplikacji mobilnych jest szczególnie ważne. W przypadku aplikacji desktopowych istnieje możliwość łatwego wyłączenia reklam, tak, więc porażkę aplikacji webowych można zdefiniować, jako błędy z rodziny kodów 400 i 500 oraz aplikacji standalone, jako niekontrolowane się wyłączenie. Jednak w przypadku aplikacji mobilnych i sposobu ich użytkowania, główną przyczyną porażki jest złe użycie reklam. [24]



Definicja porażki wg User Mobile Experience [24]

Efektywnych sposobów na umieszczanie reklam jest wiele, jednym z nich są banery. Banery są jednymi z najczęściej używanych form reklam w aplikacjach mobilnych. Mogą różnić się

zawartością i wielkością. Mogą zarówno być czysto tekstowe jak i w pełni multimedialne. Chodź nieczęsto spotykane podobnie jak w przypadku stron internetowych banery mogą być rozwijane, interaktywne i rozsuwane.



Reklamy banerowe

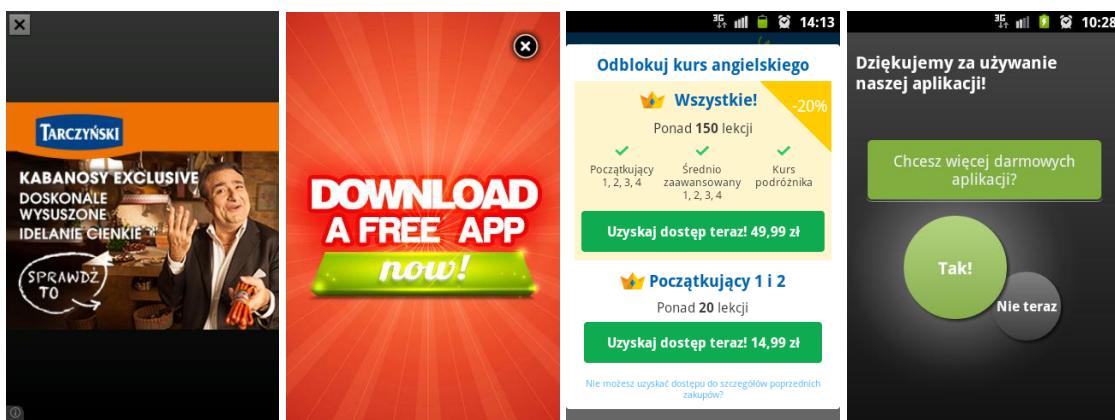
com.clevercat.craftingguide

com.librus.dziennik

com.ludi.climapoland

com.arume.whereismycar

Innym często spotykanym typem reklam są reklamy pełnoekranowe, zazwyczaj są używane przy opuszczaniu aplikacji oraz w momentach zmiany stanu aplikacji, dla przykładu podczas awansowania na nowy poziom w grze albo podczas ponawiania obecnego poziomu. Podobnie jak w przypadku banerów kreacja reklam pełnoekranowych jest dowolna, reklamy tego typu mogą być w pełni multimedialne, statyczne, tekstowe. Mogą komponować się z aplikacją lub też widocznie oddzielać się od jej konwencji. Zazwyczaj w tego typu reklamach zamknięcie jest dostępne tylko poprzez przycisk X, a każde inne kliknięcie spowoduje przejście do strony internetowej reklamodawcy.



Reklamy pełnoekranowe

com.yamlearning.geographylearning

com.arume.whereismycar

com.busuu.android.enc

com.mxdata.tube.Market

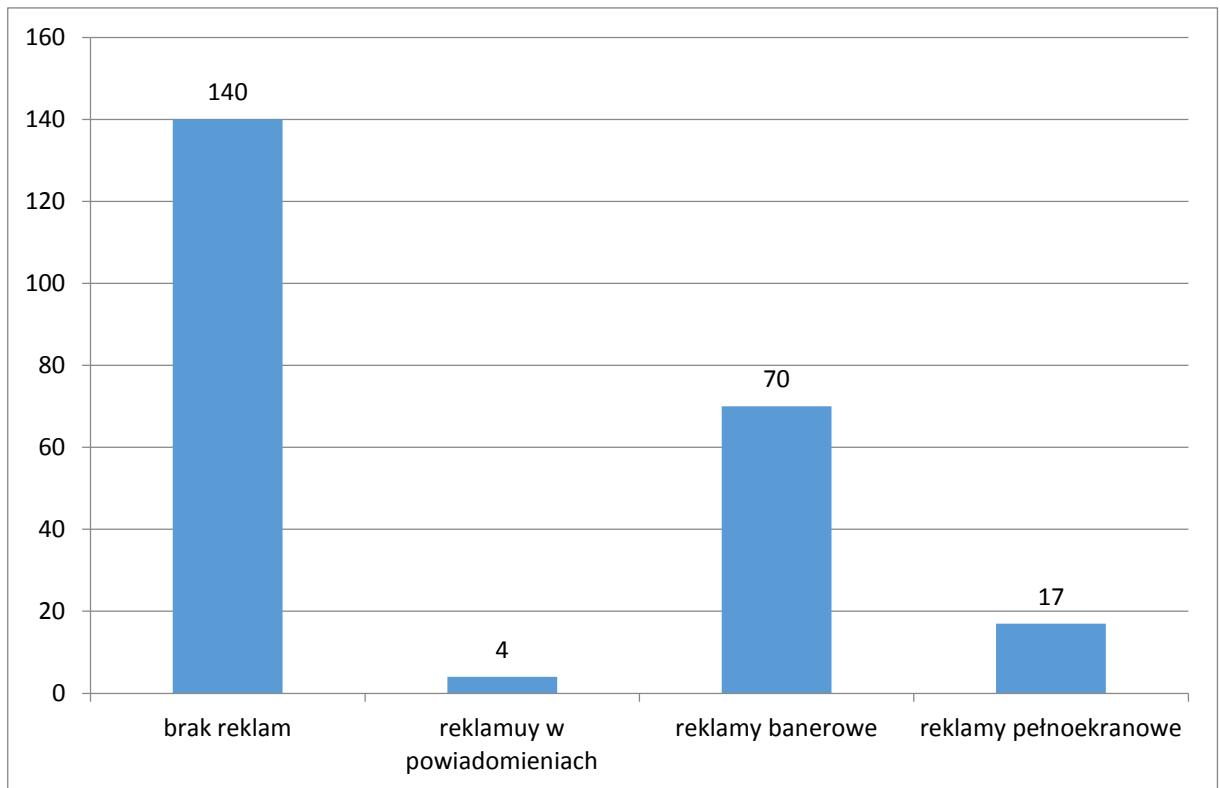
Powiadomienia (ang. Notification Ads) stanowią jedną z najbardziej natrętnych modelów reklamowania. Takie rozwiązanie zazwyczaj bardzo dobrze monetaryzuje aplikację, lecz nadużywany może prowadzić do zwiększenia się grona niezadowolonych użytkowników aplikacji. [16]



Reklama w powiadomieniach

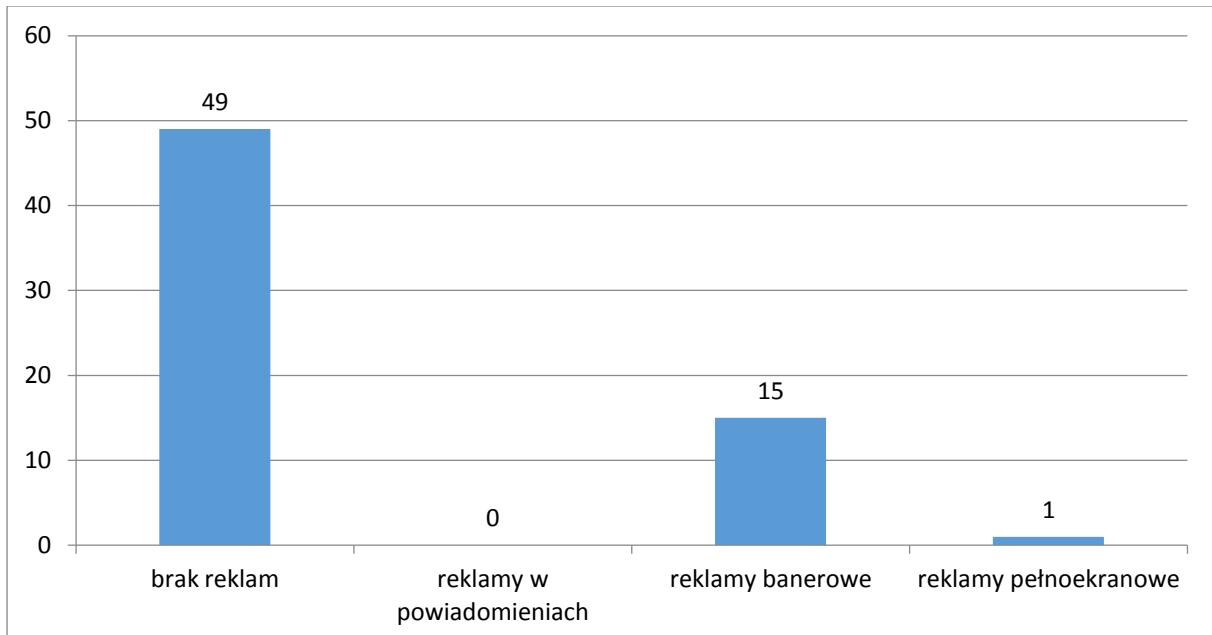
uk.co.appsunlimited.wikiapp

Wśród aplikacji używających reklamy każda używa reklam banerowych, dodatkowo część z nich używa reklam pełnoekranowych i tylko cztery korzystały z reklam umieszczonych w powiadomieniach, w tym jedna (uk.co.appsunlimited.wikiapp) robiła to natarczywie.



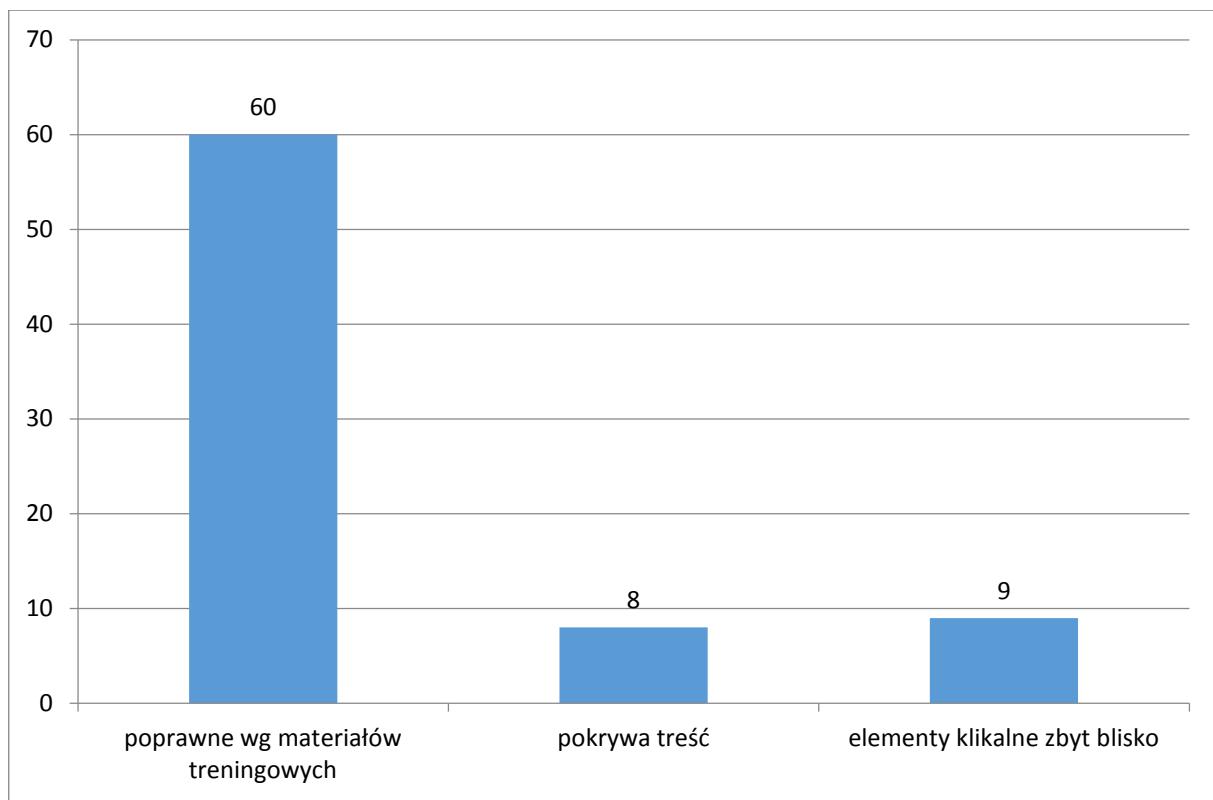
Wykres wykorzystania rodzajów reklam lub ich braku

Dla aplikacji, które zostały zainstalowane, co najmniej 5 mln razy, ilość stosujących reklamy spadła o połowę i tylko jedna posiadała reklamy pełnoekranowe. Wartym zaznaczenia jest, że żadna z aplikacji korzystającej z reklam w powiadomieniach nie osiągnęła nawet 500 tysięcy instalacji.



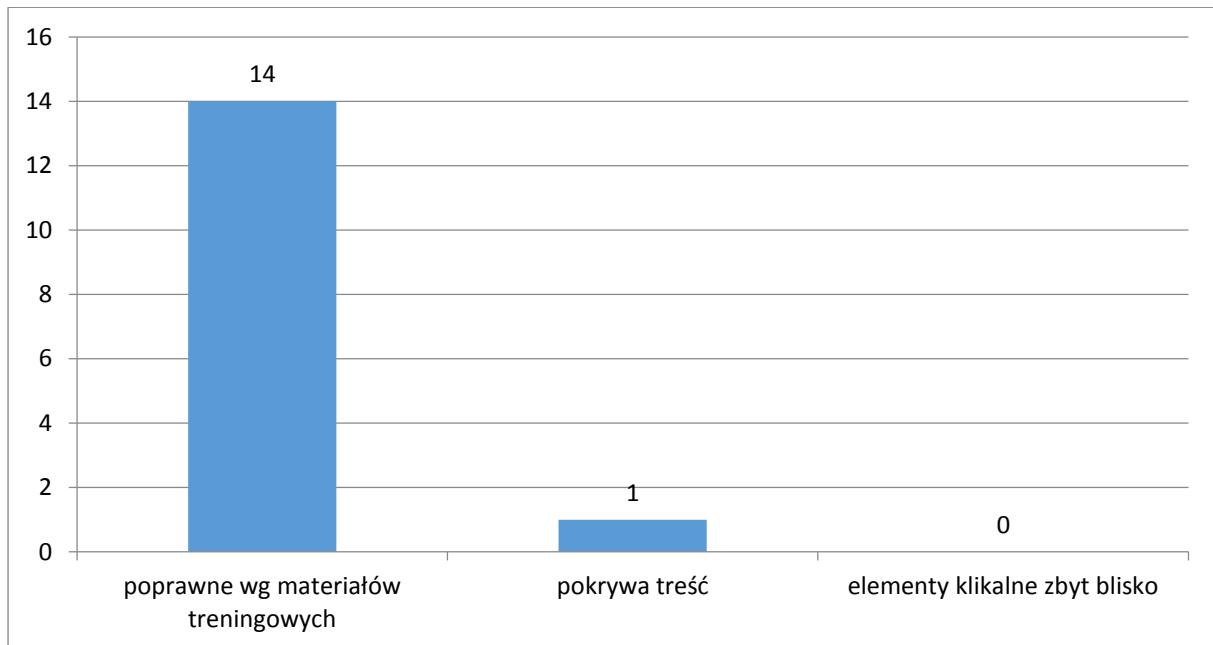
Wykres wykorzystania rodzajów reklam lub ich braku dla aplikacji powyżej 5mln instalacji

Podczas przeglądu aplikacji zwrócono też uwagę na stosowanie się do wskazówek zamieszczonych w materiałach szkoleniowych Google. Większość (60 z 70) poprawnie zastosowało się do zaleceń. Tylko 8 aplikacji reklamą pokrywało treść lub przyciski, w jednym przypadku (com.librus.dziennik) reklama całkowicie zasłaniała przycisk ustawić. Wśród aplikacji pokrywających treść 9 aplikacji również miało banery w bardzo bliskiej odległości lub stykające się z elementami klikalnymi.



Wykres realizacji wskazówek materiałów szkoleniowych Google dla reklam banerowych

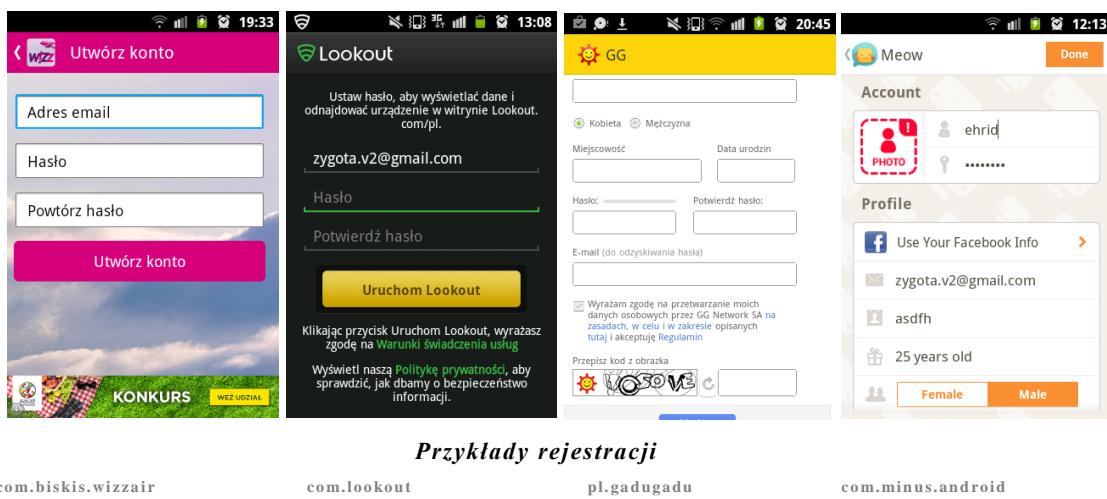
O wiele lepiej sytuacja przedstawia się dla aplikacji posiadających najmniej 5mln instalacji. W tym przypadku, tylko jedna aplikacja (com.sonyericsson.trackid) posiadała baner pokrywający prezentowaną treść.



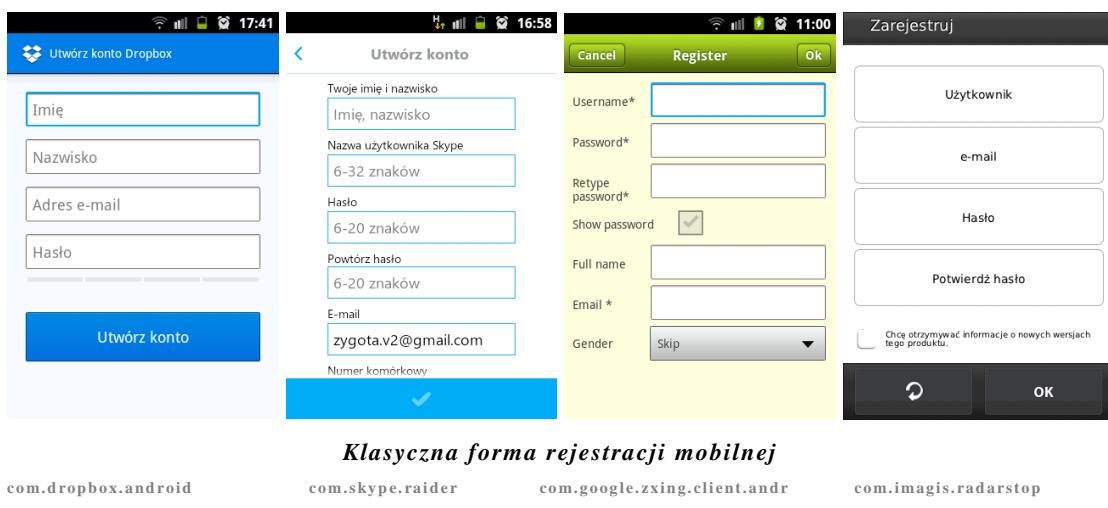
Wykres realizacji wskazówek materiałów szkoleniowych Google dla reklam banerowych dla aplikacji powyżej 5mln instalacji

5.7. Rejestracja

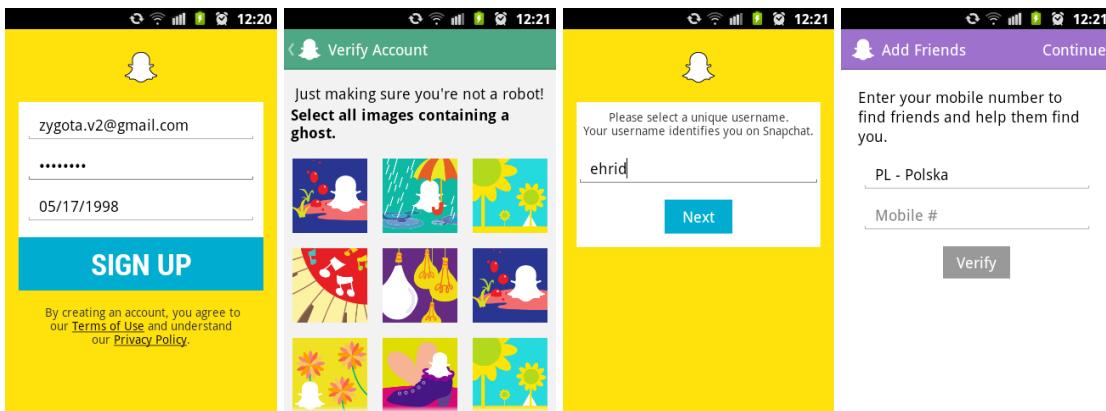
Do uzyskania pełni funkcjonalności serwisu wiele aplikacji wymaga zalogowanego użytkownika, większość z nich oferuje możliwość rejestracji poprzez aplikację. Zarejestrowanie polega na wypełnieniu szeregu danych, może być to zaledwie kilka jak i kilkanaście danych. W rzadkich przypadkach aplikacja przekierowuje na stronę Internetową.



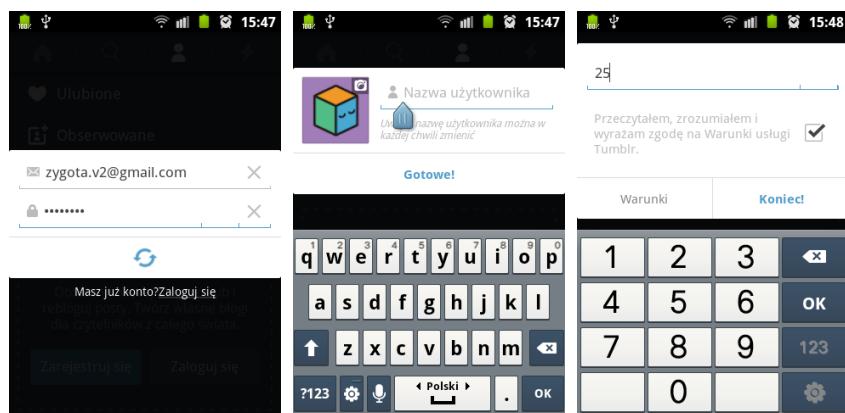
Najczęściej spotykaną klasyczną formą jest wbudowany widok umożliwiający wprowadzenie danych i utworzenie konta. Rejestracja tego typu jest najczęściej jednoetapowa i nie wymaga wiele czasu od użytkownika.



Rzadziej spotykaną formą rejestracji jest rejestracja wieloetapowa, pozornie użytkownik musi wypełnić kilka pól, jednak po kliknięciu na przycisk rejestracji musi jeszcze podać szereg danych zanim rejestracja dobiegnie końca.

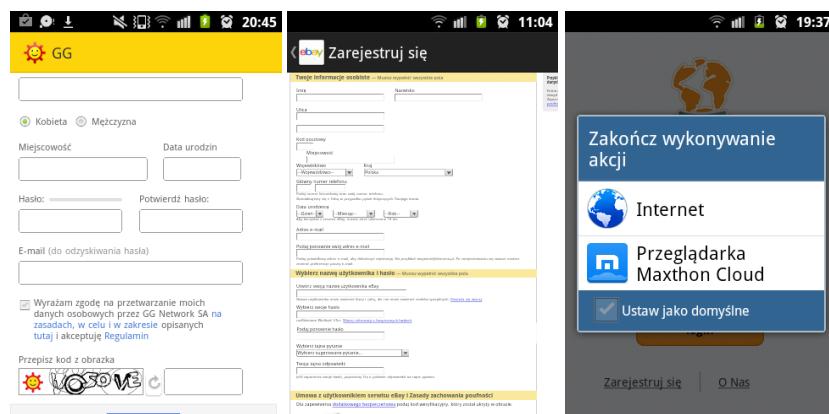


Część rejestracji wieloetapowej aplikacji com.snapchat.android



Część rejestracji wieloetapowej aplikacji com.tumblr

Również spotykana jest rejestracja z udziałem stron internetowych, taka forma przebiega w osadzonej przeglądarce w aplikacji albo z użyciem zewnętrznych przeglądarek zainstalowanych na urządzeniu mobilnym.



Rejestracja z użyciem stron internetowych

pl.gadugadu

com.ebay.mobile

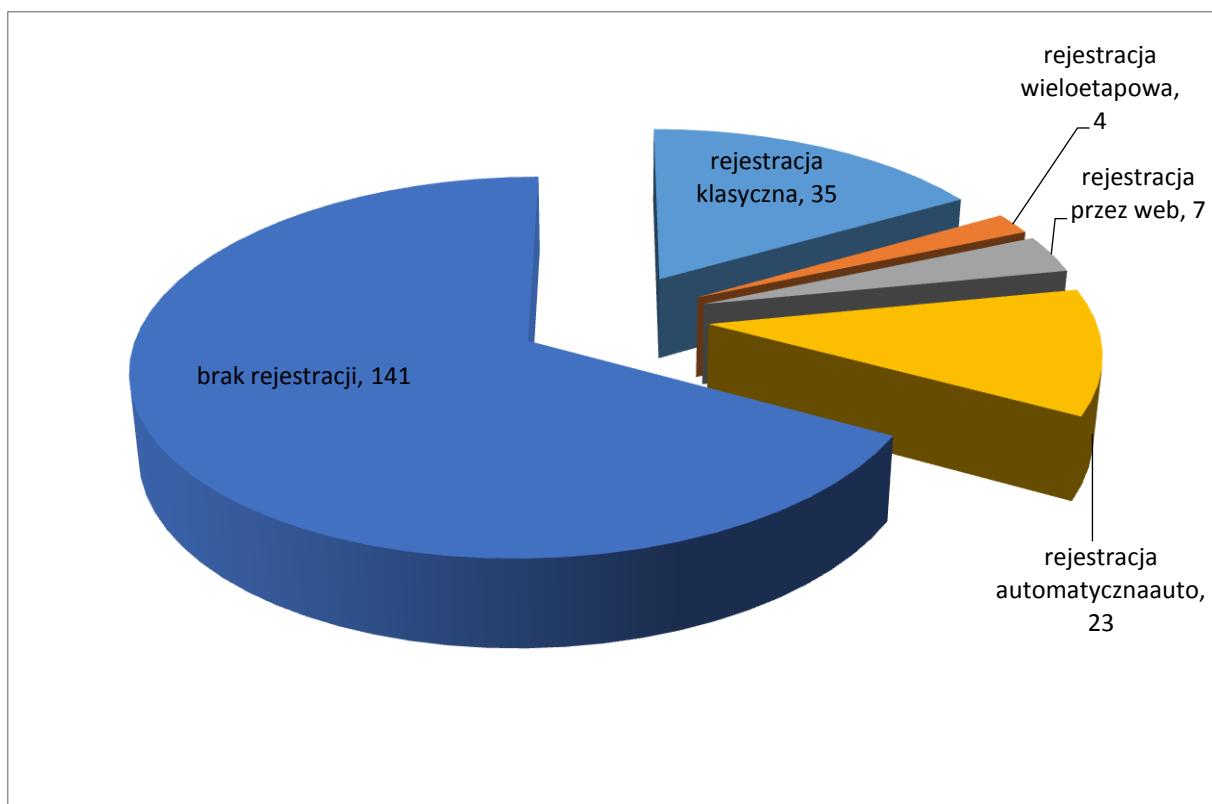
com.couchsurfing.mobile.a
ndroid

Dyskusyjną kwestią jest ilość pól wymaganych do rejestracji. Niektóre aplikacje wymagają tylko kilku pól, lecz są też takie, które wymagają ich znacznie więcej. Najwięcej pól wymagał do

wypełnienie com.ebay.mobile – 23. W aplikacjach mobilnych możliwa jest rejestracja z wykorzystaniem jednego pola, w tym celu można wykorzystać dane o zalogowanym użytkowniku do usług Google lub też za pomocą aplikacji Facebook. Możliwa jest też rejestracja automatyczna z wykorzystaniem danych zalogowanego użytkownika Google.

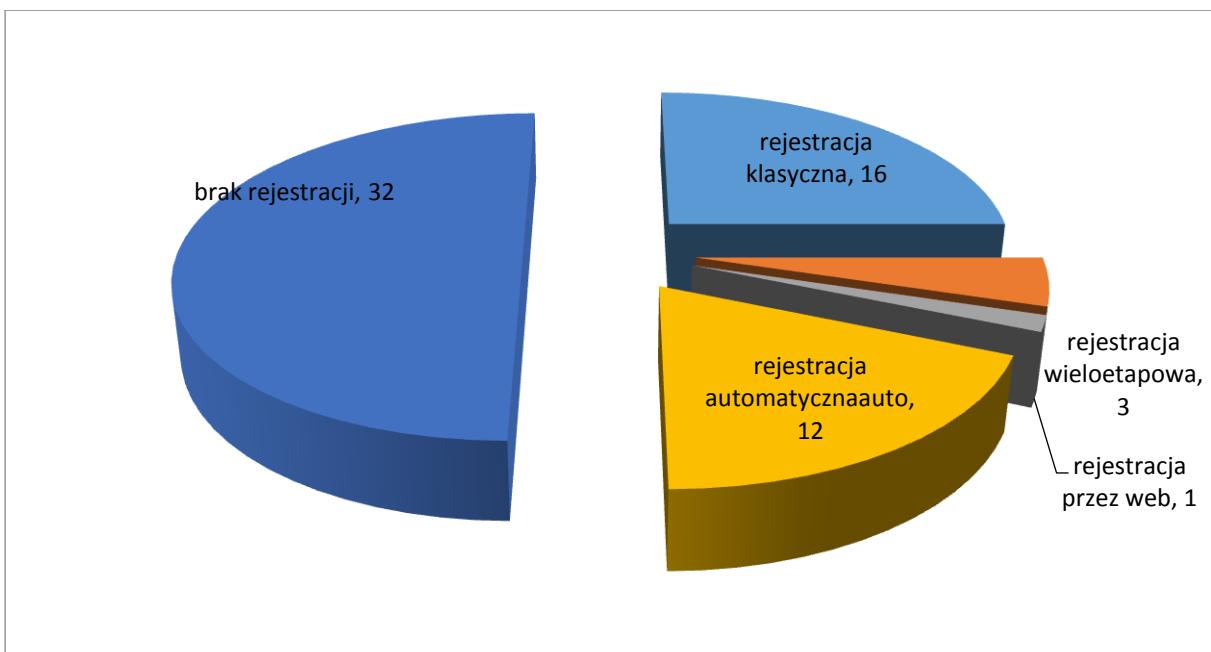
Tylko jedna aplikacja com.fsp.android.friendlocator wykorzystała dostępne informacje aktualnie zalogowanego użytkownika Google do automatycznego wypełnienia pól, jednocześnie oszczędzając czas użytkownikowi.

Z przeprowadzonego przeglądu mobilnych aplikacji biznesowych wynika, że zdecydowana większość rejestracji jest jednoetapowa, zaprojektowana bezpośrednio w aplikacji. Część też rejestruje użytkownika automatycznie, wzięto tutaj pod uwagę łączenie poprzez konto z aplikacją Facebook oraz kontem Google na telefonie. Pozostałe typy na badanej próbce okazały się sposobami marginalnymi.



Wykres wykorzystanych rodzajów rejestracji

W przypadku aplikacji, które były zainstalowane 5mln razy i więcej rejestrację umożliwia ponad 50% aplikacji. Znacznie popularniejsza jest też rejestracja automatyczna między innymi z użyciem danych kont Google i Facebook. Wartym zauważenia jest także fakt, że tylko jedna aplikacja posiadająca poniżej 5mln instalacji używała rejestracji wieloetapowej.

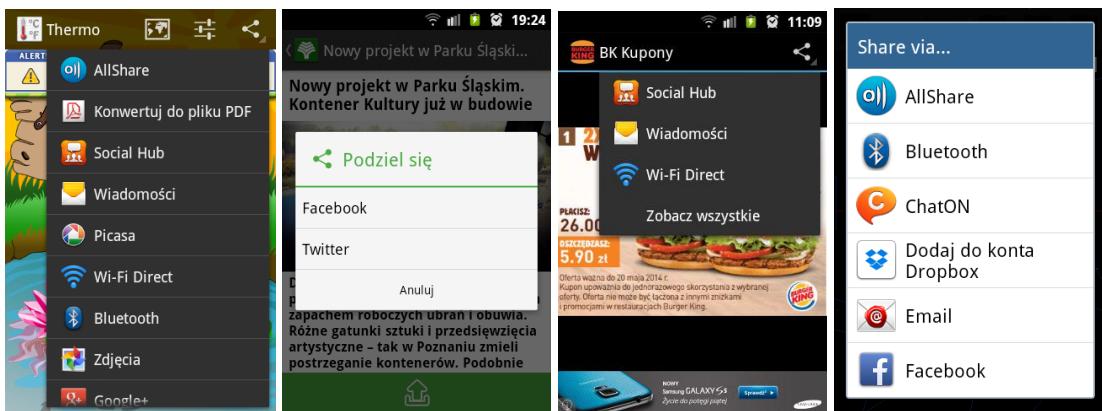


Wykres wykorzystanych rodzajów rejestracji dla aplikacji powyżej 5mln instalacji

5.8. Social media

Jednym ze spotykanych elementów w aplikacjach mobilnych jest możliwość udostępnienia w serwisach społecznościowych jak Facebook lub formach komunikacji typu Bluetooth przeglądanej treści lub aplikacji.

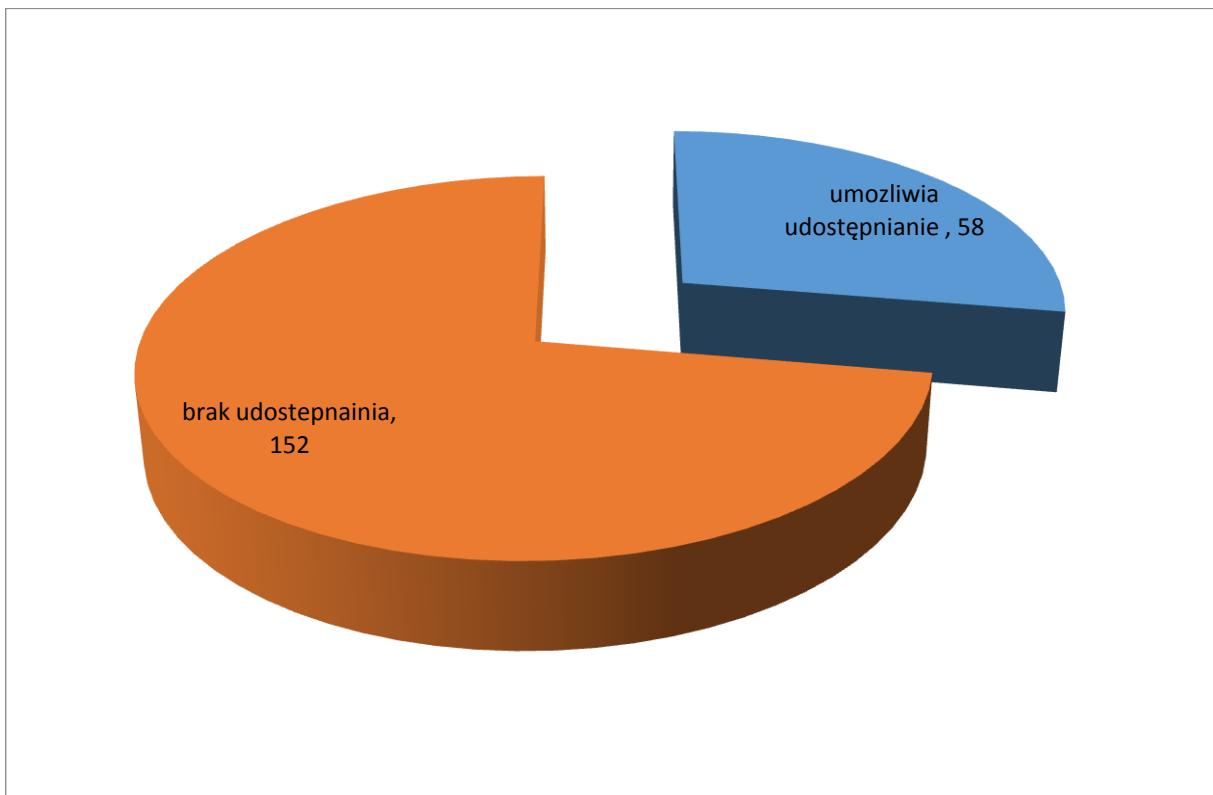
Po przyciśnięciu przycisku odpowiedzialnego za udostępnienie treści, aplikacja umożliwia wybranie wybrać medium, w którym zostanie dana treść udostępniona. Taki wybór może mieć miejsce poprzez odpowiednio przygotowane menu, komunikaty lub z wykorzystaniem natywnie przygotowanego do tego menu. Własnoręcznie przygotowane menu oraz komunikaty dostarczają większą kontrolę nad wyglądem aplikacji, lecz wymagają także własnoręcznego przygotowania obsługi usług, w których ma nastąpić udostępnianie. Alternatywą jest użycie natywnego menu zawierającego wszystkie dostępne media, możliwym jest także ograniczenie tej listy do wybranych pozycji. Wadą takiego rozwiązania jest mniejsza kontrola nad wyglądem aplikacji.



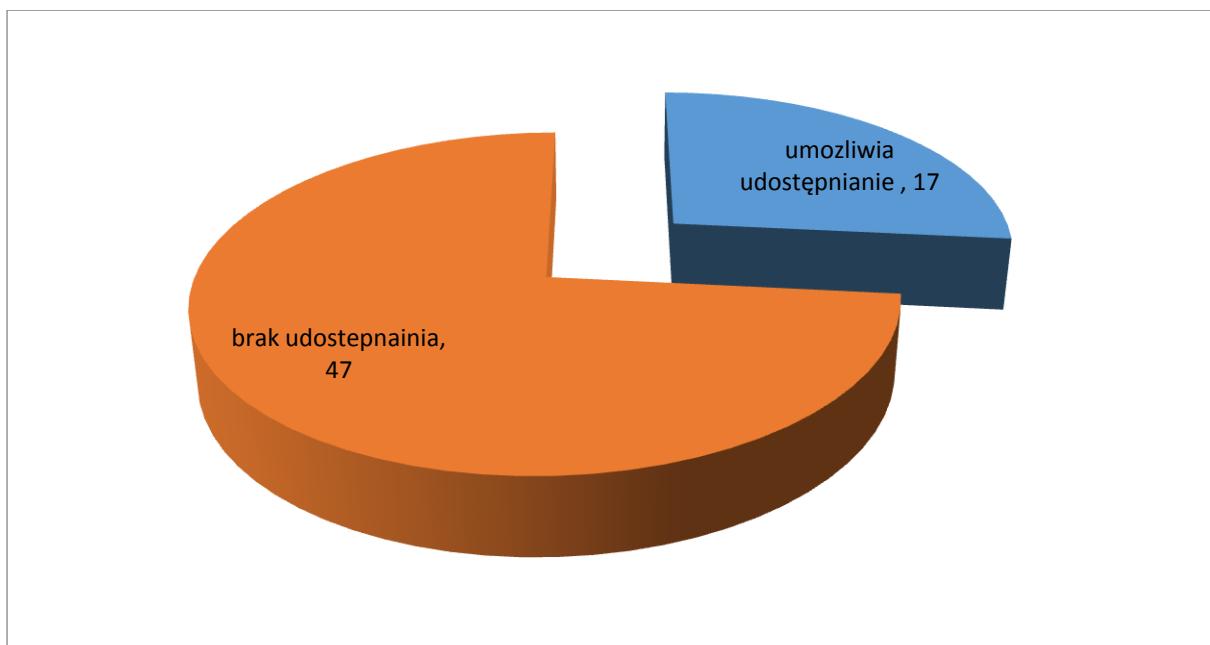
Przykłady menu udostępniania

com.xiaad.android.thermometer
com.altconnect.android.par
pl.com.androidapps.burgerki
com.escapestgames.starchart
tertrial kslaski nggutscheine

Z poniższego wykresu wynika, że tylko niewiele ponad 28% (i 27% dla aplikacji powyżej 5mln) aplikacji umożliwia udostępnianie treści. Na ten fakt może wpływać trudność w przygotowaniu odpowiedniej treści do udostępnienia, może to być także potrzeba uproszczenia aplikacji.



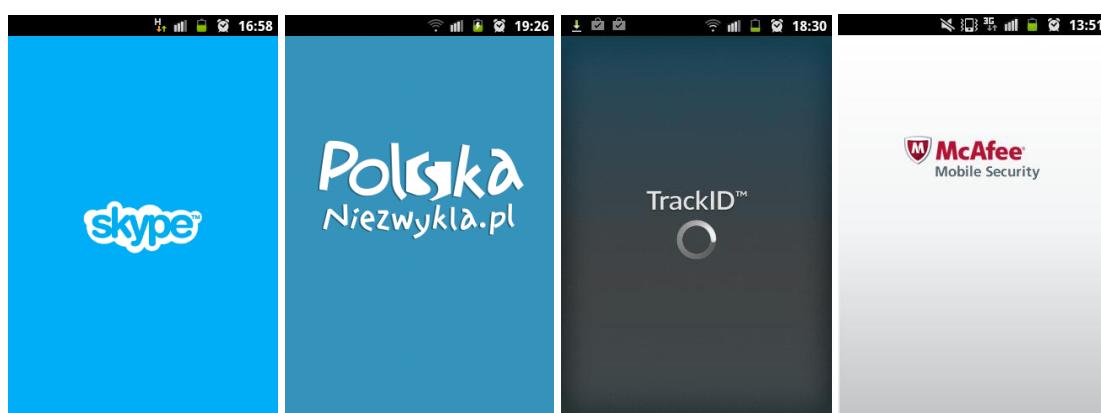
Wykres wykorzystania udostępniania



Wykres wykorzystania udostępniania dla aplikacji powyżej 5mln instalacji

5.9. Splashscreen

Splashscreen (z ang. ekran powitalny) jest widokiem wyświetlany jedynie przy starcie, po czym możliwość do jego powrotu jest jedynie poprzez ponowne uruchomienie aplikacji. Ten widok zazwyczaj trwa krótki czas i może on być wykorzystany do prezentacji między innymi logo aplikacji lub sponsora, ładowania zasobów potrzebnych do działania lub też próby automatycznego logowania z zapamiętanych danych.



Przykłady użycia splashscreen

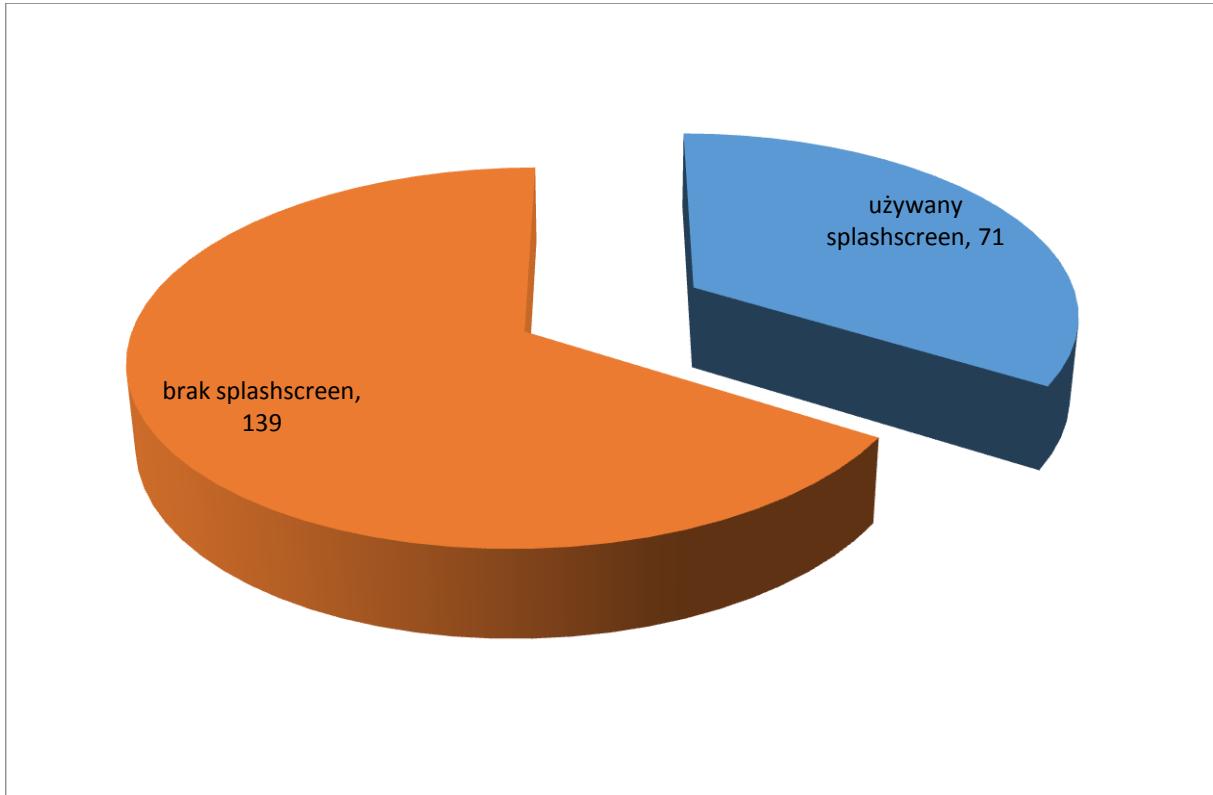
com.skype.raider

com.netmart.pnandroid

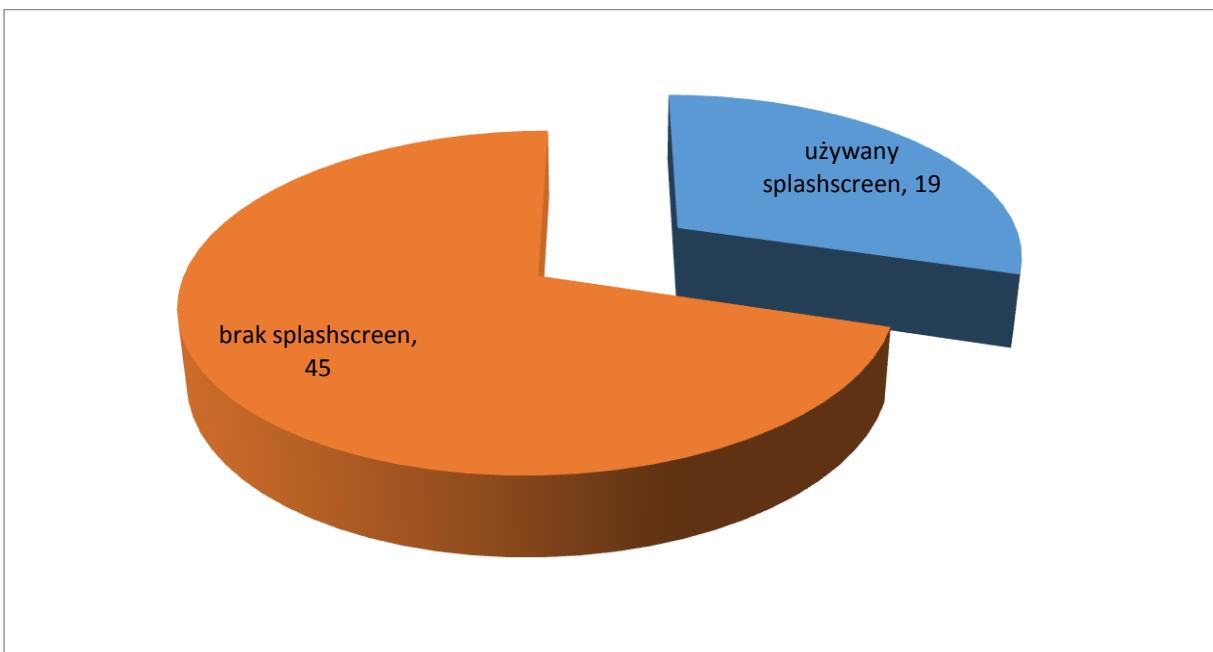
com.sonyericsson.trackid

com.wsandroid.suite

Z przeprowadzonych badań wynika, że jedna trzecia aplikacji używa Splasscreen. Użycie tego ekranu nie stanowi nawet połowy wybranych aplikacji prawdopodobnie dlatego, że nie istnieje natywny sposób tworzenia takich ekranów. Ekrany Splasscreen w platformie Android zostały zapożyczone z platformy iOS. [17] W przypadku zawężonego grona aplikacji występuje zbliżona sytuacja.



Wykres wykorzystania Splasscreen

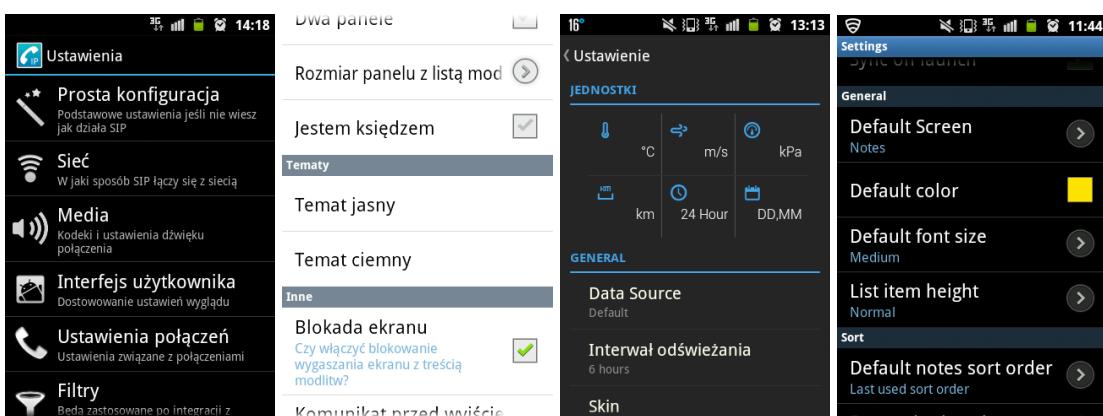


Wykres wykorzystania Splasscreen dla aplikacji powyżej 5mln instalacji

5.10. Ustawienia

Aplikacje często zawierają ustawienia, które pozwalają użytkownikom modyfikowanie ich zachowania. Zmiana takich ustawień odbywa się poprzez ekran ustawień, a przykładem ustawienia może być specyfikacja jak długi tekst może występować na stronie, stanowić o włączeniu lub wyłączeniu dźwięku.

Google, podobnie jak w przypadku wielu innych aspektów programowania na platformę Android OS, udostępnia możliwość wykorzystania wbudowanego systemu ustawień, wykonując sporą część pracy za programistę. Wygląd takich ustawień różni się w zależności od wersji platformy i może oferować zmianę takich typów parametrów jak boolean, float, int, long, String oraz String set. Najczęściej stosowanymi sposobami zmiany ustawień są CheckBox, rozsuwana lista i pole do wprowadzania tekstu. [18]



Ustawienia z wykorzystaniem API Google

com.csipsimple

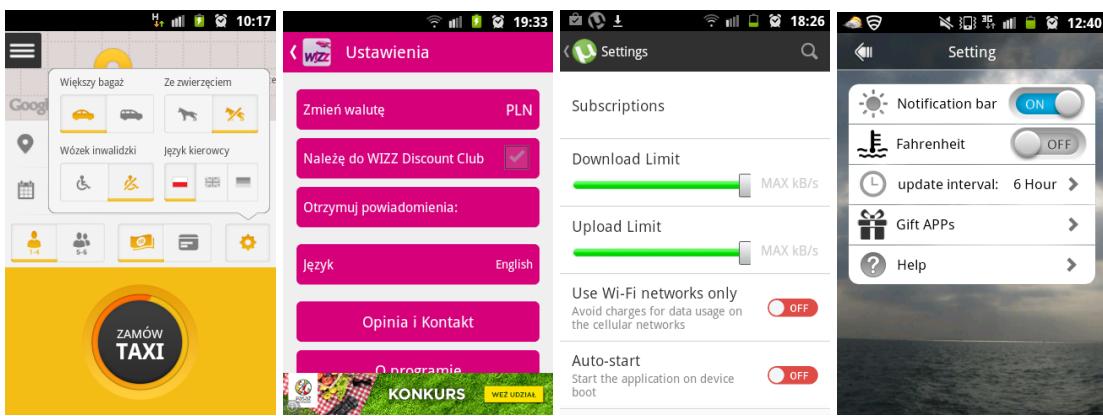
com.kodeksy

mobi.infolife.ezweather

com.socialnmobile.dictapps.notepad.co

lor.note

W przypadku potrzeby użycia niestandardowego rodzaju ekranu ustawień programista musi sam utworzyć wszystkie widoki, zwiększa to nakład pracy pogrzebny do stworzenia aplikacji, lecz jednocześnie daje większą kontrolę. W takim przypadku istnieje możliwość między innymi użycia odrębnych stylów elementów klikalnych, zastosowania reklam czy umieszczenie menu aplikacji dostępnego z poziomu ekranu ustawień.



Ustawienia projektowane bez API Google

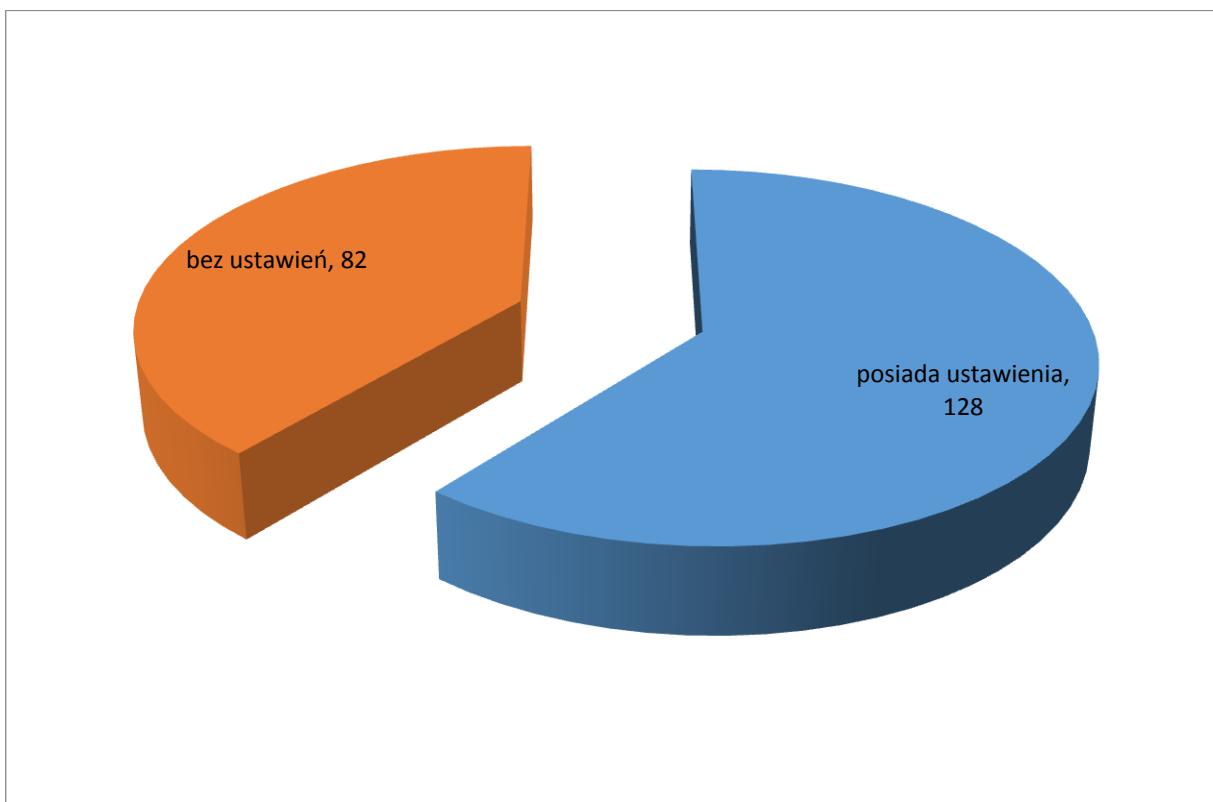
pl.taxi5

com.biskis.wizzair

com.utorrent.client

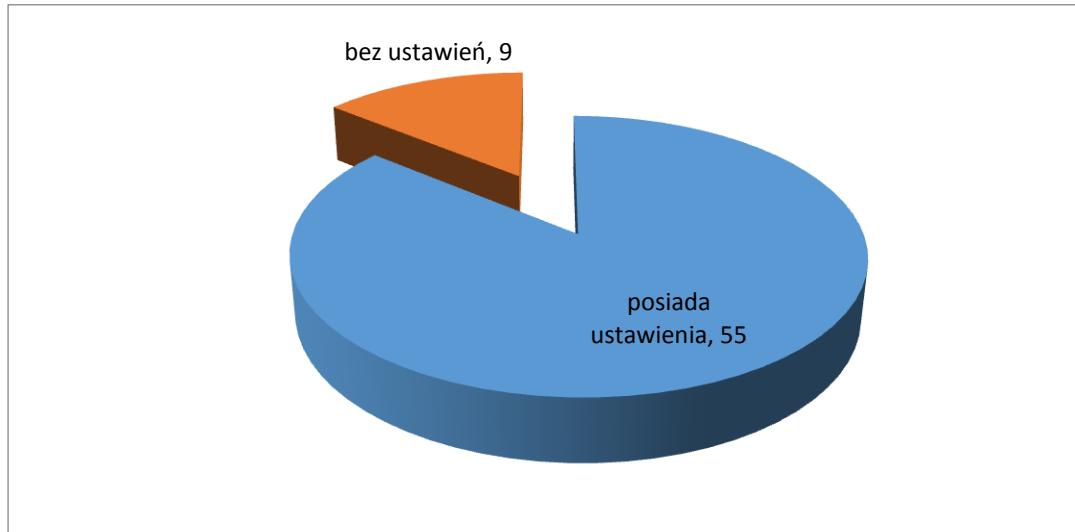
yong.tools.life.weather

Z przeprowadzonego przeglądu aplikacji mobilnych wynika, że zdecydowana większość, bo aż 61% aplikacji używa ustawień. W większości przypadków niestosowania ustawień treść aplikacji nie potrzebowała zmian ustawień albo z możliwych ustawień przydatnych użytkownikowi zastosowanie miałaby tylko jedna.



Wykres wykorzystania ustawień

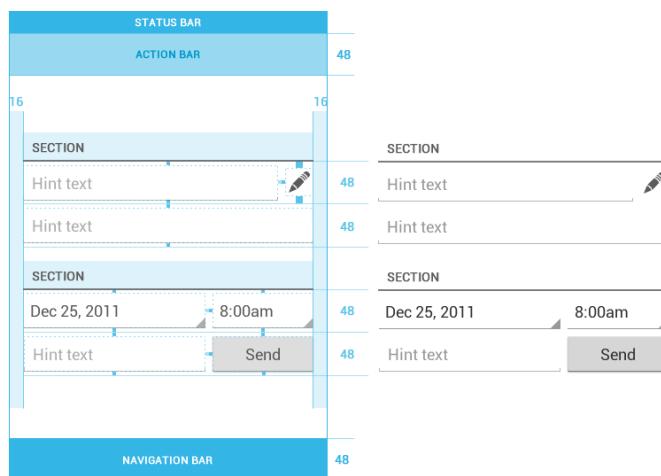
W przypadku aplikacji powyżej 5mln instalacji można zaobserwować, że jedynie 14% aplikacji nie posiada ustawień. Pozostałe oferują bardzo dużo możliwości dostosowania aplikacji do swoich wymagań.



Wykres wykorzystania ustawień dla aplikacji powyżej 5mln instalacji

5.11. Wielkość przycisków

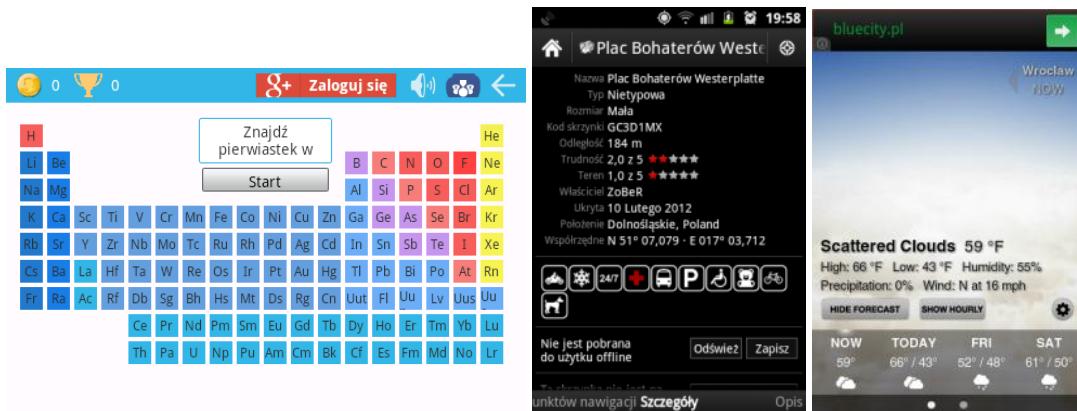
Użytkownik wchodzi w interakcje z UI opartym na dotyku za pomocą swoich palców. Tak, więc elementy interfejsu muszą być dostatecznie duże, aby nie frustrować klienta niezaplanowanymi akcjami i zbyt małymi przyciskami. W poradnikach Google możemy wyczytać, że aby elementy klikalne nie sprawiały problemów z kliknięciem nie mogą być mniejsze niż 7mm niezależnie od rodzaju urządzenia. A zalecany rozmiarem ramki jest 48dp, co odpowiada około 9mm na urządzeniu, z odległością od krawędzi ramki 8dp lub też 1,5 mm. [19] W przypadku urządzeń Apple, ta wielkość wynosi 44 piksele, co jest rozmiarem podanym bezpośrednio przez firmę Apple. [21]



Przykład prawidłowego zastosowania zalecanych metryk wg Google[19]

Przykład prawidłowego zastosowania zalecanych metryk wg Apple[21]

Wśród przebadanych aplikacji zdecydowana większość trzymała się zaleceń Google (87%), lecz używanie niektórych aplikacji wymagało większego skupienia ze względu na wielkość przycisków. Jedna z aplikacji com.bk.advance.chemik zastosowała dużą ilość małych przycisków w malej odległości od siebie, co sprawiało znaczące problemy w wybraniu odpowiedniego.

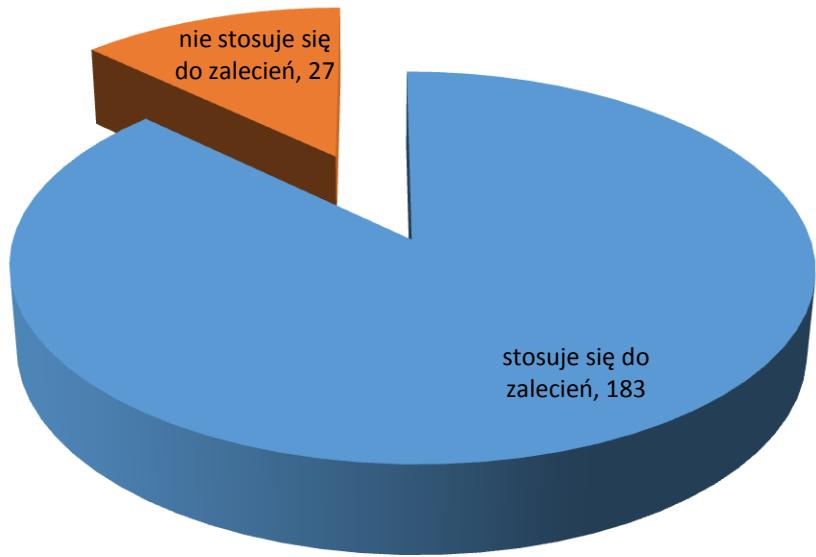


Błędne zastosowanie zalecanych metryk

com.bk.advance.chemik

geo.geocaching

com.vimov.weatherhd.android.free



Wykres zastosowania się do zaleceń Google

Bardzo podobnie, bo 89% aplikacji z zawężonego grona trzymała się zaleceń Google. W każdym z przypadków niestosowania się do zaleceń, aplikacja potrzebowała zmieścić bardzo dużo możliwości na małej powierzchni.



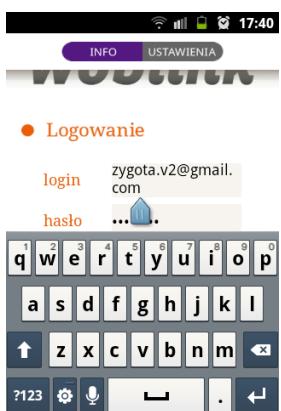
Wykres zastosowania się do zaleceń Google dla aplikacji powyżej 5mln instalacji

5.12. Wprowadzanie danych

W szerokiej gamie urządzeń używających Android OS znajdują się zarówno urządzenia z klawiaturą fizyczną, klawiaturą wyświetlana na ekranie jak i obiema na raz. Użycie klawiatury fizycznej w urządzeniach uniemożliwia jej modyfikację, natomiast modyfikacje są możliwe w przypadku klawiatur wyświetlanych na ekranie.

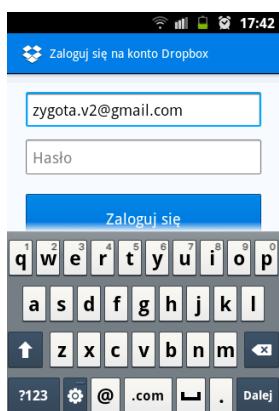
Definiując pola wprowadzania tekstu znajdujące się w aktualnie wyświetlonym widoku istnieje możliwość zdefiniowania kolejności tych pól lub akcji związanej z wypełnieniem danego pola, często w przypadku ostatniego pola. W przypadku dokonania takiej definicji przycisk nowej linii na klawiaturze zostanie zastąpiony przyciskiem odpowiedniej akcji, na przykład „dalej” w przypadku zdefiniowanej kolejności przechodzenia po wielu polach. Dobrym przykładem zastosowania jest proces logowania, gdzie po wprowadzeniu nazwy użytkownika nie odciągając palców od klawiatury możemy przejść do pola wprowadzania hasła, a następnie zalogować się.

Istnieje także możliwość zastosowania własnych klawiatur, jednak w większości przypadków jest to nieuzasadnione i o wiele lepszym rozwiązaniem jest korzystanie z wbudowanej. Przykładem dobrego zastosowania własnej klawiatury jest zaawansowany kalkulator, gdzie możliwość szybkiego wprowadzenia operacji matematycznych stanowi spory atut aplikacji. Na 210 przebadanych aplikacji tylko 9 korzystały z własnoręcznie zdefiniowanych klawiatur.



Przykład użycia standardowej klawiatury

com.woblink.reader

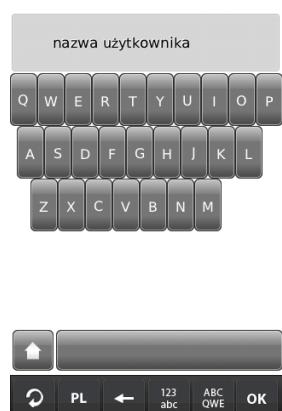


com.dropbox.android



Przykład użycia własnych klawiatur

us.mathlab.android



com.imagis.radarstop

Używanie klawiatur dostarczonych przez system operacyjny daje możliwość zdefiniowania metody wprowadzania danych. Standardowo dostępne są trzy rodzaje wprowadzania danych numeryczna, tekstowa i głosowa. W przypadku wybrania tekstowej metody wprowadzania można zdefiniować również wiele elementów takich jak autokorekta i autouzupełnianie. [20]

6. Analiza przeglądu mobilnych aplikacji biznesowych

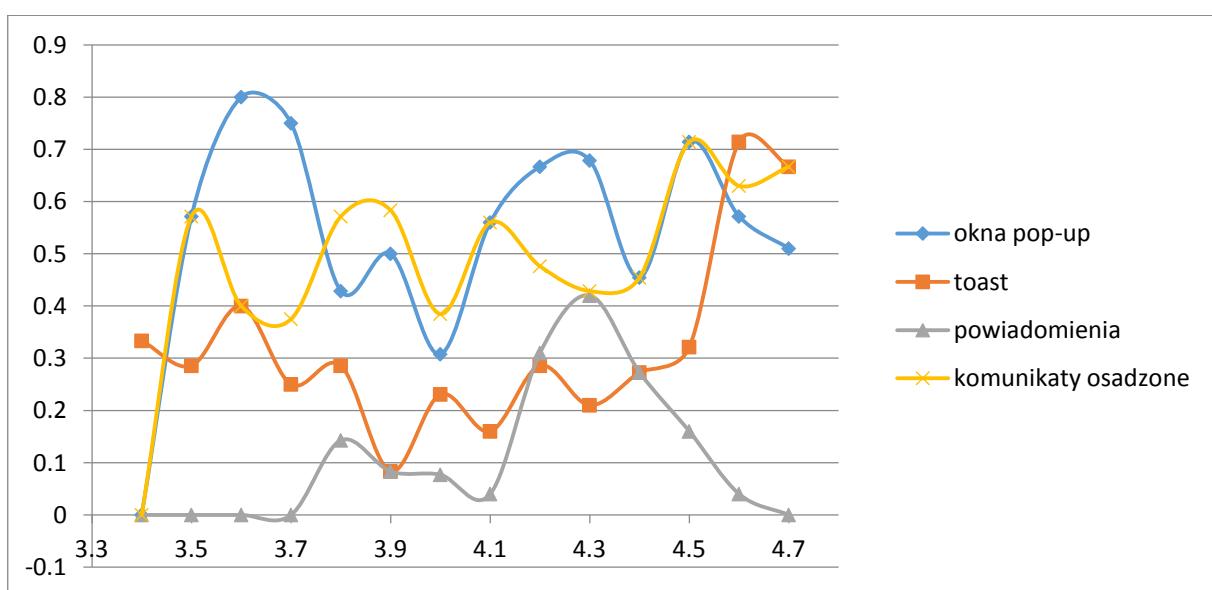
Ten rozdział poświęcony jest analizie zebranych danych przedstawionych w poprzednim rozdziale. Nastąpił podział aplikacji ze względu na ocenę na Google play, następnie utworzone grupy zostały znormalizowane tak, aby 0 oznaczało, że żadna aplikacja nie stosuje podanego kryterium, a 1 oznaczało, że wszystkie aplikacje z tej grupy to kryterium stosują. W poniższych zestawieniach odrzucono grupy, w których było mniej niż 10 aplikacji, tj. grupy powyżej oceny 4,7 oraz poniżej oceny 3,4.

Na podstawie przeprowadzonej analizy zostanie wysunięta teza, w jakich przypadkach stosowane i zalecane jest dane kryterium, co pozwoli na lepsze zaprojektowanie aplikacji potwierdzającej lub zaprzeczającej postawionym tezom.

6.1. Komunikaty

Pierwszą z analizowanych kwestii są rodzaje stosowanych komunikatów. Na poniższym wykresie można zobaczyć, że wraz ze wzrostem oceny rośnie stopień użycia komunikatów typ toast i komunikatów osadzonych, także dla zakresu ocen 4,1 do 4,4 bardzo popularne są komunikaty z użyciem mechanizmu powiadomień, ich obecność może być uciążliwa, co może wpływać na ich niskie użycie w wyżej ocenianych aplikacjach. Użycie komunikatów posługujących się oknami pop-up nie wpływa na ocenę aplikacji, ale w dużej części grup występują one powszechnie.

Na podstawie dostępnych danych można postawić tezę, że dobra aplikacja cechuje się użyciem natywnych komunikatów toast oraz duża część informacji jest też przedstawiana poprzez komunikaty osadzone.

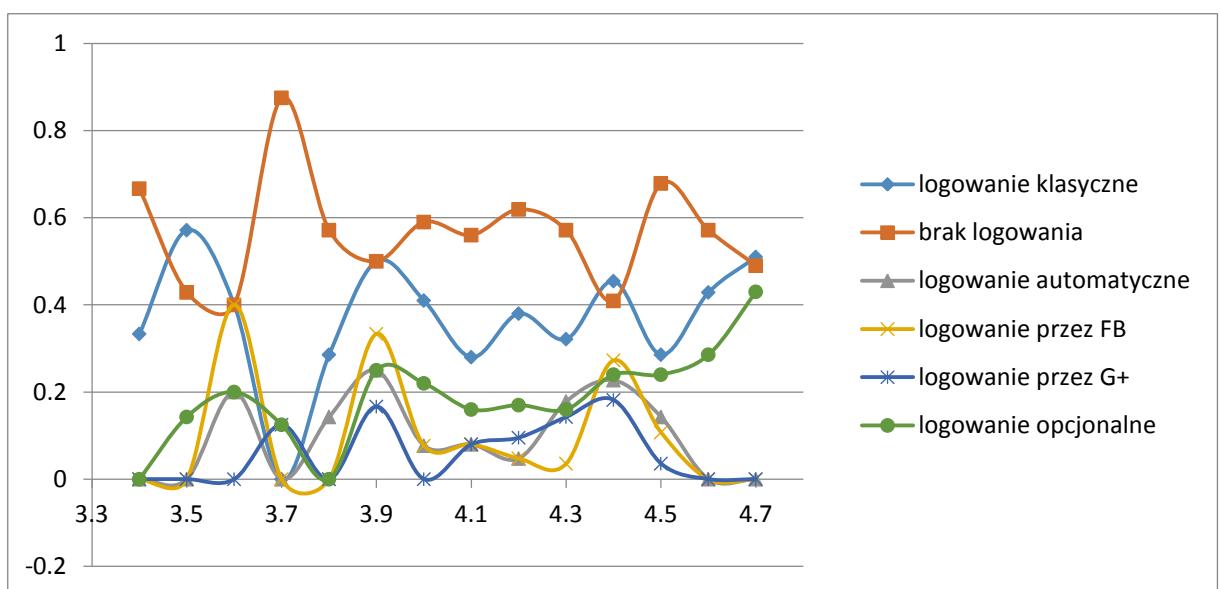


Wykres stosowania rodzajów powiadomień względem oceny aplikacji

6.2. Logowanie

Analiza kryterium logowania potwierdza dane wynikające z przeglądu, iż z niewielkimi odchyłami wśród badanych programów w większości grup aplikacje nie posiadają logowania, ale najczęstszym z sposobów jest podejście klasyczne. Wartym uwydawnienia jest fakt, że wraz ze wzrostem oceny rośnie ilość aplikacji traktujących proces uwierzytelniania, jako opcjonalny. Może to świadczyć o zadowoleniu z klientów, którzy zanim się zarejestrują mają możliwość przetestowania aplikacji lub korzystania z niej bez logowania się.

Na tej podstawie można postawić tezę, że dobra aplikacja, jeśli wymaga logowania powinna umożliwić korzystanie z aplikacji bez logowania.

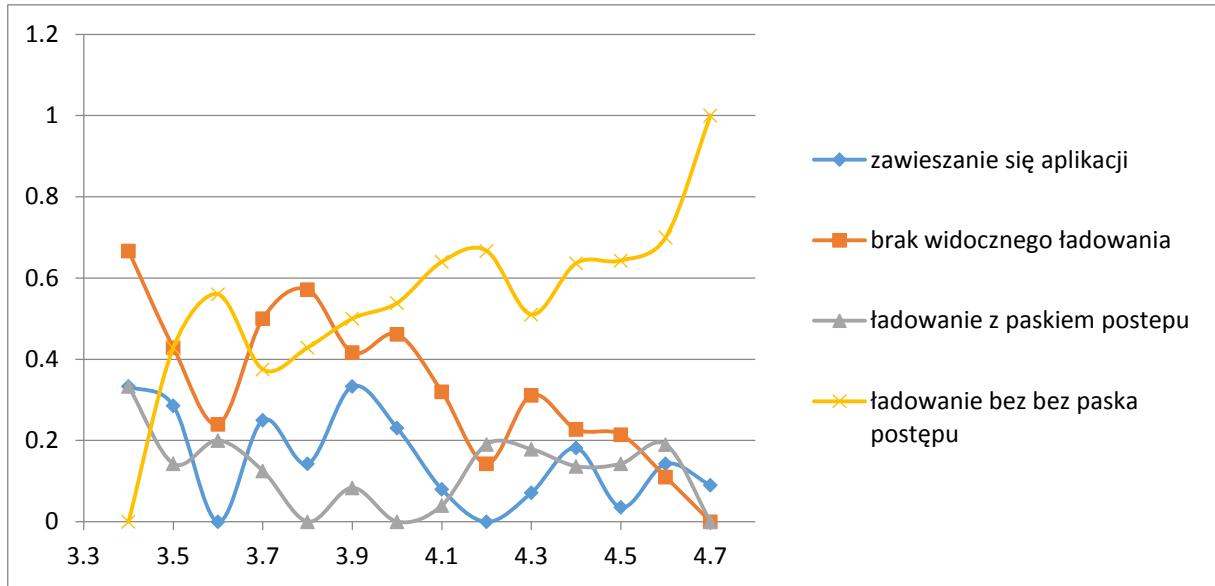


Wykres stosowania rodzajów logowania względem oceny aplikacji

6.3. Ładowanie treści

W przypadku ładowania treści warto zaznaczyć, wraz ze wzrostem oceny maleje użycie większości rodzajów informowania o pobieraniu danych. Wyjątkiem jest ładowanie bez paska postępu, świadczy to o tym, iż bardzo ważnym jest informowanie klienta o fakcie pozyskiwania treści, niskie wykorzystanie sposobu z użyciem paska postępu może oznaczać trudność w implementacji takiego rozwiązania oraz możliwe przekłamanie o czasie pozostałym do ukończenia czynności, co może negatywnie wpłynąć na ocenę aplikacji.

Na tej podstawie można postawić tezę, iż dobra aplikacja informuje o fakcie ładowania treści, ale nie używając paska postępu, jednocześnie bardzo rzadko lub w ogóle zawiesza działanie podczas ładowania treści.

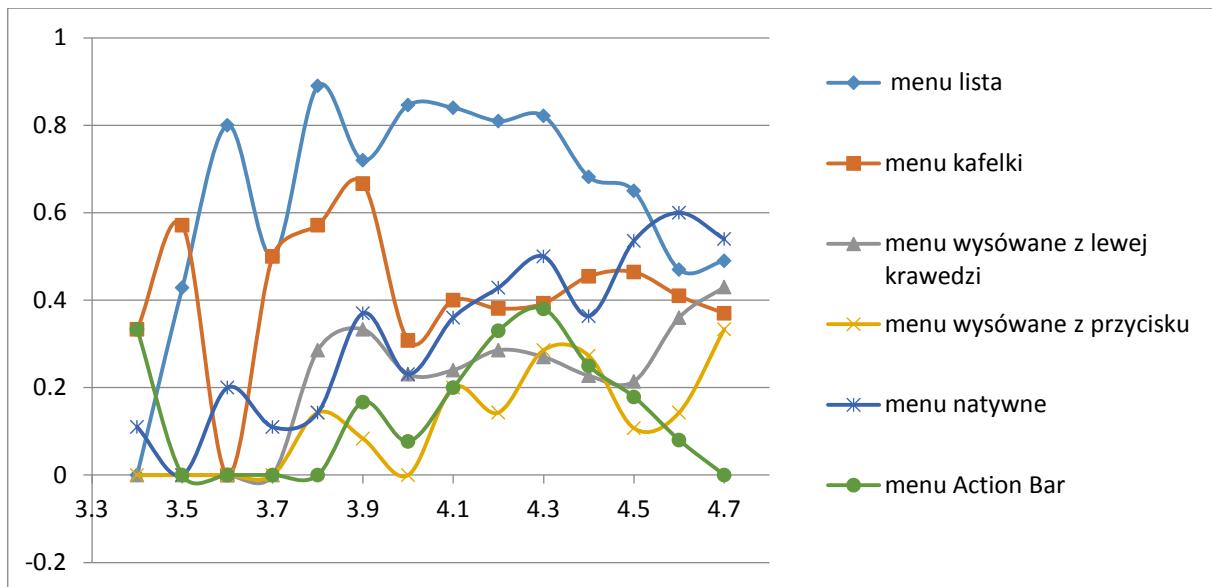


Wykres stosowania rodzajów ładowania treści względem oceny aplikacji

6.4. Menu

Analizując rodzaje menu, można zauważyc, że wraz ze wzrostem oceny częściej używane jest menu natywne androida. Ponadto kosztem bardziej klasycznych rodzajów takich jak lista i kafelki zyskują menu wysuwane z przycisku oraz wysuwane z lewej krawędzi. Przy zaznaczeniu, że użycie klasycznej listy jest wciąż najpopularniejsze. Dla przedziału ocen 4,1 do 4,4 na popularności zyskuje menu z użyciem technologii Action Bar.

Na tej podstawie można postawić tezę, że dobra aplikacja korzysta z natywnego menu, także używa bardziej nowoczesnych rodzajów takich jak menu wysuwane z krawędzi oraz przycisku.

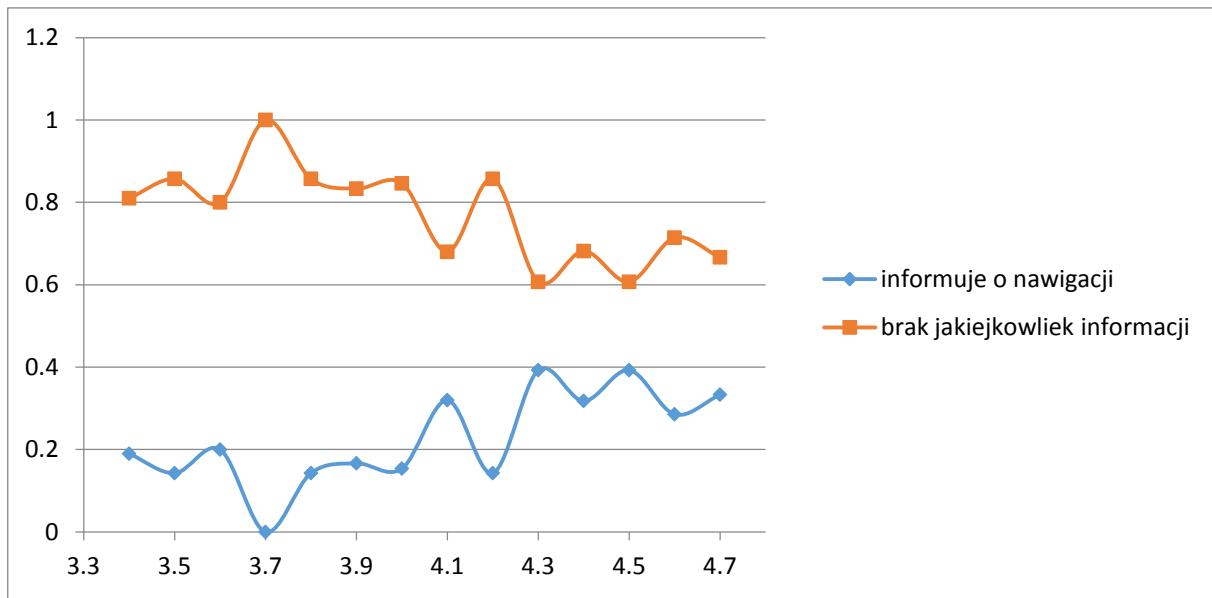


Wykres stosowania rodzajów menu względem oceny aplikacji

6.5. Nawigacja

Kryterium nawigacji jasno pokazuje, że większość aplikacji nie informuje w żaden sposób o położeniu albo możliwej nawigacji. Ale wraz ze wzrostem oceny informowanie o nawigacji jest coraz bardziej popularne.

Na tej podstawie można postawić tezę, że, jeśli istnieje taka potrzeba, dobra aplikacja informuje klienta o nawigacji, nie robiąc tego w sposób natarczywy.

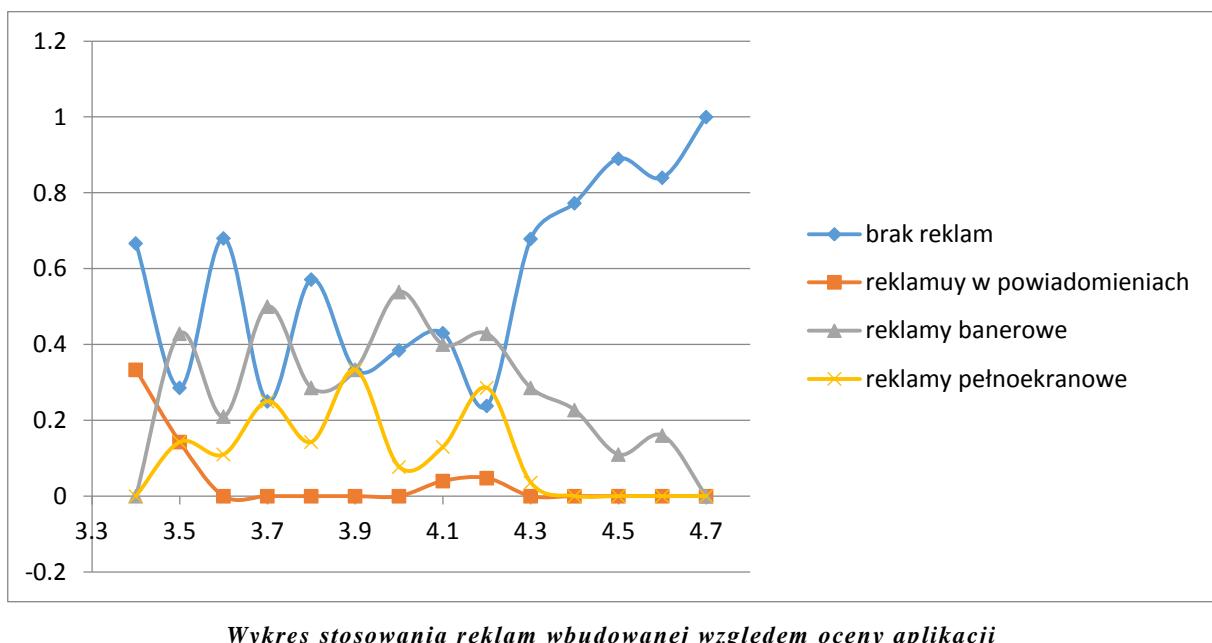


Wykres informowania o nawigacji względem oceny aplikacji

6.6. Reklamy

Reklamy są jednym ze sposobów monetaryzacji, ale zajmują one pewną powierzchnię, mogą też spowolnić działanie i te czynniki widać na poniższym wykresie, gdzie wraz ze wzrostem oceny zanika wykorzystanie reklam.

Na tej podstawie można postawić tezę, że dobra aplikacja to taka aplikacja, która nie używa reklam, albo używa ich w bardzo małym zakresie.

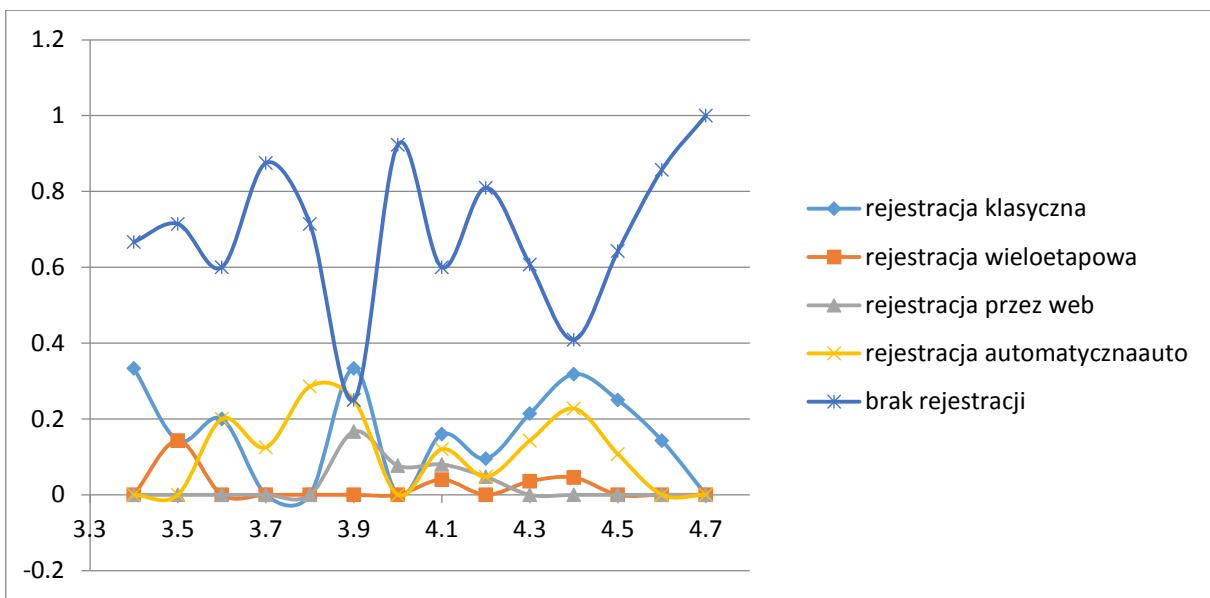


Wykres stosowania reklam wbudowanej względem oceny aplikacji

6.7. Rejestracja

Kryterium rejestracji, przedstawia, że w znaczącym stopniu rejestracja nie jest możliwa z poziomu aplikacji i wraz ze wzrostem oceny nieznacznie rośnie brak możliwości rejestracji. Użytkownicy w celu zalogowania muszą posiadać wcześniej utworzone konto w danym serwisie, wymog wypełnienia często dużej ilości danych może źle wpływać na ogólną ocenę aplikacji. Najczęstszym sposobem rejestracji jest rejestracja klasyczna, ale w przypadku wyboru użycia mechanizmu rejestracji czynnik ten nie ma znaczącego wpływu na ocenę aplikacji.

Na tej podstawie można postawić tezę, że dobre aplikacje w dużej mierze nie posiadają rejestracji.

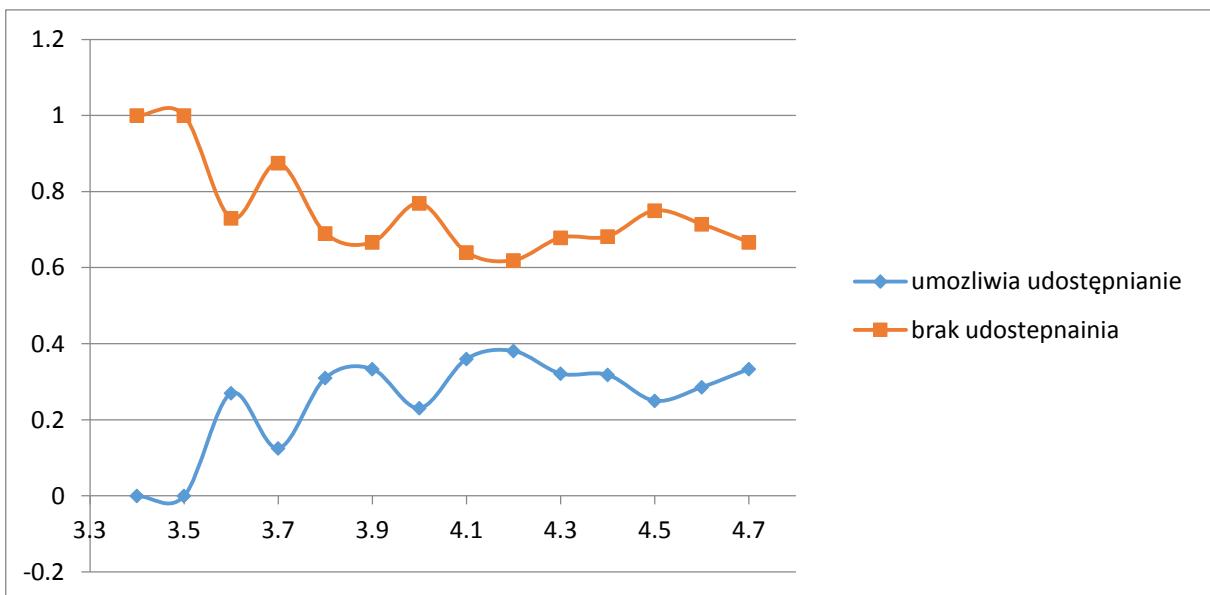


Wykres stosowania rodzajów rejestracji względem oceny aplikacji

6.8. Social media

Udostępnianie treści staje się coraz popularniejsze, ta forma aktywności spotykana jest już w wielu miejscach, podobnie jak w przypadku aplikacji mobilnych. Z dostępnych danych można zobaczyć, że wraz ze wzrostem oceny rośnie poziom wykorzystania możliwości udostępniania treści poprzez social media, jednak wciąż większa część nie daje takiej możliwości.

Na tej podstawie można postawić tezę, że dobra aplikacja to taka, która stara się wdrażyć możliwość dzielenia informacjami poprzez social media.

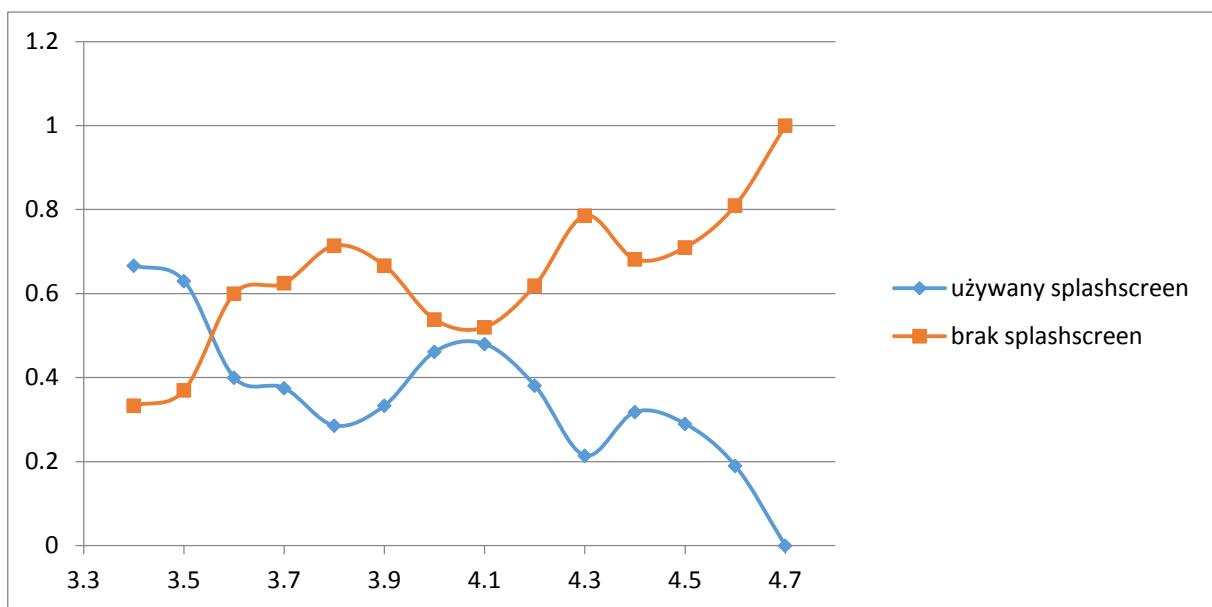


Wykres stosowania udostępniania względem oceny aplikacji

6.9. Splashscreen

Splashscreen to zapożyczony element z platformy iPhone i nie jest natywny na telefonach z Androidem, w wyniku, czego jest odradzany przez firmę Google. Użycie splashscreen powoduje spowolnienie uruchomienia aplikacji, co może być negatywnie oceniane przez klientów. Na poniższym wykresie można zaobserwować, że wraz ze wzrostem oceny maleje ilość użyć splashscreen.

Na tej podstawie można postawić tezę, że dobra aplikacja to taka, która nie używa splashscreen.

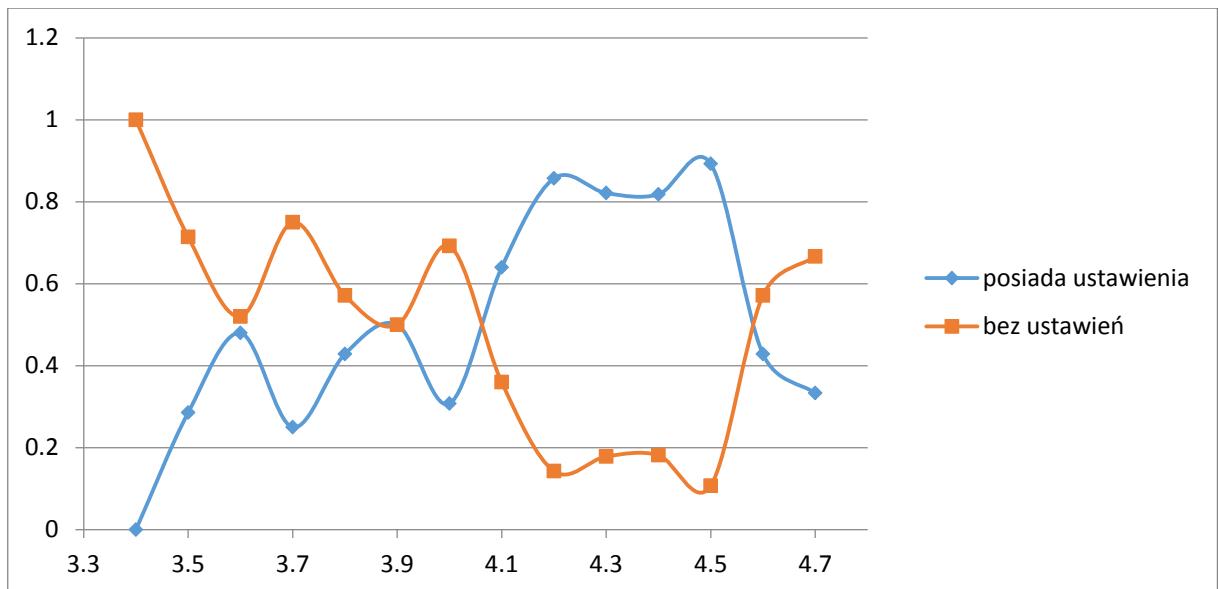


Wykres stosowania splashscreen względem oceny aplikacji

6.10. Ustawienia

Kryterium występowania ekranu ustawień posiada największy odsetek wykorzystania dla aplikacji z ocenami od 4,1 do 5,4. Jednak w przypadku najwyżej ocenianych programów zaobserwować można spory spadek. Spowodowane jest to tym, że badana grupa z tego zakresu w mniejszym stopniu miała zapotrzebowania na takie rozwiązanie, jednocześnie nie wprowadzając usilnie jakichkolwiek ustawień jak to miało miejsce w przypadku aplikacji z ocenami od 4,1 do 4,5.

Na tej podstawie można postawić tezę, że dobra aplikacja to taka, która używa ustawień, ale tylko wtedy, jeśli jest to uzasadnione.

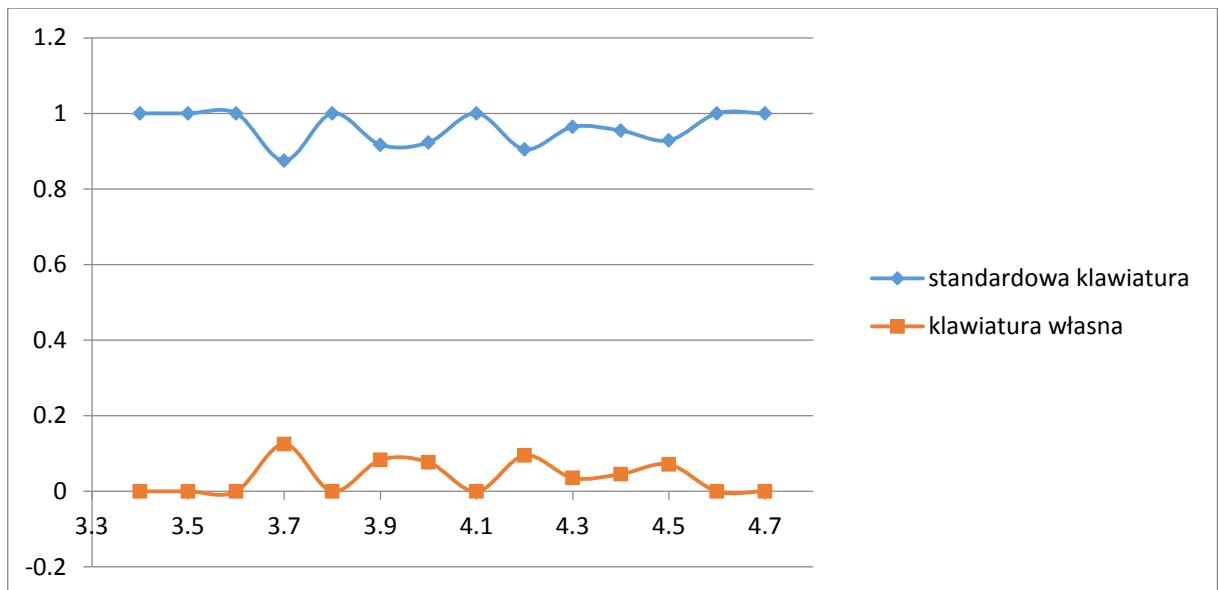


Wykres stosowania ustawień względem oceny aplikacji

6.11. Wprowadzanie danych

W przypadku wprowadzania danych, można zobaczyć na wykresie, że używanie własnych klawiatur jest elementem marginalnym i zdecydowana większość aplikacji używa klawiatur standardowych, która są znane użytkownikowi i proste w obsłudze.

Na tej podstawie można postawić tezę, że dobra aplikacja to taka, która używa standardowych klawiatur.



Wykres stosowania klawiatury wbudowanej względem oceny aplikacji

6.12. Podsumowanie postawionych tez

Rozdział ten umożliwił spójne spojrzenie na obecny stan mobilnych aplikacji biznesowych umożliwiając wygenerowanie jedenastu rad i zaleceń w projektowaniu oprogramowania na urządzenia przenośne.

Postawione tezy są następujące:

- Aplikacja powinna korzystać z komunikatów toast oraz informować za pomocą komunikatów osadzonych.
- Aplikacja używająca mechanizmu logowania, powinna umożliwić dostęp do treści bez logowania.
- Aplikacja powinna informować o fakcie ładowania treści oraz nie powinna z tego powodu zawieszać swojego działania.
- Aplikacja powinna korzystać z menu wysuwanego z krawędzi lub przycisku oraz korzystać z natywnego menu lub wykorzystywać fizyczny przycisk menu do otwierania menu właściwego.
- Aplikacja powinna informować użytkownika o możliwej nawigacji.
- Aplikacja nie powinna używać reklam.
- Aplikacja nie powinna posiadać możliwości rejestracji.
- Aplikacja powinna korzystać z możliwości dzielenia się informacjami poprzez social media.
- Aplikacja nie powinna używać spalschreen.
- Aplikacja powinna używać ekranu ustawień, ale tylko wtedy, jeśli jest to uzasadnione.
- Aplikacja powinna używać standardowych klawiatur.

7. Projektowanie mobilnej aplikacji biznesowej

Ten rozdział pracy poświęcony będzie relacji procesu projektowania aplikacji mobilnej w oparciu o przegląd dokonany w rozdziale poprzednim.

Projekt aplikacji biznesowej tradycyjnie rozpoczyna się od opracowania koncepcji struktury nawigacji, stworzenie aktywnego prototypu wspomnianej aplikacji umożliwiają takie programy jak Indigo Studio, NinjaMock, FluidUI i wiele innych. [27]

7.1. Badanie użyteczności wybranych narzędzi do prototypowania

Badanie użyteczności narzędzie do prototypowania będzie przeprowadzone przy użyciu Indigo Studio, NinjaMock oraz FluidUI. Oprogramowanie to zostanie opisane i porównane, co pozwoli zdecydować o wyborze narzędzia do prototypowania wybranej aplikacji.

7.1.1. FluidUI

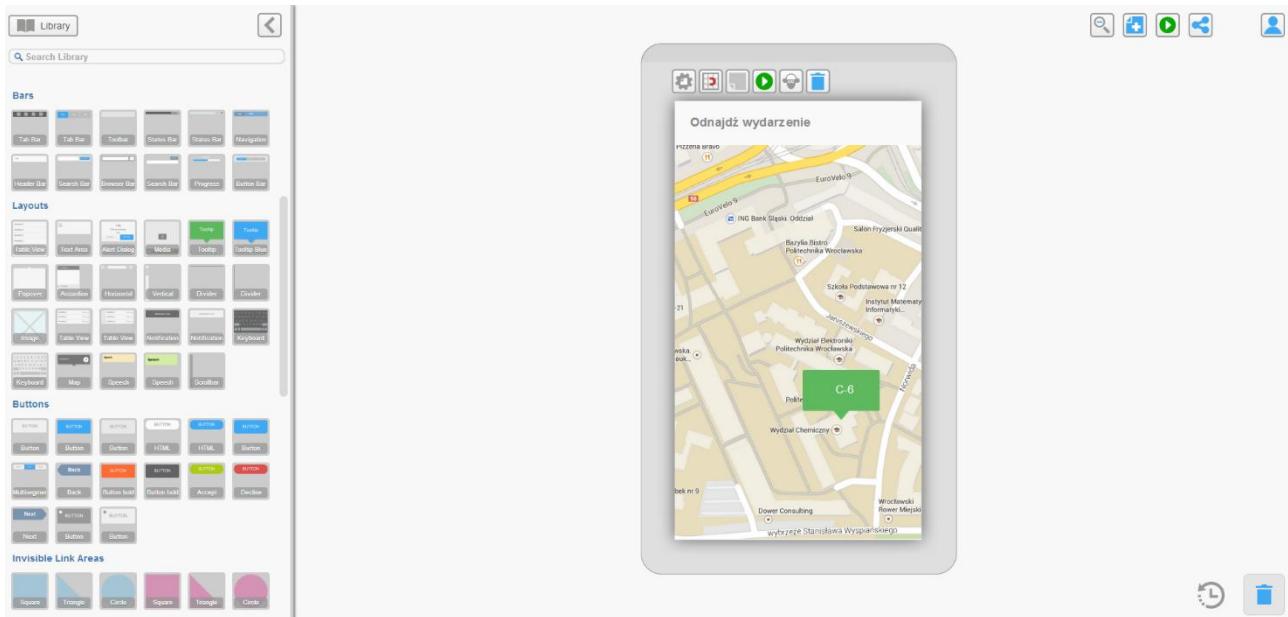
Pierwszym badanym oprogramowaniem do prototypowania jest FluidUI (<https://www.fluidui.com/>). Umożliwia ono szczegółowe prototypowanie aplikacji urządzeń mobilnych takich jak Android, Android Tablet, iPhone and iPad i szybkie zbieranie feedbacku.



Logo aplikacji FluidUI [48]

FluidUI posiada bardzo obszerną bibliotekę gotowych komponentów, ale natynie nie wspiera tworzenia własnych modułów, co okazuje się być sporym ograniczeniem, które powoduje wydłużenie czasu tworzenia prototypu z powodu potrzeby wyszukania odpowiedniego gotowego elementu widoku.

Niebywałym atutem tego narzędzia jest możliwość przetestowania zaprojektowanej aplikacji na telefonie bez potrzeby pisania ani jednej linijki kodu, co w połączeniu ze szczegółowością i ilością dostępnych komponentów daje wrażenie przeglądania gotowej aplikacji.



Zrzut ekranu oprogramowania FluidUI

Oprogramowanie jest bardzo proste w obsłudze, dzięki czemu opanowanie narzędzia stanowiło około 30 minut, niestety z powodu bardzo dużej liczby elementów i braku możliwości szybkiego tworzenia własnych, czas tworzenia aplikacji wyniósł niecałe 6 godzin i 30 minut.

7.1.2. NinjaMock

Kolejnym narzędziem jest NinjaMock (<http://nijamock.com/>). Oprogramowanie to umożliwia łatwe i szybkie projektowanie aplikacji mobilnych dla platform Android, iPhone, iPad, Windows Phone, Microsoft Surface, www oraz projektów dowolnych niezwiązanych ze wspieranymi platformami. [26]

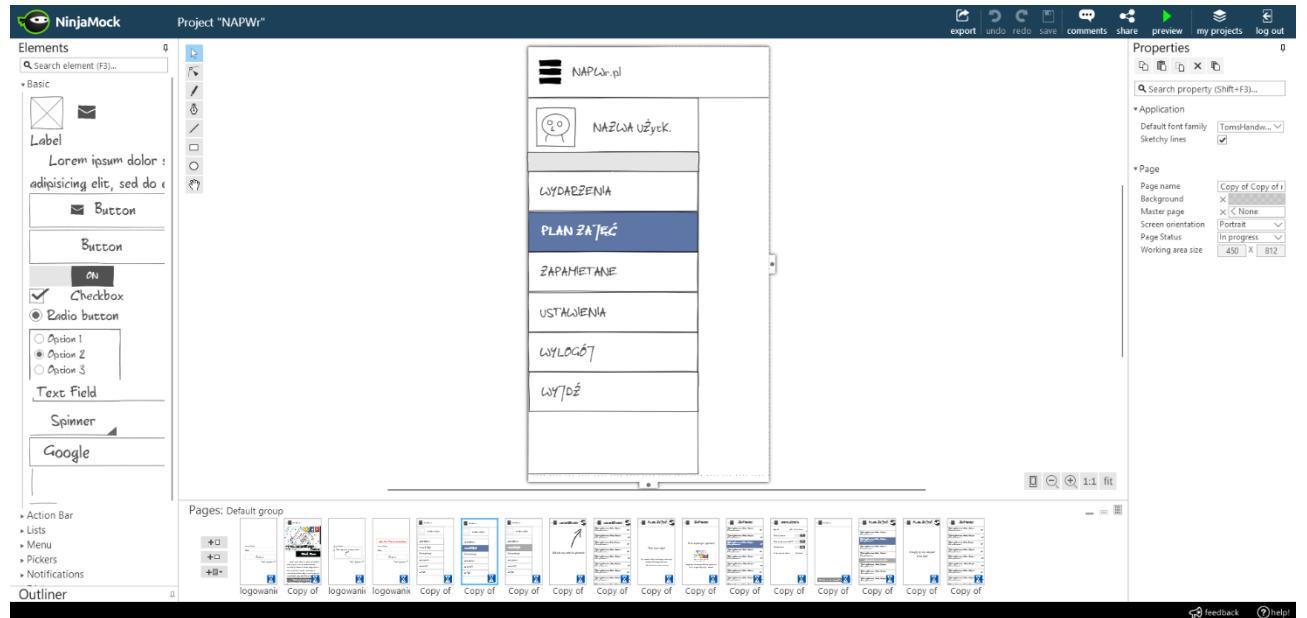


Logo aplikacji NinjaMock [26]

Wygląd NinjaMock przypomina szkice robione ka kartce, dzięki temu prototyp stworzony w oparciu o to narzędzie nie narzuca dokładnego wyglądu każdego z elementów, a stanowi jedynie zalecenia, co dynamicznie zmieniających się wymagań względem aplikacji jest ogromną zaletą.

Oprogramowanie to posiada zestaw gotowych elementów które dowolnie można modyfikować, są one pogrupowane i odpowiadają większości dostępnych standardowych elementów w kreatorze widoków w Android Developer Tools. Dodatkowo istnieje możliwość

rysowania własnych komponentów i zapisywania ich do późniejszego wykorzystania. Każdy z modułów poza standardowymi ustawieniami ma możliwość zdefiniowania przejścia po kliknięciu do innego widoku, co sprawia, że stworzony prototyp może przypominać zachowaniem prawdziwe oprogramowanie.



Zrzut ekranu oprogramowania NinjaMock

Narzędzie jest bardzo przyjemne i intuicyjne w obsłudze, czas nauki wykorzystania pełni możliwości był w tym przypadku niezwykle szybki, bez żadnych poradników wynosił on niecałą godzinę. Natomiast wykonanie prototypu zajęło 4 godziny i kilka minut.

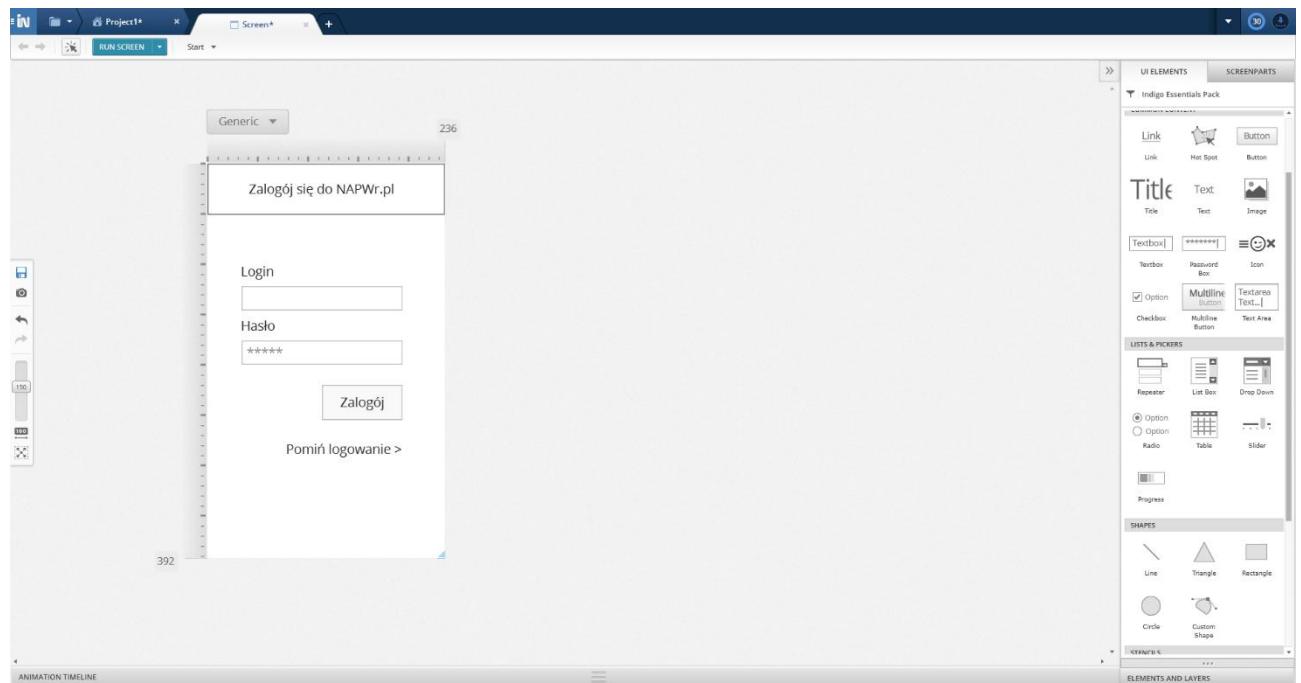
7.1.3. Indigo Studio

Ostatnim z badanych narzędzi do prototypowania jest Indigo Studio (<http://www.infragistics.com/products/indigo-studio>). Oprogramowanie to pozwala na prototypowanie funkcjonalnego i animowanego UI z nastawieniem na doświadczenie użytkownika na każdym kroku. [47]



Logo aplikacji Indigo Studio [47]

Indigo Studio jest najbardziej zaawansowanym z wybranych programów, posiada on wiele dobrze zaprojektowanych i funkcjonalnych elementów. Jednym z nich jest Storyboarding, który został przeniesiony na wyższy bardziej interaktywny poziom jak to miało miejsce wśród innych badanych narzędzi. Pozwala on na szybkie nadawanie kontekstu oraz proste grupowanie elementów. Innym komponentem wartym uwagi jest dodawanie animacji, które daje spore możliwości. Jednakże nie posiada on wielu gotowych modułów i wymaga w dużej mierze tworzenia przez użytkownika.



Zrzut ekranu oprogramowania Indigo Studio

Narzędzie to daje więcej możliwości niż poprzednie i jednocześnie jest o wiele bardziej wymagające, pełne opanowanie produktu zajęło cały dzień roboczy, a zaprojektowanie prostej aplikacji w okolicach 7 godzin.

7.1.4. Podsumowanie badania

Przebadane oprogramowanie do tworzenia aktywnych koncepcji widoków aplikacji pokazało odmienne sposoby podejścia do problemu prototypowania, każde miało swoje zalety i wady. FluidUI jest bardzo szybkim do nauczenia, ale ogromna liczba gotowych komponentów utrudnia projektowanie. NinjaMock daje możliwość tworzenia konceptów bez narzucania sztywnej

struktury, ale brak jest wizualnego połączenia ekranów. Z kolei Indigo Studio jest bardzo rozbudowane i daje sporo możliwości, ale wymaga większej ilości czasu.

Najszybszym do nauki z aplikacji do prototypowania jest FluidUI z czasem 30 minut oraz dobry wynik uzyskał NinjaMock z czasem 1 godziny. Natomiast Indigo Studia wymagało najwięcej bo aż 8 godzin.

W przypadku czasu wykonania prototypu aplikacji, najszybszym okazał się być NinjaMock, gdzie zajęło to 4 godziny, natomiast FluidUI i Indigo Studia wymagały więcej uwagi i zajęły kolejno 6 godzin 30 minut oraz 7 godzin.

Największe możliwości daje oprogramowanie Indigo Studia, natomiast pozostałe dwa narzędzia na podobnym poziomie.

Najprzyjemniej z kolei korzystało się na narzędzia NinjaMock, było najbardziej intuicyjne i elastyczne.

Biorąc zestawienie badań i własne odczucia uważam, że najlepszym oprogramowaniem z przedstawionych jest NinjaMock dający bardzo dobre doświadczenia z użytkowania i możliwość szybkiego tworzenia prototypów i wprowadzania zmian.

7.2. Zdefiniowanie projektowanej aplikacji

Podstawowym elementem projektowania każdej aplikacji jest określenie celu, zastosowania i grupy docelowej. Stanowi to fundament i proces prototypowania wymaga zdefiniowania tych elementów.

Celem tworzonej aplikacji jest promowanie Politechniki Wrocławskiej, poprzez stworzenie aplikacji na platformie android serwisu internetowego NAPWr.pl. Jest to innowacyjny, studencki projekt przepływu informacji na Politechnice Wrocławskiej z wykorzystaniem algorytmów sztucznej inteligencji i social media. Prezentuje on informacje o nadchodzących wydarzeniach na Politechnice Wrocławskiej takich jak wyjazdy wakacyjne, imprezy studenckie, spotkania kół naukowych, otwarte wykłady i wiele więcej. Grupą docelową tego serwisu są studenci i absolwenci Politechniki Wrocławskiej. [28]



Logo serwisu NAPWr.pl

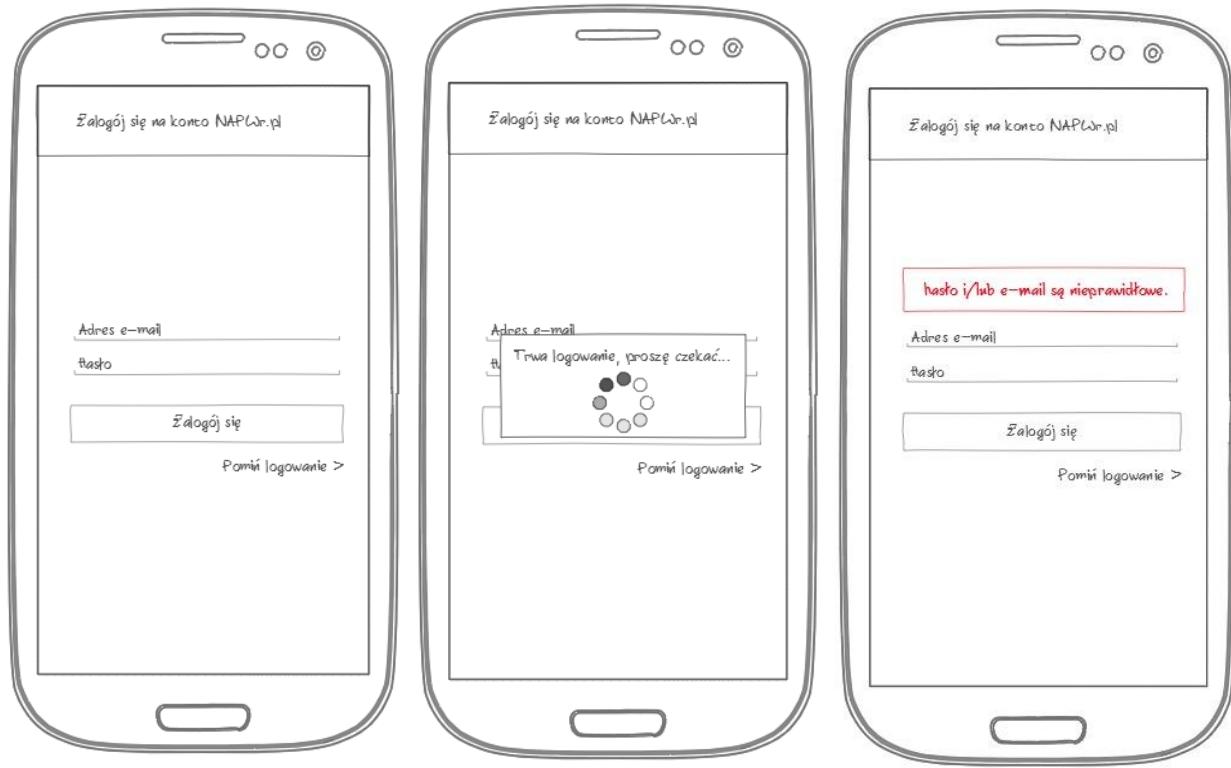
7.3. Projektowanie aplikacji w oparciu o postawione tezy

Wybranym narzędziem projektowym jest opisywany wcześniej NinjaMock, poniższe zdjęcia pochodzą z tego oprogramowania.

Serwis internetowy NaPWr.pl oferuje dodatkowe możliwości dla zalogowanych użytkowników. Są to przeglądanie swojego planu zajęć oraz zapisywanie wybranych wydarzeń, jako ulubione.

Etap opracowania koncepcji struktury nawigacji zostanie rozpoczęty od zdefiniowania ekranu logowania. Ekran logowania składać się będzie z dwóch pól do wprowadzania danych logowania, przycisku logowania oraz przycisku pozwalającego pominąć logowanie.

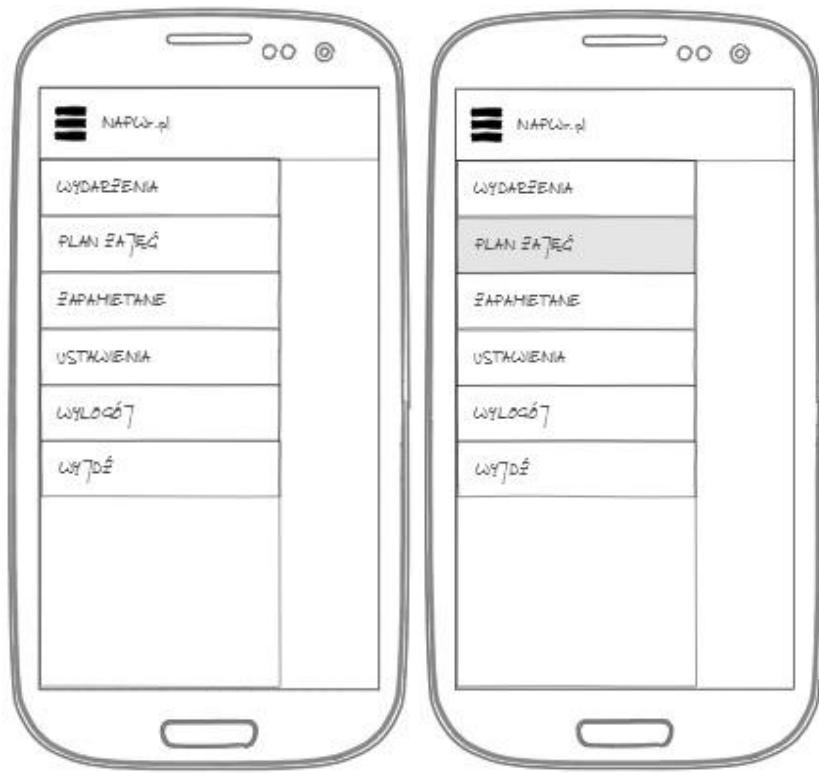
W trakcie logowania zostanie wyświetlony komunikat pop-up informujący o tej czynności, a w przypadku nieprawidłowych danych użytkownika zostanie wyświetlony odpowiedni komunikat.



Projekt ekranu logowania

Menu będzie wysuwane z lewej krawędzi, za pomocą przycisku umieszczonego w lewym górnym rogu oraz poprzez użycie fizycznego przycisku menu. Będzie złożony z informacji o aktualnie zalogowanym użytkowniku, przycisków listy wydarzeń, planu zajęć, wydarzeń zapisanych, ustawień, wylogowania się i zamknięcia aplikacji.

W trakcie działania akcji „on click” wybrana opcja będzie podświetlana, a tytuł paska action bar będzie informował o aktualnie wybranej pozycji poprzez zmianę nazwy „NAPWr.pl” na odpowiadającą aktywnemu ekranowi.



Projekt menu

Przycisk „wydarzenia” z menu będzie powodował otwarcie widoku listy wydarzeń. Przy pierwszym uruchomieniu lista będzie pusta, na ekranie wyświetlony zostanie komunikat informujący o potrzebie odświeżenia, aby pobrać dane. Po pierwszym odnowieniu wydarzeń w prawym górnym rogu będzie znajdowała się informacja o dacie i godzinie ostatniego użycia tej opcji.

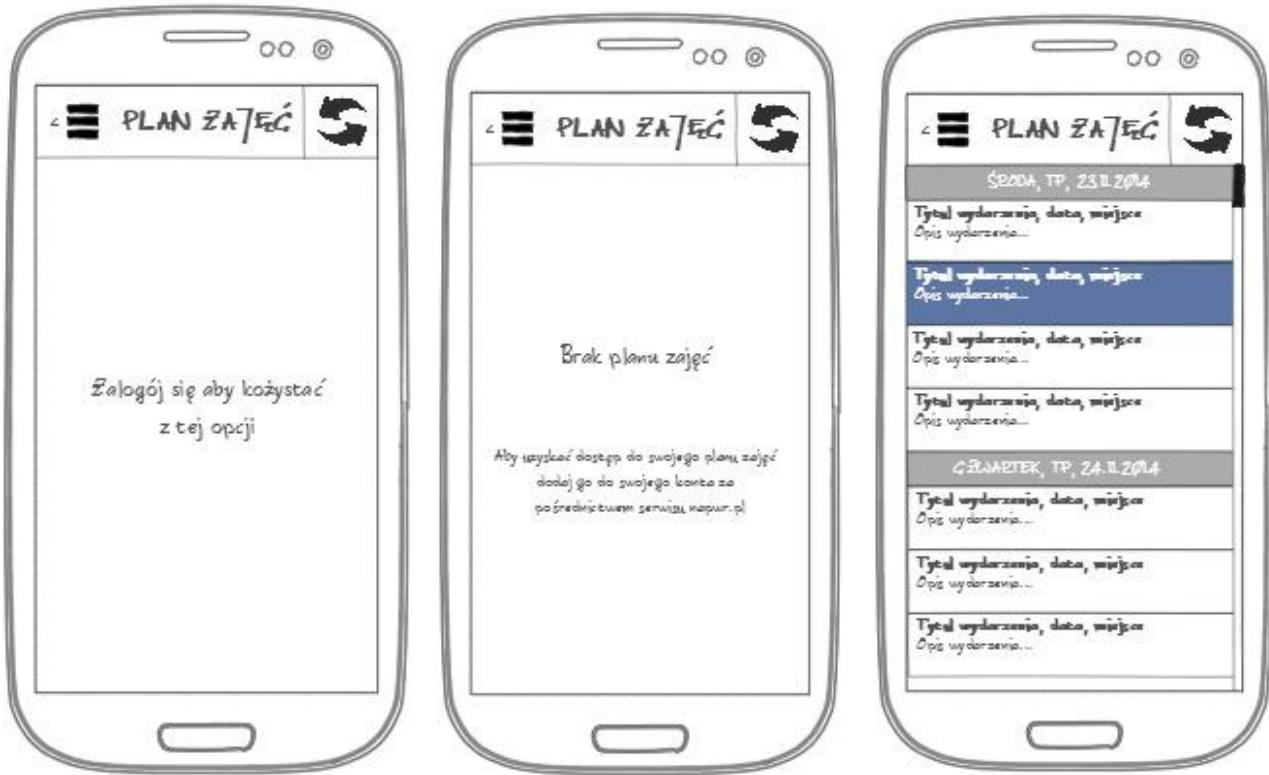
Zestawienie zapisanych elementów będzie prezentowane w formie listy wydarzeń posortowanej względem daty rozpoczęcia zdarzenia. Akcja kliknięcia na wybrany element spowoduje otwarcie szczegółowych informacji o wydarzeniu.



Projekt ekranu listy wydarzeń

Przycisk „plan zajęć” z menu będzie przenosił do ekranu wyświetlającego zapisany harmonogram w formie listy. W przypadku braku planu będzie prezentowany komunikat informujący o możliwości dodania grafiku lekcji na serwisie NaPWr.pl i odświeżeniu. Natomiast w przypadku istnienia takich danych zostanie wyświetlony dla kolejnych dni z uwzględnieniem rodzaju tygodnia (tydzień parzysty i nieparzysty) oraz aktualne zajęcia będą wyróżnione innym kolorem tła. Każdy element listy będzie informował o nazwie, typie, godzinach, sali i prowadzącym.

Do działania tej opcji potrzebny będzie status zalogowanego użytkownika, w przeciwnym razie będzie wyświetlony odpowiedni komunikat.



Projekt ekranu planu zajęć

Przycisk „zapisane” będzie powodował otwarcie widoku zapisanych wydarzeń. W przypadku braku zapisanych wydarzeń będzie prezentowana informacja o takiej możliwości i sposobie na zapisane zdarzeń.

Zapamiętanie elementy będą wyświetlane w formie listy, gdzie po kliknięciu na wybrane wydarzenie będzie otworzony ekran ze szczegółowymi informacjami.



Projekt ekranu wydarzeń zapisanych

Widok ekranu wydarzenia będzie złożony z tytułu, daty i miejsca, ilustracji oraz opisu zdarzenia. W prawym górnym rogu obrazka będą znajdować się przyciski umożliwiające dzielenie się informacją o wydarzeniu poprzez aplikacje Facebook i Google+, dodatkowo znajdzie się tam przycisk dający możliwość dodania elementu do zapisanych. Zapamiętanie wydarzenie będzie informowane poprzez zmianę koloru przycisku oraz komunikat toast.



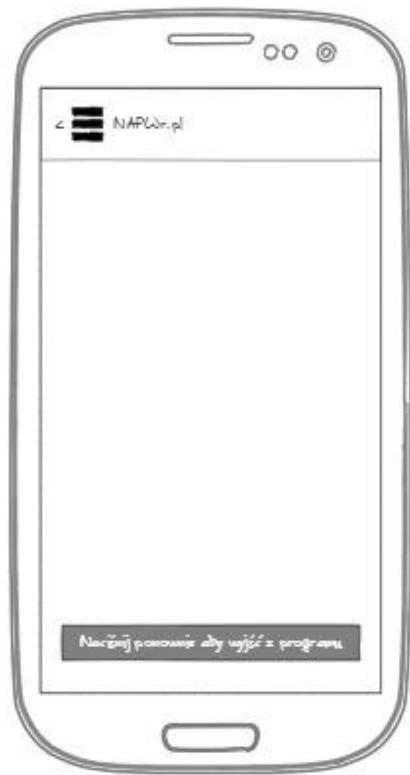
Projekt ekranu prezentacji wydarzenia

Przycisk „ustawienia” będzie powodował otwarcie ekranu ustawień. Dostępnymi możliwymi opcjami do zmiany będzie zmiana wydziału i akcji wykonywanej po naciśnięciu przycisku zamknij. Dodatkowo będzie można zmienić wyłączyć pobieranie obrazków oraz zdecydować czy transfer danych ma się odbywać jedynie poprzez sieć Wi-Fi czy także z użyciem sieci komórkowej.



Projekt ekranu ustawień

Dodatkowym elementem będzie komunikat toast w momencie naciśnięcia przycisku wstecz na pierwszym ekranie. Użytkownik będzie informowany o możliwości zamknięcia aplikacji poprzez ponowne naciśnięcie tego przycisku. Domyślnie taka akcja spowodowała by wyjście z programu już przy pierwszej próbce, ale częstą sytuacją jest opuszczenie aplikacji przez zbyt dużą ilość kliknięć mającą na celu cofnięcie się do ekranu startowego. Tego typu zabieg jest stosowany z wielokrotnością badanych aplikacji zawierających ponad 5mln instalacji, co może świadczyć o wysokiej użyteczności tego rozwiązania.



Projekt powiadomienia po naciśnięciu przycisku wstecz na pierwszym ekranie

8. Implementacja mobilnej aplikacji biznesowej

Ta część pracy magisterskiej poświęcona będzie omówieniu przeprowadzonych badań w kontekście implementowania mobilnych aplikacji biznesowych. Badania tutaj zawarte będą poruszały temat pisania kodu do obsługi bazodanowej, utrzymania czystości kodu oraz budowania produktu.

8.1. Badanie frameworków implementowania bazy danych

Pierwszym z przeprowadzonych badań jest zestawienie frameworków do utrzymywania bazy danych. Porównane będą framweorki OMRLite, GreenDAO oraz natywna obsługa SQLite.

8.1.1. Natywny SQLLite

Podstawowym rozwiązaniem problemu obsługi bazy danych jest wykorzystanie domyślnie dostępnych natywnych struktur. Wybranie tego podejścia upraszcza zależności i nie wymaga dodawania zewnętrznych bibliotek. Jednocześnie jest aktualizowane domyślnie, co pozwala na łatwą migrację do nowszych wersji bibliotek.

Podstawowym tego rozwiązania jest klasa otwierająca i zarządzająca połączeniem z bazą danych oraz obsługą zapytań. W tym przypadku te dwa zadania zostały rozdzielone upraszczając struktury, DataBaseHandler do zażądania bazą danych oraz TagHandler do zażądania obiektami.

```
public class DataBaseHandler extends SQLiteOpenHelper {
    public static final int DATABASE_VERSION = 2;
    public static final String DATABASE_NAME = "NAPWrDatabase";

    public DataBaseHandler(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + OrganizationHandler.TABLE_NAME);
        onCreate(db);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(TagHandler.getCreateSQL());
    }

    public void addTagObject(TagObject object) {
        TagHandler.addTagObject(object, getWritableDatabase());
    }
}
```

```

public TagObject getTagObject(int tagId) {
    return TagHandler.getTagObject(tagId, getWritableDatabase());
}

public class TagHandler {
    public static final String TABLE_NAME = "Tagi";
    private static final String KEY_ID = "tagId";
    private static final String KEY_NAZWA = "tagNazwa";

    public static String getCreateSQL() {
        return "CREATE TABLE IF NOT EXISTS " + TABLE_NAME + "(" + KEY_ID
            + " INTEGER PRIMARY KEY," + KEY_NAZWA + " TEXT)";
    }

    public static void addTagObject(TagObject object, SQLiteDatabase db) {
        ContentValues values = new ContentValues();
        values.put(KEY_ID, object.tagId);
        values.put(KEY_NAZWA, object.tagNazwa);

        db.insert(TABLE_NAME, null, values);
        db.close();
    }

    public static TagObject getTagObject(int tagId, SQLiteDatabase db) {
        String[] values = new String[] { KEY_ID, KEY_NAZWA };
        String where = KEY_ID + "=?";
        String[] whatValue = new String[] { String.valueOf(tagId) };
        Cursor cursor = db.query(TABLE_NAME, values, where, whatValue, null,
            null, null);

        return get(cursor);
    }

    private static TagObject get(Cursor cursor) {
        if (cursor != null) {
            cursor.moveToFirst();

            if (cursor.getCount() > 0) {

                int tagId = Integer.parseInt(cursor.getString(0));
                String tagNazwa = cursor.getString(1);

                return new TagObject(tagId, tagNazwa);
            }
        }

        return null;
    }
}

```

Struktura tego podejścia jest bardo prosta i nauczenie się jej nie jest dużym wyzwaniem. W przypadku prostych struktur bazodanowych jest to zdecydowanie dobre podejście, niestety w

przypadku dużej ilości pól, ilość wysiłku potrzebnego do zaimplementowania tego rozwiązania znacząco rośnie.

Czas nauki tego podejścia to niecała godzina, natomiast zaimplementowanie i optymalizacja pełnej bazy danych zajęło około 4 godzin 30 minut, głównie ze względu na trudność wprowadzania modyfikacji w istniejących tabelach zawierających dużo pól. Ilość wygenerowanego kodu tym rozwiązaniem to 1025 wierszy.

Sprawdzane też były czasy zapisu i odczytu danych bazując na przykładowym zastawie danych, identycznych dla wszystkich frameworków. Wszystkie sprawdzane frameworki u swoich podstaw bazują na obsłudze bazy danych poprzez SQLite, tak więc ich czasy wykonania będą porównywane do natywnego SQLite.

8.1.2. OMRLite

OMRLite jest prostym frameworkiem do szybkiego i generycznego tworzenia połączeń bazodanowych. Wybór tego rozwiązania wiąże się z dodaniem biblioteki do folderu libs oraz dodaniem zależności na OMRLite-core.jar oraz OMRLite-android.jar.

Podany framework bazuje na rozbudowanej strukturze klasy zarządzającą bazą danych, natomiast znika problem definiowania zażądania obiektami, które jest generowane wewnątrz frameworka na podstawie klas obiektów z adnotacjami.

```
@DatabaseTable
public class TagObject {

    @DatabaseField(generatedId = true)
    public int tagId;

    @DatabaseField
    public String tagNazwa;

    public int getTagId() {
        return tagId;
    }

    public void setTagId(int tagId) {
        this.tagId = tagId;
    }

    public String getTagNazwa() {
        return tagNazwa;
    }

    public void setTagNazwa(String tagNazwa) {
        this.tagNazwa = tagNazwa;
    }
}
```

```

public class DatabaseHelper extends OrmLiteSqliteOpenHelper {

    private static final String DATABASE_NAME = "notify.db";
    private static final int DATABASE_VERSION = 4;
    private final String LOG_NAME = getClass().getName();
    private Dao<TagObject, Integer> tagDao;

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase,
                         ConnectionSource connectionSource) {
        try {
            TableUtils.createTable(connectionSource, TagObject.class);
        } catch (SQLException e) {
            Log.e(LOG_NAME, "Could not create new tables", e);
        }
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase,
                         ConnectionSource connectionSource, int oldVersion, int newVersion) {
        try {
            TableUtils.dropTable(connectionSource, TagObject.class, true);
            onCreate(sqLiteDatabase, connectionSource);
        } catch (SQLException e) {
            Log.e(LOG_NAME, "Could not upgrade the tables", e);
        }
    }

    public Dao<TagObject, Integer> getTagObjectDao() throws SQLException {
        if (tagDao == null) {
            tagDao = getDao(TagObject.class);
        }
        return tagDao;
    }
}

```

Przez potrzebę ustawiania metod Get i set dla każdej zmiennej generowane jest bardzo dużo kodu, jednakże nie jest to kod skomplikowany. Jest łatwy do operowania i nauki, a dodanie nowego parametru nie niesie za sobą konieczności przepisywania sporej liczny linii kodu, tylko zmiany kosmetyczne.

Czas nauki tego framweorku był niezwykle szybki i wystarczyło niecałe 30 minut, natomiast stworzenie w pełni funkcjonalnej bazy danych wyniosło nieco ponad 30 minut. Ilość kodu nie automatycznie wygenerowanego tym rozwiążaniem to 334 wiersze.

Czas zapisu danych wyniósł 3 krotność czasu SQLite, natomiast czas odczytu 12 krotność.

8.1.3. GreenDAO

GreenDAO jest projektem open-source opracowanym do wspomagania pracy z osadzoną relacyjną bazą danych SQLite. Framework ten generuje na podstawie klasy z definicjami obiektów API zorientowane obiektowo, pozwalając na zapis, aktualizacje, usuwanie i pobieranie danych. Używanie GreenDAO wiąże się z dodaniem zależności na bibliotekę greenDAO.jar

```
public class NAPWrDatabaseGenerator {  
  
    public static void main(String[] args) throws Exception {  
        Schema schema = new Schema(1000, "pl.wroc.pwr.na");  
  
        addTagObject(schema);  
  
        new DaoGenerator().generateAll(schema, "../pl/wroc/pwr-na/database/dao/");  
    }  
  
    private static void addTagObject(Schema schema) {  
        Entity tagObject = schema.addEntity("TagObject");  
        tagObject.addIdProperty();  
        tagObject.addStringProperty("tagNazwa").notNull();  
    }  
}
```

Na podstawie przygotowanej klasy zawierającej definicję obiektów, GreenDAO generuje bazę danych, jest to bardzo szybkie i przyjemne rozwiązanie, minusem tego frameworka jest konieczność ponownego generowania w przypadku chęci wprowadzenia zmian.

Czas nauki był umiarkowany i wyniósł około godziny, natomiast czas wykonania bazy danych z pomocą generatora trwał około 10 minut. Ilość kodu nie automatycznie wygenerowanego tym rozwiązaniem to 100 wierszy.

W przypadku czasu zapisu i odczytu był on bardzo zbliżony do czasów SQLite.

8.1.4. Podsumowanie badania

Najszybszym i najbardziej przyjemnym testowanym frameworkiem jest OMRLite, też ilość napisanego kodu jest dość niska, co wpływa znaczco na czytelność. Niestety posiada najgorsze wyniki czasu zapisu i odczytu, więc z tego powodu nie nadaje się do zastosowań biznesowych, gdzie czas dostarczenia danych odgrywa kluczową rolę.

Najtrudniejszym i najbardziej czasochłonnym podejściem okazało się wybranie natywnego sposobu implementacji bazy danych, lecz czas odczytu i zapisu jest na najlepszym możliwym poziomie.

Z kolei GreenDAO, poprzez w pełni generowany kod obsługi bazy danych pozwala na równie szybki czas odczytu i zapisu, co natywnie napisany w SQLite, ale czas wymagany do stworzenia i zmiany struktury jest wielokrotnie niższy. Ponadto zapytania generowane przez GreenDAO są tworzone w sposób optymalny, gdzie w przypadku własnoręcznego tworzenia w SQLite istnieje zagrożenie nieoptymalnego stworzenia zapytań i wydłużenie czasu egzekucji.

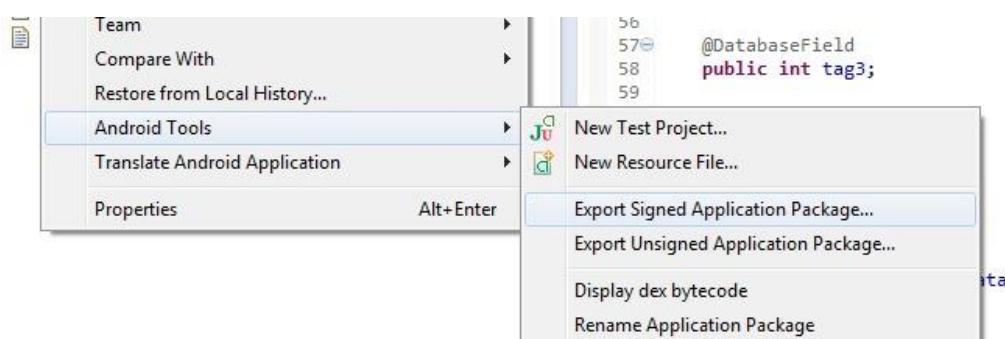
Z przebadanych frameworków najlepszym jest GreenDAO, oferuje on szybkość na bardzo dobrym poziomie, jednocześnie umożliwiając szybkie i efektowne wprowadzanie zmian.

8.2. Badanie sposobów budowania produktu

Podstawowym elementem releasu oprogramowania jest budowanie produktu. Proces ten może być robiony ręcznie albo automatycznie. W przypadku dużej liczby iteracji i potrzeby puszczania testów za każdym razem za chodzi potrzeba automatyzacji, głównymi narzędziami pozwalającymi na budowanie produktu są Ant, Maven oraz Gradle.

8.2.1. Ręczne budowanie produktu

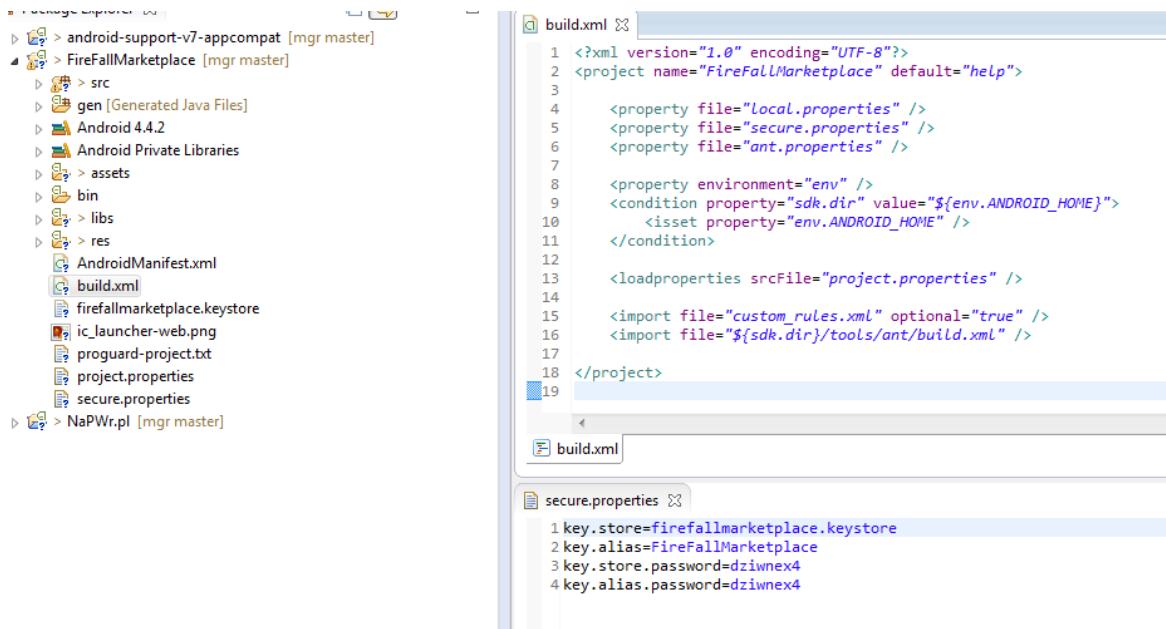
Najprostszym ze sposobów budowania produktu jest ręczne przeprowadzenie tego procesu. Klikając PPM na projekt możemy wybrać “Android Tools > Export (un)signed application package” i rozpoczęć tworzenie paczki oprogramowania. W przypadku pierwszego podejścia należy wygenerować keystore, który jest także podstawą do automatycznego budowania, a w przypadku kolejnych prób wystarczy tylko podać utworzone hasła.



Zaletą tego rozwiązania jest szybkość i brak potrzeby modyfikowania projektu. Jednak przy projekcie, którego powstawanie jest rozłożone w szerszej perspektywie czasu o wiele praktyczniejszym jest automatyzacja, która pozwala na testowanie i budowanie bez udziału programisty.

8.2.2. Ant

Apache Ant jest to biblioteka oparta na Javie, której misją jest przeprowadzenie procesu opisanego w plikach xml. Głównym znanym celem użytkowania jest budowanie produktów aplikacji Java. [49]



Widok projektu budowanego poprzez Ant

Wybór tego sposobu polega na dodaniu prostego pliku build.xml oraz pliku z danymi do keystore. Plik build.xml może być wygenerowany za pomocą prostych komend. Użytych w commandn line. Zarówno dla bibliotek, projektu i projektów testowych. Ogromną zaletą tego rozwiązania jest brak wpływu na strukturę kodu.

Czas wdrożenia rozwiązania wraz ze skryptem budującym na Continues Intergration Environment to jedynie 30 minut.

8.2.3. Maven

Apache Maven jest narzędziem do zarządzania oprogramowaniem na podstawie koncepcji obiektowego projektu POM. Maven może zarządzać nad budowaniem projektu, gromadzeniem dokumentacji, raportowaniem i centralizowaniem informacji. [50]

```

7 <groupId>com.androidexample</groupId>
8 <artifactId>androidexample</artifactId>
9 <version>1.0.0-SNAPSHOT</version>
10 <packaging>pom</packaging>
11
12
13 <modules>
14   <module>androidexample-lib</module>
15   <module>androidexample-app</module>
16   <module>androidexample-tests</module>
17 </modules>
18
19
20 <dependencyManagement>
21   <dependencies>
22     <dependency>
23       <groupId>com.google.android</groupId>
24       <artifactId>android</artifactId>
25       <version>4.1.1.4</version>
26       <scope>provided</scope>
27     </dependency>
28
29     <dependency>
30       <groupId>com.google.android</groupId>
31       <artifactId>android-test</artifactId>
32       <version>4.1.1.4</version>
33       <scope>provided</scope>
34     </dependency>
35
36   </dependencies>
37 </dependencyManagement>
38
39 <build>

```

Widok projektu budowanego poprzez Maven

Wybór tego sposobu polega na całkowitym przebudowaniu struktury projektu, należy także zadbać o dodanie zależności oraz w przypadku dodatkowych bibliotek i projektów testowych należy je dodać w parent POM. W przypadku prostych testów można użyć automatycznych skryptów przekształcających projekt, natomiast w przypadku bardziej zaawansowanych zmiana sposobu budowania musi zostać zrobiona ręcznie. Dodatkowym problemem jest utrudnione debugowanie projektu.

Zalety Mavena w postaci łatwości rozszerzania skryptu budującego o kolejne moduły, takie jak pokrycie kodu testami, nie znajduje zastosowania w projektach na Androida.

Czas wdrożenia rozwiązania wraz ze skryptem budującym na Continues Intergration Environment to ponad 4 godziny.

8.2.4. Gradle

Gradle jest opens-sourcowym systemem automatyzacji budowania. Gradle łączy zarządzanie zależnościami Mavena i elastyczność Anta. Gradle może automatyzować między innymi budowanie, testowanie, publikowanie, deployowanie i każde inne żądanie zdefiniowane w skrypcie gradowym. [51]

```

4 ##
5 ##  Gradle start up script for UN*X
6 ##
7 #####
8
9 # Add default JVM options here. You can also use JAVA_OPTS and GRADLE_OPTS to pass JVM options to this script.
10 DEFAULT_JVM_OPTS=""
11
12 APP_NAME="gradle"
13 APP_BASE_NAME= basename "$0"
14
15 # Use the maximum available, or set MAX_FD != -1 to use that value.
16 MAX_FD="maximum"
17
18 warn () {
19     echo "$*"
20 }
21
22 die () {
23     echo
24     echo "$*"
25     echo
26     exit 1
27 }
28
29 # OS specific support (must be 'true' or 'false').
30 cygwin=false
31 msys=false
32 darwin=false
33 case "$uname" in
34   CYGWIN*)

```

Widok projektu budowanego poprzez Gradle

Gradle podobnie jak Maven zarządza automatycznie zależnościami oraz wymaga przebudowania struktury projekty. Zalety, jakie pobiera z Mavena i Anta pozwalające łatwo rozszerzać skrypt budujący o nowe moduły oraz definiować własne procesy, podobnie jak w przypadku Mavena nie znajduje zastosowania w projekcie Androidowym.

Czas wdrożenia rozwiązania wraz ze skryptem budującym na Continues Intergation Environment to ponad 6 godzin.

8.2.5. Podsumowanie badania

Przeprowadzone badanie porusza temat problemu automatycznego budowania, zarówno Gradle jak i Maven wymagają zmiany struktury kodu aplikacji, co uniemożliwia swobodną modyfikację, a także przebudowywanie produktu bez użycia skryptu budującego. Pozbawiony tych wad jest Ant, który stanowi jedynie dodatek do projektu, przez co struktura pozostaje nienaruszona. Przewaga Gradla i Mavena nad Antem to automatyczne zażądanie zależnościami, które nie znajduje zastosowania w Androidzie, ponieważ wszystkie zależności muszą być dodane do folderu libs, aby móc pracować na kodzie bez błędów komplikacji.

Z powyżej wymienionych powodów najlepiej w badaniu sprawdził się Ant.

Podczas prowadzenia badań był także brany pod uwagę czas budowania, jednak różnice były nieznaczące i najprawdopodobniej wynikały z aktualnego obciążenia maszyny.

9. Prezentacja i porównanie utworzonej aplikacji serwisu napwr.pl

Ten rozdział poświęcony będzie prezentacji stworzonej aplikacji i zestawieniu jej z obecnie funkcjonującą mobilną reprezentacją serwisu napwr.pl dostępną w Google Play [29].

Aktualna aplikacja napwr.pl została stworzona w 2013 roku, powstawała ona w oparciu o doświadczenia wynikające głównie z użytkowania webowej wersji omawianego serwisu. Nie zostały poczynione żadne kompleksowe kroki w celu badania rynku z zamierzeniem wysunięcia tez dobrych praktyk projektowania mobilnych aplikacji. Dzięki identycznej treści, grupie docelowej i celu stanowić ona będzie dobry przykład sprawdzający skuteczność postawionych tez.

W projekcie kreowanego programu zrezygnowano z splashscreen bazując na wysuniętych tezach w fragmencie pracy poświęconej analizie dokonanego przeglądu. Zabieg ten pozwolił na uzyskanie szybszego ładowania właściwej treści aplikacji, względem starszej wersji jest to około 5 sekund.



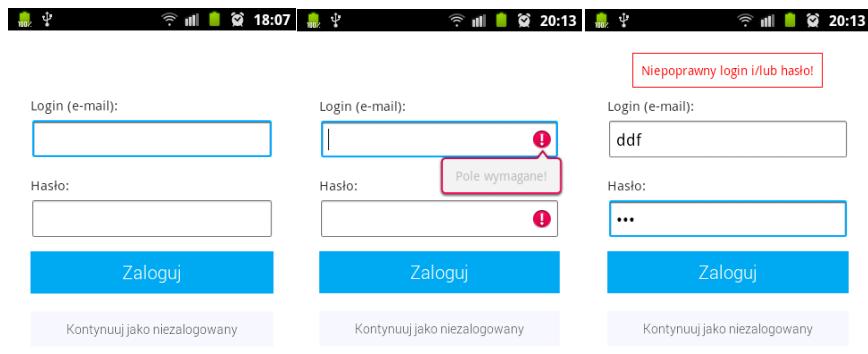
Ekran Splashscreen starej aplikacji

Ekran logowania w przygotowanym oprogramowaniu jest całkowicie niezależnym ekranem, który jest uruchamiany, jako pierwszy widok. Poza przyciskiem zaloguj znajduje się opcja kontynuowania, jako niezalogowany użytkownik oraz informacja o możliwym założeniu konta w serwisie napwr.pl. W przypadku próby logowania bez wypełniania pól hasła i/lub e-mail, klient jest zawiadamiany odpowiednim komunikatem osadzonym w danym polu. Natomiast przy wprowadzeniu błędного hasła i/lub e-maila wyświetlany jest adekwatna informacja na nad polami z danymi logowania.

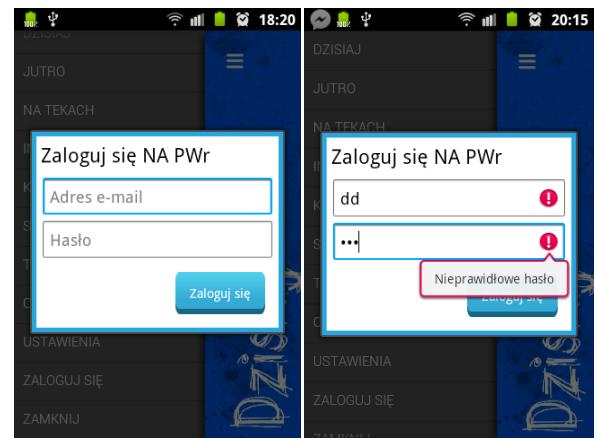
Jest to odmienne podejście względem obecnie funkcjonującej aplikacji, gdzie ekran logowania był komunikatem pop-up, a użytkownik mógł wyświetlić to okno opcjonalnie z poziomu menu, w

przypadku nieprawidłowo podanych danych była wyświetiana informacja, jakie dokładnie pole zawiera błąd.

W nowej wersji programu poprawione zostało bezpieczeństwo, gdzie niepodawane jest, która dana jest błędnie wprowadzona oraz umiejscowienie ekranu logowania w aplikacji nie sprawia problemu w dotarciu do niego, na co skarżyli się użytkownicy starszej mobilnej wersji serwisu napwr.pl.

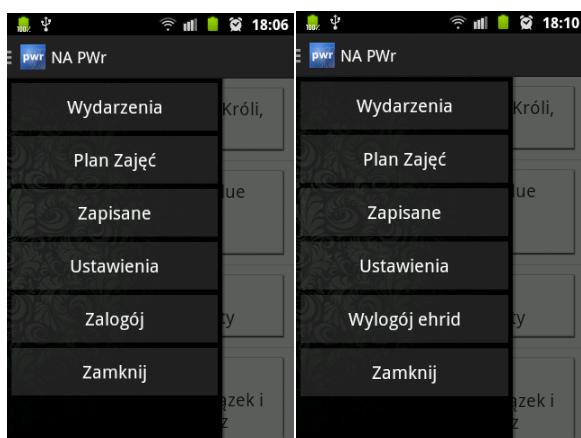


Ekran logowania w nowej aplikacji

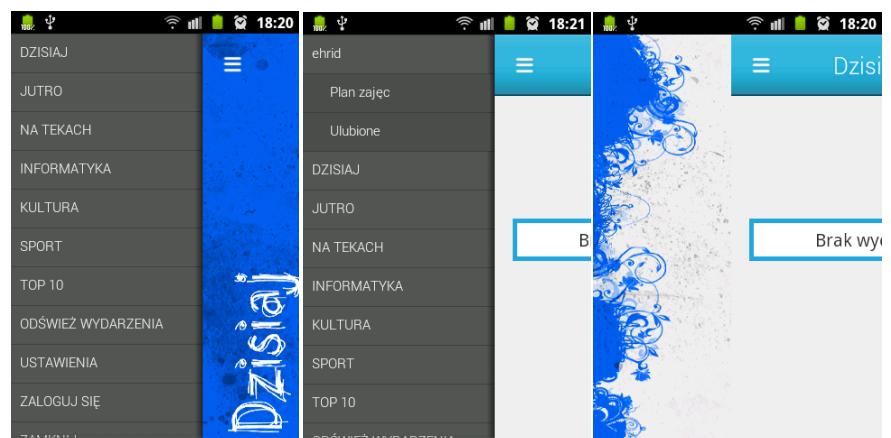


Ekran logowania w starej aplikacji

Menu w nowej aplikacji zostało znaczco uproszczone, co w znaczącym stopniu upraszcza użytkowanie oprogramowania. Zestawiając z obecną wersją zawiera jedynie 6 opcji, zamiast ponad 10. Zabieg ten polegał na połączeniu wielu kategorii wydarzeń w jednym miejscu oraz przycisk odświeżający treść został osadzony bezpośrednio w ekranach, które go wymagają. Pozwoliło to na ułatwienie nawigacji, z którą użytkownicy zgłaszały problem.



Menu w nowej aplikacji



Menu w starej aplikacji

Widok planu zajęć uległ poprawie wizualnej oraz została dodana możliwość odświeżania harmonogramu w prawym górnym rogu. Brak takiej możliwości we wcześniejszej wersji

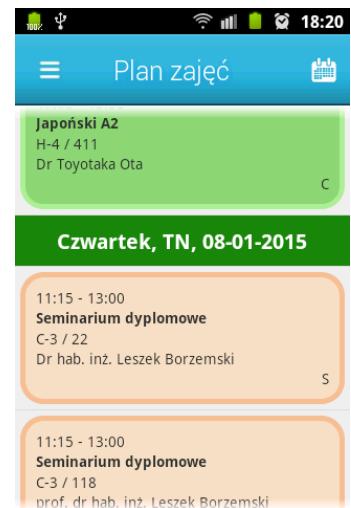
uniemożliwiał załadowanie nowego planu zajęć w nowym semestrze, co negatywnie wpływało na ocenę aplikacji.



Ekran planu zajęć w nowej aplikacji



Ekran planu zajęć w starej aplikacji



Ekran ustawień w nowej aplikacji



Ekran ustawień w starej aplikacji

Ekran ustawień został poprawiony organizacyjnie, dostępne opcje manipulacji aplikacji zostały w znaczącym stopniu uproszczone, co pozwoliło na zwiększenie czytelności.



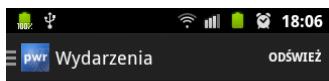
Ekran ustawień w nowej aplikacji



Ekran ustawień w starej aplikacji

W pierwotnej mobilnej reprezentacji serwisu napwr.pl wydarzenia składały się z szeregowo ustawionych ekranów z możliwością ich zmiany używając akcji przesunięcia ekranu palcem. Elementy listy zawierającej nadchodzące zdarzenia zawierały sporo informacji, co ograniczało ilość wyświetlanych pozycji.

Unowocześniona aplikacja zawiera tylko jeden widok wydarzeń posortowanych w kolejności rozpoczęcia oraz ilość prezentowanych danych została ograniczona do daty i tytułu. Ponadto został dodany przycisk umożliwiający szybkie odświeżenie treści.

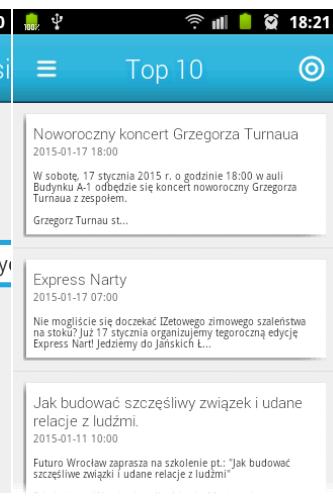
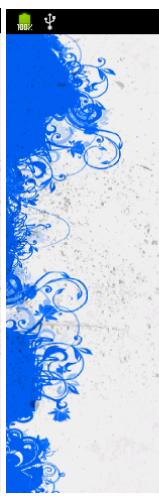


06 Święto Trzech Króli,
Styczeń brak zajęć

08 Pokaz filmu 'Blue
Velvet' Davida
Lyncha

09 Zajęcia jak we
Styczeń wtorek parzysty

Jak budować
11 szczęśliwy związek i
Styczeń udane relacje z
ludźmi



Ekran wydarzeń w nowej aplikacji

Ekran wydarzeń w starej aplikacji

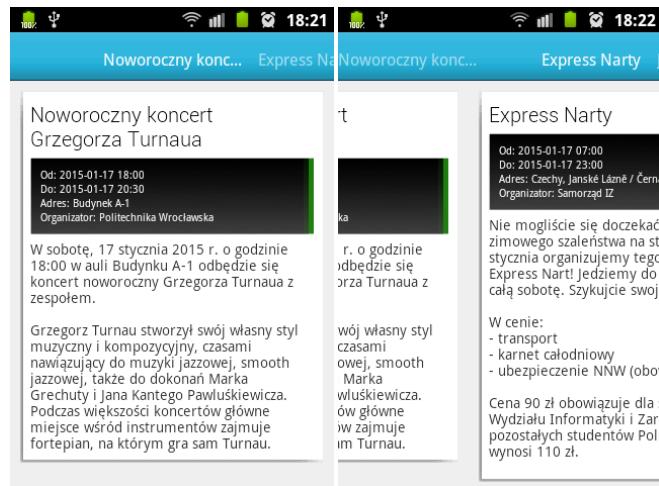
Organizacja ekranu pojedynczego wydarzenia uległa zmianie, ponadto została dodane opcje dzielenia się treścią poprzez Facebook i Google+ oraz dodania wydarzenia do zapisanych. Ponadto wyświetlany jest także element plakatu wydarzenia. Uproszczeniu uległa nawigacja, na którą narzekali użytkownicy, obecnie widok stanowi reprezentację tylko jednego wydarzenia, gdzie w poprzedniej aplikacji można było przesuwając palcem zmieniać widoki.



Zajęcia jak we wtorek parzysty
7:30
GDZIE: Politechnika Wrocławska

Piątek, 09 Styczeń

W piątek (parzysty) 9 stycznia 2015 r., zajęcia dydaktyczne odbywają się jak we wtorek parzysty.



Ekran wydarzenia w nowej aplikacji

Ekran wydarzenia w starej aplikacji

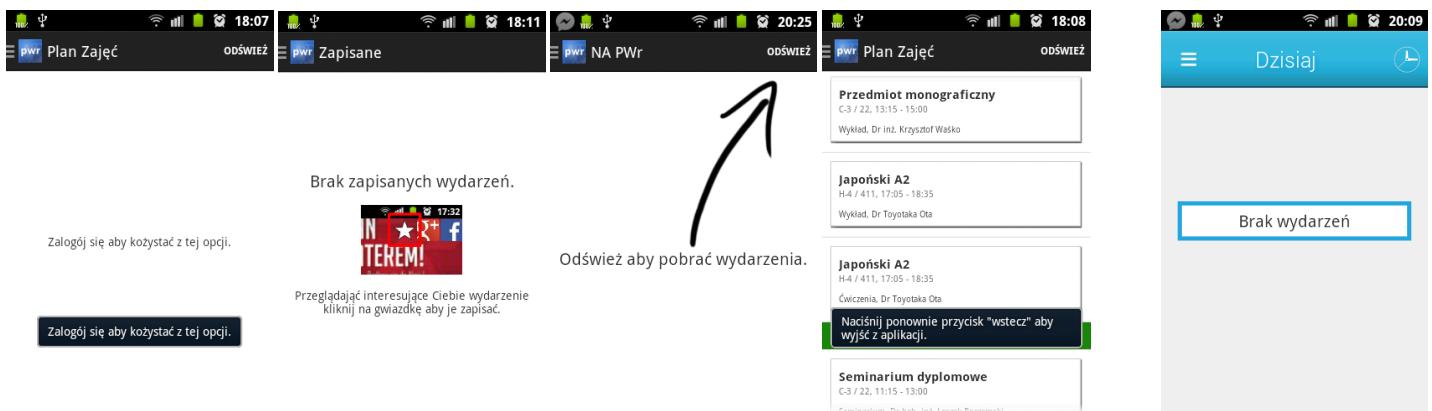
Nową funkcjonalnością obecnego oprogramowania jest możliwość zapisywania wybranych wydarzeń, co umożliwia szybkie dotarcie do potrzebnych informacji. Jest to nowe wymagania, którego potrzeba była zgłoszana w komentarzach.



Ekrany zapisanych wydarzeń w nowej aplikacji

Poważnym brakiem obecnej aplikacji jest nieinformowanie użytkownika o możliwych operacjach i stanie oprogramowania. Jedynym komunikatem, jaki użytkownik uzyskuje jest informacja o braku wydarzeń, ale brakuje powiadomienia o możliwych rozwiązań tego problemu.

Nowa mobilna wersja napwr.pl wprowadza poprawę w tej kwestii, użytkownik jest informowany licznymi komunikatami toast o wykonanych akcjach, takich jak dodanie wydarzenia do zapisanych. Komunikaty osadzone informują na wielu etapach o stanie aplikacji oraz możliwym rozwiązań problemu lub sposobie użytkowania danej funkcjonalności.



10. Podsumowanie

Niniejsza praca przedstawia istotne aspekty tworzenia mobilnej aplikacji biznesowej, a także reguły niezbędne przy jej projektowaniu. Do głównych założeń, którymi należy się kierować przy budowaniu oprogramowania jest zwrócenie uwagi na koszty produkcji oraz zdefiniowanie grupy społecznej, przez którą dana aplikacja będzie wykorzystywana.

Celem pracy jest również zweryfikowanie funkcjonalności zaleceń projektowania oraz implementacji mobilnych aplikacji biznesowych.

Do najważniejszych tez projektowych aplikacji zalicza się:

- Korzystanie z komunikatów toast i informowanie za pomocą komunikatów osadzonych.
- Umożliwianie dostępu do aplikacji bez potrzeby logowania
- Informowanie o ładowaniu treści, bez widocznego zawieszania działania
- Korzystanie z menu wysuwanego z krawędzi lub przycisku
- Wykorzystanie menu natywnego albo połączenie fizycznego przycisku z własnym menu
- Informowanie użytkownika o nawigacji
- Nie używanie reklam
- Nie posiadanie rejestracji
- Korzystanie z dzielenia się informacjami poprzez social media.
- Nie używanie ekranu splashscreen
- Korzystanie z ekranu ustawień
- Używanie standardowych klawiatur

Z kolei do tez implementacyjnych wlicza się:

- Użycie narzędzia NinjaMock do tworzenia prototypu
- Użycie frameworku GreenDAO do obsługi bazy danych
- Użycie narzędzia Apache Ant do budowania produktu

Aplikacja realizująca wymienione powyżej tezy, została zaimplementowana oraz porównana z jej poprzednikiem. Z badań można wywnioskować znaczącą poprawę jakości oprogramowania, a także wygodę w jej użytkowaniu. Dzięki zastosowaniu reguł, o których mowa w pracy, dostrzega się, iż aplikacja stała się bardziej dostępna dla użytkownika, intuicyjna i co najważniejsze czas oczekiwania na odpowiedź aplikacji uległ skróceniu.

Należy zwrócić uwagę, że zbliżone badania przeprowadzone rok temu wykazały większe znaczenie używania spalschreen, gdzie obecnie jest to rozwiązanie odradzane. Zważywszy na dynamiczność rynku mobilnego, wiele elementów stanowiących w chwili obecnej zalecenia, może w nadchodzących latach okazać się przeciwwskazaniem.

Aby w pełni uzyskać obraz obecnego stanu dziedziny projektowania należałyby przeprowadzać podobne badania cyklicznie, co pozwoliłoby także na wyznaczanie trendów i dać możliwość szybszego reagowania na nadchodzące zmiany.

Rozszerzeniem obecnego badania mogłoby być, w przypadku przeglądu rynku aplikacji, dodanie kolejnych kryteriów, takich jak zależność ekranów, ilość opcji w menu oraz wielu innych. Natomiast w przypadku problemów implementacyjnych można sprawdzić większą liczbę frameworków o różnym przeznaczeniu, a także porównać różne systemy operacyjne takie jak iOS, Windows Phone i BlackBerry OS.

11. Źródła

- [1] IDC, Q3 2014, [Smartphone OS Market Share Q3 2014](http://www.idc.com/prodserv/smartphone-os-market-share.jsp),
<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- [2] Zeroonezero, [Mobile Device Application](http://www.zeroonezero.com/glossary/mobile-phone-applications.html), <http://www.zeroonezero.com/glossary/mobile-phone-applications.html>
- [3] Cory Janssen, [Mobile Application \(Mobile App\)](http://www.techopedia.com/definition/2953/mobile-application-mobile-app),
<http://www.techopedia.com/definition/2953/mobile-application-mobile-app>
- [4] Zheng Qin, 2009, Introduction to e-commerce, ISBN 978-3540496441, Strona 7
- [5] Peter Tarasewich, Robert C. Nickerson, Merrill Warkentin, 2002, Issues in Mobile E-Commerce, Communication of the Association for Information Systems (Volume 8, 2002) 41-64, Strona 42
- [6] Peter Tarasewich, 2003, Designing Mobile Commerce Application, Communication of the ACM December 2003/Vol. 46 No. 12, Strony 57-60
- [7] Margaret Rouse, 2011, [Mobile Application Development](http://searchsoa.techtarget.com/definition/Mobile-application-development),
<http://searchsoa.techtarget.com/definition/Mobile-application-development>
- [8] Enough software, 2012, Don't Panic Mobile Developer's Guide To The Galaxy 11th Edition, Strona 10
- [9] Enough software, 2012, Don't Panic Mobile Developer's Guide To The Galaxy 11th Edition, Strony 34-40
- [10] Enough software, 2012, Don't Panic Mobile Developer's Guide To The Galaxy 11th Edition, Strona 47
- [11] Enough software, 2012, Don't Panic Mobile Developer's Guide To The Galaxy 11th Edition, Strony 76-78
- [12] Andrzej Jaskiewicz, 1997, Inżynieria oprogramowania, ISBN 83-7197-007-2, Strony 11-13
- [13] Andrzej Jaskiewicz, 1997, Inżynieria oprogramowania, ISBN 83-7197-007-2, Strony 50-51
- [14] Andrzej Jaskiewicz, 1997, Inżynieria oprogramowania, ISBN 83-7197-007-2, Strony 59-60
- [15] Google Corporation, [Advertising without Compromising User Experience](http://developer.android.com/training/monetization/ads-and-ux.html),
<http://developer.android.com/training/monetization/ads-and-ux.html>
- [16] ContractIQ, [Mobile Monetization Models – Making money out of Apps](http://blog.contractiq.com/mobile-monetization-models-making-money-out-of-apps/),
<http://blog.contractiq.com/mobile-monetization-models-making-money-out-of-apps/>

- [17] Cyril Mottier, Splash Screens Are Evil, Don't Use Them!,
<http://cyrilmottier.com/2012/05/03/splash-screens-are-evil-dont-use-them/>
- [18] Google Corporation, Settings, <http://developer.android.com/guide/topics/ui/settings.htm>
- [19] Google Corporation, Metrics and Grids, <http://developer.android.com/design/style/metrics-grids.html>
- [20] Google Corporation, Specifying the Input Method Type,
<http://developer.android.com/training/keyboard-input/style.html>
- [21] Josh Clark, 2010, Tapworthy Designing Great iPhone Apps, ISBN 978-1-449-38165-3, Strony 62-64
- [22] Theresa Neil, 2012, Mobile Design Pattern Gallery, ISBN 978-1-4493-1432-3, Strony 11-25
- [23] Javier Cuello & José Vittone, 2013, Designing Mobile Apps, Strony 7-8
- [24] Adrian Mendoza, 2014, Mobile User Experience, ISBN 978-0-12-409514-4, Strony 177-178
- [25] White Paper, The Essentials of Mobile App Marketing, Strony 3-4
- [26] Sysmagazine, NinjaMock — the free supplement for prototyping of interfaces
<http://sysmagazine.com/posts/208406/>
- [27] Grace Smith, 2013, <http://mashable.com/2013/04/02/wireframing-tools-mobile/>
- [28] Zespół NAPWr.pl, <http://www.napwr.pl/>
- [29] Aplikacja napwr.pl, <https://play.google.com/store/apps/details?id=pl.wroc.pwr.na&hl=en>
- [30] Marcus Dixon, 2013, 7 benefits of using Mobile Buisness App
<http://info.groveis.com/blog/bid/186170/7-Benefits-of-using-a-Mobile-Business-App>
- [31] Brian Geary, 2014, The Benefits of Mobile Apps For Businesses
<http://www.andplus.com/the-benefits-of-mobile-apps-for-businesses/>
- [32] Toontuts, Article on Web Technologies, 2014,
<http://toonlite.tumblr.com/post/78177523521/new-post-has-been-published-on>
- [33] Ceneo, <http://info.ceneo.pl/ceneo-na-telefon/>
- [34] TeamViewer, http://www.teamviewer.com/mailings/online/2010_06_iPad_en.html
- [35] Gazeta Małych i Średnich Przedsiębiorstw, 2007, Analiza przepływów finansowych
http://gazeta-msp.pl/?id=pokaz_artykul&indeks_artykulu=210
- [36] Janmedia, 2012, Raport uzytecznosci mobilnych aplikacji bankowych
- [37] Krystian MacKozer Kozerawski, Konto pod kciukiem, czyli przegląd bankowych aplikacji,
<http://www.zafinansowani.pl/wirtualne-pieniadze/konto-pod-kciukiem-czyli-przeglad-aplikacji-bankowych-dla-smartfonow>

- [38] KANTAL WORLDPANEL, Smarfton OS market share
<http://www.kantarworldpanel.com/global/smartphone-os-market-share/>
- [39] Chris Foresman, 2014, iOS app success is a “lottery”: 60% (or more) of developers don’t break even <http://arstechnica.com/apple/2012/05/ios-app-success-is-a-lottery-and-60-of-developers-dont-break-even/>
- [40] Tristan Louis, 2013, How Much Do Average Apps Make?
<http://www.forbes.com/sites/tristanlouis/2013/08/10/how-much-do-average-apps-make/>
- [41] Zakładanie konta programisty Google Play, <https://support.google.com/googleplay/android-developer/answer/6112435?hl=pl&rd=1>
- [42] iOS Developer Program, <https://developer.apple.com/programs/ios/>
- [43] Account types, locations, and fees, <http://msdn.microsoft.com/pl-pl/library/windows/apps/jj863494.aspx>
- [44] BlackBerry World, <http://developer.blackberry.com/devzone/blackberryworld/>
- [45] Sara Angels, 2012, How Much Does it Cost to Make an App?: An Infographic
<http://www.ideatoappster.com/how-much-does-it-cost-to-make-an-app-an-infographic/>
- [46] Dhawal Barot, How much does it cost to build an Android app?
<http://www.quora.com/How-much-does-it-cost-to-build-an-Android-app>
- [47] Indiego Studio, DISCOVER THE RIGHT DESIGN - CODE-FREE
<http://www.infragistics.com/products/indigo-studio>
- [48] FluidUI, <https://www.fluidui.com/>
- [49] Apache Ant, Welcome, <http://ant.apache.org/>
- [50] Apache maven, Welcome to Apache Maven, <http://maven.apache.org/>
- [51] Gradle, What is Gradle?, <https://www.gradle.org/>

12. Wykorzystane aplikacje mobilne

pl.electoroffline	ANGIELSKI OFFLINE	14-11-2014	Edukacja
com.bk.advance.chemik	Chemik	14-11-2014	Edukacja
org.coursera.android	Coursera	15-11-2014	Edukacja
com.clevercat.craftingguide	Crafting Guide Minecraft		Edukacja
com.librus.dziennik	e-dziennik	14-11-2014	Edukacja
Gecko.Droid.PhysicsHelper	Fizyka na 5!	14-11-2014	Edukacja
com.yamlearning.geographylearning	Gra Geografia	14-11-2014	Edukacja
us.mathlab.android	Graphing Calculator by Mathlab	14-11-2014	Edukacja
appinventor.ai_abendryczek.Historia	Historia	14-11-2014	Edukacja
com.japanactivator.android.jasensei	JA Sensei	14-11-2014	Edukacja
com.freeit.java	Java Programs With Output	14-11-2014	Edukacja
net.lusil.android.kanjidraw.kana	Kana Draw	14-11-2014	Edukacja
m.mokrzycki.prawo	Kodeksy	15-11-2014	Edukacja
air.mFiszki.mFiszki.en.Demo.apk	mFISZKI Angielski w podróży	14-11-2014	Edukacja
pl.pracadomowa	PracaDomowa.pl	14-11-2014	Edukacja
com.escapestgames.starchart	Star Chart	14-11-2014	Edukacja
pl.streszczenia.app	Streszczenia.pl	14-11-2014	Edukacja
com.willy.moonburg.lms.mtg	Tabliczka mnożenia gra	14-11-2014	Edukacja
com.ted.android	TED	14-11-2014	Edukacja
pl.greenislanddev.prawojazdy	Testy na Prawo Jazdy kat. B	14-11-2014	Edukacja
com.busuu.android.enc	Ucz się języków - busuu	14-11-2014	Edukacja
appinventor.ai_kzelski.Bramka_SMS	Bramka SMS	15-11-2022	Komunikacja
com.vladlee.easyblacklist	Calls Blacklist	15-11-2023	Komunikacja
com.csipsimple	CSipSimple	15-11-2014	Komunikacja
com.facebook.orca	Facebook Messenger	15-11-2014	Komunikacja
org.mozilla.firefox	Firefox	15-11-2015	Komunikacja
pl.gadugadu	Gadu-Gadu	15-11-2016	Komunikacja
com.google.android.gm	Gmail	15-11-2017	Komunikacja
com.google.android.talk	Hangouts	15-11-2018	Komunikacja
com.mx.browser	Maxthon	15-11-2019	Komunikacja
com.opera.browser.classic	Mobile Classic	15-11-2020	Komunikacja
net.daum.android.air	mypeople Messenger	15-11-2021	Komunikacja
flyjam.InstantTraductor	Natychmiastowa Tłumacz	15-11-2022	Komunikacja
com.ninesky.browser	Ninesky	15-11-2023	Komunikacja
com.opera.mini.android	Opera Mini	15-11-2014	Komunikacja
com.skype.raider	Skype	15-11-2014	Komunikacja
ga.smsme.smsmegauuniversal	SMS-ME.GA UNIVERSAL	15-11-2015	Komunikacja
org.telegram.messenger	Telegram	15-11-2016	Komunikacja

com.teamspeak.ts3remote	TS3 Remote	15-11-2017	Komunikacja
pl.komur.android.vencommunicator	vCommunicator	15-11-2018	Komunikacja
com.viber.voip	Viber	15-11-2019	Komunikacja
com.whatsapp	WhatsApp Messenger	15-11-2020	Komunikacja
de.phl.whoscalling	Who's Calling?	15-11-2021	Komunikacja
org.coolreader	Cool Reader	14-11-2014	Książki
wp.wattpad	Darmowe książki - Wattpad	14-11-2014	Książki
com.over.joke	Dowcipów.	14-11-2014	Książki
com.zireck.dontstarve	Food Guide for Don't Starve	14-11-2014	Książki
com.cheese2135gmail.com.gta	gta V cheats		Książki
com.google.android.apps.books	Książki Google Play	14-11-2014	Książki
com.ahker.linux	Linux Guide	14-11-2014	Książki
counterpicks.leagueoflegends	LoL Counter picker	14-11-2014	Książki
pl.mb.modlitewnik	Modlitewnik	14-11-2014	Książki
kst.NWTBibleLite	NWT Bible - Lite	14-11-2014	Książki
com.snotyak.pokedexxy.free	Pokedex XY Free		Książki
com.martinvillar.android.bibliaenpolaco	Polska Biblia Gdańsk	14-11-2014	Książki
com.kodeksy	Polskie Kodeksy	14-11-2014	Książki
com.dic_o.dico_eng_pol	Słownik angielsko - polski	14-11-2014	Książki
org.leo.android.dict	Słownik LEO	14-11-2014	Książki
com.greenleaf.android.translator.enes.a	Spanish Tutor / Translator	14-11-2014	Książki
com.google.android.apps.translate	Tłumacz Google	14-11-2014	Książki
com.egmont.TransformersCubeUS	Transformers Prime: Story Cube	14-11-2014	Książki
org.wikipedia	Wikipedia	14-11-2014	Książki
uk.co.appsunlimited.wikiapp	Wikipedia Encyclopedia	14-11-2014	Książki
com.woblink.reader	Woblink	14-11-2014	Książki
com.utorrent.client	μTorrent®	15-11-2014	Multimedia
com.akazoo.music	Akazoo	15-11-2014	Multimedia
com.zieduck.cancionesdevioletta	Canciones de Violetta		Multimedia
com.topappsonline.mp3ringtones	Darmowe Dzwonki mp3	15-11-2014	Multimedia
com.bestringtonesapps.funnysayingsringtones	Dzwonki Smieszne Powiedzenia	15-11-2014	Multimedia
com.smssoundsdar	Dźwięki SMS	15-11-2014	Multimedia
smi.android.eska	eskaGO	15-11-2014	Multimedia
pl.redefine.ipla	ipla	15-11-2014	Multimedia
com.ui.LapseIt	Lapse It	15-11-2014	Multimedia
com.skycash.multikino.android	Multikino	15-11-2014	Multimedia
pl.gadugadu.openfm	OpenFM	15-11-2014	Multimedia
PlanetaFM.app	Planeta FM	15-11-2014	Multimedia
pl.tvn.player	Player	15-11-2014	Multimedia
com.rmf	RMFon.pl	15-11-2014	Multimedia

com.mobiledev.sporttvhd	Sport TV HD		Multimedia
com.spotify.mobile.android.ui	Spotify	15-11-2014	Multimedia
com.blackbeltstudio.crazysounds	Śmieszne dzwonki	15-11-2014	Multimedia
com.sonyericsson.trackid	TrackID™	15-11-2014	Multimedia
org.videolan.vlc.betav7neon	VLC	15-11-2014	Multimedia
com.wondershare.player	Wondershare Player	15-11-2014	Multimedia
com.google.android.youtube	YouTube	15-11-2014	Multimedia
com.a0soft.gphone.aCompassPlus	3D Compass	15-11-2014	Podróże
com.alltrails.alltrails	AllTrials	15-11-2014	Podróże
emis.katalog.grzybow	Atlas grzybów	15-11-2014	Podróże
com.comuto	BlaBlaCar	15-11-2023	Podróże
cgeo.geocaching	c:geo	15-11-2014	Podróże
com.couchsurfing.mobile.android	CouchSurfing	15-11-2014	Podróże
com.google.earth	Google Earth	15-11-2014	Podróże
com.eclipsim.gpsstatus2	GPS Status & Toolbox	15-11-2014	Podróże
eu.mapof.poland	Mapa Polski	15-11-2014	Podróże
com.mapfactor.navigator	MapFactor: GPS Navigation	15-11-2014	Podróże
com.google.android.apps.maps	Mapy	15-11-2014	Podróże
com.arlean.maps	Mapy i nawigacja GPS	15-11-2014	Podróże
com.tobo.mobitrip	MobiTrip	15-11-2014	Podróże
com.altconnect.android.parkslaski	Park Śląski	15-11-2014	Podróże
info.pitupitu.app	Pitu Pitu		Podróże
com.netmart.pnandroid	Polska Niezwykła	15-11-2014	Podróże
pl.polskieszlaki.PolskieSzlaki	Polskie Szlaki	15-11-2014	Podróże
pl.trail.szlakiwgorach	Szlaki w górach	15-11-2014	Podróże
com.biskis.wizzair	Wizzair	15-11-2014	Podróże
com.hrs.b2c.android	Wyszukiwanie hoteli HRS	15-11-2014	Podróże
pl.neptis.yanosik.mobi.android	Yanosik	15-11-2014	Podróże
com.accuweather.android	AccuWeather	15-11-2014	Pogoda
mobi.infolife.ezweather	EZ Pogoda HD	15-11-2014	Pogoda
com.gau.go.launcherex.gowidget.weatherwidget	GO Weather Forecast	15-11-2014	Pogoda
air.byss.mobi.instaweatherfree	InstaWeather	15-11-2014	Pogoda
com.bestapp.weather	łatwe pogody	15-11-2014	Pogoda
air.NasaSpaceViewer	NASA Space Weather Viewer	18-11-2014	Pogoda
com.devexpert.weather	Pogoda	16-11-2014	Pogoda
easy.weather	Pogoda	16-11-2014	Pogoda
com.macropinch.swan	Pogoda - Weather	16-11-2014	Pogoda
com.audioguidia.myweather	Pogoda 5 dni	16-11-2014	Pogoda
yong.tools.life.weather	Pogoda bezpłatny	16-11-2014	Pogoda
pl.tvn.meteo	Pogoda TVN Meteo	16-11-2014	Pogoda

com.ludi.climapoland	Polska Prognoza Pogody	16-11-2014	Pogoda
com.dessoft.weather	Prognoza pogody		Pogoda
air.clockAndWeather	Prognoza pogody zegar	16-11-2014	Pogoda
com.xiaad.android.thermometertrial	Thermometer (Free)	16-11-2014	Pogoda
com.vimov.weatherhd.android.free	Weather HD Free	16-11-2014	Pogoda
au.com.weatherzone.android.weatherzonefreeapp	Weatherzone	16-11-2014	Pogoda
no.nrk.yr	yr.no	16-11-2014	Pogoda
com.alarmclock.zjkstudio.a	zegar Pogda	16-11-2014	Pogoda
com.gau.go.weatherex.systemwidgetskin.zneonskin	Z-Neon Free Theme GO Weather	16-11-2014	Pogoda
com.qihoo.msafe	360 Mobile Safe	16-11-2014	Produktywność
com.adobe.reader	Adobe Reader	16-11-2014	Produktywność
com.rechild.advancedtaskkiller	Advanced Task Killer	16-11-2014	Produktywność
com.advancedprocessmanager	Android Assistant	16-11-2014	Produktywność
com.agilesoftresource	AndroZip™	16-11-2014	Produktywność
com.rs.autorun	Autorun Manager	16-11-2014	Produktywność
com.socialnmobile.dictapps.notepad.color.note	ColorNote	16-11-2014	Produktywność
com.lookout	Darmowe Antywirus	16-11-2014	Produktywność
com.dropbox.android	Dropbox	16-11-2014	Produktywność
com.synology.DSfile	DS file	16-11-2014	Produktywność
com.google.android.apps.docs	Dysk Google	16-11-2014	Produktywność
com.latedroid.juicedefender	JuiceDefender	16-11-2014	Produktywność
com.j2eeknowledge.com	Kalkulator Mem	16-11-2014	Produktywność
com.wsandroid.suite	McAfee Antivirus & Security	16-11-2014	Produktywność
com.rhmsoft.fm.hd	Menedżer Plików HD	16-11-2014	Produktywność
com.lostpixels.fieldservice	Ministry Assistant	16-11-2014	Produktywność
com.antyzero.njumeter	Nju Meter	16-11-2014	Produktywność
la.droid.qr	QR Droid™	16-11-2014	Produktywność
com.bluetornadosf.smartypants	Skyvi	16-11-2014	Produktywność
com.teamviewer.teamviewer.market.mobile	TeamViewer	16-11-2014	Produktywność
pl.orange.labs	Test Internetu	16-11-2014	Produktywność
pl.golabek.apajacyk	aPajacyk	16-11-2014	Społeczności
com.badoo.mobile	Badoo	16-11-2014	Społeczności
com.winternetexplorer1	explorer 1	16-11-2014	Społeczności
com.facebook.katana	Facebook	16-11-2014	Społeczności
com.shebatech.Instagram.instafollower.lite	Followers+	16-11-2014	Społeczności
hashgram.net	Hashgram	16-11-2014	Społeczności
com.instagram.android	Instagram	16-11-2014	Społeczności
com.minus.android	Meow Chat	16-11-2014	Społeczności
com.funhouse.wp.minecraftgarden	Minecraft Ogród		Społeczności
pl.cyberstudio.mojepaliwo	Moje Paliwo	16-11-2014	Społeczności

pl.nk.m.android	nk.pl	16-11-2014	Społeczności
krzysiek.zimon.pajacyk	Pajacyk	16-11-2014	Społeczności
pl.prophone.bls	Ratuj Życie	16-11-2014	Społeczności
com.snapchat.android	Snapchat	16-11-2014	Społeczności
com.most_wanted	The Most Wanted App	16-11-2014	Społeczności
com.tumblr	Tumblr	16-11-2014	Społeczności
com.unleash.web.turbolikez	Turbo Likez	16-11-2014	Społeczności
com.twitter.android	Twitter	16-11-2014	Społeczności
com.fsp.android.friendlocator	Znajdź moich przyjaciół!	16-11-2014	Społeczności
com.mo7sheh.pro	fm.ask - مُحَشِّش رسمى		Społeczności
de.hafas.android.pkp	Bilkom	16-11-2014	Transport
com.inno.epodroznik.android	e-podróżnik.pl	16-11-2014	Transport
com.arume.whereismycar	Gdzie jest mój samochód?	16-11-2014	Transport
eu.gingermobile	Ginger	16-11-2014	Transport
com.citynav.jakdojade.pl.android	jakdojade.pl	16-11-2014	Transport
pl.seriousimpact.lost	LOTOS	16-11-2014	Transport
pl.mobicore.mobilempk	mobileMPK	16-11-2014	Transport
pl.wasko.mpkbus	MPK Wrocław	16-11-2014	Transport
de.nextbike	Nextbike	16-11-2014	Transport
com.fogdens	Przewodnik po metrze paryskim	16-11-2014	Transport
com.imagis.radarstop	RadarSTOP	16-11-2014	Transport
com.auto.autokennzeichen	Rejestracje samochodowe EU	16-11-2014	Transport
pl.taxi5	Taxi5	16-11-2014	Transport
org.prowl.recorder	Track Recorder	16-11-2014	Transport
com.transportoid	Transportoid	16-11-2014	Transport
com.mxdata.tube.Market	Tube Map - London Underground	16-11-2014	Transport
com.visualit.tubeLondonCity	Tube Map London Underground	16-11-2014	Transport
com.magnetic.openmaps	Twoja mapa	16-11-2014	Transport
pl.ascen.ufgmobile.android.client	UFG Baza	16-11-2014	Transport
com.codilime.vetutrip	VetuTrip	16-11-2014	Transport
jp.co.goga.walkstreet	Walk Street	16-11-2014	Transport
com.slickapps	Wyszukiwarka samochodów	16-11-2014	Transport
com.jandy.zglive	Zielona Góra MZK Live	16-11-2014	Transport
pl.allegro	Allegro	16-11-2014	Zakupy
com.google.zxing.client.android	Barcode Scanner	16-11-2014	Zakupy
pl.com.androidapps.burgerkinggutscheine	Burger King Kupony	16-11-2014	Zakupy
pl.ceneo	Ceneo	16-11-2014	Zakupy
com.ebay.mobile	eBay	16-11-2014	Zakupy
com.tippingcanoe.hukd	HotUKDeals - Official	16-11-2014	Zakupy
appinventor.ai_prycpl.kfc_kupony	KFC Kupony	16-11-2014	Zakupy

pl.booncol.mcdonald.kupony	Kupony do McDonalds	16-11-2014	Zakupy
com.mpstudio.kupony	Kupony McDonalds KFC NorthFish	16-11-2014	Zakupy
com.gezbox.android.ntzst820	Lego乐高	16-11-2014	Zakupy
de.sec.mobile	Lidl	16-11-2014	Zakupy
com.lifeboat	Lifeboat+	16-11-2014	Zakupy
com.shoppinglist	List zakupów	16-11-2014	Zakupy
pl.rtproject.android.mniamo	Mniamo	16-11-2014	Zakupy
com.biuropodrozyreklamy.obi	OBI	16-11-2014	Zakupy
pl.tablica	OLX.pl	16-11-2014	Zakupy
pl.payback	PAYBACK PL	16-11-2014	Zakupy
com.MidCenturyMedia.Shopper.light	Shopper Grocery Shopping List	16-11-2014	Zakupy
pl.skapiec.android	Skapiec.pl	16-11-2014	Zakupy
com.waclawiec.starbucksfinder	Starbucks Finder	16-11-2014	Zakupy
thecouponsapp.coupon	The Coupons App	16-11-2014	Zakupy