Politechnika Warszawska

WYDZIAŁ ELEKTRYCZNY

Instytut Elektrotechniki Teoretycznej
i Systemów Informacyjno-Pomiarowych
Zakład Elektrotechniki Teoretycznej
i Informatyki Stosowanej

Praca dyplomowa magisterska

na kierunku Informatyka
w specjalności Inżynieria oprogramowania

Porównanie progresywnych aplikacji internetowych i aplikacji hybrydowych

Marta Wiśniewska
nr albumu 252217

promotor
prof. nzw. dr hab. inż. Krzysztof Siwek

WARSZAWA 2018

PORÓWNANIE PROGRESYWNYCH APLIKACJI INTERNETOWYCH I APLIKACJI HYBRYDOWYCH

Streszczenie

Praca składa się z krótkiego wstępu jasno i wyczerpująco opisującego oraz uzasadniającego cel pracy, trzech rozdziałów (2-4) zawierających opis istniejących podobnych rozwiązań, komponentów rozpatrywanychjako kandydaci do tworzonego systemu i wreszcie zagadnień wydajności wirtualnych rozwiązań. Piąty rozdział to opis środowiska obejmujący opis konfiguracji środowiska oraz przykładowe ćwiczenia laboratoryjne. Ostatni rozdział pracy to opis możliwości dalszego rozwoju projektu.

Słowa kluczowe: progresywne aplikacje internetowe, aplikacje hybrydowe, aplikacje natywne

THESIS TITLE

Abstract

This thesis presents a novel way of using a novel algorithm to solve complex problems of filter design. In the first chapter the fundamentals of filter design are presented. The second chapter describes an original algorithm invented by the authors. Is is based on evolution strategy, but uses an original method of filter description similar to artificial neural network. In the third chapter the implementation of the algorithm in C programming language is presented. The fifth chapter contains results of tests which prove high efficiency and enormous accuracy of the program. Finally some posibilities of further development of the invented algoriths are proposed.

Keywords: progressive web apps, hybrid apps, native apps

WARSZAWA, 1 czerwca 2018

POLITECHNIKA WARSZAWSKA
WYDZIAŁ ELEKTRYCZNY

OŚWIADCZENIE

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa magisterska pt. Porównanie progresywnych aplikacji internetowych i aplikacji hybrydowych:

- została napisana przeze mnie samodzielnie,
- nie narusza niczyich praw autorskich,
- nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam, że przedłożona do obrony praca dyplomowa nie była wcześniej podstawą postępowania związanego z uzyskaniem dyplomu lub tytułu zawodowego w uczelni wyższej. Jestem świadom, że praca zawiera również rezultaty stanowiące własności intelektualne Politechniki Warszawskiej, które nie mogą być udostępniane innym osobom i instytucjom bez zgody Władz Wydziału Elektrycznego.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Marta Wiśniewska.....

Spis treści

Rozdział 1

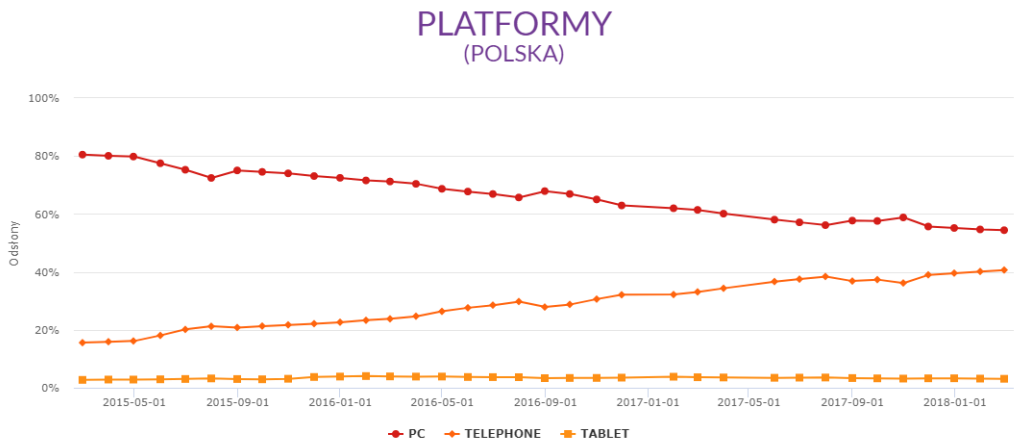
Wstęp

Użytkownicy spędzają coraz więcej czasu na urządzeniach mobilnych, więc zadbanie o to, aby witryny mobilne spełniały ich oczekiwania, jest niezwykle ważne. Z jednej strony jeśli witryna Twojego klienta ładuje się zbyt wolno, użytkownicy zrezygnują z korzystania z niej. Z drugiej strony szybko ładująca się, ale źle zaprojektowana witryna utrudnia użytkownikom wykonanie pożądanych działań. W świecie, gdzie urządzenia mobilne są coraz bardziej powszechne, oczekiwania konsumentów są wysokie. W tym module dowiesz się, dlaczego wygoda użytkowników i szybkość witryn mobilnych są ważne.

Rozdział 2

Aplikacje mobilne

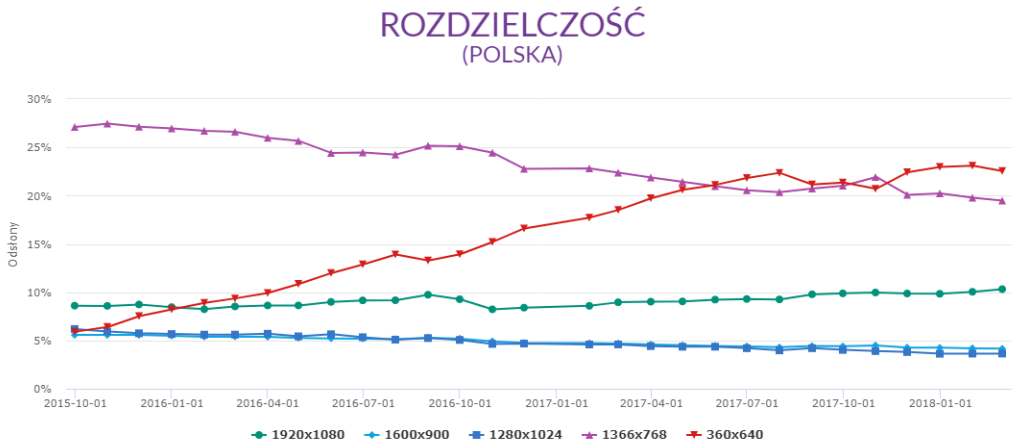
Urządzenia mobilne na przestrzeni ostanich kilku lat zdobyły ogromną popularność. Ma to duży wpływ na rozwój aplikacji internetowych i możliwości ich wykorzystania na różnych urządzeniach mobilnych. Użytkownicy aplikacji sieciowych coraz częściej używają urządzeń mobilnych, rezygnując z korzystania z przeglądarki komputera. Na wykresie 2.1 zaprezentowano jak na przestrzeni lat 2015-2018 zmieniał się udział odsłon aplikacji sieciowych z różnych platform - komputera, telefonu, tabletu.



Rysunek 2.1: Udział odsłon z różnych platform
Źródło: Gemius [1], 01/01/2015 - 01/03/2018r

Okazuje się, że zanotowano 20% spadek udziału odsłon z komputera na rzecz urządzeń mobilnych. Jeżeli aktualny trend utrzyma się można prognozować, że w przyszłym roku ilość odsłon z telefonów wyrówna się z ilością odsłon z przeglądarek komputera, a w kolejnych latach procentowy udział

odsłon z tego typu platform może być największy.
Warto przyjrzeć w jakich rozdzielczościach ekranu są wyświetlane aplikacje internetowe i przeanalizować wykres 2.2



Rysunek 2.2: Udział różnych rozdzielczości
Źródło: Gemius [2], 01/10/2015 - 01/03/2018r

Ze statystyk wynika, że aktualnie najczęściej aplikacje uruchamiane są w najmniejszej rozdzielczości (360x640). Znaczny wzrost udziału odsłon aplikacji w rozdzielczości 360x640 zaobserwowano w ciągu ostatnich kilku lat. Zaprezentowane dane potwierdzają słuszność stosowania podejścia "mobile first" przy projektowaniu interfejsu graficznego aplikacji. Zakłada on dostarczenie jak najlepszych doświadczeń użytkownikowi końcowemu dla urządzeń mobilnych. Oprócz tego dane potwierdzają konieczność rozbudowy funkcjonalności aplikacji internetowych z uwzględnieniem preferencji i wymagań użytkowników korzystających z urządzeń mobilnych.

2.1 Technologie dla urządzeń mobilnych

W odpowiedzi na potrzeby użytkowników korzystających z urządzeń mobilnych powstają nowe technologie, które umożliwiają tworzenie aplikacji przyjaznych użytkownikowi. Wyróżnić można kilka podstawowych rodzajów aplikacji. Wśród nich znajdują się te serwowane na zdalnym serwerze przez protokół HTTP/HTTPS: mobilne aplikacje sieciowe oraz progresywne aplikacje internetowe. Są one tworzone za pomocą technologii webowych, czyli języka HTML, CSS i JavaScript. Z aplikacji takich można korzystać poprzez przeglądarkę internetową dostępną na urządzeniu mobilnym. Nowym rozwiązaniem, którego celem jest jak najszybsze dostarczenie treści użytkownikowi

korzystającemu z aplikacji na urządzeniu mobilnym są przyspieszone strony mobilne (ang. Accelerated Mobile Pages). Innym rodzajem są aplikacje wymagające instalacji na danej platformie. Do tej grupy zaliczają się aplikacje natywne oraz aplikacje hybrydowe.
Poniżej zaprezentowano pięć kluczowych technologii stosowanych przy tworzeniu produktów na urządzenia mobilne, z uwzględnieniem ich wad i zalet.

Mobilne aplikacje webowe (ang. Mobile Web Apps)

Aplikacje tworzone przy pomocy hipertekstowego języka znaczników HTML, kaskadowego arkusza stylów CSS oraz skryptowego języka programowania JavaScript. Aplikacja webowa pracuje na zdalnym serwerze i jest serwowana przez protokół HTTP(ang. Hypertext Transfer Protocol). Użytkownik końcowy może korzystać z aplikacji na urządzeniu mobilnym poprzez przeglądarkę internetową zainstalowaną na danym urządzeniu. Dostęp do aplikacji możliwy jest poprzez unikalny adres URL. Zaletą tej technologii jest stosunkowo krótki czas tworzenia, brak konieczności instalacji i uniwersalność - raz napisana aplikacja działa na różnych platformach i urządzeniach. Wymaga przeglądarki i dostępu do sieci. Wadą tego typu technologii jest brak możliwości pracy w trybie Offline oraz brak dostępu do funkcjonalności typowych dla urządzeń mobilnych (np. geolokalizacji, kamery, powiadomień).

Aplikacje natywne (ang. Native Apps)

Aplikacje przeznaczone dla konkretnej platformy mobilnej (np. Android lub iOS). Wymagają stosowania przez deweloperów odpowiednich języków programowania oraz narzędzi deweloperskich (np. Eclipse i Java dla platformy Android, Xcode i język C dla iOS). Biorąc pod uwagę doświadczenia użytkownika końcowego korzystającego z urządzenia mobilnego, tego typu aplikacje wyglądają i działają najlepiej. Zaletą tej technologii jest możliwość pracy w trybie Offline, działanie w trybie pełnoekranowym, możliwość korzystania z funkcjonalności urządzeń mobilnych (takich jak kamera, geolokalizacja, bluetooth), natychmiastowy czas ładowania pierwszego ekranu, szybki dostęp z poziomu ekranu głównego. Niekorzystny z perspektywy użytkownika jest proces instalacji aplikacji, który często wymaga kilku kroków i zajmuje dużo czasu. Proces tworzenia aplikacji natywnych jest bardziej skomplikowany niż w przypadku aplikacji webowych. Wymaga od programisty doświadczenia i znajomości konkretnej technologii. W przypadku chęci uzyskania produktu wieloplatformowego konieczne jest tworzenie kilku aplikacji.

Progresywne aplikacje internetowe (ang. Progressive Web Apps)

Szczególny rodzaj mobilnych aplikacji webowych, który powstał w 2015 roku. Rozwiązanie to polega na progresywnym rozszerzeniu aplikacji sieciowej, w taki sposób, aby wyglądała i działała jak aplikacja natywna. Progresywne aplikacje webowe łączą w sobie najlepsze cechy aplikacji natywnych z najlepszymi cechami aplikacji sieciowych. Dzięki temu powstaje aplikacja, która nie wymaga instalacji. Uruchamiana jest dzięki przeglądarce zainstalowanej na urządzeniu mobilnym. Aplikacja taka jest szybka, niezawodna (działa w trybie offline lub przy niestabilnym połączeniu sieciowym) i angażująca użytkownika [3]. Zaangażowanie to uzyskuje się dzięki możliwości przypięcia ikony aplikacji do ekranu głównego urządzenia mobilnego i szybkiej możliwości uruchomienia oraz dzięki możliwości wykorzystania powiadomień typu push, co nie jest typową funkcjonalnością aplikacji sieciowych. Ten rodzaj technologii pozwala na tworzenie produktów bezpiecznych (ze zwględu na serwowanie przez protokół HTTPS) oraz przyjaznych użytkownikowi. Ponadto możliwe jest wykorzystanie funkcjonalności typowych dla urządzeń mobilnych (np. kamera, mikrofon, kalendarz, bluetooth). Problemem w tego typu aplikacjach jest wsparcie przeglądarek internetowych. Tylko nowoczesne przeglądarki umożliwiają korzystanie z pełnej funkcjonalności progresywnych aplikacji internetowych. Szczegółowy opis technologii umieszczono w rozdziale "Progresywne aplikacje internetowe".

Aplikacje hybrydowe (ang. Hybrid Apps)

Idea aplikacji hybrydowych polega na wykorzystaniu najlepszych cech aplikacji natywnych i aplikacji webowych Do stworzenia aplikacji wykorzystuje się języki typowe dla aplikacji webowych (HTML, JavaScript, CSS). Kod źródłowy aplikacji hybrydowej jest pakowany do specjalnego kontenera Widoku Sieciowego (WebView) [4]. Takie rozwiązanie umożliwia tworzenie aplikacji, które wyglądają jak aplikacje natywne i są wieloplatformowe. Produkty tego typu wymagają instalacji i są dystrybuowane przez sklepy producentów systemów operacyjnych. Aplikacje hybrydowe działają w trybie online i offline. Są bezpiecznym rozwiązaniem i umożliwiają korzystanie z funkcjonalności urządzeń mobilnych dzięki zastosowaniu pluginów i odpowiednich narzędzi. Ten rodzaj technologii bardzo się rozwija w ostatnim czasie. Ciągle powstają nowe narzędzia i frameworki pozwalające na tworzenie wieloplatformowych aplikacji za pomocą technologii webowych. Szczegółowy opis aplikacji hybrydowych zawarty jest w rozdziale "Aplikacje hybrydowe".

Przyspieszone strony mobilne (ang. Accelerated Mobile Pages)

Technologia Accelerated Mobile Pages powstała w wyniku współpracy firmy Google z wydawcami na całym świecie w ramach projektu Digital News Initiative. Główna idea tego rozwiązania to maksymalna optymalizacja czasu renderowania elementów znajdujących się na stronie internetowej oraz redukcja ładowania dodatkowych bibliotek do minimum [5]. W celu realizacji tych założeń powstała biblioteka typy open-source - AMP. Służy ona do tworzenia stron internetowych, które szybko się ładują i są atrkacyjne dla użytkowników końcowych [6]. Przy tworzeniu tego typu stron korzysta się z języków programowania typowych dla aplikacji webowych (HTML, JavaScript, CSS). W przypadku tego rozwiązania nie ma problemu z kompatybilnością przeglądarek i platform, na których strony AMP są używane. Ten rodzaj technologii nie daje możliwości korzystania z funkcjonalności typowych dla urządzeń mobilnych jak kamera, bluetooth czy kalendarz.

Rozdział 3

Progresywne aplikacje internetowe

Twórcą technologii jest firma Google. Progresywne aplikacje internetowe powstały w odpowiedzi na ograniczenia standardowych aplikacji webowych względem urządzeń mobilnych. PWA (ang Progressive Web Apps) tworzone są za pomocą hipertekstowego języka znaczników HTML, kaskadowego arkusza stylów CSS oraz skryptowego języka programowania JavaScript. Idea tego rozwiązania polega na rozszerzeniu podstawowej funkcjonalności aplikacji internetowych w sposób progresywny, aby w trakcie korzystania z PWA doświadczenia użytkownika były podobne, jak przy korzystaniu z klasycznej aplikacji natywnej.

Cechy Progresywnych Aplikacji Internetowych

- **niezawodność i niezależność od sieci** - aplikacja ma dostarczać treści użytkownikowi w sposób niezawodny, bez względu na stan sieci (również w stanie Offline). Realizacja tej cechy jest możliwa dzięki zastosowaniu mechanizmu Service Worker w połączeniu z interfejsem API magazynu do przechowywania danych po stronie klienta przeglądarki np. Cache Storage API lub IndexedDB. Szczegółowy opis mechanizmu umieszczono w podrozdziale "Service Worker",
- **szybkość działania** - odpowiedzi serwera na interakcje użytkownika powinny być natychmiastowe, animacje podczas przewijania powinny działać w sposób płynny. W celu optymalizacji wydajności wykorzystuje się ponowne wykorzystanie pobranych zasobów (buforowanie).Ma to wpływ róznież ze względów ekonomicznych. Minimalizacja kolejnych cykli wymiany danych nie blokuje przeglądarki i nie przyczynia się do większych kosztów transferu,

- **możliwość angażowania użytkownika** - aplikacje typu PWA mogą zwiększać zaangażowanie użytkowników dzięki zastosowaniu notyfikacji typu push oraz możliwości przypięcia ikony aplikacji do ekranu głównego urządzenia mobilnego. Opcja ta jest możliwa dzięki standardowi Web App Manifest. Szczegółowy opis standardu znajduje się w podrozdziale "Web App Manifest",
- **bezpieczeństwo** - wymagane jest bezpieczne połączenie, aplikacja musi być serwowana przez protokół HTTPS,
- **integrowalne** - istnieje możliwość integracji z innymi aplikacjami oraz urządzeniami smartphone'a poprzez zewnętrzne API,
- **responsywność** - aplikacja dobrze wygląda bez względu na urządzenie i ekran, z którego korzysta użytkownik,
- **progresywność** - aplikacja działa niezależnie od użytkownika i przeglądarki, z której on korzysta,
- **wykrywalność** - wykrywanie aplikacji poprzez silniki przeglądarek jest możliwe dzięki standardom W3C: manifest W3C [8] oraz Service Worker W3C [9],
- **uniwersalność** - aplikacja działa poprawnie na różnych platformach,
- **dostępność** - progresywne aplikacje internetowe są zarówno instalowalne (istnieje możliwość przypięcia ikony aplikacji do ekranu głównego z opcją szybkiego uruchomienia) jak i linkowalne tzn. do aplikacji można dostać się przez przeglądarkę za pomocą unikalnego adresu URL, dzięki temu udostępnianie treści jest łatwe,
- **aktualność danych** - dzięki funkcjonalnościom Service Workera możliwe jest odświeżanie danych, dzięki czemu dane aplikacji są aktualne.

podlinkuj https://developers.google.com/web/updates/2015/12/getting-started-pwa

3.1 Wymagania stawiane progresywnym aplikacjom sieciowym

Progresywne aplikacje webowe są serwowane na zdalnym serwerze. Dostęp do nich możliwy jest tak jak w przypadku klasycznych aplikacji sieciowych przez przeglądarkę. Z technicznego punktu widzenia zwykła aplikacja webowa jest nazywana progresywną, gdy spełnione są następujące warunki [?]:

- aplikacja serwowana jest poprzez protokół HTTPS (ang. Hypertext Transfer Protocol Secure), co zapewnia użytkownikowi bezpieczne korzystanie z aplikacji,
- aplikacja korzysta z minimum jednego mechanizmu Service Worker,
- aplikacja wykorzystuje Web App Manifest, który zawiera metadane dotyczące nazwy aplikacji, ikon, startowego adresu URL, itp.

3.2 Architektura powłoki aplikacji progresywnych

Powłoka systemowa aplikacji tworzy interfejs użytkownika i generowana jest za pomocą kodu HTML, CSS, JavaScript. Jest to minimalny zestaw plików niezbędny do załadowania strony głównej aplikacji. W przypadku zwykłej aplikacji sieciowej przy każdym wejściu na stronę pobierane są wszystkie pliki (treści dynamiczne jak i statyczne). Wiąże się to z pewnym czasem oczekiwania użytkownika na załadowanie aplikacji. Istnieją pliki, które są wykorzystywane wielokrotnie. Progresywne aplikacje internetowe umożliwiają przechowywanie takich plików w pamięci podręcznej przeglądarki (cache). Dzięki temu możliwa jest minimalizacja czasu oczekiwania użytkownika aplikacji na załadowanie strony.
Powłokę systemową aplikacji progresywnych można porównać do pakietu kodu aplikacji natywnej publikowanego w sklepie. Powłoka systemowa odpowiada za niezawodność i wydajność. Dzięki niej interfejs użytkownika zapisuje się lokalnie, a treść pobierana jest dynamicznie przez interfejs API. Wymagania stawiane powłoce systemowej to przede wszystkim szybkość wczytywania, dynamiczne wyświetlanie treści, możliwość zapisania w pamięci podręcznej.
Architektura powłoki systemowej aplikacji to pośrednik między infrastrukturą aplikacji i interfejsem, a danymi. Dzięki mechanizmowi Service Worker możliwe jest zapisanie lokalne w pamięci podręcznej infrastruktury i interfejsu.

Korzyści wynikające z modelu powłoki

Zastosowanie architektury powłoki niesie ze sobą wiele korzyści dla użytkowników aplikacji. Wśród nich można wymienić:

- Niezawodna wydajność i szybkość działania aplikacji.
  Powtórne otwieranie strony jest bardzo szybkie. Treści statyczne oraz elementy tworzące interfejs użytkownika (np. HTML, CSS, JavaScript, obrazy) są zapisywane w pamięci podręcznej przeglądarki (pamięci cache). Wykonuje się to przy pierwszym uruchomieniu aplikacji w danej przeglądarce. Treść może trafić do pamięci podręcznej przeglądarki przy pierwszym uruchomieniu aplikacji, ale jest ładowana w momencie, w którym jest potrzebna do prawidłowego działania aplikacji (określona przez programistę w kodzie źródłowym aplikacji),
- Interakcje jak w aplikacjach natywnych.
  Użytkownicy, którzy korzystają z aplikacji sieciowych z architekturą powłoki mają doświadczenia (ang. user experience) podobne jak przy korzystaniu z aplikacji natywnej. Czas interakcji na akcję użytkownika jest bardzo krótki. Nawigacja jest podobna do nawigacji w aplikacjach natywnych. Aplikacje progresywne działają w trybie Offlie, co jest nietypowe dla aplikacji sieciowych.
- Ekonomiczne użycie danych.
  Podejście zakładające minimalne użycie danych i rozsądne zapisywanie danych do pamięci podręcznej. Zapisywanie zbyt dużej ilości mało istotnych danych (np. obrazów w dużym rozmiarze, które są wyświetlane tylko na jednej stronie) powoduje pobieraniem zbyt dużej ilości danych, niż jest to potrzebne. Wpływa to na koszt połączenia i danych.

Wymagania stawiane architekturze powłoki aplikacji

Powłoka aplikacji powinna [7]:

- szybko się ładować,
- używać jak najmniejszej ilości danych,
- używać zasobów statycznych z pamięci podręcznej,
- oddzielać treści od nawigacji,
- pobierać i wyświetlać treści specyficzne dla strony(HTML, JSON, itp.),
- dynamicznie wyświetlać treści

https://blog.ionicframework.com/what-is-a-progressive-web-app/ App Shell
https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-concepts/content/docs/i
to-progressive-web-app-architectures.html

Wzorce architektoniczne PWA

PWA Architectural Patterns

3.3 Standardy progresywnych aplikacji webowych

3.3.1 Service Worker

Service Worker jest rodzajem Web Worker'a, który jest uruchamiany razem z całą aplikacją sieciową, ale jego czas życia zależy od realizacji zdarzeń, które są przypisane w aplikacji. Mechanizm zawiera sieć proxy https://w3c.github.io/ServiceWor
Some of its services include a network proxy written in JavaScript that intercepts HTTP requests made from web pages (including HTTPS). For example, a service worker can intercept outgoing network requests and provide alternatives when the network is offline (such as by serving up cached web content). Service workers can also respond to incoming events, such as push notifications. Service workers depend on two APIs to work effectively: Fetch (a standard way to retrieve content from the network) and Cache (a persistent content storage for application data. This cache is persistent and independent from the browser cache or network status.)

3.3.2 Web App Manifest

web app manifest W3C Spec https://w3c.github.io/manifest/

A JSON-formatted file named manifest.json that is a centralized place to put metadata that controls how the web application appears to the user and how it can be launched. (Do not confuse this with amanifestfile used by AppCache.) PWAs use the web app manifest to enable ądd to homescreen".

3.4 Praca w trybie Offline

http://grinninggecko.com/2011/02/24/developing-cross-platform-html5-offline-app-1/ https://www.html5rocks.com/en/tutorials/offline/storage/

3.5 Powiadomienia

Push push push

3.6 Ograniczenia

cache'owanie - limity https://cloudfour.com/thinks/ios-doesnt-support-progressive-web-apps-so-what/

3.7 Narzędzia deweloperskie

https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-concepts/content/docs/pwa-analysis-tool.html

Lighthouse

Zakładka Applications

Rozdział 4

Aplikacje hybrydowe

4.1 Porównanie natywnych aplikacji i hybrydowych

4.2 Przegląd narzędzi

4.3 Wady i zalety

Rozdział 5

Implementacja aplikacji

/

Rozdział 6

Porównanie progresywnej aplikacji internetowej i aplikacji hybrydowej

Rozdział 7

Wnioski

Bibliografia

[1] http://ranking.gemius.com/pl/ranking/platforms/

[2] http://ranking.gemius.com/pl/ranking/resolutions/

[3] https://developers.google.com/web/progressive-web-apps/

[4] https://blog.strefakursow.pl/co-musisz-wiedziec-zanim-zaczniesz-tworzyc-aplikacje-mobilne/

[5] http://antyweb.pl/strony-mobilne-jeszcze-nigdy-nie-wczytywaly-sie-tak-szybko-google-przedstawia-projekt-accelerated-mobile-pages/

[6] https://www.ampproject.org/ru/learn/overview/

[7] http://webagility.com/posts/how-progressive-web-apps-make-the-web-great-again

[8] https://w3c.github.io/manifest/

[9] https://w3c.github.io/ServiceWorker/

Opinia

o pracy dyplomowej magisterskiej wykonanej przez dyplomanta
Zdolnego Studenta i Pracowitego Kolegę
Wydział Elektryczny, kierunek Informatyka, Politechnika Warszawska
Temat pracy
TYTUŁ PRACY DYPLOMOWEJ

Promotor: **dr inż. Miły Opiekun**
Ocena pracy dyplomowej: **bardzo dobry**

Treść opinii

Celem pracy dyplomowej panów dolnego Studenta i Pracowitego Kolegi było opracowanie systemu pozwalającego symulować i opartego o oprogramowanie o otwartych źródłach (ang. Open Source). Jak piszą Dyplomanci, starali się opracować system, który łatwo będzie dostosować do zmieniających się dynamicznie wymagań, będzie miał niewielkie wymagania sprzętowe i umożliwiał dalszą łatwą rozbudowę oraz dostosowanie go do potrzeb. Przedstawiona do recenzji praca składa się z krótkiego wstępu jasno i wyczerpująco opisującego oraz uzasadniającego cel pracy, trzech rozdziałów (2-4) zawierających opis istniejących podobnych rozwiązań, komponentów rozpatrywanychjako kandydaci do tworzonego systemu i wreszcie zagadnień wydajności wirtualnych rozwiązań. Piąty rozdział to opis przygotowanego przez Dyplomantów środowiska obejmujący opis konfiguracji środowiska oraz przykładowe ćwiczenia laboratoryjne. Ostatni rozdział pracy to opis możliwości dalszego rozwoju projektu. W ramach przygotowania pracy Dyplomanci zebrali i przedstawili w bardzo przejrzysty sposób duży zasób informacji, co świadczy o dobrej orientacji w nowoczesnej i ciągle intensywnie rozwijanej tematyce stanowiącej zakres pracy i o umiejętności przejrzystego przedstawienia tych wyników. Praca zawiera dwa dodatki, z których pierwszy obejmuje wyniki eksperymentów i badań nad wydajnością, a drugi to źródła skryptów budujących środowisko.

Dyplomanci dość dobrze zrealizowali postawione przed nimi zadanie, wykazali się więc umiejętnością zastosowania w praktyce wiedzy przedstawionej w rozdziałach 2-4. Uważam, że cele postawione w założeniach pracy zostały pomyślnie zrealizowane. Proponuję ocenę bardzo dobrą (5).

(data, podpis)

Recenzja

pracy dyplomowej magisterskiej wykonanej przez dyplomanta
Zdolnego Studenta i Pracowitego Kolegę
Wydział Elektryczny, kierunek Informatyka, Politechnika Warszawska
Temat pracy
TYTUŁ PRACY DYPLOMOWEJ

Recenzent: **prof. nzw. dr hab. inż. Jan Surowy**
Ocena pracy dyplomowej: **bardzo dobry**

Treść recenzji

Celem pracy dyplomowej panów dolnego Studenta i Pracowitego Kolegi było opracowanie systemu pozwalającego symulować i opartego o oprogramowanie o otwartych źródłach (ang. Open Source). Jak piszą Dyplomanci, starali się opracować system, który łatwo będzie dostosować do zmieniających się dynamicznie wymagań, będzie miał niewielkie wymagania sprzętowe i umożliwiał dalszą łatwą rozbudowę oraz dostosowanie go do potrzeb. Przedstawiona do recenzji praca składa się z krótkiego wstępu jasno i wyczerpująco opisującego oraz uzasadniającego cel pracy, trzech rozdziałów (2-4) zawierających bardzo solidny i przejrzysty opis: istniejących podobnych rozwiązań (rozdz. 2), komponentów rozpatrywanychjako kandydaci do tworzonego systemu (rozdz. 3) i wreszcie zagadnień wydajności wirtualnych rozwiązań, zwłaszcza w kontekście współpracy kilku elementów sieci (rozdział 4). Piąty rozdział to opis przygotowanego przez Dyplomantów środowiska obejmujący opis konfiguracji środowiska oraz przykładowe ćwiczenia laboratoryjne (5 ćwiczeń). Ostatni, szósty rozdział pracy to krótkie zakończenie, które wylicza także możliwości dalszego rozwoju projektu. W ramach przygotowania pracy Dyplomanci zebrali i przedstawili w bardzo przejrzysty sposób duży zasób informacji o narzędziach, Rozdziały 2, 3 i 4 świadczą o dobrej orientacji w nowoczesnej i ciągle intensywnie rozwijanej tematyce stanowiącej zakres pracy i o umiejętności syntetycznego, przejrzystego przedstawienia tych wyników. Drobne mankamenty tej części pracy to zbyt skrótowe omawianie niektórych zagadnień technicznych, zakładające dużą początkową wiedzę czytelnika i dość niestaranne podejście do powołań na źródła. Utrudnia to w pewnym stopniu czytanie pracy i zmniejsza jej wartość dydaktyczną (a ta zdaje się być jednym z celów Autorów), ale jest zrekompensowane zawartością merytoryczną. Praca zawiera dwa dodatki, z których pierwszy obejmuje wyniki eksperymentów i badań nad wydajnością, a drugi to źródła skryptów budujących środowisko. Praca zawiera niestety dość dużą liczbę drobnych błędów redakcyjnych, ale nie wpływają one w sposób istotny na na jej czytelność i wartość. W całej pracy przewijają się samodzielne, zdecydowane wnioski

Autorów, które są wynikiem własnych i oryginalnych badań. Rozdział 5 i dodatki pracy przekonują mnie, że Dyplomanci dość dobrze zrealizowali postawione przed nimi zadanie. Pozwala to stwierdzić, że wykazali się więc także umiejętnością zastosowania w praktyce wiedzy przedstawionej w rozdziałach 2-4. Kończący pracę rozdział szósty świadczy o dużym (ale moim zdaniem uzasadnionym) poczuciu własnej wartości i jest świadectwem własnego, oryginalnego spojrzenia na tematykę przedstawioną w pracy dyplomowej. Uważam, że cele postawione w założeniach pracy zostały pomyślnie zrealizowane. Proponuję ocenę bardzo dobrą (5).

(data, podpis)