POLSKO-JAPOŃSKA WYŻSZA SZKOŁA TECHNIK KOMPUTEROWYCH

Wydział Informatyki

Katedra Inżynierii Oprogramowania

Inżynieria Oprogramowania i Baz Danych

Marcin Czerwiak

Nr albumu 7881

Kreator Stron Mobilnych

Praca magisterska napisana
pod kierunkiem:
dr inż. Mariusz Trzaska

Warszawa, czerwiec 2012

Streszczenie

Praca traktuje o potrzebie tworzenia mobilnych stron internetowych. Autor zwraca uwagę na problemy z jakimi zmagają się deweloperzy przy tworzeniu stron www na smartfony oraz inne urządzenia przenośne. Analizuje dostępne gotowe narzędzia pod kątem ich funkcjonalności, a następnie szczegółowo omawia technologie mobilne i dokonuje ich oceny pod kątem możliwości wykorzystania w procesie budowy nowoczesnych mobilnych stron internetowych. Praca nad problemem pozwoliła stworzyć prototyp kreatora takich stron. Narzędzie oparte na technice „przeciągnij i upuść" pozwala wygenerować stronę kompatybilną z najpopularniejszymi przeglądarkami mobilnymi. Kreator jest aplikacją webową bazująca na technologiach takich jak: PHP5, jQuery Mobile, Extjs.

Spis treści

1. Wstęp

W 2011 roku po raz pierwszy odnotowano wyższą sprzedaż smartfonów względem komputerów osobistych. Według danych firmy Canalys było to odpowiednio 488 mln sztuk wobec 415 mln sprzedanych urządzeń [1]. Prognozy firmy Juniper Research [2] podają, że w 2013 roku 530 mln użytkowników będzie korzystało z bankowości mobilnej, a za 5 lat liczba użytkowników mobilnego Internetu przewyższy ilość osób łączących się z globalną siecią w sposób tradycyjny (jak przewiduje Międzynarodowa Unia Telekomunikacyjna (ITU) [3]). Tak gwałtowny wzrost liczby użytkowników korzystających z Internetu mobilnego stawia olbrzymie wyzwanie przed projektantami i twórcami stron internetowych.

Cieszące się ogromną popularnością mobilne strony internetowe są prawdziwym wyzwaniem dla programistów prześcigających się w udoskonalaniu narzędzi, które ułatwią tworzenie stron na urządzenia przenośne. Najpoważniejszym problemem tych kreatorów jest uniwersalność kodu, który musi działać na wielu modelach urządzeń mobilnych. Kolejną trudnością staje się kompatybilność. Oprócz tego, że strony muszą współpracować z różnymi platformami, każde urządzenie może korzystać z dowolnej liczby mobilnych przeglądarek internetowych. Na przykład, użytkownik może uzyskać dostęp do witryny Android za pomocą rodzimej przeglądarki Android, ale może również zainstalować Opera Mini lub Firefox Mobile. Aplikacje mobilne można podzielić na dwa główne typy:

- aplikacje natywne, które instalowane są na telefonach i które działają bezpośrednio na urządzeniu;
- aplikacje webowe, do których dostęp jest możliwy poprzez przeglądarkę internetową w urządzeniu.

Aplikacje natywne stają się coraz bardziej popularne, o czym może świadczyć finansowy sukces App Store należącego do Apple'a oraz Google Play firmy Google. Dzieje się tak nie bez powodu - aplikacje natywne mają szereg zalet: są szybkie, mają dostęp do wszystkich zasobów danego urządzenia mobilnego i w pełni wykorzystują jego możliwości. Mają one jednak jedno poważne ograniczenie: nie są przenośne. Dlatego, jeśli zależy nam na dostępności dla wielu platform, taka aplikacja musi zostać napisana w wielu językach (w wyniku czego otrzymujemy wiele wersji kodu do utrzymania), albo przy użyciu warstwy pośredniczącej (abstrakcji) jaką mogą być biblioteki takie jak: Titanium Mobile lub PhoneGap.

Liczba użytkowników korzystających z urządzeń mobilnych oraz ilość treści dostosowanej do wyświetlania za pomocą tych urządzeń rośnie w błyskawicznym tempie. Wiele wskazuje na to, że w najbliższym czasie ten trend będzie kontynuowany. Coraz częściej abonenci usług telefonii komórkowej przeglądają mobilne witryny internetowe i stale zaopatrują się w coraz bardziej zaawansowane urządzenia mobilne oferujące przeglądarki, które obsługują standardy właściwe tradycyjnemu Internetowi.

1.1. Cel pracy

Nadrzędnym celem pracy będzie szczegółowa analiza wymogów i technologii używanych przy tworzeniu stron internetowych na urządzenia mobilne. Autor poruszy zagadnienia techniczne leżące po stronie dewelopera jak i sensoryczne podyktowane preferencjami użytkownika. Uwzględni ograniczenia wynikające z wielkości wyświetlacza urządzeń oraz mocy obliczeniowej ich procesorów.

Efektem analizy poruszonych w pracy problemów będzie prototyp narzędzia - kreator mobilnych stron internetowych.

1.2. Rozwiązania przyjęte w pracy

Stworzony prototyp jest aplikacją opartą na technologiach webowych. Wykorzystuje on biblioteki takie jak:

- Po stronie serwera (Server Side)
    - PHP 5
    - Framework Symfony 2
- Po stronie klienta (Client Side)
    - Bibloteka Extjs
    - Framework jQuery Mobile, jQuery

1.3. Rezultaty pracy

Bezpośrednim wynikiem pracy jest zaprezentowanie dostępnych narzędzi, bibliotek oraz technologii niezbędnych do stworzenia nowoczesnej strony internetowej na urządzenia mobilne. Pośrednim wynikiem jest działająca aplikacja internetowa będąca funkcjonalną odpowiedzią na problemy stawiane w pracy.

1.4. Organizacja pracy

W rozdziale pierwszym po krótce omówiona będzie tematyka problemu analizowanego w niniejszej pracy magisterskiej. W rozdziale drugim i trzecim zostaną zaprezentowane istniejące na rynku gotowe rozwiązania oraz technologie używane przy tworzeniu stron mobilnych. Rozdział czwarty opisywał będzie narzędzia i technologie na bazie których powstanie prototyp kreatora szczegółowo zaprezentowany w rozdziale piątym. W ostatniej części pracy zebrane zostaną uwagi i wnioski wypływające z analizy tematu pracy oraz wskazane zostaną możliwe kierunki zmian prowadzące do optymalizacji procesu tworzenia stron internetowych na urządzenia mobilne.

2. Istniejące rozwiązania

W obecnym momencie na runku znajduje się kilka rozwiązań, które za pomocą gotowych kreatorów proponują tworzenie stron internetowych na urządzenia mobilne. W tym rozdziale omówiono najpopularniejsze z nich.

2.1. Designer 2.0 Sencha

Designer 2.0 jest najnowszą wersją kreatora tworzącego strony mobilne oparte na Java Script. Umożliwia tworzenie aplikacji na urządzenia desktopowe i mobilne w oparciu o MVC, jak i pozwala na zbudowanie samego interfejsu użytkownika. Jest to narzędzie typu "przyciągnij i upuść", które gwarantuje szybkie zarządzanie elementami interfejsu aplikacji. Export danych oraz generowany kod są zorientowane obiektowo. Za pomocą tej aplikacji można:

- szybko i sprawnie budować formularze
- zmienić układ komponentów i wymienić rodzaje kontrolek za pomocą kliknięcia
- łatwo zmienić kontrolki oraz model danych w aplikacji
- generować dobrej jakości kod w oparciu o Sench Touch lub Extjs

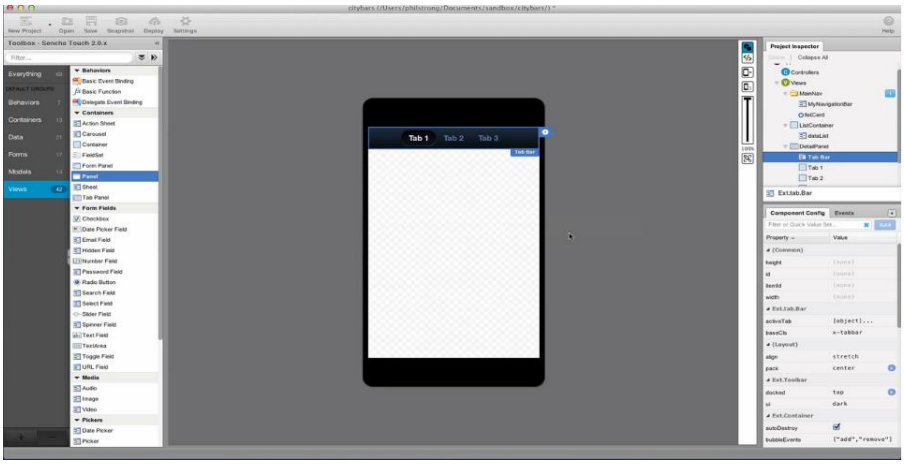
Innym ważnym elementem jest edytor, który umożliwia dodawanie własnego kodu. Zapewnia przełączanie się pomiędzy widokiem projektu, a kodem źródłowym i gwarantuje określenie typu generowanego kodu :

- Extjs
- Senach Touch

Dodatkowymi funkcjonalnościami oferowanymi przez Designera są:

- inspektor kodu - zapewnia graficzną prezentację wszystkich części aplikacji
- możliwość dodawania obsługi zdarzeń w kreatorze zamiast w plikach zewnętrznych
- generowanie przejrzystej struktury katalogów w tworzonej aplikacji
- export plików

Na rysunku 1 zaprezentowano widok główny kreatora Designer 2.0

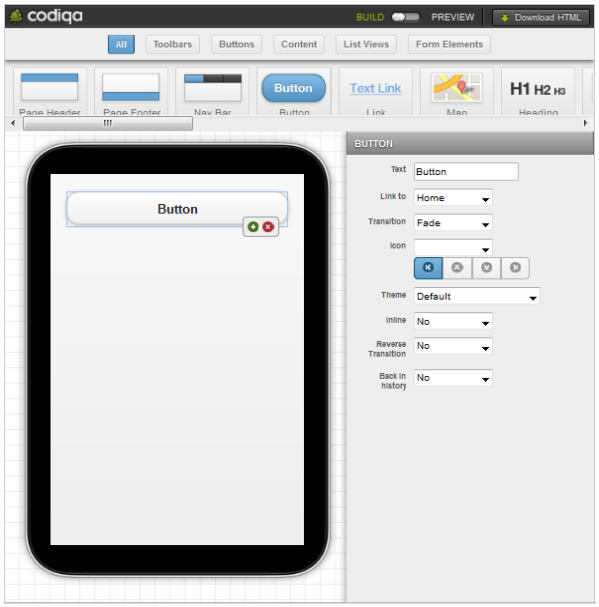


Rysunek 1. Widok główny kreatora Designer 2.0

źródło: http://www.sencha.com/blog/sencha-designer-2-beta-announcement

2.2. Codiqa

Codiqa jest narzędziem bardzo chętnie wybieranym przez developerów, designerów oraz agencje interaktywne tworzące strony na urządzenia mobilne. W 100% procentach bazuje na komponentach jQuery Mobile i korzysta z jego funkcjonalności. Nie korzysta z bibliotek zewnętrznych. Pisanie kodu od podstaw jest zajęciem niezwykle żmudnym i czasochłonnym, a przy wykorzystaniu Codigi jest o wiele prostsze. Nowatorska technika "przeciągnij i upuść" znacząco skraca czas budowy aplikacji, a moduł podglądu pozwala na testowanie i ocenę widoku strony w każdym momencie jej tworzenia. Na rysunku 2 zaprezentowano zdjęcie omawianego kreatora.



Rysunek 2. Widok Codiga- kreatora stron mobilnych

źródło: http://www.codiqa.com/

3. Przegląd dostępnych technologii mobilnych

Poniższe podrozdziały omawiają szczegółowo dostępne języki oraz technologie, które zostały wykorzystane pośrednio lub bezpośrednio w pracy.

3.1. WAP (Wirelless Aplication Protocol)

Wirelless Aplication Protocol (WAP) jest protokołem umożliwiającym użytkownikowi dostęp do danych z wykorzystaniem urządzeń bezprzewodowych takich jak smartfony, tablety, pagery i inne. Jest obsługiwany przez wszystkie systemy operacyjne a wśród nich najpopularniejsze: PALM OS, Windows, Java OS. Jednocześnie obsługuje większość sieci bezprzewodowych miedzy innymi :

- GSM (*Global System for Mobile Communications*)
- CDMA (*Code Division Multiple Access)*

Wszystkie urządzenia Wapowe mające dostęp do Internetu nazywane są mikro przeglądarkami o małych wielkościach pliku, które mają różną pojemność pamięci zależną od urządzenia przenośnego, a także ograniczone pasmo sieci bezprzewodowej. Protokół obsługuje języki takie jak HTML, XML, WML, ale szczególnie dedykowany jest małym wyświetlaczom bez klawiatury. Obsługuje także WMLScript, podobny do JavaScript, ale z zachowaniem tylko niezbędnych funkcji. Podsumowując: WAP umożliwia dostęp do stron WWW w takiej jakości, na którą pozwalają dane techniczne urządzenia, a także zależy od łącza danych (które może być realizowane np. przez GPRS). Omawiana technologia została wyparta przez nowsze, warto jednak o niej wspomnieć, gdyż była jedną z popularniejszych.

3.2. Mobilne języki znaczników

Przeglądarki internetowe dostępne na urządzenia mobilne umożliwiają przeglądanie treści w wielu językach mobilnych znaczników, jednak nie wszystkie z nich są w pełni implementowane Najczęściej używanymi językami mobilnych znaczników są:

- Wireless Markup Language (WML)
- XHTML Mobile Profile (XHTML-MP)
- HTML5

3.2.1. Wireless Markup Language (WML)

Jest to jeden ze starszych, prostszych języków znaczników mobilnych. WML został opracowany przez organizację Wireless Application Protocol Forum w roku 1998 i został opisany w pierwszej specyfikacji protokołu WAP [10]. Stąd często jest określany mianem WAP1. Dokładnie rzecz ujmując, język ten jest odmianą XML (Extensive Markup Language), którego uzupełniono o metaforę kart. Karta w tym znaczeniu to ekran interfejsu użytkownika (przy czym pojedynczy dokument znaczników może obejmować wiele kart). Przykładową stronę WML prezentuje listing 1.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card>
  <br/>
  <p><b><i>Marcin Czerwiak</i></b></p>
  <p><strong>Introduction to WML.</strong></p>
  <do label='Click' type='accept'>
  <go href='http://mobile-site.pl'/></do>
</card>
</wml>
```

Listing 1. Przykładowa strona w języku WML

WML został zaprojektowany do wyświetlania tekstu na monochromatycznych ekranach urządzeń przenośnych o ograniczonej pamięci i mocy procesora, dlatego w niektórych sieciach mobilnych jest on kompilowany na kod binarny, dzięki czemu jest szybciej przesyłany między serwerem, a urządzeniem. W takim przypadku przeglądarka w urządzeniu mobilnym dekompiluje i wizualizuje otrzymany kod. Język ten występuje w dwóch wersjach 1.1 oraz 1.3, przy czym dopiero wersja 1.3 wprowadza obsługę kolorowych obrazków.

WML został uznany za przestarzały na tyle, że jego obsługa została wycofana z przeglądarki telefonu iPhone firmy Apple. Z drugiej strony jednak jest szeroko obsługiwany przez urządzenia mobilne i właśnie z tego względu nadal znajduje zastosowanie przy tworzeniu prostych aplikacji webowych, czy stron www adresowanych do użytkowników starszych telefonów. Można powiedzieć, że został on wyparty przez inne języki mobilnych znaczników, które miały większe możliwości (jak XHTML) i w efekcie okazały się popularniejsze.

3.2.2. XHTML Mobile Profile

Język XHTML Mobile Profile (XHTML-MP) został opracowany przez organizację Open Mobile Alliance. Bardzo szybko uznano go za standardowy język znaczników mobilnych. W nazwie produktu zawarto sposób jego działania - jest podzbiorem języka XHTML

dostosowanym do możliwości urządzeń przenośnych. Od 2009 roku włączono go do specyfikacji protokołu WAP2.

Język ten postuluje rozdzielenie warstwy prezentacji danych od struktury dokumentu, poprzez dołączenie mobilnych kaskadowych arkuszy styli CSS (Cascading Style Sheets) w standardach:

- Wireless CSS
- CSS Mobile Profile
- CSS2,

które szerzej będą opisane w sekcji 3.3. Na listingu 2 przedstawiono przykład zastosowania języka XHTML-WP.

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"
"http://www.wapforum.org/DTD/xhtml-mobile10.dtd" >
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Test</title>
</head>
<body>
<p align="center">
      <b>Location</b>
</p>
<p align="center">
      Warsaw/>Seattle<br/>09123
</p>
<p align="center">
      <a href="mailto:m.czerwiak@gmail.com">Email Us</a>
</p>
</body>
</html>
```

Listing 2. Przykładowa strona w XHTML-WP

Język XHTML-MP ma następujące wersje:

- XHTML-MP 1.0
- XHTML-MP 1.1
- XHTML-MP 1.2
- XHTML-MP 1.3

W wersji 1.1 została dodana obsługa znacznika <script> i mobilnego JavaScriptu, a do wersji 1.2 kolejne znaczniki formularzy i tryby wprowadzania tekstu. Wersja 1.3 wykorzystuje XHTML Basic 1.1, a zdarzenia w tej wersji specyfikacji zostały zaktualizowane do DOM Level 3, tj. nie są one zależne od platformy i języka naturalnego. Obowiązkowo, na początku każdego dokumentu napisanego w języku XHTML-MP, powinna się znaleźć deklaracja DOCTYPE (patrz listing 3).

```
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.2//EN "
"http://www.wapforum.org/DTD/xhtml-mobile10.dtd">
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.1//EN"
"http://www.openmobilealliance.org/tech/DTD/xhtml-mobile11.dtd">
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.2//EN"
"http://www.openmobilealliance.org/tech/DTD/xhtml-mobile12.dtd">
```

Listing 3. Przykładowe deklaracja DOCTYPE

Język XHTML-WP zostaje powoli wypierany przez HTML5, który jest bardziej elastyczny, dzięki czemu oferuje programistom więcej możliwości i przynosi nowoczesne rozwiązania.

3.2.3. HTML5

Jedną z największych zalet języka HTML - jest jego dobra znajomość i popularność wśród projektantów interfejsów oraz programistów. HTML5 jako nowoczesna technologia, wywodząca się z niego, daje podstawę nowej generacji dla aplikacji webowych. HTML5 w odniesieniu do zwykłego HTMLa został rozszerzony o nowe znaczniki, przeznaczone bezpośrednio do opisywania treści, które pozwalają łatwiej zdefiniować strukturę strony [11]. Wspomnianymi znacznikami są: „<header>”, „<footer>”,” <nav>”, „<section>”,” <article>”. Dzięki nim dokument HTML5 jest bardziej czytelny dla programisty i dostępny dla aplikacji odczytujących zawartość ekranu.

W listingu 4 zamieszczono przykład prostej strony wykorzystującej nowe znaczniki strukturalne :

```
<!doctype html>
<html>
 <head>
    <title>Tytuł strony</title>
 </head>
 <body>
    <header> <h1>Tytuł</h1> </header>
    <nav>  <!-- Nawigacja -->
    </nav>
    <section id="intro"> <!-- Wstęp --> </section>
    <section> <!--Główna treść -->  </section>
    <aside> <!--Panel boczny --> </aside>
    <footer> <!-- Stopka --> </footer>
 </body>
</html>
```

Listing 4. Przykładowa strona zbudowana w oparciu HTML5.

Do HTML5 dodano również obsługę multimediów, która stała się bardziej niezależna od dodatkowych wtyczek: Flash i Silverlight. Osadzanie multimediów w technologii HTML5 będzie omówione w sekcji 3.9 .

Za wadę tego języka może być uznane to, że nie wszystkie przeglądarki internetowe wspierają go w pełni. Sytuacja na rynku najpopularniejszych przeglądarek zmienia się jednak dynamicznie, co skutkuje tym, że coraz więcej elementów specyfikacji języka HTML5 jest implementowanych jak pokazano w tabeli 1.

Aby szybko sprawdzić stopień implementacji HTML5 w przeglądarkach mobilnych, najlepiej jest zajrzeć na stronę http://html5test.com. Odwiedzając tę stronę z urządzenia mobilnego, można otrzymać pełny raport o konkretnej przeglądarce z danego urządzenia. W informacjach podany będzie stopień implementacji HTML5 i CSS3. Jest to niezbędne do prawidłowej optymalizacji funkcjonalności strony mobilnej projektowanej na daną przeglądarkę.

Tabela 1. Kompatybilność HTML5 w przeglądarkach mobilnych.

Przeglądarka	iOS	Android Browser	BlackBerry Browser	Nokia Browser	Opera Mobile	Firefox mobile	Internet Explorer
Platforma	iPhone, iPad	Telefony	Telefony	MeeGo - N9	Android & Symbian	Android	Windows Phone 7.5
Obsługa Sesij	✔	✔	✔	✔	✔	✔	✔
Web SQL	✔	✔	✔	✔			
Lokalizacja	✔	✔	✔	✔	✔	✔	✔
Multimedia	✔	✔	✔	✔	✔	✔	✔
Gniazda sieciowe	✔		✔		✔	✔	
Canvas API	✔	✔	✔	✔	✔	✔	✔
SVG - skalowanie grafiki	✔	✔	✔	✔	✔	✔	✔

Czujniki ruchu np. Żyroskop	✔	✔		✔			
HTML5 nowe kontrolki np. data	✔		✔		✔	✔	
Zdarzenia dotykowe	✔	✔	✔		✔	✔	
FullScreen API							

Źródło: http://mobilehtml5.org

3.3. Mobilne arkusze styli

Kaskadowe arkusze styli (CSS - Cascading Style Sheets) na urządzenia mobilne mają za zadanie ułatwić developerom formatowanie stron WWW. Ze względu na różnorodność dostępnych na rynku urządzeń mobilnych, większość przeglądarek obsługuje po kilka standardów arkuszy, choć zdarza się, że starsze przeglądarki nie wspierają żadnej z nich. W takim przypadku najlepiej opracować stronę w czystym WMLu.

Standard Wireless CSS został opracowany przez konsorcjum Open Mobile Alliance. Jego ostatnia wersja oznaczona numerem 1.2 została zatwierdzona w sierpniu 2008 r. CSS Mobile Profile został zaakceptowany przez organizację W3C (World Wide Web Consortium) i na jej stronach została umieszczona specyfikacja (z grudnia 2008 r.) w wersji 2.0. Oba te standardy posiadają pewne właściwości z CSS2, które są obsługiwane opcjonalnie lub w sposób ograniczony, zatem niezbędne staje się porównanie pomiędzy poszczególnymi standardami. Przykładowo, w standardzie Wireless CSS obsługa właściwości „inherit" jest opcjonalna, podczas gdy w CSS MP jest ona w pełni zaimplementowana.

CSS ma wiele aktualizacji. Standard CSS2 w zastosowaniach mobilnych można podzielić na dwa podzbiory:

- Wireless CSS
- CSS Mobile Profile,

przy czym podgrupa Wireless CSS jest nieco węższym podzbiorem CSS2, przeznaczonym dla urządzeń mobilnych o ograniczonych zasobach: pamięci i szybkości procesora. Obie powyższe podgrupy zostały stworzone z myślą o stosowaniu ich w dokumentach napisanych w języku XHTML-MP, ale obecnie powoli się od nich odchodzi. Na popularności zyskuje standard CSS3 obsługiwany przez nowoczesne smartfony.

Najnowsze CSS wnikają w najmniejsze szczegóły strony i potrafią dokonać mikro zmian. Najważniejszymi nowościami kolejnej odsłony CSS o numerze 3 są:

- podział na moduły
- nowe selektory, pseudo klasy i pseudo elementy
- nowe style i właściwości

Stopień implementacji CSS3 w wybranych przeglądarkach mobilnych zebrano w tabeli 2.

Tabela 2.Implementacja CSS3 w przeglądarka mobilnych.

Przeglądarka	iOS	Android Browser	BlackBerry Browser	Nokia Browser	Opera Mobile	Firefox mobile	Internet Explorer
Platforma	iPhone, iPad	Telefony	Telefony	MeeGo - N9	Android & Symbian	Android	Windows Phone 7.5
CSS 3 Podstawowy	✔	✔	✔	✔	✔	✔	✔
CSS 3 2D	✔	✔	✔	✔	✔	✔	✔
CSS 3 3D	✔	✔					
CSS 3 Przejścia	✔	✔	✔	✔	✔	✔	
CSS 3 Animacje	✔	✔	✔	✔		✔	

Źródło: http://mobilehtml5.org

3.4. Przeglądarki w urządzeniach mobilnych

Przeglądarki mobilne są, przeglądarkami internetowymi, maksymalnie zoptymalizowanymi pod kątem wyświetlania stron internetowych na małych ekranach urządzeń przenośnych. Ich charakterystycznymi cechami są:

- prostota obsługi,
- brak zaawansowanych opcji,
- fizycznie nieduży rozmiar.

Dynamiczny rozwoju Internetu mobilnego swoim zasięgiem nie ominął rynku przeglądarek internetowych. Większość producentów przeglądarek desktopowych przygotowało ich wersje przeznaczone na urządzenia mobilne. I tak na rynku dostępne są następujące przeglądarki mobilne:

- Android browser - obecnie popularna mobilna przeglądarka internetowa dostarczana razem z mobilnym systemem operacyjnym Android
- Safari - oparta na silniku WebKit - stworzona przez firmę Apple dla systemu operacyjnego OS X, od roku 2007 została udostępniona także dla systemów operacyjnych firmy Microsoft
- Opera Mini – mobilna przeglądarka internetowa dla telefonów komórkowych wyposażonych w Java Platform Micro Edition, stworzona przez norweską firmę Opera Software ASA.
- Safari Mobile – mobilna wersja przeglądarki Safari dla urządzenia iPhone i iPod
- WebOS – adresowana dla urządzenia Palm Pre
- przeglądarka dla urządzeń BlackBerry
- przeglądarka Nokia Web dla urządzeń Nokia z systemem Series 60
- Internet Explorer Mobile dla systemu Windows Mobile

Różne przeglądarki nie zawsze w ten sam sposób wyświetlają strony mobilne ze względu na różnice w implementacji niektórych znaczników i atrybutów CSS, co czyni testowanie jeszcze bardziej wymagającym.

3.5. Java Script w aplikacjach mobilnych

Przebudowa tradycyjnych stron internetowych, na ich odpowiedniki mobilne, stanowi poważny problem. Ma to szczególnie duże znaczenie dla urządzeń o ograniczonych ilościach pamięci RAM i wolnych procesorach - które istotnie spowalniają "serfowanie po Internecie". Z pomocą przychodzi nam język programowania jakim jest JavaScript ,który do niedawna kojarzono tylko z WWW. Aktualnie stanowi obiecującą alternatywę w programowaniu na urządzenia mobilne. Mnogość systemów operacyjnych wykorzystywanych w platformach przenośnych (iOS, Android, Windows Mobile, BlackBerry, Symbian, Palm OS) stała się olbrzymim utrudnieniem dla programisty. Współczesne frameworki JavaScript mogą przeistoczyć ten chaos w koherentny i przyjazny dla dewelopera świat. Język ten tworzony jest za pomocą skryptów, których dużą zaletą jest możliwość zastosowania interaktywności (poprzez reagowanie na zdarzenia) lub sprawdzania poprawności wypełniania formularzy np. podczas przechodzenia użytkownika z jednego pola do następnego.

Zaletą JavaScript jest także jego szybkość, ponieważ aplety w nim zawarte są interpretowane i uruchamiane przez przeglądarkę bezpośrednio z kodu strony internetowej.

Jednak warunkiem takiego działania jest konieczność obsługi JavaScript przez przeglądarkę na której jest wyświetlana dana strona WWW. Dokładniej mówiąc: musi posiadać ona wszystkie niezbędne biblioteki potrzebne do uruchomienia skryptu. Należy zaznaczyć także, że skrypty napisane w JavaScript mogą być umieszczane bezpośrednio na tworzonych stronach HTML, co sprawia, że zostają one skompilowane przez przeglądarkę dopiero po ściągnięciu całej strony z serwera. Istotną rzeczą jest, aby w trakcie budowy strony internetowej, wzbogacając jej funkcjonalność zwrócić uwagę na to, aby żaden jej element nie stał się niedostępny po wyłączeniu obsługi JavaScriptu w przeglądarce. Może się tak wydarzyć, ponieważ skrypty napisane w tym języku mają znacznie ograniczony dostęp do komputera użytkownika.

Język JavaScript jest alternatywą w programowaniu urządzeń mobilnych. To właśnie z jego pomocą programista w trakcie tworzenia projektu buduje aplikację, poszerzając i dostosowując poszczególne jej składniki do wymagań realizowanego projektu. Tym sposobem powstaje gotowa aplikacja, która może być używana wielokrotnie.

3.6. Frameworki mobilne

Wydaje się, że można stwierdzić , iż framework jest narzędziem służącym do tworzenia aplikacji, gdyż zawiera najczęściej używane narzędzia, biblioteki, oraz wiele innych elementów, po to aby programista mógł skupić się na tworzeniu docelowego programu. W trakcie tworzenia aplikacji webowej dostosowuje on poszczególne jej składniki (komponenty) do wymagań realizowanego projektu, tworząc w ten sposób gotową aplikację.

Warty podkreślenia pozostaje fakt, iż frameworki wymuszają na programiście pisanie w ściśle określony sposób, co przekłada się na jakość kodu źródłowego. Kod ten jest dużo bardziej zrozumiały dla innego programisty, który zna budowę frameworku.

Mogło by się wydawać, że stosowanie frameworków przynosi same korzyści, ale jednak tak nie jest. Jak większość rozwiązań idących drogą na skróty, wiąże się z wadami, które wraz z rosnącą wielkością aplikacji i liczbą użytkowników mogą przysparzać problemów. Mimo to należy zaznaczyć, że wszystkie opisane poniżej wady nie są w stanie przysłonić zalet płynących ze stosowania frameworków.

Zalety stosowania frameworków:

- poprawa efektywności — tworzenie aplikacji z wykorzystaniem frameworków pozwala na znaczne zmniejszenie ilości kodu źródłowego,
- poprawa jakości kodu — ze względu na dopracowaną architekturę wewnętrzna i logikę, ponieważ frameworki są projektowane z myślą o elastyczności,

- niezawodność — uzyskiwana dzięki dobremu przetestowaniu przez rzeszę programistów w różnych aplikacjach, co pozwala na usunięcie znaczących błędów powstałych w trakcie tworzenia danego frameworku.

Wady stosowania frameworków:

- potrzeba większej ilości zasobów – frameworki są niedużym, ale jednak dodatkowym obciążeniem dla serwera,
- możliwość powstania niezabezpieczonych przestrzeni – jeśli we frameworku pojawi się błąd, to wiązać się to będzie z zagrożeniem dla wszystkich aplikacji, w których jest on wykorzystany,
- uzależnienie programisty od gotowych rozwiązań – programując aplikacje za pomocą gotowych rozwiązań jesteśmy zależni od ich producenta, programista może rozbudować framework, ale w sposób taki, który nie zmieni domyślnego kodu.

Ze względu na popularność, użyteczność oraz łatwość z jaką mogą być zastosowane należy wymienić przede wszystkim dwa mobilne frameworki:

1. jQuery Mobile - wbrew temu co można sądzić nie jest mobilną wersją standardowego frameworku o tej samej nazwie, lecz rozszerzeniem oryginalnego jQuery o dodatki, które są niezwykle pomocne i niezbędne, do tworzenia aplikacji webowych z poziomu urządzeń mobilnych. Za pomocą tego frameworku można budować mobilne strony w języku HTML z tą sama biblioteką jQuery, która jest używana do tworzenia standardowych stron internetowych.

Twórcy jQuery Mobile jako efekt końcowy swojej pracy uzyskali framework, który działa w ten sam sposób na wielu platformach (różnego rodzaju urządzeniach), to znaczy w sposób niezależny od przeglądarki internetowej w jakiej zostanie otworzona strona. Bez wątpienia ten stan rzeczy powoduje, że zbędne jest pisanie osobnych aplikacji na różne urządzenia.

2. SenchaTouch to framework pozwalający na szybkie i łatwe tworzenie aplikacji mobilnych bazujących na HTML5, które są przeznaczone do współpracy takimi systemami jak Android, iOS, czy Blackberry. Programując z pomocą Sencha Touch można korzystać ze specjalnych tematów stworzonych dla tych urządzeń.

Mówiąc w inny sposób: jest to uniwersalny, wieloplatformowy framework programistyczny dla urządzeń z ekranami dotykowymi. Zawiera on gotowe zestawy ikon, przycisków, karuzel, suwaków oraz bibliotek konfigurowalnych animacji, przejść pomiędzy widokami, które niezwykle prosto poddają się stylizacji programisty. Dzięki zaś komponentowi Google Maps i interfejsowi geolokalizacji łatwo wzbogacić aplikacje o dane geograficzne. Sencha Touch posiada także obsługę zdarzeń dotykowych, takich jak na przykład popularny multitouch. Framework ten wykorzystuje na urządzeniach mobilnych silnik przeglądarki WebKit. To wszystko

sprawia, że za jego pomocą można programować niezwykle ciekawe i interesujące wizualnie interfejsy.

Można powiedzieć, iż Sencha Touch jest rozwiązaniem dla przedsiębiorstw, którym niezbędne są przenośne aplikacje w prowadzeniu działalności gospodarczej, a ponieważ stworzono w nim rygorystyczne standardy kodowania oraz klasycznego modelu dziedziczenia, to może być łatwo wykorzystywana do pracy przez małe i duże zespoły programistyczne. Licencja tego frameworku jest zgodna z GNU GPL v3 i pomimo, że ma ona wiele warunków do spełnienia to istotna cechą jest to, że należy podać kod źródłowy aplikacji dla jej użytkowników. Dzięki temu aplikacje takie można dowolnie modyfikować wedle własnych potrzeb.

Sencha Touch wyposażony jest w niesamowicie potężny pakiet danych, do których programiści mogą łatwo uzyskać dostęp, na przykład na temat konkretnych komponentów wizualnych lub szablonów. Następnie dane te można odczytać w trybie offline lub z lokalnego magazynu pamięci. Proces pozyskania tego typu informacji jest niezwykle wartościowy w trakcie tworzenia aplikacji.

W celu przybliżenia technologii oraz obsługiwanych platform, autor zamieścił tabelę 3, porównującą frameworki jQuery Mobile oraz Sencha Touch2.

Tabela 3. Zestawienie obsługiwanych platform przez frameworki mobilne.

Biblioteka	jQuery Mobile	Sencha Touch 2
Technologie	HTML5, jQuery	HTML5, CSS
Obsługiwane Platformy	iOS, Android, BlackBerry, Palm WebOS, Nokia/Symbian, Windows Phone 7, MeeGo, Opera Mobile, Firefox Mobile, Kindle, Nook	iOS, Android, BlackBerry, Kindle Fire

3.7. Rozpoznawanie urządzeń mobilnych

Ze względu na to, że rynek urządzeń mobilnych, a także przystosowanych do nich przeglądarek zdecydowanie się zwiększa, istnieje potrzeba tworzenia internetowych stron mobilnych. Dzieje się tak dlatego, że dla wielu z tych urządzeń standardowe strony są zbyt duże do przetworzenia. Idealnym rozwiązaniem jest tworzenie specjalnie zaprojektowanych stron mobilnych w taki sposób, aby użytkownicy tego typu urządzeń poprzez wcześniejsze ich rozpoznanie byli automatycznie przekierowywani na strony internetowe w wersji mobilnej.

Rozpoznawanie urządzeń mobilnych w sieci to nic innego, jak dostosowywanie treści wyświetlanych stron internetowych do możliwości fizycznych - sprzętowych urządzenia, które je

wyświetla. W celu takiej optymalizacji potrzebne są informacje o urządzeniach oraz przeglądarkach internetowych na nich zainstalowanych, z których w danej chwili korzysta użytkownik. Informacje takie można najlepiej wykorzystać, gdy zebrane są w postaci bazy danych o urządzeniach i przeglądarkach. Będąc w posiadaniu takiej wiedzy bez najmniejszego problemu można dopasować wyświetlaną treść strony do potrzeb urządzenia. Działanie takie uzyskuje się poprzez dobór właściwego języka znaczników, języka skryptowego, czy też stylów, w trakcie budowy gotowej strony internetowej. Zbiór danych można otrzymać dzięki wykorzystaniu informacji zawartych w żądaniach identyfikacyjnych wysyłanych przez strony internetowe, które służą do identyfikacji przeglądarek i urządzeń mobilnych.

Analiza danych zawartych w nagłówkach protokołu HTTP umożliwia zgromadzenie informacji niezbędnych do dostosowania wyświetlanej treści na stornie internetowej w trakcie przeglądania jej na urządzeniu mobilnym. Najistotniejszym nagłówkiem tego typu jest nagłówek User Agent.

User Agent sam w sobie jest aplikacją napisaną w języku HTTP (po stronie klienta), która w trakcie komunikacji z danym serwerem WWW wysyła informacje zawierające dane dotyczące urządzenia i przeglądarki internetowej. Aplikacja ta bardzo często służy także do tworzenia statystyk dotyczących częstotliwości odwiedzin danej strony przez użytkowników korzystających z różnych przeglądarek i urządzeń, co pozwala optymalizować wyświetlaną treść pod kątem najczęściej odwiedzających klientów.

W listingu 5 przedstawiono przykład nagłówka User Agent wysyłanego przez urządzenie iPhone marki Apple, posiadające system operacyjny w wersji 3.1.1 w języku angielskim, z zainstalowaną przeglądarką Safari /525.20.

```
Mozilla/5.0 (iPhone; CPU iPhone OS 2_2_1 like Mac OS X; en-us)
AppleWebKit/525.18.1 (KHTML, like Geco) Version/3.1.1 Mobile/5H11 Safari/525.20
```

Listing. 5. Przykładowy User Agent generowany przez urządzenie iPhone

Przy pomocy powyższych informacji można wyszukać więcej danych na temat telefonu iPhone, korzystając z bazy danych zawierającej szczegółową specyfikację tego urządzenia. Niezbędną jest wiedza na temat rozdzielczości ekranu, czy też możliwości jego rotacji, gdyż są to aspekty mające znaczenie w trakcie wyświetlania treści na urządzeniu.

Należy dodać, że nagłówek User Agent jest przesyłany obecnie przez niemal wszystkie przeglądarki internetowe. Jednak nie jest on obowiązkowy. Na szczęście ingerencja użytkownika w jego strukturę jest dość prosta – co powoduje możliwość podszywania się za boty w trakcie przeglądania stron internetowych, czy też forum. Istotnym aspektem jest także fakt: pomimo iż User Agent dostarcza informacje na temat urządzenia oraz przeglądarki z której ono korzysta,

to i tak nie pozwala na określenie konkretnych cech takiej pary - urządzenie i przeglądarka. W celu dokładniejszej analizy niezbędna okazuje się baza danych, w której można znaleźć potrzebne informacje. Współpraca z bazami danych o urządzeniach jest możliwa dzięki specjalnym interfejsom API (Interfejs Programowania Aplikacji).

Jedną z największych tego typu baz jest DeviceAtlas.com, firmowana przez dotMobi, która zawiera dokładne charakterystyki wielu modeli urządzeń mobilnych. W bazie tej można znaleźć oprócz standardowych parametrów takich jak: rozmiar, rozdzielczość ekranu i system operacyjny, także informacje o obsługiwanych kodekach dźwięku i obrazu, maszynie wirtualnej Java czy dostępnych protokołach sieciowych (GPRS, EDGE, UMTS, HSDPA itp.). Dość znaczącą wadą tej bazy jest jednak to, że można ją tylko przeglądać za darmo (online), natomiast pobranie jej jest możliwe tylko po uiszczeniu opłaty, która waha się pomiędzy 100$ a 300$.

Tworząc mobilne witryny nie ma bezwzględnego przymusu korzystania z komercyjnych baz danych. Rozwiązaniem w tym przypadku może być w pełni darmowa i otwarta baza WURFL [5]. Korzenie jej sięgają początku roku 2002. Od tego czasu ulega stałemu powiększaniu dzięki szerokiej społeczności programistów z nią związanych. Dzieje się tak za sprawą licencji open source, dzięki której użytkownicy indywidualni oraz przedsiębiorstwa mogą tworzyć zespoły programistyczne mające za zadanie dodawanie, weryfikację i ewentualne korygowanie danych zawartych w tej bazie. Należy zaznaczyć, że w trakcie tworzenia bazy Wurfl zastosowano technologię polegającą na hierarchii urządzeń w niej opisanych. Polega to na tym, iż urządzenia potomne (tego samego typu co pierwowzór) dziedziczą pewne cechy po urządzeniach macierzystych (np. starszy model), tworząc w ten sposób całe rodziny urządzeń. Dzięki temu poprawianie lub aktualizowanie danych może dotyczyć poszczególnych urządzeń, lub całych rodzin, co powoduje poprawę jakości danych i istotnie skraca czas ich dodawania.

Należy zaznaczyć, że pomimo iż baza danych Wurfl dzięki opisanym powyżej działaniom społeczności programistów zawiera dane (w postaci parametrów technicznych) o niemal wszystkich urządzeniach mobilnych, to i tak zdarzają się przypadki błędnej identyfikacji tych urządzeń. Z drugiej jednak strony zdarzenia tego typu to rzadkość, a w razie ich zaistnienia programiści nadzorujący projekt po otrzymaniu informacji o nieprawidłowościach bezzwłocznie je eliminują.

Sama baza fizycznie występuje pod postacią niedużego, około kilkunastomegabajtowego pliku, zapisanego w formacie xml lub jako archiwum zip. Pliki te można bezpłatnie pobrać ze strony http://wurfl.sourceforge.net, gdzie można znaleźć także przykłady jej użycia dla różnych języków programowania – np. Java, PHP, czy Perl. Natomiast w razie potrzeby podejrzenia jakiejś konkretnej wartości warto odwiedzić repozytorium znajdujące się pod adresem http://wurflpro.com/, gdzie można szukać i sprawdzać możliwości poszczególnych urządzeń.

Wspomniany wcześniej interfejs API podczas współpracy z Wurfl ma za zadanie odszukiwanie parametrów poszczególnych urządzeń. Jest to o tyle łatwe, że pola tekstowe

reprezentujące poszczególne wartości danych podzielone są na grupy, które jednocześnie tworzą kryterium wyszukiwania.

Korzystając z bazy Wurfl należy wiedzieć jakie parametry są w niej opisane, a właściwie które z tych parametrów są stosowane najczęściej, gdyż taka wiedza pozwoli na sprawne i szybkie wykorzystanie informacji w niej zawartych. Tabela 4 zawiera wybrane parametry tej bazy.

Tabela 4. Wybrane parametry urządzeń mobilnych zawarte w bazie Wurfl [5].

Nazwa parametru	**Opis**
brand_name	marka urządzenia np. LG, Apple, Nokia, Palm
model_name	nazwa modelu urządzeniach np. VX9100, iPhone, N96
is_wireless_device	informacja określająca czy urządzenie jest mobilne; posiada wartość false dla tradycyjnych przeglądarek i robotów sieciowych
device_claims_web_support	informacja określająca deklaracje obsługi standardów HTML/JavaScript/AJAX przez przeglądarkę
ajax_support_javascript	informacja określająca właściwą obsługę JavaScript
preferred_markup	preferowany przez przeglądarkę język znaczników
resolution_width	parametr określający rozdzielczość, w tym przypadku szerokość ekranu w pikselach
resolution_height	parametr określający rozdzielczość, w tym przypadku wysokość ekranu w pikselach

W dużym skrócie poszczególnym parametrom przypisane są funkcje:

- pierwsze trzy tzn.: brand_name, model_name i is_wireless_device są parametrami, które mają za zadanie opisać samo urządzenie i określić, czy jest ono mobilne,
- kolejne trzy parametry opisują podstawowe właściwości przeglądarki zainstalowanej na urządzeniu,
- ostatnie dwa parametry podobnie jak pierwsze opisują urządzenie, ale tutaj już pod kątem technicznym, gdyż rozdzielczość ekranu jest niezwykle ważną wartością pozwalającą na czytelne wyświetlanie treści znajdującej się na stronie.

Trzeba powiedzieć, że w trakcie tworzenia internetowych stron mobilnych nieoszacowanej pomocy dostarcza baza danych o urządzeniach i przeglądarkach. W trakcie wyboru jednej z nich

nie trzeba wcale kierować się sugestią mówiącą, że to co drogie jest lepsze, ponieważ informacje jakie dostarcza Wurfl są w pełni wystarczające. Niektórzy nawet twierdzą, że baza ta jest kopiowana przez twórców komercyjnych produktów tego typu. Świadczy to o wysokiej jakość i bieżącej aktualizacji tej bazy, mimo że tworzona jest przez społeczność programistyczną, a nie przez dobrze opłacane zespoły ludzi.

3.8. Projektowanie mobilnych stron internetowych

W trakcie projektowania mobilnej strony internetowej, jako główny cel powinno obrać się jej użyteczność, a przede wszystkim łatwość obsługi. Aspekty te powodują, że właściciele urządzeń mobilnych chętnie korzystają z tej formy serfowania po sieci. Liczy się tu wrażenie prostoty, gdyż strony zaprojektowane nieprawidłowo i nieergonomicznie same zniechęcają do ich przeglądania. Bardzo często zirytowany użytkownik wybiera wtedy tradycyjną formę łączenia się z siecią, gdyż jedną z podstawowych zalet Internetu mobilnego, jaką jest szybkość uzyskania informacji, zostaje zachwiana.

Wbrew pozorom zaprojektowanie użytecznej, mobilnej strony internetowej nie jest prostym zadaniem. Przyczyną tego jest fakt, że zarówno same witryny (np. wyszukiwarki lub serwisy informacyjne), jak i urządzenia, potrafią być bardzo różnorodne, gdyż posiadają pełen wachlarz charakterystycznych dla nich cech. Każda witryna w swojej "pełnej wersji" może posiadać szereg funkcji i odsyłaczy, a także elementów komercyjnych, które dla internauty mobilnego są nie tylko mało istotne, ale przede wszystkim mogą spowolnić pracę urządzenia. Wiąże się z tym także czas dostępu do samej informacji, którą użytkownik chciałby otrzymać w taki sposób, aby nie musiał poświęcać zbyt wiele swojej uwagi i energii w trakcie odnajdywania interesującej go treści w gąszczu funkcji [5]. Warto pamiętać, że użytkownik niejednokrotnie przebywa na ulicy, w tramwaju lub autobusie, gdzie zdolność koncentracji jest ograniczona na skutek dynamiki otaczającego świata.

Projektowana mobilna strona internetowa ma trafić do jak największej liczny użytkowników, posiadających różnorodne urządzenia – od telefonów GSM, przez smartfony po tablety włącznie. Tworząc projekt przeznaczony dla takich urządzeń należy zadbać o następujące aspekty:

- Prawidłowość wyświetlania treści nawet na "słabych technicznie" urządzeniach. Najlepszym rozwiązaniem jest stworzenie wielu wersji mobilnych, pod kątem różnych parametrów technicznych urządzeń – np. telefony tradycyjne z klawiszami, lub dotykowe smartfony.
- Przejrzystość projektowanego interfejsu – rezygnacja ze zbędnych odnośników i elementów wideo (np. materiały filmowe w serwisach informacyjnych), czy ograniczenie do minimum elementów graficznych (zdjęcia) po to, aby zapewnić maksimum treści i minimum obciążenia dla sprzętu.

- Brak zaawansowanych funkcji. Można np. zrezygnować z edytowania kompozycji panelu użytkownika (np. witryny wymiany plików multimedialnych), czy tworzenia nowych obszernych artykułów (np. Wikipedia). Żaden użytkownik nie będzie zainteresowany zaawansowanymi opcjami graficznymi, czy też tworzeniem tekstu za pomocą nieporęcznej klawiatury urządzenia przenośnego.
- Wyświetlana treść powinna być prezentowana w postaci kolumn.
- Zachowanie najbardziej istotnych dla potencjalnych użytkowników treści – typu panel logowania (portale), czy wyszukiwania (wyszukiwarki, encyklopedie).
- Brak poziomego paska przesunięcia ekranu – sam pasek pionowy pozwala na większą swobodę poruszania się, niż pionu i poziomu razem.
- Ograniczenie do minimum elementów komercyjnych na stronie, wyświetlanych w postaci reklam. To właśnie one powodują zamieszczanie największej ilości niechcianych treści, a co za tym idzie spowolnienie korzystania z danej strony.

Dostosowywanie treści na stronach mobilnych

Dostosowywanie treści na stronach mobilnych, to nic innego, jak zbiór działań mających na celu dopasowanie strony internetowej do specyficznych wymagań urządzeń dla których jest przeznaczona. Mówiąc inaczej, to określenie schematu kreowania konkretnej strony internetowej, która jest przekształcana w zależności od potrzeby – czyli właściwości technicznych urządzeń i przeglądarek na których ma być wyświetlana.

Polega to w szczególności na dopasowaniu odpowiedniego języka programowania (w tym przypadku języka znaczników), skryptów, czy też arkuszy styli. Należy zaznaczyć, że obecnie na rynku oferowanych jest wiele urządzeń, które znacząco różnią się od siebie. Posiadają rozmaite ekrany (o różnych wielkościach, rozdzielczościach, głębiach kolorów), swoiste systemy operacyjne, przeglądarki internetowe, czy też sposoby obsługi i wprowadzania danych, co powoduje utrudnienia w trakcie tworzenia strony [5].

W celu właściwej identyfikacji urządzeń, niezbędnym elementem jest zastosowanie opisanej w sekcji 3.7 bazy danych zawierającej informacje na temat urządzeń i przeglądarek mobilnych. Na podstawie takich danych, należy wyodrębnić grupy urządzeń, które będą spełniać określone kryteria dotyczące obsługi poszczególnych języków i skryptów. Grupy takie musi określić sam programista za pomocą wyodrębnienia konkretnych parametrów technicznych, które będą stanowić kryterium przynależności. Po wyłonieniu grup urządzeń, należy zaprojektować stronę w taki sposób, aby była prawidłowo wyświetlana na starszych technicznie urządzeniach, a na nowszych (bardziej zaawansowanych) smartfonach wykorzystywała optymalnie ich możliwości techniczne – nie powodując tym samym zbytecznego obciążenia. Zastosowanie rozwiązań tego typu powoduje, że zaprojektowana witryna będzie skutecznie wykorzystywała

zalety różnych platform mobilnych, ale jednocześnie wymusza znaczny nakład pracy w trakcie jej tworzenia.

Kolejną ważną rzeczą w trakcie projektowania mobilnej witryny internetowej (następującą po utworzeniu grup urządzeń) jest dostosowanie projektu i jego funkcji do możliwości technicznych urządzeń znajdujących się w tych grupach. Do sprawnego wykonania takiego kroku należy przeanalizować wszystkie elementy projektu pod kątem poszczególnych grup platform mobilnych, celem ich optymalizacji. Ma to za zadanie zastosowanie technik, które mogą w pełni wykorzystywać możliwości zarówno bardziej zaawansowanych smartfonów, jak i zwykłych telefonów. Uściślając chodzi tu o modyfikacje i usuwanie elementów projektowanej witryny, jak też dostosowywanie arkuszy CSS, czy dynamicznych elementów pod kątem właściwości urządzeń w danej grupie. Za przykład mogą posłużyć tu telefony, które posiadają obsługę Java Script, gdzie np. można zastosować - w odpowiedzi na kliknięcie ekranu - rozwijane menu. Efekt taki tworzy pozytywne doznania użytkownikom urządzeń z taką obsługą, lecz jednocześnie wymusza zastosowanie innego rozwiązania w telefonach które nie obsługują skryptów, ponieważ taka strona nie wyświetlałaby się na nich prawidłowo [12].

Dostosowanie treści witryny mobilnej może technicznie polegać na zmianie skali wyświetlanej strony na urządzeniu (w zależności od wielkości ekranu), zmiany długości akapitów tekstu, czy też zastosowaniu innego arkusza stylów dla urządzeń obsługujących poziomą orientację ekranu.

3.9. Multimedia: audio i video

Osadzenie plików audio i wideo na stronie www przysparzało programistom wielu problemów. Na początku osadzali oni na swoich stronach pliki MIDI w sposób zaprezentowany w listingu 6:

```
<embed src="test.mp3" autostart="true"></embed>
```

Listing 6. Osadzanie plików MIDI na stronach WWW.

Jednak powyższe rozwiązanie nie było zgodne ze standardem "World Wide Web Consortium", co spowodowało że zaczęto zamiast niego używać znacznika *object*, który następnie został uznany za standard. Czasami można spotkać się z sytuacją, że znacznik *embed* jest zagnieżdżony w znaczniku *object* w sposób jak na listingu 7.

```
<object>
 <param name="src" value="test.mp3">
 <param name="autoplay" value="false"
 <embed src="test.mp3" autostart="true"></embed>
```

```
</object>
```

Listing 7. Osadzanie plików multimedialnych przez znacznik „Object”.

Takie rozwiązanie nie było idealne, gdyż nie każda przeglądarka potrafi obsługiwać dane osadzone w podany sposób i nie każdy serwer był tak konfigurowany, aby skutecznie udostępniać takie pliki. Sytuacja jeszcze bardziej się skomplikowała, kiedy na rynku pojawiły się telefony komórkowe, a wraz z nimi nowe przeglądarki oraz urządzenia do obsługi danych audio i wideo wśród nich: QuickTime, RealPlayer czy Windows Media. Poza tym każda większa firma proponuje własny sposób obsługi danych mulitimedialnych, co w efekcie prowadzi do mnożenia się sposobów osadzania plików audio lub wideo na stronach. Twórcy specyfikacji HTML5 wychodzą z założenia że przeglądarki powinny posiadać własne rozwiązanie wspierające odtwarzanie plików multimedialnych na stronach WWW [4].

3.9.1. Kontenery i kodeki

Poruszając temat udostępniania danych video w Internecie stosuje się pojęcie kontenerów i kodeków. W bardzo dużym uproszczeniu można założyć, że nagrany kamerą cyfrową lub telefonem film jest równoznaczny z pikiem AVI lub MPEG. Kontenery to pewien rodzaj kopert w których znajdują się strumienie audio, strumienie wideo oraz metadane np. napisy do filmów. Strumienie audio i wideo muszą być zakodowane za pomocą tzw. kodeków. Dane wideo i audio można kodować na setki różnych sposobów, lecz tak naprawdę w przypadku plików wideo udostępnianych przy użyciu HTML5 istotne znaczenie ma tylko klika z nich. Zakres obsługi kodeków audio w poszczególnych przeglądarkach prezentuje poniższa tabela 5:

Tabela 5. Wsparcie kodeków przez przeglądarki internetowe.

Typ kodeku	Przeglądarki
Advanced Audio Coding (AAC)	Safari 4, Chrome 3, IOS
MP3	IE9, Safari 4, Chrome 3, IOS
Vorbis (OGG)	Firefox 3, Chrome 4, Opera 10

Advanced Audio Coding (AAC) - to format audio wykorzystywany przez firmę Apple w sklepie iTunes Store. Został on zaprojektowany w taki sposób, aby zapewnić jak największą jakość dźwięku, przewyższającą MP3, przy jednoczesnym porównywalnym rozmiarze plików. Podobnie jak w H.264, tak i w AAC, dostępnych jest kilka różnych profili. Ponadto analogicznie jak H.264 nie jest to kodek darmowy, a za jego wykorzystanie należy uiszczać odpowiednie opłaty

licencyjne. Warto zaznaczyć, że wszystkie produkty firmy Apple potrafią odtwarzać pliki w formie AAC. Jest on obsługiwany również w programie Flash Player firmy Adobe oraz w odtwarzaczu VLC o otwartym dostępie do kodu źródłowego.

MP3 - jest bardzo popularnym i powszechnie używanym formatem danych dźwiękowych, ale niestety z racji ochrony patentowej, nie jest obsługiwany przez przeglądarkę Firefox i Opera.

Vorbis (OGG) - Jest to format o otwartym dostępie do kodu źródłowego, całkowicie zwolniony z opłat licencyjnych i obsługiwany przez Firefox, Opera, Chrome. Ponadto format OGG jest używany wraz z kodekami wideo Theora i VP8. Pliki w formacie Vorbis charakteryzują się bardzo dobrą jakością dźwięku, lecz liczba obsługiwanych je odtwarzaczy muzycznych jest wciąż niewielka.

Kontenery i kodeki działają razem. Każdy kontener to plik metadanych, który identyfikują i przeplatają pliki audio lub wideo. Kontenery nie zawierają żadnej informacji na temat tego, jak zakodowane zostały zawarte w nich dane. Zasadniczo kontener stanowi opakowanie dla strumieni audio i wideo. W większości przypadków mogą one zawierać dowolną kombinację zakodowanych strumieni. Jeśli chodzi o udostępnianie wideo w Internecie używane są następujące kombinacje kodeków:

- Kontener OGG, z formatem obrazu Theora i formatem dźwięku Vorbis – obsługiwany w przeglądarkach Firefox, Chrome, Opera.
- Kontener MP4, z formatem obrazu H.264 i formatem dźwięku AAC - obsługiwany w przeglądarkach Safari i Chrome, a także w Adobe Flash Player oraz urządzeniach iPhone, iPad, iPod.
- Kontener WebM, z formatem obrazu VP8 i formatem dźwięku Vorbis - obsługiwany w przeglądarkach Firefox, Chrome i Opera oraz przez Adobe Flash Player.

3.9.2. Osadzanie plików audio i wideo

Jedną z najbardziej, od dawna oczekiwanych możliwości w HTML5 jest osadzanie plików audio i video, które umożliwia natywne odtwarzanie utworów muzycznych w przeglądarce. Określana jest znacznikiem audio.

Według specyfikacji może przyjmować pięć atrybutów:

- autoplay – określa czy odtwarzać automatycznie dźwięk, gdy zostanie on załadowany,
- controls – określa czy wyświetlić odtwarzacz w standardowym układzie,
- loop– określa czy powtarzać odtwarzanie,
- preload– wyznacza, czy załadować zasoby w trakcie ładowania strony,

- src – adres URL do odtwarzanego pliku.

W praktyce pojawiają się jednak problemy, związane z implementacją różnych wersji kodeków w przeglądarkach. Z tego powodu, należy definiować dodatkowe zasoby w tagu source. Atrybuty które obsługuje to:

- media – zdefiniowanie typu zasobu, domyślnie odpowiada wszystkim typom mediów,
- src – adres URL do źródła pliku,
- type – definiuje typ zasobu.

Korzystając z powyższych informacji możemy otrzymać listing 8, który będzie działał poprawnie w większości przeglądarek :

```
<audio id="id">
<source src="test.ogg" type="audio/ogg"></source>
<source src="test.mp3" type="audio/mpeg"></source>
</audio>
```

Listing 8.Osadzanie plików audio za pomocą HTML5.

Drugi, ważny element HTML5 oparty jest o znacznik video. Przyjmuje te same atrybuty co audio, a ponadto:

- height – określa wysokość okna playera,
- poster – adres URL alternatywnej grafiki, wyświetlanej jeżeli domyślny zasób nie będzie dostępny,
- width – określa szerokość okna odtwarzacza.

Jeśli pliki wideo zostały udostępnione w formatach H.264 Theora i VP8 osadzenie pliku video na stronie www jest bardzo proste, gdyż sprowadza się tylko do użycia znacznika *video* (patrz listing 9).

```
<!DOCTYPE HTML>
<html>
 <body>
    <video width="320" height="240" controls="controls">
       <source src="movie.mp4" type="video/mp4" />
       <source src="movie.ogg" type="video/ogg" />
       <source src="movie.webm" type="video/webm" />
       Your browser does not support the video tag.
    </video>
 </body>
</html>
```

Listing 9.Osadzanie plików wideo za pomocą HTML5.

Wszystkie wymienione atrybuty jak i przykłady w całości zostały zaimplementowane w przeglądarkach mobilnych.

3.10. Testowanie mobilnych stron internetowych

Ważnym momentem w trakcie tworzenia trony mobilnej jest jej testowanie na różnych urządzeniach i przeglądarkach internetowych. Potężna liczba powstającego sprzętu mobilnego uniemożliwia wykonania odpowiednich testów. Z pomocą przychodzą wówczas narzędzia, które zminimalizują czas trwania tego żmudnego, ale bardzo ważnego procesu. Poniżej zaprezentowano przykładowe narzędzie wykorzystane przy budowie prototypu.

Adobe Shadow jest darmowym narzędziem przydatnym podczas testowania stron mobilnych. Umożliwia deweloperom jednoczesny podgląd tej samej strony internetowej na wielu urządzeniach mobilnych. Aplikacja mobilna pozwala dodatkowo przeprowadzić analizę projektu i debugować kod, podczas gdy zmiany wprowadzane są na komputerze jak i w Internecie.
Funkcjonalności:

- zsynchronizowane wyszukiwanie i odświeżanie,
- zdalny nadzór nad kodem i debugowanie,
- autoryzacja urządzeń,
- podgląd i inspekcja kodu strony,
- URL monitoring,
- wsparcie dla Amazon Kindle Fire.

4. Opis narzędzi oraz technologii zastosowanych w pracy

W tym rozdziale zostaną przedstawione technologie oraz rozwiązania, które zostały użyte w niniejszej pracy.

Do stworzenia prototypu aplikacji zostały użyte technologie takie jak:

- Symfony 2 - framework PHP na którym został oparty projekt kreatora stron mobilnych,
- Extjs - biblioteka JavaScript, która umożliwia tworzenie aplikacji internetowych,
- JqueryMobile framework javaScript
- Them Roller - narzędzie, które umożliwia dowolne projektowanie stylów w aplikacji lub stronie mobilnej. Zapewnia bogatą paletę barw, a także możliwość podzielenie się nimi poprzez URL, pobranie i umieszczenie na stronie.

4.1. Symfony 2

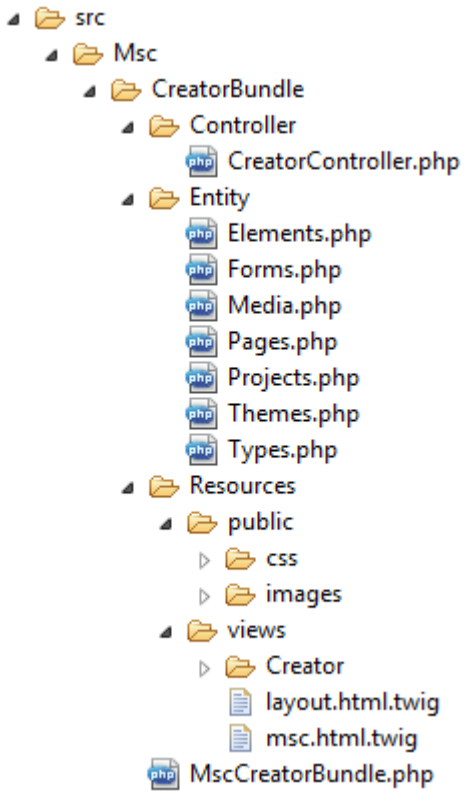
Celem tego podrozdziału jest przybliżenie najnowszych elementów frameworka Symfony 2, który został użyty przy projektowaniu prototypu.

Symfony jest framworkiem PHP na licencji MIT opartym o architekturę MVC ze wsparciem dla ORM (Object Relational mapping layer . Wykorzystując ten framework, użytkownik otrzymuje dostęp do wszystkich funkcjonalności charakteryzujących Doctrine 2, Twiga oraz wiele innych rozwiązań. Dzięki nim organizacja kodu pozwala aplikacji łatwo się rozrastać w czasie, unikając mieszania wywoływań zapytań bazy danych, HTML i logiki biznesowej w tym samym skrypcie [8].

4.1.1. Bundles

Bundle jest rodzaj plaginu z tą różnicą, że w Symfony 2 wszystko jest oparte o “bundle” włączając w to rdzeń oraz biblioteki frameworków oraz kod, który pisze sam deweloper. Bundle jest prostą strukturą plików, prezentującą poszczególne funkcjonalności, na podstawie których można stworzyć: BlogBundle, ForumBundle itd. Każda struktura zawiera pakiet danych związanych z funkcjonalnością włączając w to pliki php, css, Java Script czy testy.

Rysunek 3 prezentuje bundle na przykładzie prototypu kreatora stron, który zawiera kontroler, zdefiniowane klasy z encjami do wygenerowania bazy danych oraz pliki css i js.



Rysunek 3. Przykładowy bundle – CreatorBundle.

4.1.2. ORM - Doctrine 2

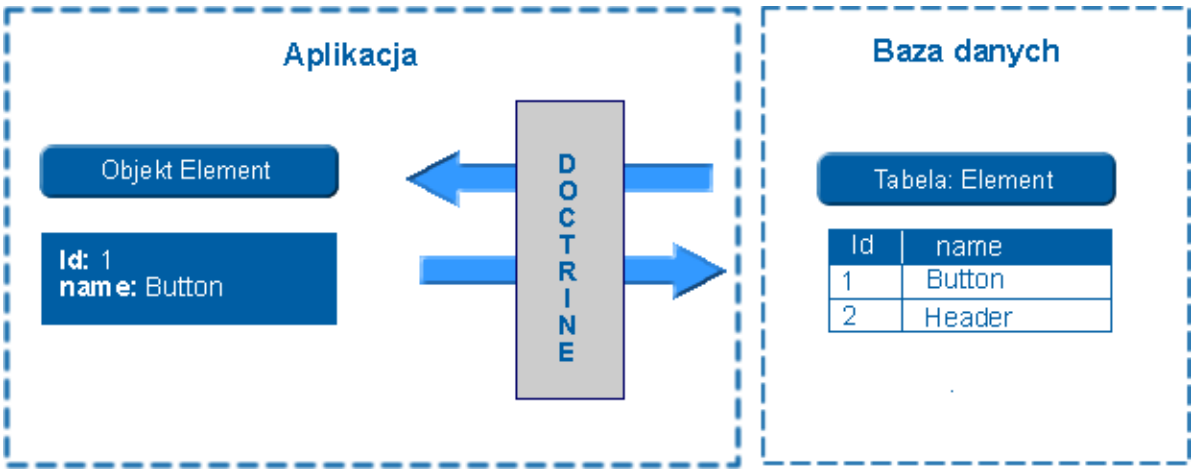
Doctrine (Database Abstraction Layer DBAL) jest to ORM dla języka PHP. Wykorzystuje schemat mapowania danych, dążąc do całkowitego zniesienia impedancji pomiędzy relacyjną bazą danych, a kodem aplikacji. Korzyści z tego ORM są takie, że jest on abstrakcyjną warstwą ulokowaną na szczycie (PDO -PHP Data Objects), umożliwiającą nam intuicyjne oraz elastyczne API do komunikacji z większością popularnych relacyjnych baz danych. Innymi słowy, biblioteka DBAL ułatwia wywoływanie zapytań oraz wykonywanie innych akcji bazy danych [9].

W dokumentacji Doctrine możemy przeczytać o dwóch sposobach podejścia do projektowania aplikacji i tworzenia modelu bazy danych. W większości przypadków, model bazy danych jest zaprojektowany i zbudowany od podstaw (tak jak to miało miejsce przy tworzeniu niniejszego prototypu). Czasami jednak pracuje się z istniejącym i nie podlegającym zmianie modelem bazy danych. Doctrine posiada kilka narzędzi, które pomogą wygenerować modele klas z istniejącej już bazy danych, ale z pewnymi uwagami. Inżynieria odwrotna (reverse engineering) jest jednorazowym procesem. Doctrine jest w stanie zinterpretować około 70-80% z ważnych informacji do zmapowania bazując na polach, indeksach oraz kluczach obcych [9]. Doctrine nie poradzi sobie z takimi zagadnieniami takimi jak:

- odwrotne połączenie (inverse associations),
- dziedziczenie (inheritance),

- encje, których klucze obce są kluczami głównymi
- semantyczne operacje na połączeniach takich jak kaskady lub cykle wydarzeń.

W powyższych przypadkach będzie potrzebna pewna dodatkowa praca, aby dopasować specyficzne wymagania modelu bazy do wygenerowanych encji. Na rysunku 4 został zaprezentowany diagram wyjaśniający zasadę działania ORM.



Rysunek 4. przedstawia zasadę działania ORM (Doctrine 2).

Na listingu 10 został przedstawiony fragment kodu z aplikacji. Jest to definicja klasy, opisanej encjami. Na ich podstawie zostanie wygenerowana tabela w bazie danych oraz odpowiednie funkcje, które będą umożliwiały dostęp do danych zapisanych w bazie.

```
<?php
// src/Msc/CreatorBundle/Entity/Elements.php
namespace Msc\CreatorBundle\Entity;

use Doctrine\ORM\Mapping as ORM;
/**
* @ORM\Entity
* @ORM\Table(name="elements")
*/
class Elements
{
    /**
    * @ORM\Id
    * @ORM\Column(type="integer")
    * @ORM\GeneratedValue(strategy="AUTO")
    */
    private $id;
    /**
    * @ORM\Column(type="string", length="255")
    */

    private $name;
```

```
}
```

Listing 10. Przykładowa klasa opisująca obiekt mapowania.

Listing 11 pokazuje przykładowe zapytanie do bazy wykorzystujące funkcje udostępniane przez Doctrine 2.

```
$repository = $this -> getDoctrine();
$repository-> getRepository('MscCreatorBundle:Elements');
$repository-> find($elementId);
```

Listing 11. Przykładowe zapytanie wykonane z poziomu aplikacji przy użyciu Doctrine.

4.1.3. Twig

Twig jest nowym narzędziem w języku PHP zaprojektowanym do tworzenia template'ów (wzorów). Wyróżniają go 3 ważne cechy:

- szybkość - kompiluje widoki do czystego zoptymalizowanego kodu PHP,
- bezpieczeństwo- posiada sandbox-tryb do wyszukiwania nieprawidłowego kodu. Ta właściwość jest cenna o tyle, że Twig może być używany jako wzór dla wielu aplikacji i jego wygląd może być dowolnie modyfikowany,
- elastyczny - jest wspierany przez laxer i parser, co umożliwia deweloperowi stworzenie własnego DSL (nowe tagi, filtry).

Template nie jest niczym innym jak plikiem tekstowym. Nie posiada żadnego specyficznego rozwinięcia. Może to być html lub xml. Może być wygenerowany jako dokument tekstowy taki jak: HTML, XML, CSV, LaTeX i inne).

Omawiany template posiada parametry, które mogą być zastąpione przez nowe wartości wraz z kolejną aktualizacją oraz tagi, które kontrolują jego spójność. Prosty wzór z listingu 12 przybliża podstawy działania Twiga.

```
<!DOCTYPE HTML>
<html>
   <head>
       <title>Kreator stron Mobilnych</title>
   </head>
   <body>
       <ul id="menu">
       {% for item in menu %}
        <li><a href="{{ item.href }}">{{item.caption }}</a></li>
       {% endfor %}
       </ul>
       <h1>Kreator</h1>
       {{ name }}
```

```
    </body>
</html>
```

Listing 12. Przykładowy template twig.

4.2. Extjs - interfejs użytkownika

Extjs jest to ciekawy framework JavaScript, służący do tworzenia zaawansowanych interfejsów przypominających wyglądem i zachowaniem interfejsy okien w systemach operacyjnych.

Kwestia wyglądu aplikacji internetowych zawsze pozostawała niedoceniana przez twórców różnych technologii programistycznych. To jednak powoli się zmienia. Powstają frameworki JavaScript tworzone specjalnie z myślą o prostym tworzeniu efektownych interfejsów. Jednym z nich jest Ext JS, który został przygotowany jako narzędzie do tworzenia interfejsów dla wszelakich systemów internetowych. W mniejszym stopniu przyda się on podczas tworzenia klasycznych stron internetowych, ale doskonale nadaje się do paneli administracyjnych, CMS-ów oraz aplikacji internetowych [6].

W takim właśnie charakterze został zastosowany Extjs - by stworzyć przejrzysty i funkcjonalny interfejs użytkownik. Opis oraz funkcjonalność prototypu zostanie zaprezentowania w kolejnym rozdziale.

4.3. JQuery Mobile

JQuery Mobile jest przeznaczony do tworzenia aplikacji webowych działających na szerokiej gamie urządzeń mobilnych. Pozwala on na szybkie wytworzenie aplikacji, które będą wyglądały i działały w ten sam sposób we wszystkich obsługiwanych urządzeniach. Biblioteka ta nie jest jedynie mobilną wersją jQuery, wręcz przeciwnie: rozszerza ona macierzyste jQuery o elementy interfejsu i API do wyświetlenia na urządzeniach mobilnych, dostosowane do obsługi palcem bądź rysikiem. Oferuje deweloperowi układ interfejsu użytkownika i możliwości interakcji (przyciski, listy wyboru, pola tekstowe itp.), które są odpowiednio duże na ekranie i przez to łatwe do używania. Biblioteka ta wykorzystuje HTML5 i CSS3, a od stycznia 2012 dostępna jest w stabilnej wersji 1.0.1.

Należy zaznaczyć, że w znaczącej większości urządzeń mobilnych, posiadających dostęp do Internetu (np. smartfony czy tablety) oraz przeglądarek stron internetowych, jQuery Mobile jest obsługiwane. Istotnym atutem tego frameworku jest to, że jest on zoptymalizowany pod kątem urządzeń posiadających ekrany dotykowe, gdyż posiada całą masę elementów gotowych do wykorzystania przy layout'ach oraz duży zbiór komponentów służących do obsługi formularzy i interfejsu użytkownika. Wartym podkreślenia jest fakt, że pozostaje kompatybilny ze starszymi

modelami urządzeń mobilnych i przeglądarek, stając się jedną z popularniejszych platform umożliwiających pisanie aplikacji na telefony.

Poziom na którym jQuery Mobile wspiera poszczególne platformy, można podzielić na 3 klasy: A, B i C. Dane zaprezentowano w tabelach 6,7,8,9.

Klasa A -pełne wsparcie dla Ajaxa, animacji stron oraz różnych kierunków zapisu tekstu (RTL)

Tabela 6. Lista urządzeń które w pełni wspierają Ajax-a i animacje.

System Operacyjny	Testowne urządzenia
Apple iOS 3.2 - 5.0	iPad (4.3 / 5.0), iPad 2 (4.3), iPhone (3.1), iPhone 3 (3.2), 3GS (4.3), 4 (4.3 / 5.0), 4S (5.0)
Android 2.1-2.3	HTC Incredible (2.2), Droid (2.2), HTC Aria (2.1), Google Nexus S (2.3)
Android 3.1 (Honeycomb)	Samsung Galaxy Tab 10.1, Motorola XOOM
Android 4.0 (ICS)	Galaxy Nexus S
Windows Phone 7-7.5	HTC Surround (7.0) HTC Trophy (7.5), LG-E900 (7.5), Nokia Lumia 800
Blackberry 6.0 - 7.0	Torch 9800 , Style 9670
Blackberry Playbook (1.0-2.0)	PlayBook
Palm WebOS (1.4-2.0)	Palm Pixi (1.4), Pre (1.4), Pre 2 (2.0)
Palm WebOS 3.0	HP TouchPad
Firebox Mobile (10 Beta)	Android 2.3
Chrome for Android (Beta)	Android 4.0
Skyfire 4.1	Android 2.3
Opera Mobile 11.5	Android 2.3
Meego 1.2	Nokia 950 and N9
Samsung bada 2.0	Samsung Wave 3
UC Browser	Android 2.3 device

Kindle 3	WebKit

Tabela 7. Przeglądarki internetowe – desktopowe.

Przegladarka	System Operacyjny
Chrome Desktop 11-17	Windows 7
Safari Desktop 4-5	Windows 7
Firefox Desktop 4-9	Windows 7
Internet Explorer 7-9	Windows XP, Vista , Windows 7
Opera Desktop 10-11	Windows 7

Klasa B-brak wsparcia dla nawigacji Ajax-owej.

Tabela 8. Lista urządzeń w których brak wsparcia dla nawigacji Ajax-owej.

System Operacyjny	Testowane urządzenia
Blackberry 5.0	Storm 2 9550, Bold 9770
Opera Mini (5.0-6.5)	iOS 3.2/4.3 and Android 2.3
Nokia Symbian	Nokia N8, N97

Klasa C-podstawowa wersja

Tabela 9. Podstawowe wsparcie.

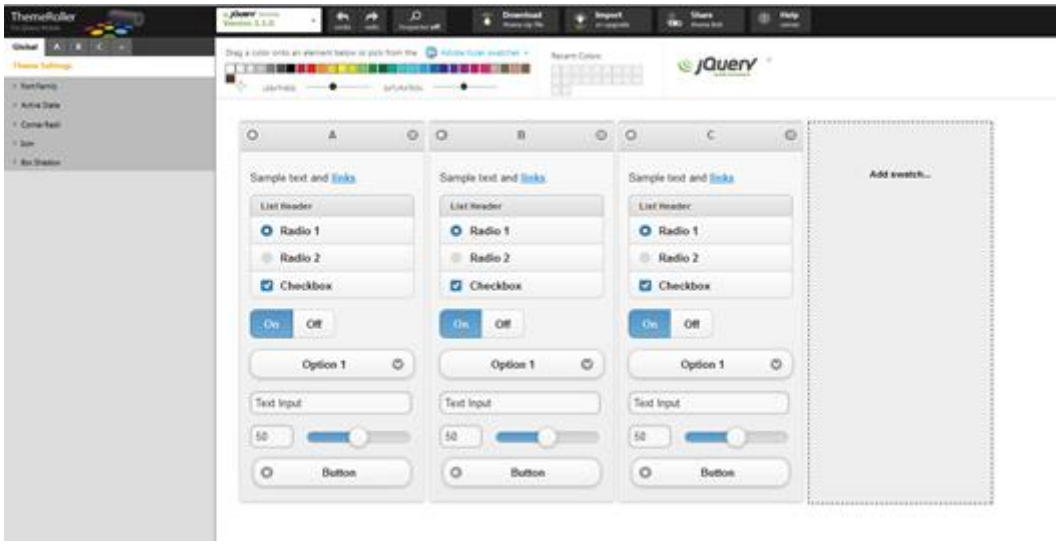
System Operacyjny	Testowane urządzenia
Blackberry 4.x	Curve 8330
Windows Mobile	HTC Leo (WinMo 5.2)

Wszystkie urządzenia mobilne, które nie obsługują języka Java Script będą zaklasyfikowane jako kategoria C.

4.4. Theme Roller

Za pomocą tego narzędzia można stworzy dowolny styl strony z możliwością zaimportowania do prototypu kreatora. Jest to wersja beta narzędzia dewelopera, zatem wspierania najnowsze wersje popularnych przeglądarek desktopowych: Chrome, Firefox, Safari. Narzędzie działa w IE9, ale nie obsługuje gradientów CSS, więc nie poleca się korzystania z tej przeglądarki w celu tworzenia tematów.

Poniżej została zaprezentowana krótka charakterystyka narzędzia, które współpracuje z kreatorem. Interfejs ThemeRoller. Składa się z 3 głównych stref: lewa kolumna zawiera panel inspektora, na górze jest QuickSwatch / Kuler bar, a poniżej okno podglądu. Za pomocą panelu inspektora można ustawić globalne ustawienia tematyczne na pierwszej karcie i dostosować poszczególne opcje stylu dla każdej próbki. Powyżej zakładek, znajdują się linki do pobrania. Tak wygenerowany plik można zaimportować do prototypu kreatora stron mobilnych. Na rysunku 5 przedstawiono widok narzędzia Theme Roller.



Rysunek 5. Widok narzędzia Theme Roller.
Źródło: http://jquerymobile.com/themeroller/index.php

5. Proponowane rozwiązanie

W poniższym rozdziale autor omówi założenia funkcjonalne prototypu, odwołując się do wymogów stawianych przez użytkowników jak i technicznych możliwości urządzeń mobilnych.

5.1. Motywacja przyjętego rozwiązania

Podczas tworzenia stron mobilnych warto pamiętać, że potencjalny użytkownik oczekuje od nich różnic w stosunku do ich tradycyjnych odpowiedników. Osoby korzystające ze stron mobilnych w dużej mierze potrzebują dostępu do ściśle określonych treści i przeważnie nie są zainteresowane dodatkami w postaci zbędnych funkcjonalności, trudnych do wyłączenia reklam czy też ozdób graficznych. Wymusza to na programiście dokonania właściwej selekcji i umieszczenia na stronie jedynie wartościowych informacji i niezbędnych funkcji. Istotnymi różnicami pomiędzy stroną mobilną, a tradycyjną wpływającymi na ich odmienną architekturę są:

- Czas dostępu do treści strony - przykładowo, użytkownik chcący sprawdzić dostępność określonego produktu w sieci, nie będzie chciał zbyt długo czekać na załadowanie strony z wynikami wyszukiwania.
- Koszty – miejsc z darmową siecią WiFi jest bardzo niewiele, a umowy abonamentowe z płatnym Internetem mają wciąż ograniczoną ilość transferu danych. W takiej sytuacji należy wziąć pod uwagę koszty, które można ograniczyć tworząc małe objętościowo strony mobilne, które znacznie szybciej będą mogły się wyświetlać.
- Nawigacja na stronie - wyświetlając stronę internetową na ekranie telefonu komórkowego posługujemy się zupełnie inną techniką nawigacji niż ta, z której korzystamy podczas przeglądania tradycyjnej strony internetowej.
- Estetyka - małe wyświetlacze telefonów komórkowych i smartfonów sprawiają, że strona tradycyjna może nie być czytelna na tych urządzeniach. Najczęstszą przyczyną może być zbyt duża ilość elementów, których nie pomieści mały ekran. Pewne elementy zostaną pomniejszone do tego stopnia, że użytkownik ich nie zauważy.
- Wykorzystanie oprogramowania - nie wszystkie urządzenia mobilne obsługują te same technologie np. flash czy też implementują w podobnym stopniu HTML5 czy CSS3.

Należy podkreślić, że coraz więcej liczących się serwisów i portali, w trakcie otwierania głównej strony na urządzeniu przenośnym, daje nam możliwość wyboru wersji mobilnej lub tradycyjnej. Stwarza tym samym alternatywę dla wielu użytkowników, którzy nie potrafią przyzwyczaić się do odmiennego wyglądu strony, którą odwiedzali na komputerze.

Reasumując, w myśl reguły KISS (*Keep It Simple, Stupid*) przyjazną użytkownikowi stronę cechują:

- lekkość,
- uproszczona funkcjonalność,
- minimum treści,
- szybkie ładowanie się,
- intuicyjność,
- jakościowe dostosowanie treści.

5.2. Założenia funkcjonalne prototypu

Zadaniem prototypu kreatora jest tworzenie mobilnych stron internetowych. Ważne jest, aby były poprawnie wyświetlane na najpopularniejszych przeglądarkach mobilnych, na większości urządzeń przenośnych. Wygląd strony musi być dopasowany do rozdzielczości ekranów telefonów i tabletów, przy zachowaniu optymalnej użyteczności. Poniżej zostały zaprezentowane wszystkie funkcjonalności prototypu:

- Narzędzie typu „przeciągnij i upuść" - użycie takiej techniki podyktowane jest bezpośrednio oczekiwaniami użytkowników, którzy cenią sobie wygodne i łatwe w użyciu narzędzia. Przeciąganie elementów na stronie daje możliwość swobodnego projektowania widoku tworzonej strony.
- Eksport kodu do pliku - skorzystanie z tej opcji pozwala deweloperowi na odłączenie projektu od domeny kreatora. Wyeksportowany plik z gotowym projektem strony mobilnej, jako gotowy produkt, może być przeniesiony na dowolny serwer i własną domenę.
- Hostowanie stron w aplikacji - funkcjonalność adresowana do tych użytkowników, którzy nie dysponują własnym serwerem lub kontem hostingowym. Kreator umożliwia przechowywanie wygenerowanej strony mobilnej na jego serwerze.
- Kreator formularzy - dedykowany jest wszystkim tym twórcom stron, którzy zainteresowani są gromadzeniem dodatkowych danych o osobach odwiedzających ich strony. W formularzu można poprosić o dane takie jak: imię, wiek, adres e-mail, a następnie wykorzystać powstałą bazę danych np. do wysyłki newslettera.
- Edytor kodu CSS - umożliwia zmianę styli opisujących wygląd strony i elementów z których ją zbudowano. Z funkcjonalności tej skorzysta bardziej zaawansowany użytkownik.
- Możliwość podłączenia kodów analitics - autor kieruje taką opcję do wszystkich tych, którzy chcą mieć kontrolę nad popularnością ich strony. Z analizy różnego rodzaju statystyk można wywnioskować np.: z jakiego urządzenia najczęściej korzystają

użytkownicy strony, w jakim regionie kraju strona wyświetlana jest najczęściej czy za pomocą jakich słów kluczowych pozyskuje najwięcej odwiedzających.

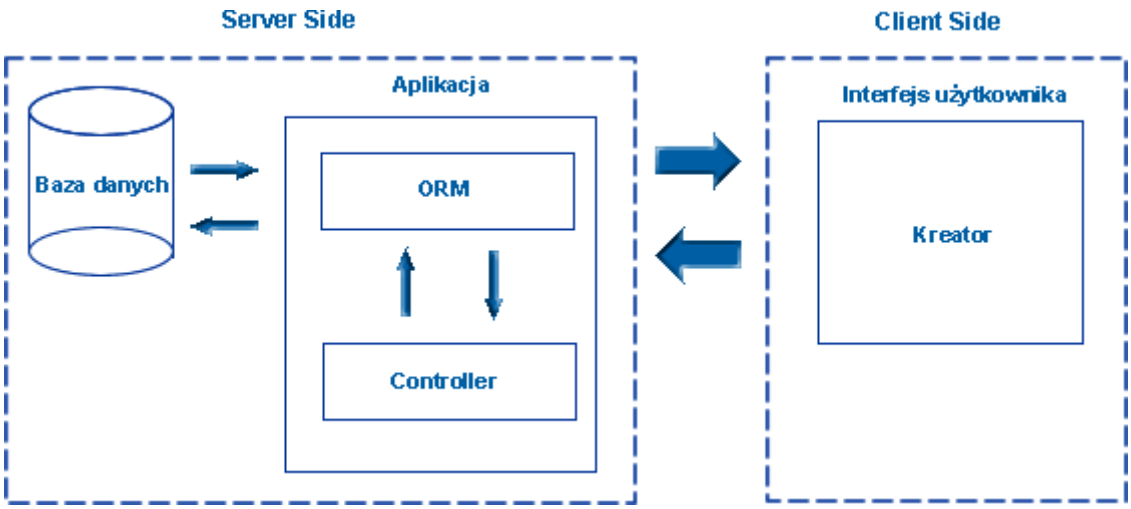
- Możliwość tworzenia wielu stron dla jednego użytkownika - kreator nie ogranicza ilości projektów jaką mogą mieć użytkownicy.
- Edytor kodu źródłowego - dla bardziej zaawansowanych użytkowników ważna jest edycja kodu. Dzięki tej funkcji mogą oni dodać własne elementy wzbogacające stronę mobilną o nowe funkcjonalności.
- Podgląd stron w różnych rozdzielczościach pozwala użytkownikowi na dopasowanie strony do dowolnie wybranej rozdzielczości (technika resize).
- Możliwość importu gotowych tematów z ThemeRoller - dla tych użytkowników, którzy czerpią z domyślnych styli kreatora istnieje opcja wygenerowania nowych tematów w osobnym narzędziu ThemeRoller, a następnie zaimportowania ich do prototypu kreatora.
- Obsługa dynamicznych treści na stronie mobilnej, która sprowadza się do elastyczniejszego zarządzania tekstami na stronie.

6. Prototyp kreatora stron mobilnych

Na potrzeby niniejszej pracy powstał prototyp kreatora stron mobilnych wykorzystujący framework jQuery Mobile. W tym rozdziale autor zaprezentował architekturę aplikacji, dokładny opis interfejsu narzędzia, a następnie ocenił stopień implementacji API jQuery Mobile w tym narzędziu. Na koniec, porównał funkcjonalności działającego prototypu z założeniami projektu opisanymi szerzej w rozdziale 5.

6.1. Architektura aplikacji

Omawiany w tym rozdziale prototyp kreatora jest aplikacją webową opartą o architekturę trójwarstwową typu klient serwer, w której interfejs użytkownika, przetwarzanie i składowanie danych są rozwijane w osobnych modułach. Rysunek 6 schematycznie prezentuje architekturę aplikacji.



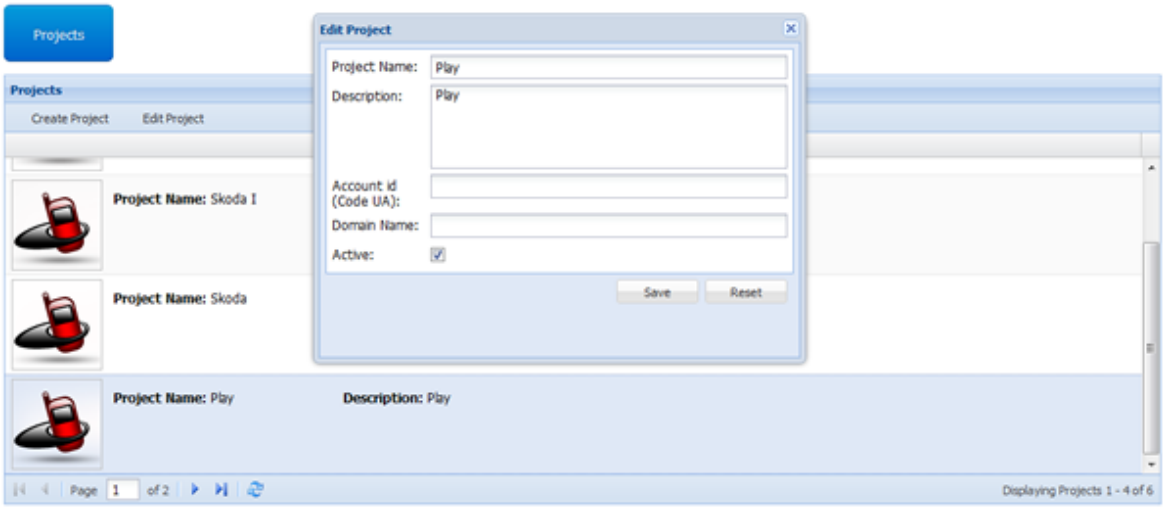
Rysunek 6. Schemat blokowy architektury aplikacji.

6.2. Interfejs użytkownika

Całość interfejsu użytkownika została stworzona przy użyciu biblioteki Extjs. Można podzielić ją na kilka części :

- Menadżer projektów
- Menadżer multimediów
- Widok podglądu strony
- Kreator stron
- Kreator formularzy

Menadżer projektów jest tą częścią interfejsu użytkownika, która pozwala na dodawanie nowych projektów. Za pomocą tej funkcjonalności można zarządzać właściwościami projektu takimi jak: nazwa , opis, jego aktywacja lub deaktywacja. W tym miejscu możemy również zdefiniować kody analitics'a, które umożliwiają śledzenie ruchu na stronie mobilnej. Rysunek 7 ilustruje omawianą treść.



Rysunek 7. Widok menadżera projektów.

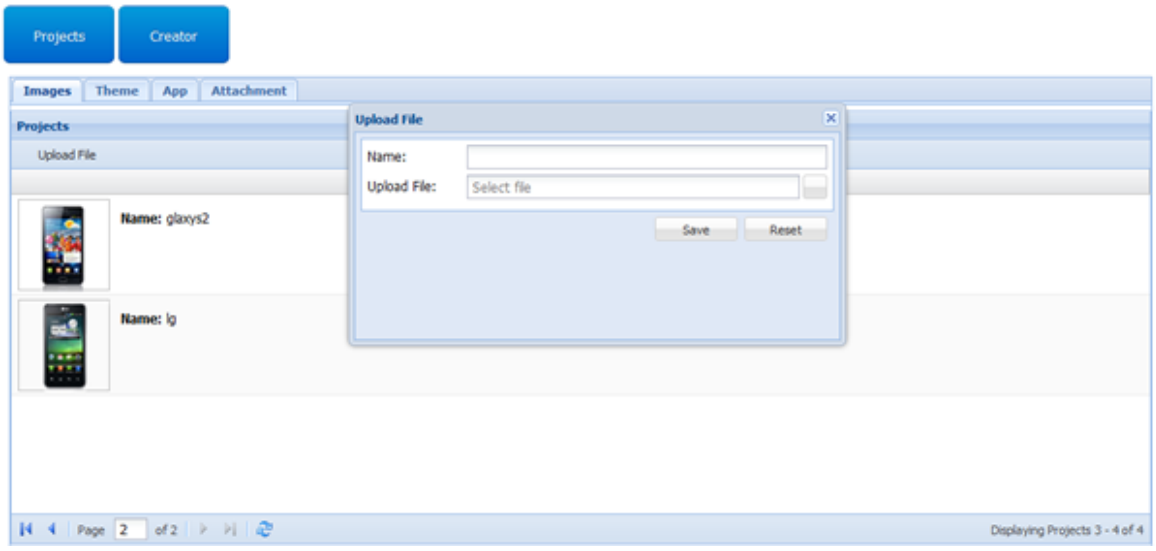
Zaprezentowany w pracy kreator narzuca schemat budowy każdej strony mobilnej. Punktem wyjściowym jest projekt, do którego dobudowywane są kolejne strony (patrz rysunek 8).

Rysunek 8. Przykładowa struktura projektu.

Menadżer multimediów pozwala na zarządzanie plikami takimi jak :

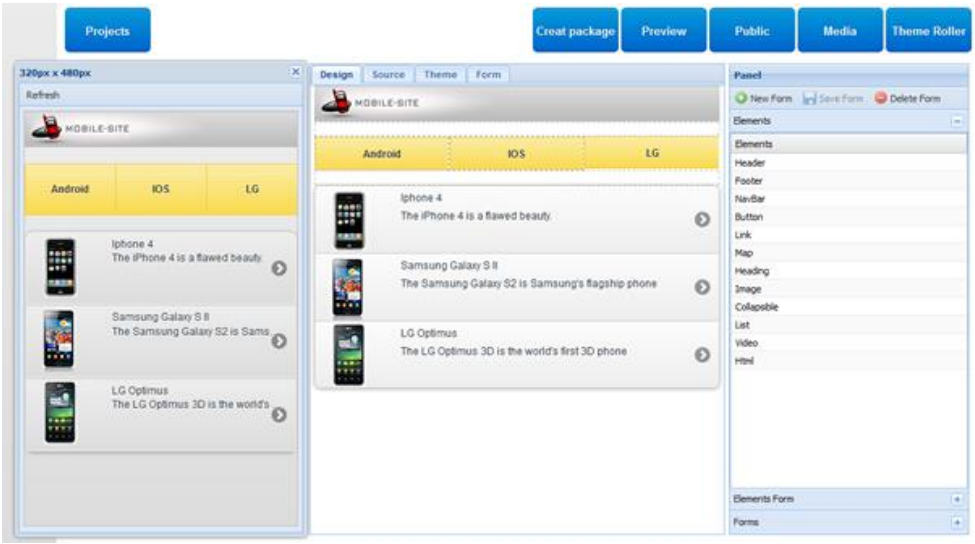
- obrazki (jpg, png),
- arkusze styli (css),
- pdf,
- inne (np. apk).

Pozwala na pobranie wyżej wymienionych plików i umieszczenie na serwerze, a następnie na wykorzystanie ich na stronie mobilnej. Omawianą funkcjonalność przedstawia rysunek 9.



Rysunek 9. Menadżer multimediów.

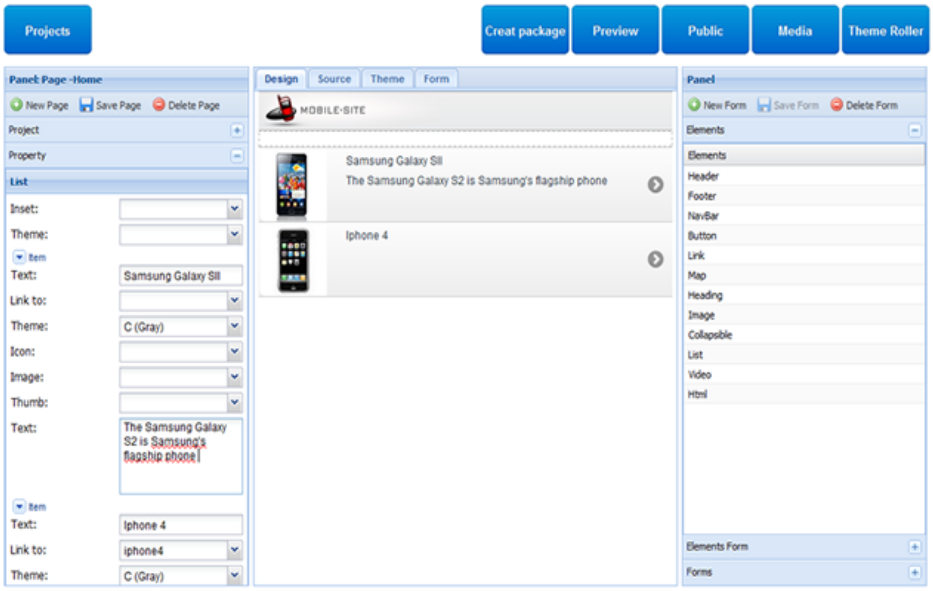
Za pomocą podglądu interfejsu można zobaczyć wygląd stron na urządzeniach mobilnych o różnych rozdzielczościach. Opcja taka znacznie skraca czas pracy nad stroną mobilną. Umożliwiając podgląd stron w różnych rozdzielczościach eliminuje przesyłanie strony na fizyczne urządzenie. Na rysunku 10 zaprezentowano podgląd przykładowej strony.



Rysunek 10. Widok podglądu przykładowej strony.

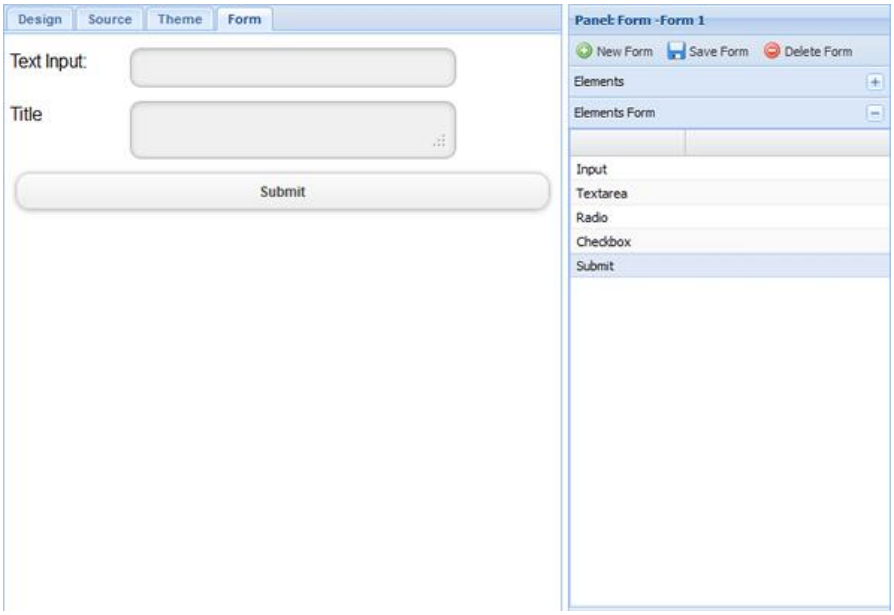
Kreator stron można podzielić na dwie części: jedna z nich umożliwia budowanie wizualnego aspektu strony, druga pozwala tworzyć formularze, które można do niej dołączyć. Przy budowie strony użytkownik ma do dyspozycji zdefiniowane elementy, takie jak: przyciski, nagłówki, mapy, obrazki, listy, wideo, które za pomocą techniki „przeciągnij i upuść" w łatwy sposób pobiera. Dodatkowo istnieje możliwość zdefiniowania własnych tagów HTML.

Po zakończeniu tworzenia strony można wyeksportować paczkę, w której będzie się znajdował wygenerowany kod oraz wszystkie potrzebne biblioteki i pliki. Jeśli użytkownik nie ma możliwości umieszczenia strony na własnym serwerze, może skorzystać z opcji opublikowania strony w domenie aplikacji. Widok kreatora przedstawia rysunek 11.



Rysunek 11. Widok kreatora stron mobilnych.

Kreator formularzy umożliwia użytkownikowi pozyskiwać dane od osób odwiedzających stronę. Za pomocą kontrolek np.: „input", „textarea" generuje pola formularza pokazane na rysunku 12.



Rysunek 12. Kreator formularzy.

6.3. Poziom implementacji

Prototyp kreatora stron mobilnych jest oparty w całości o bibliotekę jQuery mobile, która udostępnia szereg funkcji oraz atrybutów wykorzystanych przy budowie strony. W tabeli 10 autor zebrał elementy i atrybuty udostępniane przez jQuery, a następnie dokonał procentowego obliczenia ilości zaimplementowanych atrybutów w prototypie kreatora.

Tabela 10. Stopień implementacji API jQuery Mobile w prototypie kreatora.

Nazwa Elementu	Atrybuty	Stopień implementacji [%]
Button	data-corners data-icon data-iconpos data-inline data-theme	80
Collapsible	data-collapsed data-content-theme data-iconpos data-mini data-theme	80
Footer	data-id data-position data-fullscreen data-theme	50
Header	data-position data-fullscreen data-theme	66
Link	data-ajax data-direction data-dom-cache data-rel data-transition	80
Listview	data-inset data-split-icon data-split-theme data-theme data-icon	80
Navbar	data-icon data-iconpos data-theme	100
Checkbox	data-mini data-theme	50

Radio	data-theme	100
Select		0
Input	data-mini data-theme data-track-theme	33
Textarea	data-theme	100
Submit	data-type data-value	100

W działającym prototypie kreatora udało się zaimplementować większość funkcjonalności opisanych szerzej w sekcji 5.2. Zgodnie z założeniem, kreator jest narzędziem typu „przeciągnij i upuść" z możliwością eksportu kodu do pliku (patrz Rysunek 11, zakładka „Create package") lub hostowania na serwerze kreatora (patrz Rysunek 11, zakładka „Public"). Pozwala tworzyć formularze (patrz Rysunek 12), edytować kod źródłowy jak i podłączyć kody analitics'a (patrz Rysunek 7). Nie ogranicza ilości stron, jaką może wygenerować użytkownik i pozwala mu importować gotowe tematy z ThemeRoller. Możliwość podglądu stron, na różnych rozdzielczościach urządzeń mobilnych, zaprezentowano na rysunku 10.

Obsługa dynamicznych treści na stronie mobilnej jest jedyną założoną funkcjonalnością, której nie zrealizowano w projekcie. Z doświadczenia autora wynika, że mobilne strony internetowe nie zawierają dużej ilości tekstu, są zwykle wizytówkami stron, stronami produktowymi lub reklamowymi o większej zawartości multimediów. Z tego powodu ta funkcjonalność nie została zaimplementowana w prototypie kreatora.

Oprócz podstawowych elementów udostępnionych prze API jQuery Mobile, autor dodał kilka dodatkowych, które zostaną przedstawione w rozdziale 6.4.

6.4. Komponenty kreatora

Poniżej zostały zaprezentowane wszystkie dostępne elementy w wersji podstawowej, z których może skorzystać użytkownik w trakcie tworzenia strony mobilnej wykorzystując prototyp kreatora. Elementy te można podzielić na dwie grupy: jedna z nich służy do tworzenia wizualnego aspektu stron, druga zaś to elementy formularza.

Do grupy pierwszej należą:

Header - nagłówek - jest to pasek narzędzi na górze strony, który zazwyczaj zawiera tytuł strony (patrz listing 13).

```
  <div data-role="header">
    <h1>Page Title</h1>
</div>
```

Listing 13. Przykładowy kod nagłówka.

Footer - stopka ma taką samą podstawową strukturę jak nagłówek. Przykład w listingu 14.

```
<div data-role="footer">
    <h1>footer Title</h1>
</div>
```

Listing 14. Przykładowy kod stopki.

NavBar - jQuery Mobile posiada pasek nawigacyjny, który jest przydatny przy tworzeniu nawigacji na stronie. Umożliwia on dodanie do 5 przycisków oraz opcjonalnych ikon w pasku, zwykle używany w nagłówku lub stopce, co przedstawia listing 15.

```
  <div data-role="navbar">
    <ul>
        <li><a href="#home" class="ui-btn-active">Home</a></li>
        <li><a href="#contact">Contact</a></li>
    </ul>
</div>
```

Listing 15. Przykładowy kod paska nawigacji.

Button - przycisk ten jest stworzony ze standardowego HTML-a, a następnie wsparty przez dodatkowe ostylowanie wygenerowane przez jQuery Mobile tak, aby był bardziej atrakcyjny i użyteczny na urządzeniu przenośnym. Używany jako element nawigacji oraz przy wysyłaniu formularzy (patrz listing 16).

```
  <a href="#start" data-role="button"> button</a>
```

Listing 16. Przykładowy kod przycisków nawigacyjnych.

Link - jest to zwykły odnośnik HTML-owy rozszerzony o dodatkowe atrybuty takie jak "data-rel" umożliwiający użycie zewnętrznych linków**.** Przykładowy kod linku podano w listingu 17.

```
<a href="http://mobile-site.pl"> data-rel="external">Link</a>
```

Listing 17. Przykładowy kod linku.

Map – jest elementem stworzonym przez autora, który umożliwia wyświetlenie dowolnego miejsca na mapie jak w listingu 18.

```
<img src="https://maps.googleapis.com/maps/api/staticmap?
center=location&zoom=zoom&size=width x height&sensor=false">

data-role="map"/>
```

Listing 18. Przykładowy kod mapy.

Heading - umożliwia dodanie nagłówków HTML-owych. Przykład użycia zamieszczono w listingu 19.

```
<h1 data-role="heading"> Heading </h1>
```

Listing 19. Przykładowy kod nagłówków.

Image - umożliwia dodanie obrazka do strony według wzorca z listingu 20.

```
<img data-role="image" src="image.png" />
```

Listing 20. Przykładowy kod umożlwiający osadzenie obrazków.

Collapsible - jest to element umożliwiający zwijanie i rozwijanie treści, jak w załączonym listingu 21.

```
<div data-role="collapsible">
  <h3>Header</h3>
  <p>Content text</p>
</div>
```

Listing 21. Kod umożliwiający animację treści.

List - Dodaje do strony listę z danymi (patrz listing 22).

```
<ul data-role="listview">
    <li><a href="#home>Home</a></li>
    <li><a href="#android">Android</a></li>
    <li><a href="#ios">iOS</a></li>
</ul>
```

Listing 22. Przykładowy kod listy.

Video (youtube) - Umożliwia osadzenie filmów video z youtube jak w przykładzie listingu 23.

```
<div data-role="video">
<object class="VideoPlayback" type="application/x-shockwave-flash">
  <param value="http://youtube.com/asFerxCD" name="movie">
  <param value="autoplay=true" name="flashvars">
```

```
 </object>
</div>
```

Listing 23. Przykładowy kod osadzania Video.

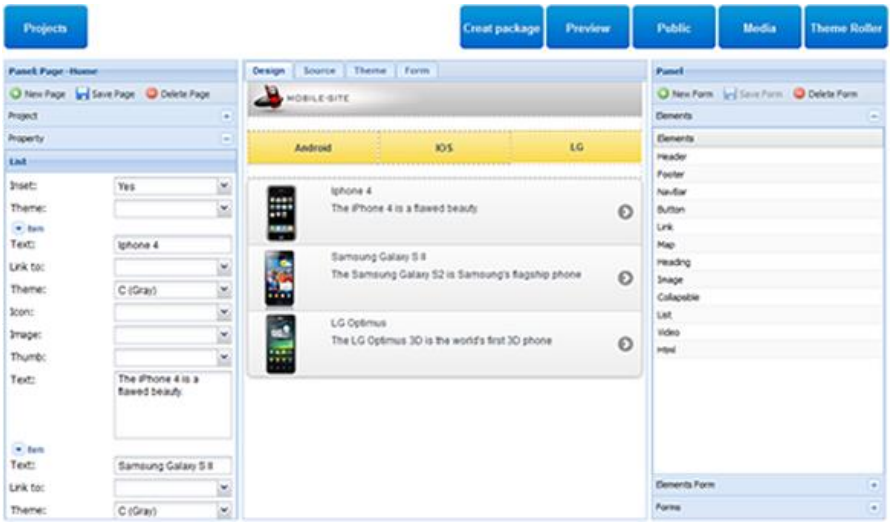
Html - umożliwią użytkownikowi dodanie dowolnego kodu Html do strony.

Do drugiej grupy elementów zaliczone zostały elementy formularza takie jak: input, textarea, radio, checkbox. Są to standardowe elementy HTML wzmocnione tylko poprzez jQuery Mobile, aby wyświetlały się w prawidłowy sposób na urządzeniach mobilnych.

6.5. Kreator - przykładowa strona

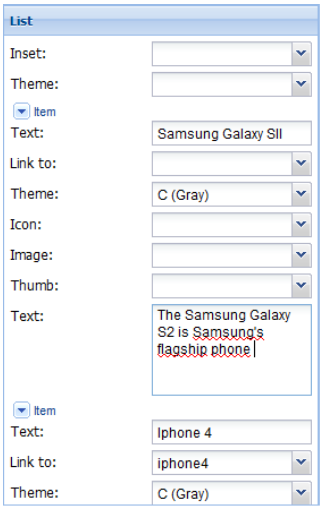
W tym rozdziale, na przykładzie dwóch stron, zostanie zaprezentowany efekt, jaki można uzyskać za pomocą prototypu kreatora stron mobilnych.

Pierwszy projekt z rysunku 13, w całości został wygenerowany przy pomocy elementów dostępnych w kreatorze oraz przy użyciu domyślnych styli dla elementów na stronie. Tworzenie projektu należy rozpocząć od funkcji "create project". W tym momencie kreator poprosi o wypełnienie pól takich jak: nazwa projektu i jego opis (opcjonalnie). Dane można poszerzyć o kody analitics, bardzo przydatne do śledzenia ruchu internautów na działającej już stronie mobilnej. Po zapisaniu należy przystąpić do dodawania stron do już istniejącego projektu. W tym przypadku są to strony takie jak home, android, lg, iphone. Kolejnym etapem jest dobór elementów spośród dostępnych po prawej stronie w panelu (rysunek 13). Za pomocą techniki "przeciągnij i upuść" przenosimy elementy do naszej strony. Do budowy zaprezentowanej strony „Mobile-site" użyto elementów takich jak: listy, nagłówki, znaczniki Html, video oraz element image.



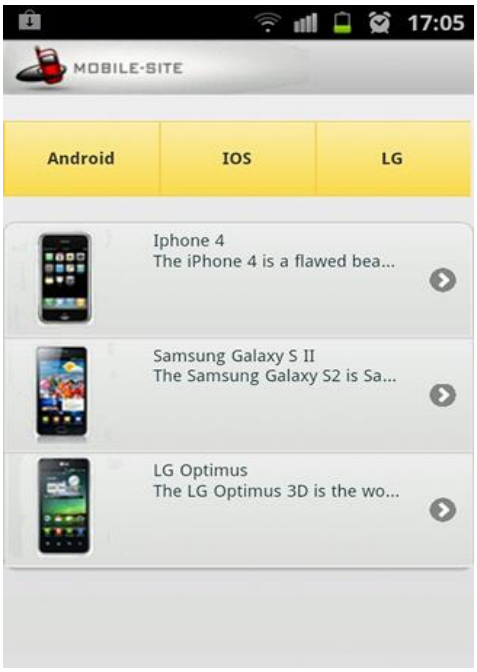
Rysunek 13. Widok kreatora w trakcie tworzenia przykładowej strony.

Po przeciągnięciu każdego z wyżej wymienionych elementów i jego zaznaczeniu, po lewej stronie pojawia się formularz konfiguracji danego elementu. Przykładowy formularz konfiguracji "listy" został pokazany na rysunku 14.



Rysunek 14. Widok formularza właściwości listy.

Po skonfigurowaniu każdego elementu można zapisać stronę i podejrzeć jej wygląd klikając na zakładkę "Preview". Jeżeli wygląd odpowiada oczekiwaniom, można przystąpić do budowy kolejnych podstron. Po uzupełnieniu wszystkich danych, kreator pozwala wygenerować cały projekt strony, klikając na zakładkę "Create package". Opcja ta umożliwia przeniesienie strony na serwer. Można również skorzystać z opcji "public", która generuje stronę i umieszcza ją pod konkretnym adresem strony danego projektu. Wspomniana możliwość jest niezwykle istotna dla użytkowników, którzy nie mają własnego serwera. Zrzuty ekranów z urządzenia Samsung Glaxy S2 (patrz rysunek 15,16,17) przedstawiają kolejne strony wygenerowane przez kreator.

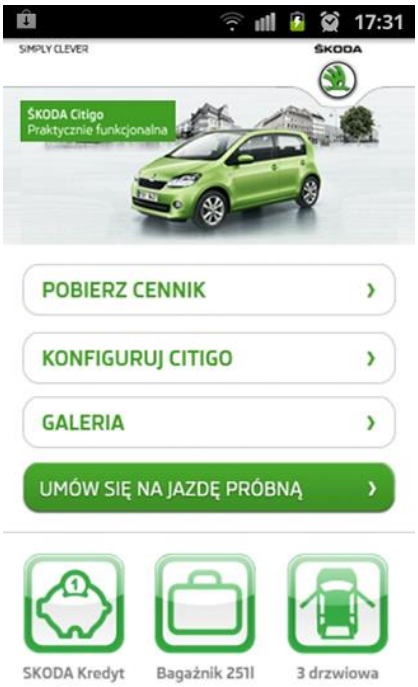


Rysunek 15. Widok strony głównej na urządzeniu Samsung Glaxy S2.

Rysunek 16. Widok podstrony z listą danych na urządzeniu Samsung Glaxy S2.

Rysunek 17. Widok podstrony z osadzeniem pliku wideo na urządzeniu Samsung Glaxy S.

W projekcie drugim, w wygenerowanej przez kreator stronie, zostały nadpisane wszystkie domyślne style dla elementów występujących na stronie (patrz rysunek 18 i 19). Autor skorzystał z edytora kodu źródłowego, rozszerzył wybrane właściwości elementów (np. przycisków) oraz podłączył zewnętrzną galerię zdjęć. Podjęte działanie było konieczne, gdyż kreator nie daje takiej możliwości.



Rysunek 18. Widok strony z modyfikacjami w źródle strony na urządzeniu Samsung Glaxy S.

Rysunek 19. Widok podstrony z modyfikacjami w źródle strony z osadzeniem pliku wideo na urządzeniu Samsung Glaxy S.

Jak widać na powyższych zrzutach ekranów przy niewielkim nakładzie pracy wykorzystując zakładkę „Source” strona może wyglądać bardzo efektownie.

7. Podsumowanie

W tym rozdziale zostaną opisane wady i zalety przyjętego rozwiązania problemu tworzenia stron na urządzenia mobilne, proponowane kierunki rozwoju projektu, a także wnioski końcowe.

7.1. Wady i zalety rozwiązania

Istotną wadą zaprezentowanego prototypu jest ograniczenie użytkownika, poprzez narzucenie typów elementów z których może zbudować swoją stronę mobilną. Nie ma także możliwości definiowania nowych re-używalnych elementów. Wygenerowany kod jest sformatowany specyficznie, co utrudnia jego analizę (domena większości generatorów kodu). Kreator nie pozwala na podłączenie własnych zdarzeń (event'ów) do elementów budujących stronę.

Wśród zalet takiego rozwiązania na pierwszym miejscu należy wymienić łatwość użytkowania. Nowoczesne narzędzie typu „przeciągnij i upuść" pozwala szybko tworzyć strony mobilne z gotowych i łatwo konfigurowalnych elementów. Za pomocą funkcji „Preview" można podejrzeć wygląd strony bez konieczności wrzucania na urządzenie przenośne. Ponadto, kreator wypełnia swoje podstawowe zadanie, mianowicie wygenerowany kod jest kompatybilny z większością przeglądarek obsługiwanych przez smartfony. Powstała przy użyciu prototypu strona, może być eksportowana do pliku oraz hostowana w aplikacji.

7.2. Proponowany plan rozwoju

Poniżej zostały zaprezentowane funkcjonalności jak i elementy kreatora, które można rozszerzyć.

Zaproponowany prototyp kreatora jest konstrukcją otwartą poddającą się modyfikacjom i rozbudowie. Użyteczność kreatora można zwiększyć poprzez zmianę organizacji i umiejscowienia poszczególnych komponentów. Na przykład: definiując właściwości wybranego elementu, można zapewnić łatwiejszy dostęp do widoku wszystkich stron w projekcie. Każda treść w wygenerowanej stronie jest statyczna, czyli nie można jej modyfikować inaczej niż poprzez edycję w kreatorze. Dodanie kolejnego elementu w postaci listy artykułów umożliwiłoby dynamiczne zarządzanie treścią na stronie. Kolejnym rozszerzeniem projektu może być zwiększenie ilości walidatorów w celu wyeliminowania błędów użytkownika. Przykładowo, ograniczenie ilości plików multimedialnych na stronie zabezpieczy przed długotrwałym ładowaniem się strony. W kreatorze można zaimplementować większą ilość gotowych elementów, dzięki czemu tworzone strony będą wizualnie i funkcjonalnie ciekawsze.

7.3. Wnioski końcowe

Napisanie kodu strony, która byłaby w jednakowy sposób wyświetlana na wszystkich urządzeniach mobilnych i tym samym kompatybilna ze wszystkimi przeglądarkami, jest zadaniem bardzo trudnym. Absolutnie idealną sytuacją dla twórców stron internetowych na urządzenia mobilne byłoby narzucenie pewnych standardów producentom przeglądarek internetowych, a także deweloperom programującym urządzenia mobilne. Szanse na taką sytuację są niewielkie, dlatego innym kierunkiem optymalizacji procesu jest rozbudowa istniejących już bibliotek frameworków jak i tworzenie nowych, aktualizowanych przez coraz szerszą rzeszę deweloperów. Dzięki temu, jedna wersja kodu zapewnia podobny widok projektu na większości urządzeń i przeglądarek obsługujących tego typu biblioteki.

8. Bibliografia

[1]. A. Maciejewski. „*Z bankiem w kieszeni*". Computerworld. 14 luty 2012, str. 26.

[2]. Juniper Research *Mobile Markets & Strategies.* [Online] [19 03 2012].

[3]. *"E-sklep to już za mało"* artykuł z Rzeczpospolitej z dnia 22.03.2012.

[4]. B. P. Hogan. **„***HTML5 i CSS3 standardy przyszłości"*, 2011 Helion S.A [ISBN:978-83-246-3028-8].

[5]. G. R. Frederick, R**.** Lal. „*Projektowanie witryn internetowych dla urządzeń mobilnych"*, 2010 Helion S.A [ISBN: 978-83-246-2729-5].

[6]. „*Dokumentacja Ext JS 4.0.7"* http://docs.sencha.com/ext-js/4-0, kwiecień 2012.

[7]. T.Parker. *„Dokumentacja Jquery Mobile 1.0.1"* http://jquerymobile.com, kwieceń 2012.

[8]. *„The Book - Symfony"* http://symfony.com, maj 2012.

[9]. *„Dokumentacja Doctrine 2.0"* http://www.doctrine-project.org , czerwiec 2012.

[10]. M.Frost "Learning WML, and WMLScript. Programming the Wireless Web", 2000 O'Reilly.

[11]*." HTML5 is The New HTML Standard"* http://www.w3schools.com/html5/default.asp, czerwiec 2012.

[12] J. Pearce. *„Programowanie mobilnych stron internetowych z wykorzystaniem systemów CMS"*, 2011 Helion S.A [ISBN:978-83-246-3028-8].