

Matematički fakultet
Univerzitet u Beogradu

Tema master rada:
**„Razvoj aplikacije Elastyc za
mobilne uređaje korišćenjem
jQuery Mobile okruženja“**



Mentor:
prof. dr. Dušan Tošić

Student:
Ana Miladinović

U Beogradu, septembar 2012.

Sažetak

JQuery Mobile je višeplatformsko mobilno okruženje kreirano da pojednostavi i unapredi razvoj mobilnih veb aplikacija integriranjem HTML5, CSS3, jQuery i jQuery UI u jedno okruženje. JQuery Mobile API (Application Program Interface – interfejs za programiranje aplikacija) ne podržava samo događaje na dodir, već i uobičajene događaje mišem, što ga čini kompatibilnim sa raznim platformama i pregledačima veba. Moguće je kombinovati JQuery Mobile sa tehnologijom PhoneGap da bismo kreirali matičnu (*native*) aplikaciju. U radu je iskorišćeno JQuery Mobile okruženje za kreiranje mobilne veb aplikacije na osnovu već postojeće Elastic aplikacije za elektronsko poslovanje.

Abstract

JQuery Mobile is a cross-platform mobile framework designed to simplify and enhance the development of mobile web applications by integrating HTML5, CSS3, jQuery and jQuery UI into one framework. JQuery Mobile's API supports not only Touch events, but also the usual Mouse, making it completely cross-browser and platform compatible. It is possible to combine JQuery Mobile with PhoneGap technology to create native applications.

In this work, JQuery Mobile framework is used for development of mobile web application based on existing Elastic application for e-busines.

1. Sadržaj

2. Predgovor	5
3. Uvod.....	6
3.1 Mobilni svet	6
3.1.1 Mobilni uređaji	7
3.1.2 Operativni sistemi i pregledači veba	7
4. JQuery Mobile.....	9
4.1 JQuery Mobile, HTML5 i CSS3.....	9
4.1.1 Kreiranje jQuery Mobile aplikacije	10
4.2 Glavne karakteristike jQuery Mobile aplikacije su:	11
4.2.1 Progresivno proširenje	12
4.3 Kompatibilnost jQuery Mobile i pregledača veba	12
4.4 Arhitektura i navigacija	13
4.4.1 Dijalozi	18
4.4.2 Tranzicija između strana.....	18
4.5 Elementi strana.....	19
4.5.1 Sadržaj za formatiranje.....	22
4.5.2 Sadržaj koji se sažima	22
4.5.3 Rešetka (<i>grid</i>).....	23
4.5.4 Dugmad.....	25
4.5.5 Liste.....	28
4.6 Forme I jQuery Mobile.....	30
4.6.1 Radio dugmad (<i>radio buttons</i>) i polja za potvrđivanje (<i>check buttons</i>)	32
4.7 JavaScript I jQuery Mobile	32
4.7.1 Konfiguracija.....	33
4.7.2 Metode za upravljanje stranama	38
4.7.3 Metode definisane za platformu.....	40
4.7.4 Metode definisane za putanje i Url-ove	41
4.7.5 Metode za korisnički interfejs	41
4.8 Događaji	42

4.8.1 Rukovanje događajima strana	42
4.9 JQuery mobile teme za stil	44
5. Rešenja slična jQuery Mobile-u	47
6. Elastyc	49
7. Teorijski opis mobilne Elastyc aplikacije.....	52
7.1 Struktura aplikacije	52
7.2 ASP	60
8. Zaključak	61
9. Literatura	62

2. Predgovor

Svi smo svedoci sve veće ekspanzije mobilnih uređaja i njihovog korišćenja za pristup veb sadržajima. Iz godine u godinu raste broj preuzimanja mobilnih aplikacija sa takozvanih prodavnica mobilnih aplikacija (npr. sa Android market-a). Od 2010. do 2011. godine taj broj je porastao sa 8,2 na 17,2 milijarde, a očekuje se da će 2014. godine dostići čak 185 milijardi. Iz tog razloga se pojavila moja zainteresovanost za razvoj kako mobilnih veb aplikacija, tako i matičnih mobilnih aplikacija.

jQuery Mobile je okruženje koje omogućava kreiranje veb aplikacija za mobilne uređaje (pametne telefone i tablične računare) prilagođene mnogim platformama i pregledačima veba. Osim toga, u kombinaciji sa drugim tehnologijama, kao što je na primer Phone Gap, omogućuje kreiranje matičnih mobilnih aplikacija, pa kao takav predstavlja univerzalno rešenje.

U samom radu biće opisano jQuery Mobile okruženje i aplikacija Elastyc koja je „preseljena“ u mobilni svet. Ideja je da se nakon čitanja ovog rada stekne slika o konceptima jQuery Mobile-a, kao i značaj mobilnih veb aplikacija u poslovnom svetu.

3. Uvod

Na početku je bitno znati da ne postoji mobilni internet, već samo internet. Na mobilnom uređaju mi koristimo isti *Facebook* ili *Tweeter*, sa istim nalogom i istim smislom kao na desktop računaru. Međutim, razdvajaju ih način na koji ih koristimo i očigledne razlike, poput veličine ekrana. Skoro svaki današnji takozvani „pametni telefon“ (*smartphone*) može da prikaže izgled svakog veb sajta, ali ne dobija se isto iskustvo. I pored toga, mnogi se ipak odlučuju za desktop, pre nego za mobilne verzije, uglavnom jer su mobilne verzije loše urađene.

Ideja jeste da se kreira jedinstveni veb (One Web), ali za sad ostaje cilj u budućnosti, jer postoji puno mobilnih uređaja sa lošom konekcijom, brzinom i ograničenim mogućnostima, koji u teoriji mogu da čitaju bilo koji dokument, ali u praksi ne mogu uvek da ga prikažu na valjan način zbog nekompatibilnosti između uređaja, pregledača veba i aplikacije.

Višeplatformski razvoj aplikacija je poprilični izazov u matičnom okruženju, jer svaka platforma zahteva drugačiji SDK (*Software Development Kit*), pa je zato mobilni veb razvoj za sada dobro rešenje za većinu situacija.

3.1 Mobilni svet

Mobilne veb aplikacije (*webapp*) nastaju korišćenjem veb tehnologija (HTML, CSS, AJAX, JavaScript), ali korisniku nude doživljaj korišćenja matičnih (*native*) mobilnih aplikacija (aplikacija koje su izgrađene koristeći SDK i programski jezik određene mobilne platforme). Mogu biti korišćene na više načina :

- Preko mobilnog pregledača veba
- Instalirane kao mobilne veb aplikacije preko celog ekrana (*full-screen webapp*)
- Kao mobilna veb aplikacija instalirana preko paketa od strane prodavca
- Kao mobilna veb aplikacija ugrađena u matičnu aplikaciju (takozvani hibrid)

3.1.1 Mobilni uređaji

Danas ljudi ne pristupaju veb sadržajima samo sa svojih desktop računara, već postoji niz drugih uređaja različitih veličina ekrana, mehanizama... Svi mobilni uređaji su podeljeni u 5 kategorija :

- **Mobilni telefoni** u najužem smislu. Imaju mogućnost slanja SMS poruka, poziva i primanja poziva. Nemaju mogućnost korišćenja interneta i brzo će nestati sa tržišta.
- **Mobilni uređaji nižeg kvaliteta** imaju mogućnost korišćenja interneta, neki jednostavni muzički plejer, kameru. Ima dosta korisnika, ali su im ograničene mogućnosti rada na vebu.
- **Mobilni uređaji srednjeg ili visokog kvaliteta** trenutno su najrasprostranjeniji. Imaju ekran srednje veličine, osnovni HTML pregledač veba, pristojnu kameru, muzički plejer, WiFi bežičnu mrežu. Omogućuju pristojan rad na vebu.
- **Pametni telefoni** u koje spadaju iPhone, Android, Symbian, BlackBerry. Imaju višeprogramski operativni sistem, moderni HTML5 pregledač veba, 3G konekcije, GPS, tv izlaz, osetljivi su na dodir... Definicija da li je telefon pametan ili nije se vrlo često menja, tako da razlika između ovih i prethodnog tipa nije jasno određena.
- **Tablični računari** se ne razlikuju puno od prethodne kategorije. Karakteriše ih veći ekran (6-12 inča), WiFi bežična mreža, 3G, potpuni HTML5 pregledač veba, osetljivi su na dodir... primeri su AppleIPad, Samsung Galaxy Tab, BlackBerry PlayBook...

3.1.2 Operativni sistemi i pregledači veba

U mobilnom svetu, operativne sisteme možemo da podelimo u dve grupe :

- **Prepoznatljivi operativni sistemi** (*identifiable operating systems*)
- **Privatni operativni sistemi** (*proprietary operating systems*)

Drugu grupu operativnih sistema imaju prve dve grupe mobilnih uređaja i oni nas ne zanimaju.

Postoji dosta operativnih sistema i pregledača veba dostupnih pametnim telefonima i tabličnim računarima i dati su i sledećoj tabeli :

Operativni sistem	Kreirao	Podrazumevani pregledač veba	Drugi pregledači veba
Ios	Apple	Safari	Opera Mini i pseudo pregledač veba
Andrioid	Google	Android browser	Firefox, Opera Mini, Amazon Silk, Opera Mobile
Symbian	Nokia	Symbian browser	Opera Mini, Opera Mobile
webOS/Open webOS	HP(formalno Palm)	WebOS browser	
Windows Phone	Microsoft	Internet Explorer	Opera Mobile
Windows Mobile	Microsoft	Internet Explorer	Firefox
MeeGo	Nokia	Micro browser/Nokia browser	
BlackBerry OS	RIM	BlackBerry OS browser	Opera Mini
Tablet OS	RIM	Tablet OS browser	
S40	Nokia	Nokia browser	
Bada	Samsung	Samsung browser	

- Tabela 1 -

4. JQuery Mobile

JQuery Mobile je okruženje koje omogućava kreiranje veb aplikacija za mobilne uređaje (pametne telefone i tablične računare). Nastao je avgusta 2010. godine od strane istog tima koji je pravio jQuery okruženje, na čelu sa Džonom Resigom (*John Resig*). JQuery Mobile se, kao i JQuery UI, pojavio kao projekat otvorenog koda (*open source*).

Kompletno JQuery Mobile okruženje se sastoji od :

- jQuery biblioteke (.js datoteka)
- JQuery Mobile biblioteke (.js datoteka)
- JQuery Mobile opisa stilova (.css datoteka)
- skupa slika za korisnički interfejs.

Jquery Mobile biblioteka je izgrađena na bazi jQuery biblioteke. Ona omogućuje kreiranje potrebnih dodataka (trake sa alatima, liste, dugmad...) i primenu osluškivača događaja koji su neophodni da bi mobilna veb aplikacija funkcionisala. JQuery Mobile .css datoteka definiše izgled i prikaz elemenata strana mobilne aplikacije.

4.1 JQuery Mobile, HTML5 i CSS3

Uz pomoć HTML5, CSS3 i drugih modernih tehnologija, sa ili bez JQuery Mobile podrške, mogu se obezbediti stvari poput: pristupa u isključenom režimu, skladištenje u isključenom režimu, geolokacijski pristup, animacije, 2D i 3D transformacije, vizuelni efekti, opcije za zumiranje u mobilnim pregledačima veba, multimedijalna podrška, grafičko crtanje, integrisanje sa matičnim aplikacijama...(Za detaljniji opis pogledati [1]) ¹

JQuery Mobile koristi mnoge HTML5 osobine kako bi omogućio pravi doživljaj aplikacije na mobilnim pretraživačima veba, čak i na onima koji ne podržavaju HTML5 u celosti. Osim toga, koristi i CSS3 kad god je to moguće, za animacije, dodatke, prikaz korisničkog interfejsa i drugih efekata.

¹ U uglastim zagradama se navodi redni broj literature za više informacija

4.1.1 Kreiranje jQuery Mobile aplikacije

Da bismo kreirali jQuery Mobile aplikaciju, prvo je potrebno kreirati HTML5 dokument. Uključivanjem jQuery Mobile okruženja (jQuery biblioteke, jQuery Mobile biblioteke i jQuery Mobile CSS datoteke) u `< head > tag` HTML5 dokumenta, kreirali smo praznu jQuery Mobile aplikaciju. Pomenute datoteke mogu biti uključene na 2 načina :

- postavljanjem svih datoteka u okvir našeg projekta

Potrebno je preuzeti kompresovan paket jQuery Mobile datoteka sa adrese (<http://jquerymobile.com/download>) i datoteku za jQuery sa (<http://jquery.com>). Svaki paket sadrži po dva tipa .js i .css datoteka – sa sufiksom .min (minimizovane, bez razmaka, novih redova i komentara pa su pogodne za mobilne aplikacije) i bez sufiksa (pogodne za izmenu). U trenutku pisanja, jQuery Mobile 1.1.0 zahteva jQuery 1.6.4 ili 1.7.1, zato ne treba uključivati novije verzije jer može doći do nekih nekompatibilnosti.

- upotrebom CDN (*Content Delivery Network*)

jQuery mobile CDN je javni server koji čuva pomenute dokumente za nas. Potrebno je sa adrese <http://jquerymobile.com/download/> iskopirati direktive:

```
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.css" />
<script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>
<script src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js">
</script>
```

Ovaj način ima i svoje prednosti i mane. Prednosti su što možemo bez preuzimanja datoteka da koristimo jQuery Mobile, biće manje datoteka na našem serveru, prednost keš memorije (ukoliko je korisnik posetio neku drugu jQuery Mobile veb aplikaciju sa istim CDN-om, onda će već imati potrebne resurse u kešu), u većini slučajeva će aplikacija raditi brže...

Ukoliko kreiramo matičnu aplikaciju, nije preporučljivo koristiti CDN jer u slučaju gubitka konekcije, nećemo moći ni da kreiramo obaveštenje o grešci.

Takođe je moguće iskopirati direktive koje uvek upućuju na poslednju verziju jQuery Mobile, ali to nije preporučljivo, jer bez testiranja ne znamo kako će se naša aplikacija ponašati :

```
<link href="http://code.jquery.com/mobile/latest/jquery.mobile.min.css"
rel="stylesheet" type="text/css" />
<script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
<script
src="http://code.jquery.com/mobile/latest/jquery.mobile.min.js"></script>
```

Osim jQuery okruženja, u *head* tag-u jQuery Mobile aplikacije se definiše i *viewport*. *Viewport* predstavlja oblast u kojoj je strana vidljiva. Možemo da definišemo širinu i visinu tako da budu veće ili manje od ukupne veličine ekrana. Najčešće želimo da vidljiva oblast bude jednaka veličini ekrana:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Dakle, kôd prazne jQuery Mobilne aplikacije bi izgledao ovako:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Demo preduzeće</title>
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-
1.1.0.min.css" />
    <script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js">
</script>
    <meta name="viewport" content="width=device-width, initial-scale=1">
  </head>
  <body>
  </body>
</html>
```

4.2 Glavne karakteristike jQuery Mobile aplikacije su:

- Prilagođenost za više platformi, više uređaja i više pregledača veba
- Optimizacija korisničkog interfejsa za uređaje osetljive na dodir
- Tematski i prilagodljiv dizajn
- Progresivno proširenje

4.2.1 Progresivno proširenje

Termin "progresivno proširenje" (*progressive enchancement*) je uveo Stiven Čempion (*Steven Champeon*) 2003. godine. To je jednostavna, ali veoma moćna tehnika čija je ideja da se kreira osnovni dokument koji je zasnovan na zajedničkim funkcionalnostima svih pregledača veba, a da se zatim proširuje i poboljšava prikaz i ponašanje strana, koristeći savremene tehnologije kao što su CSS, JavaScript, Flash, Java applets... Sva poboljšanja se nalaze u spoljnim datotekama koje se uključuju u osnovni dokument. Na taj način se odvajaju sadržaj i prikaz strane, što omogućuje da sadržaj strana bude pristupačan svim pregledačima veba.

JQuery Mobile aplikacija funkcioniše upravo na ovakav način pa će se zato prikazivati i na prostom pregledaču veba, bez CSS ili JavaScript podrške. (Za detaljniji opis pogledati [7])

4.3 Kompatibilnost jQuery Mobile i pregledača veba

GBS (*Graded Browser Support*) razdvaja pregledače veba u 3 kategorije:

A-kategorija, B-kategorija, C-kategorija.

A - kategorija

Pregledači veba koji su u najmanju ruku sposobni za medijske upite. Potpuno su testirani od strane jQuery tima. Opcije koje uređaj ne podržava biće automatski onemogućene. Okruženje daje potpuni doživljaj sa animacijama zasnovanim na AJAX-u.

B - kategorija

Poboljšani pregledači veba bez AJAX navigacionih funkcija.

C – kategorija

Pregledači veba koji su nekompatibilni sa jQuery Mobile okruženjem. To znači da korisnik neće primiti CSS ili JavaScript kôd iz okruženja, ali će videti običan HTML sadržaj.

AJAX (Asynchronous JavaScript and XML) je tehnika koja sa klijentske strane omogućuje pretraživaču da pošalje zahtev serverskoj strani bez menjanja URL-a. Na taj način je učitavanje brže jer se učitava samo deo, a ne cela strana sa servera.

Na sledećoj slici je prikazana kompatibilnost jQuery Mobile i pregledača veba :

Mobile browser support [excl]

Platform	Version	Native	Phantom Gap				Opera Mobile				Opera Mini		Fennec		Oszone	Net front						
			0.9	0	0.5, 0.65	0	9.5	0	10	0	4.0	0	5.0	0	1.0	0	1.1*	0	0.9	0	4.0	0
IOS	v2.1	B	A																			
	v3.1.3	A	A								A											
	v3.2	A	A								A											
	v4.0	A	A								A											
Symbian S60	v3.1	C		C			B		C		B							C		C		
	v3.2	A	A	C			A		C		A											
Symbian UIQ	v3.0			C														C				
	v3.1			C														C				
	v3.2					C												C				
Symbian Platform	v3.0	A																				
BlackBerry OS	v4.5	C							C		C											
	v4.6	C	C						C		B											
	v4.7	C	C						C		A											
	v5.0	B	A						C		A											
Android	v1.5	A	A																			
	v1.6	A	A																			
	v2.1	A	A						A'		C'				A'							
	v2.2	A	A						A'		C'				A'							
Windows Mobile	v6.1	C		C	C	C	C	B	C		B										C	
	v6.5.1	C		C	C	C	A	A	C		A											
	v7.0	A					A	C	A													
webOS	1.0.1	A	A																			
bada	1.0	A																				
MeeGo	5.0	B					B						C		B'							
MeeGo	1.1*	A'					A'								A'							

- Slika 1 -

Mnogi popularni desktop pregledači veba, poput *Google Chrom-a*, *Firefox-a*, *Safari-ja*, *Internet Explorer-a* su takođe kompatibilni sa jQuery Mobile okruženjem. Iako jQuery Mobile nije namenjen desktop aplikacijama, ova mogućnost može biti korisna za testiranje.

4.4 Arhitektura i navigacija

JQuery Mobile na jednostavan i moćan način definiše sadržaj veb aplikacije. Osnovna jedinica je **strana** koja nije običan HTML dokument. JQuery Mobile koristi HTML5 *data-** attribute za kreiranje elemenata korisničkog interfejsa, za njihov prikaz i definisanje strukture strana. Tako se svaka strana kreira u okviru *div* taga sa atributom *data-role="page"*. Uobičajeno je da se strana sastoji od zaglavlja (*header*), sadržaja (*content*) i podnožja (*footer*). Svaki spomenuti deo je predstavljen *div* tagom sa odgovarajućim *data-role* atributom.

```
<div data-role="page">
<div data-role="header"> Demo preduzeće d.o.o. </div>
<div data-role="content">
<p> Dobrodošli na našu malu mobilnu veb aplikaciju.
Možete se upoznati sa našim timom. </p>
</div>
<div data-role="footer">
<i> Demo 2012</i>
</div>
```

Glavni sadržaj strane nalazi se u *content* delu. Shodno definiciji *viewport* meta tag-a, sadržaj strane se menja pri orijentaciji samog uređaja.

Veb aplikacije se uglavnom sastoje od više povezanih stranica. Zbog načina definisanja, moguće je kreirati više stranica u okviru istog HTML dokumenta dodajući im *id* attribute pomoću kog im pristupamo. Ono što je zanimljivo je da jQuery Mobile prikazuje samo jednu stranu i to onu prvu na koju naiđe.

Sposobnost da se više strana kreira u okviru istog dokumenta je stara preko 10 godina. Sada zastareo Wireless Markup Language (WML) standard je imao ovu sposobnost. Svaka strana je nazivana karticom (card zbog korišćenja card taga) a dokument u kojem su strane definisane se nazivao špil (desc).

jQuery Mobile je realizovao ovu ideju koristeći HTML i JavaScript, pri čemu se strane definišu u *div* tagu.

Nekada će se veb aplikacija sastojati od velikog broja strana koje će, smeštene u jedan dokument, doprineti da inicijalno otvaranje aplikacije bude sporo. Zato to nije uvek najbolje rešenje. Moguće je i kombinovati ove dve tehnike.

Data-role nisu novi atributi u okviru HTML5, već oni predstavljaju implicitni dogovor između okruženja i nas. Ovakvi atributi se zovu prilagođeni data atributi (*custom data attributes*) i omogućuju nam da definišemo attribute kakvi god da su nam potrebni, a da zadržimo validnost HTML dokumenta.

Role atributi dostupni u jQuery Mobile 1.1.0 prikazani su u sledećoj tabeli.

Role	Opis
page	Definiše stranu
header	Definiše zaglavlje strane
footer	Definiše podnožje strane
content	Definiše sadržaj strane
navbar	Definiše navigacioni bar, često korišćen u podnožju strane
button	Definiše dugme
controlgroup	Definiše komponentu
collapsible	Definiše panel sa nekim sadržajem koji može da se prikaže ili sakrije
collapsible-set	Definiše grupu prethodno spomenutih panela
fieldcontain	Definiše kontejner za polja forme
listview	Definiše listu
Dialog	Definiše dijalog
Slider	Definiše klizač (<i>slider</i>)
Nojs	Definiše element koji će biti sakriven na jQuery Mobile kompatibilnim pregledačima veba

- Tabela 2 -

Za povezivanje strana najčešće se koriste *a* elementi. Možemo napraviti više vrsta veza :

- Unutrašnja veza ka drugoj strani u istom dokumentu.
- Spoljašnja veza ka drugim jQuery Mobile dokumentima.
- Apsolutna veza ka dokumentima koji nisu jQuery Mobile.
- Specijalne veze.

Treba razlikovati povezivanje unutrašnjih i spoljašnjih strana gde unutrašnjim zovemo strane iz istog, a spoljašnje iz drugog domena ili one sa atributom *rel="external"* ili *target="_blank"*.

Unutrašnja veza se ostvaruje pomoću *id* atributa strane.

```
<a href="#id_strane" >druga strana</a>
```

Strani koja nije u istom dokumentu pristupamo isto *a* tagom:

```
<a href="naziv_dokumenta.html" >druga strana</a>
```

Svaki put kad je pozvana veza, jQuery Mobile radi sledeće:

- Proverava da li je veza spoljašnja ili unutrašnja.
- Ukoliko veza nema atribut *target*="blank" ili *rel*="external", onda zahteva URL AJAX-om, tj. neće otvoriti novu stranu već će je učitati u DOM postojeće strane, a suprotnom učitava novu stranu.
- Menja *location.hash* u relativanu adresu nove strane kako bi omogućio obeležavanje i referenciranje.
- Primeni efekat tranzicije: menja klasu *ui-page-active* sa trenutne na sledeću stranu.

Kako ne bi bilo problema pri učitavanju spoljašnje strane, treba paziti na sledeće :

- Strana treba da bude isto jQuery Mobile dokument
- Strana treba da bude u istom domenu
- Dokument treba da sadrži samo jednu stranu
- Ukoliko je url direktorijum, onda *href* treba da se završi na sledeći način:
href="/ime_direktorijuma/
- *Target* atribut ne treba da bude deklarisan

Ukoliko učitavamo spoljašnju stranu koja se nalazi u dokumentu koji sadrži više strana, onda treba da se navede puna putanja. Razlog je što jQuery Mobile renderuje prvu stranu na koju naiđe u dokumentu i pri tome bi potpuno ignorisao *head* tag u kojem definišemo naslov strane i uključujemo razne skriptove i stilove.

Nekada ne želimo da koristimo spoljašnje stranice jer želimo da se strana prikaže odmah kada korisnik izabere tu opciju, a sa druge strane želimo da smanjimo broj AJAX poziva. Tada uhvatimo (dodajemo, engl. *fetch*) stranu u DOM na sledeći način: `dodata_str`.

Nakon što se strana sakrije, jQuery Mobile briše iz DOM-a spoljašnju stranu. Da se to ne bi dogodilo, potrebno je dodati još jedan atribut:

`dodata_str`.

Treba paziti da se ova opcija koristi kada smo zaista sigurni da će ta strana biti često prikazivana.

Primeri apsolutne spoljašnje veze :

```
<a href="http://www.novastrana.com" data-rel="external">nova strana</a>  
<a href="http://www.novastrana.com" target="_blank">nova strana</a>  
<a href="novastrana.html" data-ajax="false">nova strana</a>
```

Specijalne veze se odnose na veze sa mobilnim uređajem. To uključuje slanje poruka i pozive ukoliko se radi o mobilnom telefonu. U zavisnosti od izabrane veze, koriste se funkcionalnosti mobilnog telefona, tako da se pritiskom dugmeta za poziv otvara numerička tastatura, pritiskom dugmeta za sms se otvara uređivač teksta, pritiskom dugmeta za skajp se aktivira skajp nalog itd:

```
<a href="tel:+381605775872" data-role="button" name="tel">Telefon:  
+381605775872</a>  
<a href="sms:+381605775872" data-role="button">SMS</a>  
<a href="mailto:anchejk@gmail.com?subject=Zdravo%20from%20the%20web" data-  
role="button">Mail : anchejk@gmail.com</a>  
<a href="skype:ana?call" data-role="button">Skype</a>  
<a href="http://www.elastyc.net/demo/" data-role="button"  
target="_blank">www.elastyc.net/demo</a>
```



- Slika 2 -

4.4.1 Dijalozi

Dijalozi nisu nova stvar, već samo drugačiji način prikaza strane. To su mali prozori koji se prikazuju preko trenutne strane i pogodni su za prikaz nekog objašnjenja, pitanja i slično. Definišu se u okviru *div* elementa kome je dodat atribut *data-rel="dialog"*. Osim toga, već postojeće strane možemo da prikažemo kao dijalog tako što *a* tagu, koji definiše vezu, dodamo već pomenuti atribut *data-rel="dialog"*.

Glavne razlike između normalne strane i dijaloga :

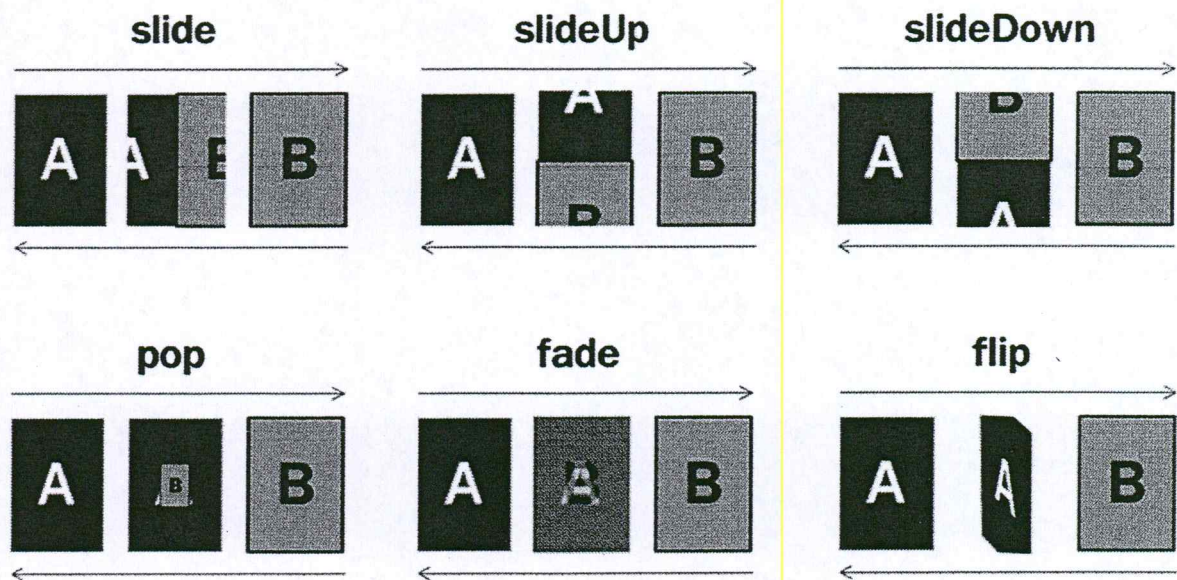
- Dijalog ima u gornjem levom uglu dugme za zatvaranje, na istom mestu gde je dugme za povratak ukoliko je definisan na običnoj strani
- Dijalog se otvara u iskačućem prozoru (engl. pop-up), a ne preko celog ekrana
- Dijalog se ne čuva u navigacionom steku kao nova strana.

4.4.2 Tranzicija između strana

Pod tranzicijom između strana podrazumevamo način prikaza stranica kada korisnik prelazi sa jedne na drugu. Za definisanje koristi se atribut *data-transition* koji može uzeti vrednosti :

- *slide* – prikaz strane sa desna 0u levo
- *slideup* – prikaz strane odozdo nagore
- *slidedown* – prikaz strane odozgo nadole
- *pop* – prikaz strane koja raste iz male tačke na sredini ekrana
- *fade* – prikaz od zamućene do jasne slike
- *flip* – strana koja se okreće

Grafički prikaza je dat sledećom slikom.



- Slika 3 -

Tranzicija se odnosi na veze, ne na strane, što znači da prikaz iste strane može biti različit ukoliko postoji više veza koje vode do nje.

Ukoliko želimo povratak na prethodnu stranu na isti način, ali u obrnutom smeru, dodajemo atribut `data-direction="reverse"`.

4.5 Elementi strana

jQuery Mobile ima veliki skup komponenti korisničkog interfejsa koje se mogu koristiti u veb aplikacijama :

- Traka sa alatima (*toolbar*)
- Komponente za formatiranje (*formatting components*)
- Dugmad (*buttons*)
- Liste (*lists*)
- Forme (*forms*)

Traka sa alatima je traka sa dugmadima, nekim tekstom, vezama... Postoje dva standardna tipa: zaglavlje (*header*) i podnožje (*footer*) čije je kreiranje već pomenuto. Zaglavlje se nalazi na početku svake strane i najčešće sadrži naslov i, zbog prostora, do dva dugmeta koja se predstavljaju a tagom. Podnožjem se obično završava strana i u njemu imamo veću slobodu postavljanja elemenata, ali najčešće

sadrži kombinaciju teksta i dugmadi. Dugmad u zaglavlju možemo da pozicioniramo pridružujući im klase „*ui-btn-right*“ za prikaz na desnoj ili „*ui-btnleft*“ za prikaz na levoj strani zaglavlja, mada ukoliko se klase ne navedu, prvo dugme će biti po definiciji postavljeno sa leve, a drugo sa desne strane. Uobičajeno je da se na levoj strani postavlja dugmad sa negativnim akcijama (*cancel, back, no, exit, logout*) a sa desne sa pozitivnim (*done, save, yes, ok, send*). Postoje definisane ikonice koje se pridružuju dodavanjem atributa *data-icon* a njegova vrednost zavisi od namene dugmeta. Umesto naslova, moguće je postaviti logo u zaglavlju i to uobčajenim dodavanjem *img* taga. U zavisnosti od veličine logoa i zaglavlje će se proširiti, ali nije preporučljivo da slika bude veća od 125px. Ukoliko ne želimo standardno zaglavlje, kakvo je do sada opisano, moguće je definisati sadržaj koji želimo u posebnom bloku (obično *div* bloku) koji se uključi u zaglavlje, ali tada je potrebno ručno definisati stil.

```
<div data-role="header">  
<h1> Demo preduzeće d.o.o. </h1>  
</div>
```

```
<div data-role="header">  
<div>  
<h1> Demo preduzeće d.o.o. </h1>  
</div>  
</div>
```

Takođe je moguće postaviti i dugme za povratak. Neki uređaji, kao sto su npr. telefoni sa Android operativnim sistemom, imaju fizičko dugme za povratak, i ono će automatski raditi u našoj aplikaciji. Međutim, neki, npr. iPhone uređaji, nemaju. Zato je poželjno svakako kreirati dugme za povratak. To se može uraditi na dva načina. Prvi je dodavanje atributa *data-add-back-btn="true"* strani, što će automatski dodati dugme za povratak u header. Da bismo promenili tekst na dugmetu, koristimo *data-back-btn-text="tekst"* . Drugi način je dodavanje *a* elementa u *content* div-u sa atributom *data-rel="back"*. Moguće je postaviti i dugme za povratak globalno, a njegov tekst promeniti JavaScript-om, ali o tome će biti reči kasnije.

Svaki *a* element u podnožju će biti predstavljen kao dugme, pri čemu se oni ne raspoređuju levo i desno kao u zaglavlju, već se ređaju redom i nije im ograničen broj. Kako bi bili definisani razmaci između dugmadi i ivica trake sa alatima, podnožju se dodeljuje klasa „*ui-bar*“.

Pozicioniranje trake sa alatima, bilo da se radi o zaglavlju ili podnožju, može da se izvrši u četiri režima rada:

- Redni režim (*inline mode*)
- Standardni fiksirani režim (*standard fixed mode*)

- Puni ekran-fiksni režim (*full screen-fixed mode*)
- Tačno fiksirani režim (*true fixed mode*)

Redni režim je podrazumevan. To znači da će zaglavlje biti vidljivo samo na vrhu strane, a podnožje na dnu i ukoliko je strana duža od visine ekrana do njih bi se dolazilo listanjem stranice. Nekada je zgodno da su nam zaglavlje i podnožje stalno vidljivi pa se zato koriste drugi režimi.

Fiksirani režim omogućuje da zaglavlje i podnožje uvek budu na vrhu, odnosno na dnu, i da se gube dok listamo stranicu. Kada korisnik prestane sa listanjem, zaglavlje i podnožje se ponovo prikazuju na vrhu i dnu strane. Fiksiranje trake sa alatima se vrši dodavanjem atributa *data-position="fixed"*. Pritiskom na neaktivni deo strane u *content* div-u moguće je izvršiti promenu iz radnog u fiksni režim i obrnuto.

Puni ekran-fiksni režim je fiksni režim u kojem se pritiskom na neaktivni deo strane zaglavlje i podnožje sakrivaju i ponovnim pritiskom pojavljuju. Ovaj režim je pogodan kada gledamo neku sliku ili neki dugačak tekst ili formu. Da bismo ga koristili, potrebno je traci sa alatima definisati atribut *data-position="fixed"* i *data-fullscreen="true"*.

Tačno fiksirani režim se odnosi na mobilne pregledače veba koji dopuštaju definisanje *position:fixed* i *overflow:auto* CSS-om. Po definiciji, u verziji 1.0 je onemogućeno postaviti ovaj stil, ali to se može omogućiti JavaScriptom *\$.mobile.touchOverflowEnabled="true"*.

Osim dodavanja dugmića u zaglavlje i podnožje, jQuery Mobile nudi čistije rešenje. Radi se o tzv. **navigacionoj tabli** (*navigation bar, navbar*). Ona se sastoji od neuređene liste koja je ugnježdjena u div sa atributom *data-role="navbar"*. Omogućeno je izdvajanje aktivnog dugmeta dodavanjem klase *"ui-btn-active"* na prvo polje. Do pet elemenata jQuery Mobile lepo raspoređuje u jedan red i podešava im širinu automatski zauzimajući celu širinu strane. Ako imamo više, prebaciće ih u novi red, ali generalno se smatra nepreglednim dodavati više od 5 polja. Ukoliko postoji tekst pre *navbara*-a, cela navigaciona tabla će biti smeštena u novi red. Ideja je da se *navbar* koristi kao osnovni metod navigacije naše veb aplikacije. Kako bi se izbegle UI greške, nije preporučljivo da se koristi za dugmad sa akcijom (*save, cancel...*). (Za detaljniji opis videti [2])

Primer :

```
<div data-role="navbar">
  <ul>
    <li><a href="#index" data-icon="home" ></a></li>
    <li><a href="#zaposleni" data-icon="myapp-zap" ></a></li>
    <li><a href="#kontakti" data-rel="dialog" class="ui-btn-active ui-state-persist" data-icon="myapp-kont"></a></li>
  </ul>
</div>
```

Ukoliko nam se veb aplikacija sastoji iz više strana, ono što svakako treba da važi je:

- Podnožje mora biti prisutno na svim stranama
- Podnožje mora imati isti *data-id* na svim stranama
- Moramo koristiti dve klase *ui-state-persist* i *ui-btn-active* na aktivnim stranama iz navigacione tabele

4.5.1 Sadržaj za formatiranje

Glavni sadržaj svake strane je smešten u *content* delu. Osnovno što treba znati je da će svaki HTML kôd biti prikazan unutar `<div data-role="content">`. Osim osnovnih elemenata HTML-a, postoje i neke komponente koje su definisane *div* tagovima sa *data-role* atributima, kao što su npr. zaglavlje i podnožje. U okviru njih smo već primetili da se određeni elementi prikazuju na poseban način (npr. a tagovi kao dugmad). To nije u potpunosti slučaj sa *content* komponentom. U okviru svake teme u jQuery Mobile (o čemu će biti reči malo kasnije) postoji definisana podešavanja za neke elemente (margine, udaljenost od ivica, veličina, boje...). Ukoliko želimo da definišemo svoj stil, treba voditi računa o temama koje po definiciji postoje u jQuery Mobile-u.

4.5.2 Sadržaj koji se sažima

Kako govorimo o mobilnim veb aplikacijama, prostor nam je jako važan. Ovaj dodatak predstavlja sadržaj koji može da se sažme i da se raširi. Pravi se lako, ubacujući labelu i sadržaj koji želimo da prikažemo ili sakrijemo u *div* sa atributom *data-role="collapsible"*. Labela će uvek biti prikazana i pored nje sa leve strane

imaćemo ikonicu + za otvaranje skrivenog sadržaja ili – za sakrivanje, i za sada nije moguće promeniti ove ikonice. Ikonice mogu da se pomere na desnu stranu dodavanjem *data-iconpos="right"*. Po definiciji sadržaj je prvo sakriven, ali može se postaviti da bude inicijalno prikazan dodavanjem atributa *data-collapsed="false"*.

Primer :

```
<div data-role="content">
  <div data-role="collapsible">
    <h2>Prvi blok</h2>
    <p>Prvi blok je zatvoren</p>
  </div>
  <div data-role="collapsible" data-collapsed="false">
    <h2>Drugi blok</h2>
    <p>Drugi blok je otvoren</p>
  </div>
</div>
```

Moguće je napraviti ugnježdene blokove koji se sažimaju. Nije preporučljivo praviti više od dva podnivoa kako bi se izbegla složenost korisničkog interfejsa i DOM-a. Ono što je posebno zanimljivo je da se može napraviti region sa više ovakvih komponenti koje se grupišu i imaju osobinu da je uvek samo jedan sadržaj prikazan i da se prikazivanjem novog prethodni sadržaj sakriva. Ovakav prikaz se još naziva harmonika prikaz (*accordion*). Ovo se omogućuje postavljanjem svih *divova* sa atributom *data-role="collapsible"* u *div* sa atributom *data-role="collapsible-set"*. Po definiciji otvoren je poslednji blok koji je u setu pomenutih blokova. Ukoliko želimo drugačiji raspored, panelu koji želimo da bude otvoren dodeljujemo atribut *data-collapsible="false"*, a ostalima sa vrednošću „*true*“.

Ugnježdjeni sadržaji koji se sažimaju nisu mogući ako su grupisani jer pri zatvaranju blokova jQuery Mobile sve otvorene blokove posmatra na isti način pa dolazi do poklapanja.

4.5.3 Rešetka (*grid*)

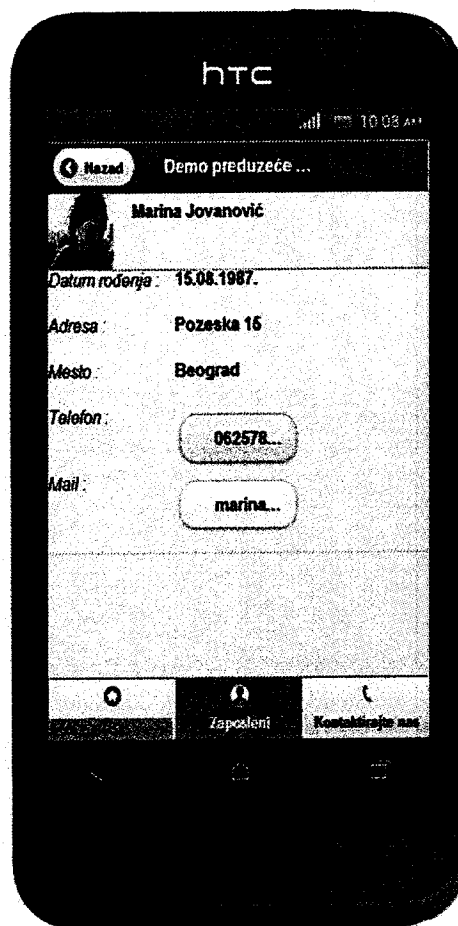
Rešetka predstavlja formu koja ne koristi *data-rel* attribute, već se generiše koristeći CSS. Zato se koriste klase *ui-grid-x* gde se *x* zamenjuje sa a,b,c,d ukoliko želimo redom 2,3,4,5 kolona, mada nije poželjno koristiti rešetke sa 5 kolona za

aplikacije na telefonima zbog veličine ekrana. Svaka ćelija može imati svoju klasu *ui-block-x*, pri čemu x menjamo sa a,b,c,d na već poznat način.

Treba stalno imati na umu da govorimo o mobilnim uređajima i da je prostor jako bitan, zato u kolone rešetke treba da stavljamo male elemente kao što su dugmad, veze...

Za kreiranje višelinijске rešetke samo je potrebno ponoviti blokove:

```
<div data-role="content">
<section class="ui-grid-b" >
<div class="ui-block-a" style="height:40px"><i>Datum rođenja :</i></div>
<div class="ui-block-b" style="height:40px"><b>15.08.1987.</b></div>
<div class="ui-block-a" style="height:40px"><i>Adresa :</i></div>
<div class="ui-block-b" style="height:40px"><b>Pozeska 15</b></div>
<div class="ui-block-a" style="height:40px"><i>Mesto :</i></div>
<div class="ui-block-b" style="height:40px"><b>Beograd</b></div>
<div class="ui-block-a"><i>Telefon :</i></div>
<div class="ui-block-b"><a href="tel:06257844545" data-role="button">
06257844545</a></div>
<div class="ui-block-a"><i>Mail :</i></div>
<div class="ui-block-b"><a href="mailto:marina@yahoo.com" data-role="button">
marina@yahoo.com</a></div>
</section>
</div>
```



- Slika 4 -

4.5.4 Dugmad

Već smo videli da svaki *a* element može da posluži za kreiranje veza među stranama. Treba imati na umu da je u tom slučaju veza samo tekst *a* elementa, što i nije mnogo praktično za mobilne uređaje. Zato se koriste dugmad koja zauzimaju određen prostor na strani sa nekim tekstom ili opciono sa nekom ikonicom. Dugme može da se kreira na više načina :

- Koristeći *button* element
- Koristeći *input* element sa atributom *type* koji može da uzme vrednosti: *button*, *submit*, *reset*, *image*
- Koristeći svaki element sa atributom *data-role="button"*.

Po definiciji dugme zauzima celu širinu ekrana. Ukoliko želimo da postavimo više njih u jednom redu, potrebno je dodati atribut *data-inline="true"*.

Primer :

```
<a href="#" data-role="button" data-inline="true">Dugme 1</a>
```

`Dugme 2`

`Dugme 3`

Ovim nećemo postići da sva tri dugmeta zajedno popune celu širinu ekrana, već svako dugme zauzima prostor tačno onoliki koliki mu je potreban, tj. kolika mu je dužina teksta. Da bismo ravnomerno raspodelili dugmad u jednom redu, možemo da koristimo rešetku.

Ukoliko imamo dugmad koja su na neki način povezana, možemo da ih grupišemo u jedan kontejner. To se zove grupna kontrola. Kreira se postavljanjem elemenata u jedan div sa atributom `data-role="controlgroup"`. Po definiciji elementi će biti vertikalno raspoređeni. Ukoliko želimo horizontalni prikaz, dovoljno je da dodamo atribut `data-type="horizontal"`.

Podrazumevani prikaz dugmeta je sa zaobljenim ivicama i sa senkom. Moguće je promeniti ove opcije dodavanjem atributa `data-corners` i `data-shadow` sa vrednostima `true` ili `false`.

Osim velikog broja tema, (o kojima će biti reči nešto kasnije), jQuery Mobile sadrži i veliki izbor ikonica za stilizovanje dugmadi. Ikonica se definiše atributom `data-icon` koji uzima kao vrednost ime ikonice. Svaka ikonica je slika i postoji skup predefinisanih koje su već uključene u okruženje, a mogu se i ručno definisati nove.

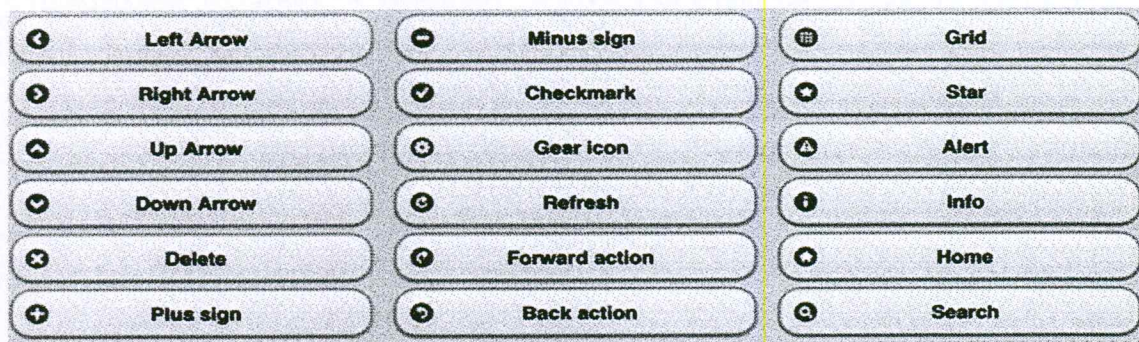
Tabela predefinisanih ikonica :

Naziv	Vrednost
Left arrow	arrow-l
Right arrow	arrow-r
Up arrow	arrow-u
Down arrow	arrow-d
Delete (x)	delete
Plus sign	plus
Minus sign	minus
Checkmark	check
Gear icon (settings)	gear
Refresh	refresh
Forward action	forward
Back action	back
Grid	grid

Star	star
Alert (warning icon)	alert
Info (i)	info
Home icon	home
Search icon	search

- Tabela 3 -

Grafički prikaz :



- Slika 5 -

Kada samostalno definišemo ikonicu, potrebno je zadati joj jedinstveno ime i dodeliti ga atributu *data-icon*. Takođe je potrebno definisati klasu *ui-icon-ime* i odrediti pozadinsku sliku. Kako bi se izbegli problemi, ikonica treba da bude na beloj i transparentnoj pozadini, veličine ne veće od 18x18px.

Na uređajima sa visokom rezolucijom (iPhone4 i iPod 4G) potrebna je i ikonica veće rezolucije. Moguće je istom slikom i CSS-om rešiti problem, ali se savetuje da se koriste dve slike i dve definicije CSS3 korišćenjem *media* upita. Primer:

```
<style>
.ui-icon-myapp-zap {
background-image: url(icons/zap.jpg);
}
@media only screen and (-webkit-min-device-pixel-ratio: 2) {
.ui-icon-myapp-zap {
background-image: url(icons/zap.jpg) !important;
background-size: 18px 18px;
}
}
</style>
```

Pozicioniranje ikonice se takođe može podesiti. Podrazumeva se poravnanje sa leve strane, ali se dodavanjem atributa *data-iconpos* može postaviti na levo, desno, na vrh, na dno (*left, right, top, bottom*).

4.5.5 Liste

Lista u jQuery Mobile je uređena (*ol* Html element) ili neuređena (*ul* html element) lista sa barem jednim elementom (*li* html element) smeštena u *div* element sa atributom *data-role="listview"*.

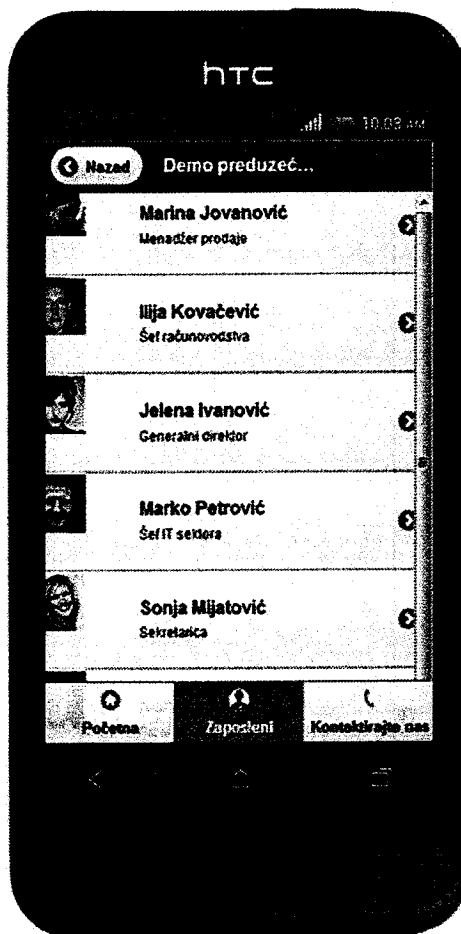
jQuery Mobile prikazuje listu tako da zauzme celu stranu i optimizovana je za dodir. Glavna odlika listi je što mogu imati interaktivne elemente. Ubacivanjem a elementa u *li* element liste, ceo red liste postaje aktivan na dodir, bez obzira gde je postavljen, tj. nije potrebno dodati tekst a elementa već je celo polje veza. Kreiranjem interaktivnih elemenata menja se i izgled liste, polja dobijaju veću visinu. Ukoliko tekst nekog reda ne staje u jedan red, tekst će biti skraćen i dodaju mu se tri tačke ... Osim toga, svaki red dobija i ikonicu sa strelicom na desnoj strani koja može da se promeni kao što je već pomenuto, promenom vrednosti atributa *data-icon*.

Ukoliko želimo na strani još neki sadržaj osim liste, tada listi dodajemo atribut *data-inset="true"* čime se dobija i elegantniji izgled liste sa udaljenošću od ivice, zaobljenim okvirom... Moguće je grupisati elemente liste po određenim svojstvima i odvojiti ih separatorima. To postizemo dodavanjem uobičajenih *li* elemenata listi sa posebnom ulogom definisanom atributom *data-role="list-divider"*. Moguće su i ugnježdene liste koje se kreiraju postavljanjem *ul* elemenata u okviru *li* elementa liste. Pri učitavanju strane sa listom, vidi se samo prvi nivo, a svaki podnivo se prikazuje na novoj strani. Nije ograničen broj podnivoa, ali je preporučljivo ako postoji veliki broj stranica da se definišu u više datoteka kako bi se smanjio DOM i inicijalno učitavanje. U ugnježdenim listama, jQuery Mobile će koristiti tekst iz reda kao naslov nove strane, tako da treba voditi računa da se navede i da ne bude predugačak.

Svakom redu možemo da dodamo sliku koja bi ga grafički definisala i to na dva načina, kao umanjeni prikaz ili ikonicu (koju ne treba mešati sa onom za interaktivni red sa desne strane ili za podeljene redove). Za dodavanje slika dovoljan je *img* tag unutar veze i jQuery Mobile će sam lepo rasporediti sliku levo, a tekst desno od njega. Što se ikonice tiče, potrebno je dodati klasu *img* tagovima *ui-li-icon*. Iako će jQuery Mobile sam napraviti ikonice, poželjno je da sličice budu što manje

(16x16px) kako bi se što pre učitale i kako bi aplikacija bila brža. Obično se koriste za definisanje akcije reda (*edit, delete, save...*). Moguće je podeliti tekst u dve kolone tako što ćemo dodati bilo koji html element sa klasom *ui-li-aside*, kao što su span ili div. To može biti neki tekst, oblačić sa brojem u kojem možemo da prikažemo npr. broj interaktivnih redova u podnivou i slično....

Nekada liste mogu biti predugačke što korisniku otežava da dođe da stavke koja je njemu potrebna, pa je zato potrebna neka pretraga listi. Interesantno je da se u jQuery Mobile—u ovaj problem rešava dodavanjem atributa listi, tj. *ul* tagu i to *data-filter="true"* . Pretraživač će automatski biti dodan na početku liste, sa ikonicom lupe sa leve strane i dugmetom sa desne za čišćenje pretraživača. Pretraživač je neosetljiv na velika i mala slova. (Za detaljniji opis pogledati [4])



- Slika 6 -

4.6 Forme I jQuery Mobile

Pod standardnom veb formom podrazumevamo poseban prozor sa grupom kontrola kao što su *input*, *textarea*, *select* elementi – sa akcionim atributom koji je definisan kao URL i koji dobija podatke forme.

Sve kontrole forme su poboljšane i prilagođene tako da bolje izgledaju na mobilnom uređaju i da budu osetljive na dodir. Opisuju se u aplikaciji na sledeći način:

- Dugmad, korišćenjem *button* i *input* elemenata
- Tekst polja, korišćenjem *input* i *textarea* elemenata
- Kontrolna i radio dugmad korišćenjem *input* elemenata
- Meniji korišćenjem *select* i *option* elemenata
- Slajderi korišćenjem nove kontrole *input* sa atributom *type="range"*

Jako je bitno da svaka kontrola ima jedinstveni *id* atribut jer ukoliko koristimo AJAX, sve strane će biti u istom DOM-u pa samim tim i kontrole forme.

Labele su jako bitne za svaku kontrolu jer se dodiranjem labele daje fokus na kontrolu, što je pogodno za mobilne uređaje jer se tako povećava površina koju sama kontrola zauzima.

Preporučljivo je da svaka forma bude ugnježdjena u `<div data-role="fieldcontain"></div>`. To omogućuje da jQuery Mobile uskladi labelu i polje forme, npr. u zavisnosti od širine ekrana, labela stoji iznad ili pored *input* polja.

Postojeća tekst polja su :

- `<input type="text">`
- `<input type="password">`
- `<input type="email">`
- `<input type="tel">`
- `<input type="url">`
- `<input type="search">`
- `<input type="number">`
- `<textarea>`

Svaki *input* element tipa *button*, *submit*, *clear* i *image* biće prikazan kao dugme. *Email*, *tel*, *url*, *search* su novi tipovi *input* polja u HTML5 i veoma su bitni za mobilne uređaje. Znamo da većina uređaja sa ekranima na dodir nema fizičku tastaturu, pa

se definisanjem ovih tipova, dodiranjem input polja, u zavisnosti od njegovog tipa, pojavljuje i tip virtuelne tastature.

Dobar primer poboljšanja forme je i *textarea* polje koje može biti nepregledno na mobilnom uređaju. JQuery Mobile omogućuje prikaz forme koja automatski raste upisivanjem teksta u nju. Tako se mnogo lakše čita nego kad bi se uključivala opcija listanja po formi. Što se tiče *input* polja tipa *password*, postoji polemika da li ga treba i na koji način koristiti. Standardno je da se umesto karaktera prikazuju kružići ili zvezdice kao način zaštite lozinke od nekoga ko, na primer, stoji pored korisnika. Međutim, zbog male veličine mobilnih uređaja, to i nije baš mnogo verovatno. Još jedna loša strana je to što ne-qwerty tastature mogu biti nezgodne jer nekada nismo sigurni koji karakter smo upisali.

Novi HTML5 atributi su bezbedni za korišćenje jer neće javiti grešku na nekompatibilnim uređajima ili na starijim pregledačima veba, prosto neće raditi ništa. Tu spadaju *required* (obavezno polje), *placeholder* (tekst unutar polja), *pattern*, *min* i *max* (korisni samo za *input* polje tipa *number*). Koristeći nove CSS3 pseudoklase možemo da kreiramo stilove za obavezna, validna, nevalidna, opciona polja...bez korišćenja JavaScript koda.

HTML5 podržava polja datuma kao *input* elemente tipa :

- *Date* – dan,mesec, godina
- *Datetime* – dan, mesec, godina, sati, minuta(uključujući GMT zonu)
- *Time* – sati,minuta
- *Datetime-local* – kompletan datum bez GMT zone
- *Month* – padajuća lista meseca
- *Week* – padajuća lista rednih brojeva nedelja

Ukoliko ne prihvati tip *input* polja, prikazaće ga kao obično tekst polje.

Numeričke vrednosti je najlakše predstaviti klizačem (*slider*), (*input type="range"*) kome možemo da definišemo *min*, *max* i *step* attribute koji bi predstavljali najmanju, najveću vrednost i korak pri pomeranju slajdera. Kada je komponenta u fokusu, moguće je upravljati njom tastaturom.

4.6.1 Radio dugmad (*radio buttons*) i polja za potvrđivanje (*check buttons*)

Dobra strana radio dugmadi je što se prikazuju automatski, bez nekog velikog posla za nas. Jedino što je potrebno je :

- Svaka opcija mora biti `<input type="radio">`
- Svaka opcija u istoj grupi mora imati istu vrednost za atribut *name*
- Svaka opcija mora imati jedinstven *id* i labelu

Labela za celu grupu radio dugmadi definiše se *legend* HTML elementom. Moguće je grupisanje, postavljanje grupe horizontalno ili vertikalno ...

Polja za potvrđivanje se koriste na sličan način kao i radio dugmad, samo što dozvoljavaju selektovanje više vrednosti ukoliko su grupisana.

4.7 JavaScript I jQuery Mobile

jQuery Mobile omogućuje JavaScript API za komunikaciju sa okruženjem (HTML5 okruženje) i omogućuje promenu sadržaja korišćenjem JavaScript-a, pri čemu treba biti oprezan jer može doći do nekompatibilnosti sa nekim pregledačima ili starijim platformama.

jQuery Mobile već sam radi dosta toga za nas, ali naravno, nekad će biti potrebno da nešto ne uradi ili da uradi nešto dodatno i za to nam služi JavaScript. Oblikovanje jQuery Mobile veb aplikacije počinjemo pisanjem koda za osluškiivač događaja *mobileinit*, koji se mora izvršiti nakon učitavanja jQuery biblioteke, ali pre nego što se učita jQuery Mobile. Za oblikovanje jQuery Mobile veb aplikacije, koristimo *\$.mobile* objekat i to možemo odraditi na dva načina koja su podjednako ispravna:

- ```
$(document).bind("mobileinit", function(){
 $.mobile.nekopodešavanje="neka vrednost";
});
```
- ```
$(document).bind("mobileinit", function(){
    $.extend($.mobile, {
        Nekopodešavanje:"neka vrednost"
    });
});
```

4.7.1 Konfiguracija

Svim dodacima (*widgets*) moguće je promeniti predefinisne vrednosti korišćenjem `$.mobile./dodatak/.prototype.options` objekata.

Tabela podešavanja za svaki dodatak :

Dodatak	Podešavanje	Vrednost (predefinisana)	Opis
Svi	theme	od a do z	Boja uzorka koja se primenjuje na svaku instancu dodatka
listview	filter	true/(false)	dozvoljava pretragu liste
listview	filterPlaceHolder	od a do z	placeholder u filter tekst polju
listview	filterTheme	(top)/bottom/left/right	tema filter tekst polja
navbar	iconpos	od a do z	pozicija ikonice stavki u navbaru
slider	trackTheme	icon value (arrow-d)	tema trake klizača
selectmenu	icon	top/bottom/left/(right)	ikonica za otvaranje selekt menija
selectmenu	iconpos	(true)/false	pozicija ikonice
selectmenu	corners	(true)/false	zaobljeni dugmići dugmeta za otvaranje
selectmenu	shadow	(true)/false	senka dugmeta za otvaranje
selectmenu	iconshadow	od a do z (b)	senka ikonice dugmeta za otvaranje
selectmenu	closeText	(true)/false	tekst za dugme za zatvaranje menija
selectmenu	nativeManu	niska	definiše da li će selekt meni biti matični ili neće

dialog	closeBtnText	niska	tekst dugmeta za zatvaranje dijaloga
dialog	overlayTheme	od a do z (a)	tema za dijalog
collapsible	expandCueText	string	tekst dugmeta za otvaranje sadržaja
collapsible	collapseCueText	string	tekst dugmeta za zatvaranje sadržaja
collapsible	collapsed	(true)/false	definiše da li blok treba da bude zatvoren ili otvoren po definiciji
collapsible	heading	vrednosti odvojene zarezom	lista tagova koji će biti korišćeni kao zaglavlje ovog bloka
collapsible	contentTheme	od a do z	tema za sadržaj
collapsible	iconTheme	od a do z	tema za ikonicu u zaglavlju
button	icon	icon value	ikonica po definiciji za svako dugme
button	iconpos	icon position	pozicija ikonice za svako dugme
button	inline	true/(false)	definiše da li svako dugme treba da bude i redu ili ne
button	corners	(true)/false	definiše zaobljenost ćoškova na dugmićima
button	shadow	(true)/false	definiše senku za svako dugme
button	iconshadow	(true)/false	definiše senku za svaku ikonicu dugmeta

- Tabela 5 -

Promenom *prototype* opcija svake strane menjamo predefinisana podešavanja. Na primer, ukoliko želimo na svakoj strani dugme za povratak,

potrebno je da postavimo `$.mobile.page.prototype.options.addBackBtn="true"`. Njegov tekst i stil menjamo sa `$.mobile.page.prototype.options.backBtnText` i `$.mobile.page.prototype.options.backBtnTheme`. Na isti način možemo da promenimo stil zaglavlja, podnožja i sadržaja koristeći `headerTheme`, `footerTheme`, `contentTheme`.

Opcije globalne konfiguracije koje se definišu u okviru `mobileinit` događaja prikazane su u sledećoj tabeli.

Podešavanje	Upotreba
ns	Koristi se ako želimo da promenimo naziv data atributima, npr. želimo da ubacimo prefiks jqm u data-role atribut i tada bismo dobili data-jqm-role="page".
activeBtnClass	Podešava ime klase koja se koristi za aktivnu dugmad. Po definiciji je to <i>ui-btn-active</i> .
activePageClass	Podešava ime klase strane na kojoj smo trenutno. Po definiciji je to <i>ui-page-active</i> .
ajaxFormsEnabled	Omogućava ili onemogućava obradu formi AJAX-om. Po defijiciji je <i>true</i> .
ajaxLinksEnabled	Omogućava ili onemogućava rukovanje vezama AJAX-om. Po defijiciji je <i>true</i> .
ajaxEnabled	Omogućava ili onemogućava asinhronne funkcije. Po defijiciji je <i>true</i> .
allowCrossDomainPages	Omogućava zabranu ili dozvolu udaljenim stranama da budu učitane sa <i>\$.mobile.loadPage</i> . Obično se zahteva samo za PhoneGap aplikacije koje učitavaju sadržaj sa drugog servera.
nonHistorySelectors	Veze sa ovom vrednošću atributa data-rel ili strane sa ovom vrednošću atributa data-role neće biti moguće pratiti u istoriji (neće menjati <i>location.hash</i> pa samim tim neće biti zabeležene). Vrednost po definiciji je <i>"dialog"</i> .
metaViewportContent	Podešavanje vrednosti atributa <i>meta viewport</i> taga. Po definiciji je <i>"width=device-width, minimum-scale=1, maximum-scale=1"</i> . Ukoliko se postavi na <i>false</i> , ni jedan meta tag neće biti uključen u DOM.
autoInitializePage	Po definiciji jQuery Mobile inicijalno prikazuje prvu stranu u dokumentu nakon što je DOM spreman. Ukoliko to ne želimo postavljamo <i>autoInitializePage = "false"</i> , ali tada će biti potrebno ručno postaviti inicijalnu stranu koristeći <i>\$.mobile.initializePage</i> .
defaultDialogTransition	Označava na koji način će se prikazivati i sklanjati

	dijalozi. Po definiciji mu je vrednost <i>pop</i> , a može biti i <i>slideup</i> , <i>slidedown</i> , <i>fade</i> , <i>flip</i> , <i>slide</i> .
<i>defaultPageTransition</i>	Isto važi kao i za prethodno podešavanje iskorišćeno za strane. Razlika je i u tome što je po definiciji vrednost <i>slide</i> .
<i>hashListeningEnabled</i>	Odnosi se na sposobnost osluškivanja promena <i>location.hash</i> podešavanja pretraživača. Po definiciji vrednost je <i>true</i> , a ako se vrednost postavi na <i>false</i> , onda je potrebno ručno napisati kod koji prati promene.
<i>loadingMessage</i>	Predstavlja podešavanje za poruku prilikom učitavanje stranice. Inicijalno je „ <i>loading...</i> “, ali je i inicijalno tekst poruke sakriven (pogledati sledeće podešavanje).
<i>loadingMessageTextVisible</i>	Po definiciji je postavljeno da se tekst pri učitavanju strane ne vidi, tj. vrednost mu je <i>false</i> , a možemo ga postaviti na <i>true</i> .
<i>loadingMessageTheme</i>	Podešavanje teme dijaloga na kojem se nalazi poruka pri učitavanju strane. Po definiciji je <i>a</i> .
<i>minScrollBack</i>	JQuery Mobile će se truditi da upamti poziciju na strani i pri povratku na istu da vrati.
<i>pageLoadErrorMessage</i>	Poruka koja se javlja ukoliko dođe do greške pri učitavanju stranice. Po definiciji je „ <i>Error Loading Page</i> “.
<i>pageLoadErrorMessageTheme</i>	Tema poruke o greški. Po definiciji je <i>e</i> .

- Tabela 6 -

Pri učitavanju spoljašnje stranice AJAX-om, koriste se neki atributi po definiciji. Do njih možemo doći objektom `$.mobile.loadPage.defaults` čija su podešavanja data u sledećoj tabeli.

podešavanje	vrednosti	vrednost po definiciji	opis
type	"get"/"post"	"get"	definiše tip Ajax poziva
data	object/string		ukoliko je type="post", ovde se može postaviti objekat
reloadPage	true/false	false	definiše da li strana treba ponovo da se učita iako je u kešu u DOM-u
role	string	<defined by data-role>	definiše ulogu stranice koju učitava
showLoadMsg	true/false	true	definiše da li poruka pri učitavanju treba da bude vidljiva ukoliko učitavanje traje duže od definisanog vremena
loadMsgdelay	milliseconds	50	vreme čekanja u milisekundama pre prikaza poruke
theme	a to z	c	tema definisana za sve strane
domCache	true/false	false	definiše da li strane treba da se keširaju u DOM ili ne

- Tabela s -

4.7.2 Metode za upravljanje stranama

- *\$.mobile.activePage* : upućuje na trenutno aktivnu stranu.
- *\$.mobile.pageContainer* : kontejner trenutno aktivne strane.
- *\$.mobile.addResolutionBreakpoints* : kao vrednosti uzima broj ili niz brojeva koji predstavljaju minimalnu visinu, odnosno širinu i maksimalnu visinu, odnosno širinu strane.

- *\$.mobile.changePage(page,options)*: ova metoda se koristi kako bi se promenila tekuća strana. Prvi parametar upućuje na novu stranu i može biti niska (tj. URL) ili može biti jQuery DOM objekat. Drugi parametar je opcioni i predstavlja par ključ / vrednost i može biti :
 - *allowSamePageTransition* : obično jQuery Mobile ne dozvoljava prelazak na istu stranu ali ako postavimo vrednost *true* i ovo će biti dopušteno.
 - *changeHash*: označava da li URL treba da se promeni (po definiciji je *true*).
 - *data*: niska ili neka vrednost objekta koja se prosleđuje drugoj strani.
 - *data-url*: vrednost url-a koja se prosleđuje pregledaču veba. Obično je to strana na koju se korisnik navodi, ali ovde može biti promenjeno.
 - *pageContainer* : jQuery Mobile smešta sve stanice u DOM, ali moguće je zaobići ovaj skup stranica i napraviti poseban
 - *reloadPage*: ukoliko je strana postoji u pregledaču veba, jQuery Mobile je uzima iz memorije tj. iz DOM-a. Ako se ova vrednost postavi na *true*, onda će je ponovo učitati(po definiciji je *false*).
 - *role*: jQuery Mobile će tražiti *data-role* atribut učitane strane ako želimo da gleda neki drugi ovde navodimo (moguće vrednosti *page/dialog*).
 - *showLoadMsg*: jQuery Mobile prikazuje poruku kad je strana uključena . Postavljanjem na *false* isključujemo ovu opciju.
 - *transition*: na koji način se prikazuje strana. Može se podesiti i globalno (po definiciji je *slide*)
 - *type*: jQuery Mobile učitava strane AJAX pozivom. Ova opcija omogućava da se definisanje HTTP metoda za učitavanje strane koja je po definiciji *get* (moguće vrednosti *get/post*).
 - *reverse*: definiše, obično kod dugmeta za povratak, da li se treba primeniti prikaz strane definisan sa *transition* (moguće vrednosti su *true/false*, po definiciji je *false*).

Način na koji se strana prikazuje možemo i ručno da definišemo koristeći CSS3 animaciju ili JavaScript. Kada definišemo *data-transition* atribut, jQuery Mobile proverava da li vrednost ovog atributa postoji definisana u okruženju. Ukoliko ne postoji, gleda da li postoji u kolekciji zvanj *\$.mobile.transitionHandlers* gde se nalaze i ručno definisani načini prikaza strane. Ukoliko ni tu ne pronade, traži CSS3 animaciju, i na kraju, ako ni to ne pronade, postavlja prikaz po definiciji.

Svaka tranzicija je JavaScript funkcija koja kao parametre mora da sadrži *ime tranzicije,reverse* (gore naveden), *toPage* – jQuery DOM objekat strane na koju prelazi, *fromPage* – jQuery Mobile objekat originalne strane.

CSS animacija se definiše primenom klasa (nešto više kasnije o njima). Ime tranzicije se koristi kao ime klase. Tako da, ako napravimo tranziciju *card*, potrebno je definisati klase *.card.in* – za novu stranu, *.card.out* za trenutnu stranu. Na isti način treba definisati klase za *reverse* : *.card.in.reverse*, *.card.out.reverse*.

- *\$.mobile.loadPage(page,options)* : Ovo je funkcija učitava stranu u DOM, ali ne vodi do nje. Prvi parametar je isti kao kod prethodno navedene funkcije, a opcije su ograničene na *data,pageContainer,reloadPage,role,type* .
- *\$.mobile.showPageLoadingMsg()* : Služi za prikazivanje poruke prilikom učitavanja strane.
- *\$.mobile.hidePageLoadingMsg()* : Služi za sakrivanje poruke prilikom učitavanja strane.

4.7.3 Metode definisane za platformu

- *\$.mobile.orientationChangeEnable*: ukoliko je matični događaj *orientationchange* dostupan
- *\$.mobile.gradeA()* : vraća *true* ukoliko je pretraživač tipa A grade
- *\$.mobile.urlHistory* : kolekcija strana koje su pregledane u okviru jQuery Mobile, a da nisu reučitavane. Svaki element sadrži *pageUrl,title* i *transition*.
- *getDocumentUrl()* : vraća originalni URL (prvog učitavanja)
- *keyCode* : konstante koje definišu šifre za rukovanje događajima dugmića sa tastature, uključujući : ALT, BACKSPACE, COMMAND, COMMAND_LEFT, COMMAND_RIGHT, DELETE, DOWN, UP, RIGHT, LEFT, END, ENTER, ESCAPE,HOME, INSERT, MENU, PAGE_DOWN, PAGE_UP, PERIOD, SHIFT, SPACE, TAB, WINDOWS.
- *getScreenHeight()* : uzima trenutnu visinu ekrana.

4.7.4. Metode definisane za putanje i Url-ove

- *\$.mobile.path.isAbsoluteUrl* i *\$.mobile.path.isRelativeURL*: ove dve funkcije gledaju URL i dozvoljavaju da proverimo da li je adresa apsolutna ili relativna.
- *\$.mobile.path.isSameDomain(first url, second url)* : dozvoljava nam da upoređujemo URL-ove i da vidimo da li pripadaju istom domenu. Ovaj metod će prepoznati da li je http ili https i razdvojiće ih u dva domena .
- *\$.mobile.path.makePathAbsolute(relative path, absolute path)*: uzima relativnu putanju, upoređuje je sa apsolutnom i vraća apsolutnu putanju koja odgovara zadatoj relativnoj...
- *\$.mobile.path.makeUrlAbsolute(relative url, absolute url)*: slično prethodnom metodu, samo što barata sa url-ovima.
- *\$.mobile.path.parseUrl()*: URL se sastoji iz raznih delova, i ovaj metod izvlači objekat koji sadrži, svejedno koji je tip url-a, podešavanja: *hesh, host, hostname, href, pathname, protocol, search*. Osim njih, vraća i dodatna podešavanja :
 - *Authority* : sadrži korisničko ime, lozinku i podešavanja host-a.
 - *Directory* : vraća deo putanje iz url-a koji se odnosi na direktorijum.
 - *Domain* : sadrži delove url-a koji se odnose na autorizaciju i protokol
 - *Filename* : vraća deo url-a koji se odnosi na ime datoteke
 - *username, password* : korisničko ime i lozinku ukoliko postoje u url-u

4.7.5 Metode za korisnički interfejs

- *\$.mobile.fixedToolbar.hide()* i *\$.mobile.fixedToolbars.show()*: prikazuju ili sakrivaju trake sa alatima koji su fiksirani.
- *\$.mobile.silentScroll(position)*: lista stranu do određene pozicije. Reč *silent* ukazuje na to da ova metoda neće oslušivati događaje vezane za listanje.

Ukoliko dinamički želimo da promenimo dodatak, potrebno je pozvati običnu jQuery funkciju *ime_dodatka()*. Ako je već kreiran dinamički, a želimo da ga promenimo, onda se funkciji kao parametar postavlja niska „*refresh*“. Ako smo promenili veliki broj html elemenata, dodali potrebno je ponovo učitati celi kontejner strane. *\$("#content").html(newHTMLcontentWithWidgets);*

4.8 Događaji

Događaji se realizuju preko metoda koji se koriste za registrovanje korisničkih akcija u pregledaču veba, delovanje nakon istih i za manipulaciju registrovanim akcijama. Osim osnovnih jQuery događaja, jQuery Mobile dodaje skup događaja specifičnih za mobilne uređaje (tap, taphold, swipe...). Događaje je moguće vezati sa `.bind()`, `.live()`, `.delegate()` ili `.on()` metodama.

Na određenu akciju korisnika, `.bind()` metoda vezuje funkciju za izvršenje za neki element. Isto to radi i `.live()`, samo što vezuje događaje i budućim elementima (npr. ako smo vezali događaj nekom elementu sa određenom klasom, svakom novom elementu sa tom klasom biće dodeljen isti događaj). Metoda `.delegate()`, osim što vezuje događaje budućim elementima, može da ih veže za dinamički kreirane elemente. Verzija 1.7 jQuery-ja uvodi metodu `.on()` koja, u zavisnosti od poziva, može da zameni prethodne tri.

Fizički događaji:

- *tap* i *taphold* : predstavljaju kratak i duži dodir ekrana
- *swipe*, *swipeleft* i *swiperight* : prevlačenje po ekranu. Ne postoji događaj za prelaženje ekrana gore i dole u trenutku pisanja.
- *scrollstart* i *scrollstop* : obrađuje početak i kraj skrolovanja.
- *orientationchange* : aktivira se kada se menja orijentacija uređaja
- *vclick*, *vmousedown*, *vmouseup*, *vmousemove*, *vmousecancel*, *vmouseover* : svi ovi virtuelni događaji služe da na apstraktan način provere događaje dodiranjem ili klikom mišem.

4.8.1 Rukovanje događajima strana

jQuery Mobile ima poseban koncept rada sa stranama. Kako bi omogućio veću kontrolu korisnicima, postoji veliki broj događaja definisanih za strane. Možemo ih podeliti u nekoliko grupa:

- **load**: događaji koji se odnose na učitavanje strana. Tu spadaju `pagebeforeload`, `pageload` i `pageloadfailed`. `Pagebeforeload` je dešava pre

nego što se zatraži ta strana, `pageload` kad se zatraži, a `pageloadfailed` logično ukoliko strana nije učitana.

- **change:** događaji koji se odnose na promenu sa jedne na drugu stranu, tu ubrajamo `pagebeforechange`, `pagechange`, `pagechangefailed`. Slično prethodnom slučaju, `pagebeforechange` služi da definišemo promenu strane, i dešava se pre `pagebeforeload`. `Pagechange` se izvršava nakon što je strana prikazana
- **transition:** odnose se na događaje prilikom prelaska na drugu stranu. `Pagebeforeshow` (pre nego što će strana biti prikazana), `pageshow` (kad je strana prikazana), `pagebeforehide` (dok je strana još uvek vidljiva), `pagehide` (kad je strana sakrivena).

Pagebeforeshow() i *pagebeforehide()* se izvršavaju pre njihovih povezanih događaja *pageshow* i *pagehide*, dok *pagebeforeload* i *pagebeforechange* zabranjuju sledeći događaj.

- **init:** događaji pri inicijalizaciji strane. Tu spadaju: `pagebeforecreate`, `pagecreate`, `pageinit`. `Pagebeforecreate` se izvršava nakon postavljanja strane u DOM ali pre kreiranja UI komponenti. `Pagecreate` – nakon kreiranja strane ali pre renderovanja komponenti. `Pageinit` – nakon što je strana potpuno učitana.
- **remove:** ovde spada `pageremove` događaj koji se dešava pre nego što se izbriše strana iz DOM-a. Može se oslušivati ovaj događaj kako bi se sprečilo brisanje strane.
- **layout:** `updatelayout` - Koristi se pri promeni neke komponente na stranici kako bi strana mogla da se obnovi.

`$(document).ready` gotovo sve jQuery aplikacije koriste za inicijalizaciju nekih bitnih podešavanja. Ovde ovo ne igra istu ulogu jer se strane učitavaju AJAX-om pa bi imalo smisla samo pri učitavanju prve strane. Zato se ovde koristi *pageinit* za takva podešavanja.

4.9 JQuery mobile teme za stil

JQuery Mobile omogućava podešavanje celog interfejsa pomoću tema i CSS-a. Svaka tema predstavlja skup šema boja kojima je definisana:

- *Boja teksta*
- *Pozadinske boje i gradijenti*
- *Font*

Moguće je kreirati 26 šema, definisanih slovima od a do z, a predefinisano je pet označenih slovima od a do e. Teme se dodeljuju elementima atributom *data-theme="a"* (b,c,d,e). Možemo da definišemo stil za celu stranu, a možemo i pojedinačno svakom elementu unutar nje da promenimo temu. Postoje i neka globalna podešavanja koja se odnose na neke efekte vezane za tekst ili polja (npr. senka i zaobljene ivice), aktivno stanje dugmad i dr. Tako, na primer, svako aktivno dugme će biti plave boje svedjedno koja mu je tema dodeljena.

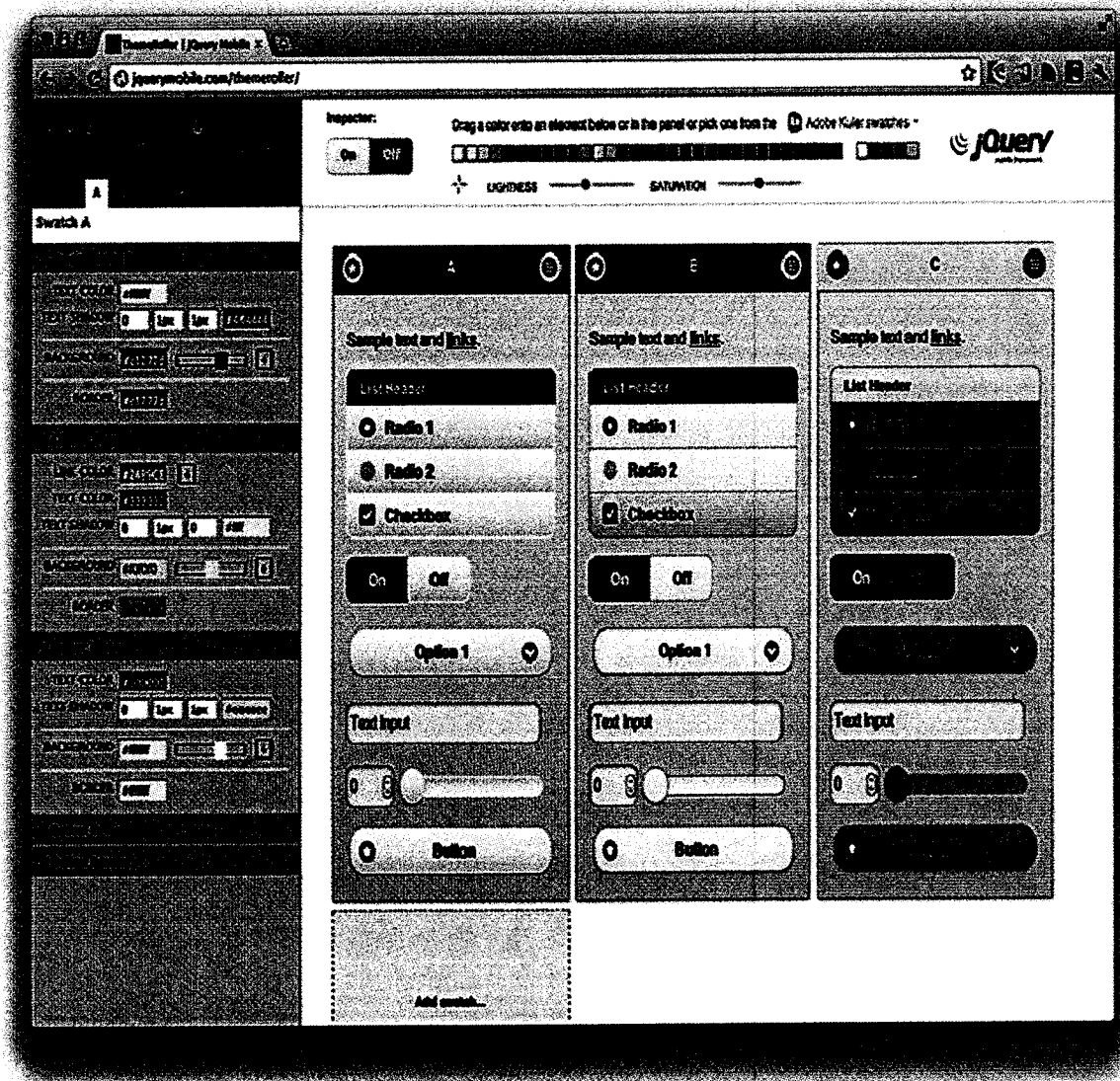
Tema je definisana u okviru CSS datoteke koju obezbeđuje okruženje. Nije poželjno menjati je, već ukoliko želimo da promenimo neka podešavanja, treba da ih uključimo u novu CSS datoteku koju uključujemo u naš html dokument, ali nakon već pomenute CSS datoteke. Kako bismo menjali stil, potrebno je razumeti kako jQuery Mobile definiše elemente i stilove. JQuery Mobile koristi klase za definisanje stilova. Tabela klasa koje možemo da definišemo u temi kako bismo prilagodili korisnički interfejs (<x> označava uzorak boja – a, b, c, ..., z) :

ime klase	opis
ui-bar-<x>	zaglavlje, podnožje
ui-btn-up-<x>	dugmići u normalnom stanju
ui-btn-hover-<x>	dugmići iznad kojih je kursor
ui-btn-down-<x>	dugmići koji su pritisnuti
ui-btn-active	dugmići koji su aktivni
ui-body-<x>	telo strane
ui-link-<x>	veze
ui-icon-<x>	ikonice za dugmiće i ostale dodatke

ui-corner-all	odnosi se na sve kontrole sa zaobljenim ivicama
ui-corner-<tl/tr/bl/br>	odnosi se na gore levo/gore desno/dole levo/dole desno zaobljene ivice
ui-corner-<top/bottom>	odnosi se na gore /dole zaobljene ivice
ui-corner-<left/right>	odnosi se na levo /desno zaobljene ivice
ui-shadow	odnosi se na sve elemente koji imaju senku
ui-disabled	odnosi se na sve elemente koji su onemogućeni HTML-om

- Tabela 8 -

Moguće je napraviti svoju temu i to pomoću ThemeRoller-a (<http://jquerymobile.com/themeroller/>):



- Slika 7 -

ThemeRoller je online besplatna aplikacija. Omogućava lako postavljanje tema prostim povlačenjem boja, fontova... promene se odmah vide i moguće je nakon toga sačuvati kreiranu temu.

Kao što se može videti sa prethodne slike, u središnjem delu se nalaze tri strane, na kojima možemo definisati tri teme. Iznad njih se nalaze boje koje možemo prevući na željeni element. Ugrađen je i Adobe Kuler, pomoću kojeg možemo da pretražujemo palete boja i koristimo ih za kreiranje tema. Korisno je i što imamo mogućnost da napravimo paletu boja od zadate slike (na primer logoa). (<http://kuler.adobe.com>). Sa leve strane se nalaze kartice: Global, A, B, C... Global kartica sadrži podešavanja koja će biti primenjena na celu aplikaciju i tu možemo definisati font, aktivno stanje dugmad i drugih kontrola, zaobljenje ivice za kontrole i grupu kontrola, podešavanje ikonica, senka polja. Kartice A, B, C ... predstavljaju teme koje kreiramo. Za njih možemo da definišemo boju zaglavlja i podnožja, pozadinu, senku, okvir i boju sadržaja, okvir i boju dugmadi...

U ThemeRoller-u možemo da izaberemo verziju jQuery Mobile-a, možemo da ubacimo postojeću temu koju bismo menjali. Kreirane teme možemo da sačuvamo. Svaka tema se sastoji od .css datoteke i od kompresovane .min.css datoteke koja se koristi za mobilne aplikacije.

Korisnicima Adobe Fireworks verzije CS5.1 na raspolaganju je jQuery Mobile Theme editor. Potrebno je instalirati besplatni dodatak „Fireworks CSS3 Mobile Pack“. Na taj način se obezbeđuje mogućnost svih podešavanja kao u već pomenutom ThemeRoller-u.

5. Rešenja slična jQuery Mobile-u

Postoje i jQuery biblioteke koje imaju slične osobine i upotrebu kao i jQuery Mobile :

- *jQTouch*
- *Sencha Touch*
- *iUI*
- *iWebKit*

jQTouch se pojavio 2009. godine kao jQuery dodatak za razvoj mobilnih aplikacija za iPhone. Zbog velike sličnosti i zavisnosti od jQuery biblioteke, ovaj dodatak je prilično težak zbog toga što je deo koda nasleđen od desktop pregledača veba. Zato se jQTouch koristi za uređaje sa malim ekranima, WebKit uređajima (iPhone, iPod Touch, Palm WebOS, Android). Preporučuje se za veb dizajnere i početnike u razvoju veb aplikacija, a profesionalnim programerima se preporučuje Sencha Touch. Glavne osobine jQTouch-a su to da je lak za podešavanje, pogodan za matične WebKit aplikacije, pogodan za podešavanje tema, ali samo za uređaje sa malim ekranima.

Sencha Touch, je nastao od istog autora kao i jQTouch Davida Kanede, ali je potpuno drugačiji paket od jQTouch. On ne zavisi od jQuery-ja, ali ima i dosta sličnih osobina sa jQTouch. Poboljšanje je to što se može koristiti na tabletima, lakši je, nudi čist JavaScript API za pravljenje aplikacija, ali još uvek radi samo na WebKit uređajima.

iUI nije, bar ne u svojoj osnovnoj postavci, biblioteka JavaScripta. Nastaje kao rezultat pokušaja Joe Hewitt-a da stvori aplikacije za iPhone. Cilj mu je korišćenje standardnog HTML-a za kreiranje uglađenog interfejsa koji zadovoljava visoke standarde Apple-ovih iPhone matičnih aplikacija. Nedostatak mu je taj što podržava samo iPhone, iPod Touch i možda iPad. Glavne osobine: uključuje html, malo funkcionalnosti, ne zahteva profesionalno znanje JavaScript.

iWebKit je paket koji omogućava kreiranje veb sajtova ili veb aplikacija kompatibilnih za iPhone, iPod Touch i iPad. Ne zahteva puno znanja html-a, veoma je lak za upotrebu, brz je, kompatibilan i proširiv. Pogodan je za početnike. Podržava WebKit uređaje.

Sve ovo navodi na zaključak da je jQuery Mobile, iako najnovije rešenje, ima najveću primenu, donekle zahvaljujući i sponzorima (Mozilla, Nokia, Blackberry, Palm...), ali i velikoj zainteresovanosti za njegovim unapređenjem.

Zbog svega navedenog, ukoliko želimo da napravimo aplikaciju za određenu platformu, u redu je koristiti neku od gore navedenih biblioteka, ali ukoliko želimo aplikaciju za više platformi, laku za korišćenje, koju podržavaju skoro svi pregledači veba, onda je jQuery mobile pravi izbor.

6. Elastyc

Elastyc je softver za elektronsko poslovanje malih i srednjih preduzeća, koji obuhvata

- WCMS (Web Content Management System) – sistem za upravljanje sadržajem veb sajta
- Korporativnu socijalnu mrežu
- CRM (*Customer Relationship Management*) – upravljanje odnosima sa kupcima
- Upravljanje projektima
- ERP (*Enterprise Resource Planning*) integracija – integracija *Pantheon* softvera za podršku i automatizaciju poslovnih sistema.

Elastyc nudi jednostavno definisanje strukture veb sajta i upravljanje njegovim sadržajem kroz predefinisane sadržajne elemente (tekstovi, meniji, forme, slajderi, galerije slika...) koji se nazivaju komponentama. Kompletna definicija izgleda veb sajta je bazirana na CSS (*Cascading Style Sheet*) tehnologiji, a Elastyc CSS editor omogućuje laku i brzu izmenu izgleda sajta. Osim toga, omogućeno je upravljanje korisnicima i korisničkim grupama, kao i njihovim pravima pristupa određenom sadržaju.

Korporativna mreža, nalik poznatoj društvenoj mreži Facebook, je uvedena u Elastyc kao osnovni vid komunikacije. Karakteriše je jednostavnost i, što je najvažnije za samo preduzeće, sva komunikacija je dokumentovana. Omogućeno je informisanje i učestvovanje u komunikaciji po svim relevantnim pitanjima podjeljenim u nekoliko celina:

- Sve aktivnosti vezane za kreirane projekte
- Svi klijentski zahtevi i komunikacija po njima
- Sve aktivnosti zaposlenog po svim predmetima u koje je uključen
- Sve aktivnosti vezane za određeno odeljenje u organizaciji ...

CRM predstavlja model za upravljanje interakcijama između potrošača, klijenata i prodaje. To podrazumeva organizaciju, automatizaciju i sinhronizaciju poslovnih procesa, prvenstveno prodajne aktivnosti. Automatizaciju službe prodaje Elastyc postiže nizom funkcionalnosti :

- Upravljanje kontaktima - centralna baza kako fizičkih tako i pravnih lica koja, pored osnovnih podataka neophodnih za evidenciju i komunikaciju, sadrži i definicije ugovorenih cenovnika, dokumente klijenta... Sistem omogućava čuvanje svih individualnih interakcija sa sopstvenim kontaktima, pružajući kompletnu istoriju prethodnih konverzacija.
- Prodajni tunel – predstavlja faze svakog poslovnog procesa kroz koje treba proći da bi se došlo do uspešnog zaključenja posla.
- Dokumenti narudžbina – jednostavno kreiranje i čuvanje svih dokumenata. Najvažniji dokumenti su zahtev kupca, ponuda kupcu i faktura. Ovi dokumenti sadrže sve neophodne parametre - osnovnih podataka, uslova plaćanja, povezanost sa specifikacijama po projektima, rezervacijom robe...

Automatizacija klijentskih zahteva je omogućena od postavke zahteva do njegove realizacije. Početak je uvek prijem zahteva (putem veb sajta, e-mail-a, telefona, lično) i distribucija istog u bazu podataka. Zatim sledi delegiranje – raspoređivanje zadataka zaposlenima u preduzeću i praćenje njihovih aktivnosti putem korporativne mreže.

Upravljanje projektima predstavlja planiranje, organizovanje i obezbeđivanje resursa u cilju realizacije određenog posla koji svojom kompleksnošću nameće praćenje po fazama. U Elastyc-u je, omogućeno kroz sledeće funkcionalnosti :

- Definisane projekta sa svim potrebnim elementima – klijent, rok završetka, ljudski resursi, povezani ugovori...
- Definisane faza kojim se dobija i dokument koji predstavlja specifikaciju posla i može se koristiti kao ponuda, ugovor...
- Definisane ljudskih resursa odakle se dobija projekcija vremenskog utroška zaposlenih u preduzeću i njihov plan rada
- Kalendar omogućuje uvid u sve termine događaja po projektu i po ljudima koji u njemu učestvuju
- Planiranje troškova

- Evidencija aktivnosti i kolaboracija kroz pomenuti mehanizam zida gde svi korisnici angažovani na projektu mogu imati uvid u realizaciju projekta, kašnjenje, mogu da evidentiraju svoju aktivnost i da komuniciraju na lak i jednostavan način.

7. Teorijski opis mobilne Elastyc aplikacije

Aplikacija o kojoj će ovde biti reči predstavlja mobilnu verziju već pomenute Elastyc aplikacije. Ona ne obuhvata sve funkcionalnosti Elastyc-a već samo najosnovnije kao što su korporativna mreža, upravljanje zahtevima, definisanje partnera i kontakata... Njen značaj je što omogućuje bržu i lakšu upotrebu Elastyc-a pomoću mobilnih uređaja, posebno mobilnih telefona.

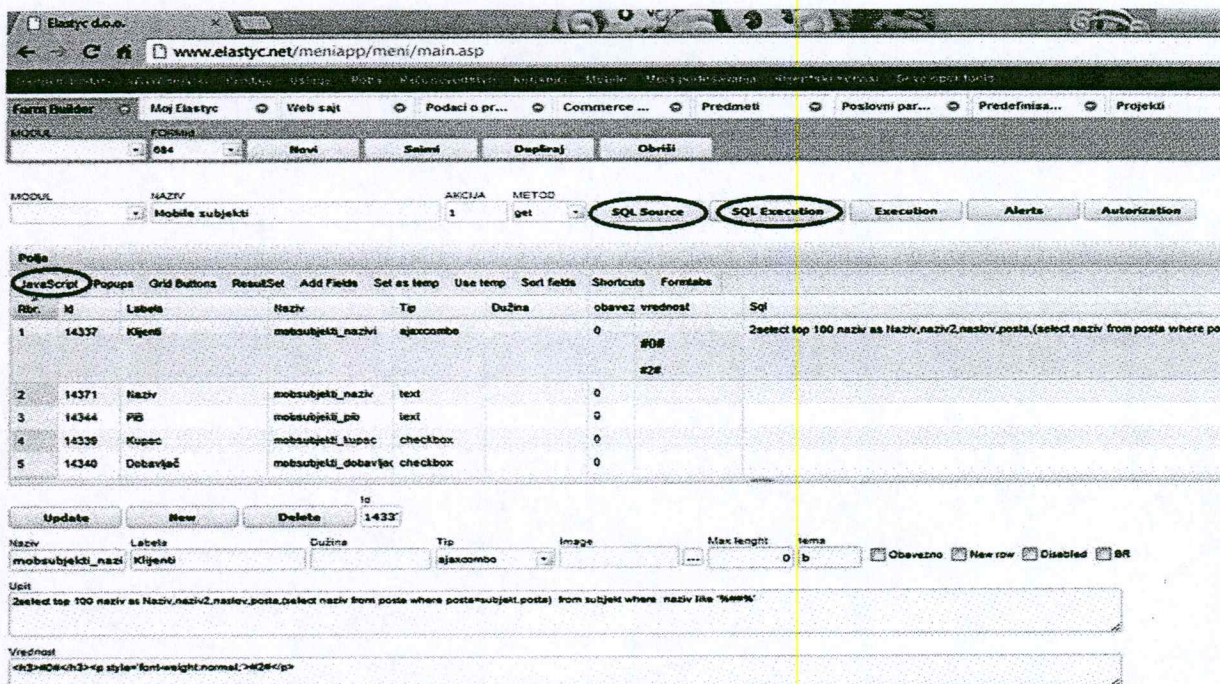
Zbog kreiranja dinamičke aplikacije, pored jQuery Mobile-a, korišćena je i ASP tehnologija. Iako je ASP donekle zastarela tehnologija, korišćena je za veći deo osnovne Elastyc aplikacije, pa je ispoštovana doslednost. Na kraju poglavlja biće par reči i o ASP-u.

U ovoj aplikaciji primenjena je 1.1.0 verzija jQuery Mobile-a. Kako je neophodno i jQuery okruženje, primenjena je 1.7.1 verzija jQuery-ja. jQuery i jQuery Mobile su uključeni direktivama u aplikaciju u *head* tag-u glavne strane aplikacije *main.asp* :

```
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.css" />
<script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>
<script src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js"></script>
```

7.1 Struktura aplikacije

Za upravljanje sadržajem aplikacije korišćene su forme. Svaku stranu aplikacije definiše jedna forma. Zbog toga je prvi korak u izradi aplikacije bio kreiranje formi za mobilnu verziju. Samo definisanje formi se obavlja na isti način kao za osnovnu aplikaciju, u Form Builder-u.



- Slika 8 -

Klikom na *SQLSource* otvara se prozor u kojem se definišu upiti kojima se forme popunjavanju podacima. Poziv upita se izvršava pozivom JavaScript funkcije *fillForm* koja je definisana u datoteci **formularimobile.js** zajedno sa ostalim funkcijama koje koriste sve forme.

Klikom na *SQLExecution*, otvara se prozor u kojem se definišu upiti kojima se menjaju podaci u bazi. Funkcijom *submitForm* izvršava se akcija ove vrste.

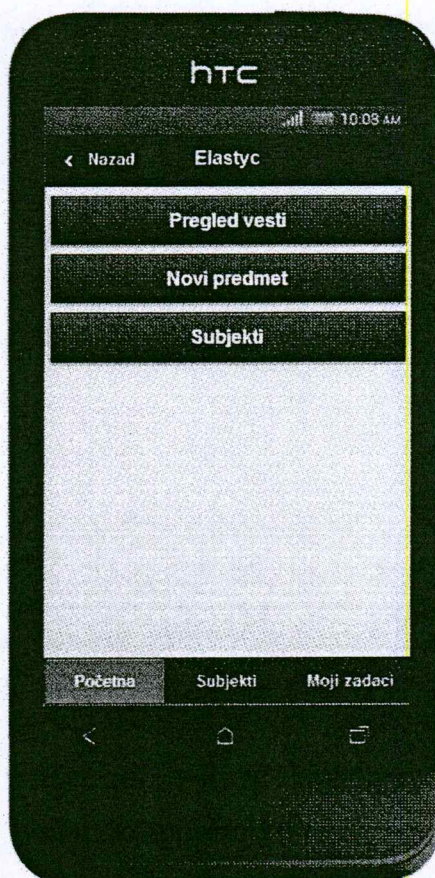
Za svaku komponentu forme moguće je definisati JavaScript kôd klikom na JavaScript.

Komponente forme su definisane u skladu sa jQuery Mobile komponentama u okviru datoteke **formularimobile.asp**.

Kako je aplikacija namenjena isključivo registrovanim korisnicima Elastyc-a, na početnoj strani **index.asp** se proverava da li je sesija korisnika (*session(„userid“)*) prazna ili ne. Ukoliko jeste, prelazi se na stranu logovanja **login.asp**, a ukoliko nije, na glavnu stranu aplikacije (**main.asp**). Strana **login.asp** sadrži polja za prijavljivanje i tu se proverava da li su ta polja popunjena, a zatim se ide na stranu **logcheck-fb.asp** radi provere korisnika. Datoteka **logcheck-fb.asp** je iskorišćena iz osnovne aplikacije.

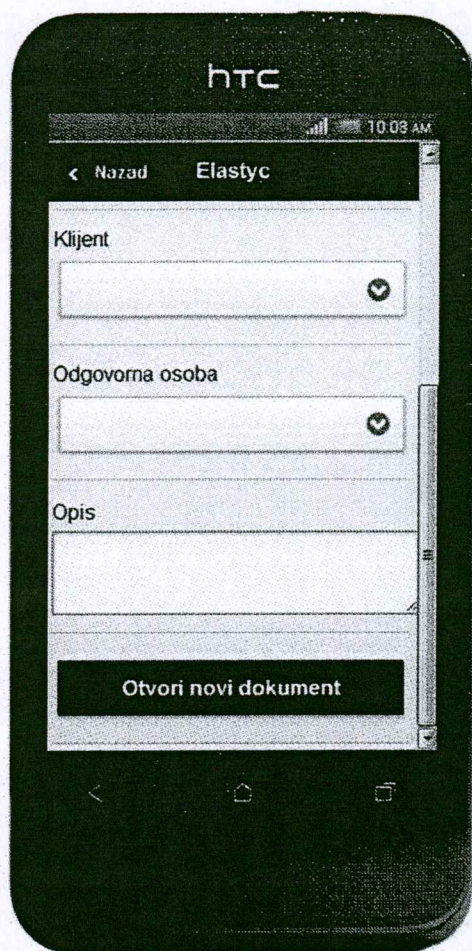
Kao što je već pomenuto, jQuery Mobile pruža mogućnost definisanja više strana unutar jedne datoteke. U main.asp datoteci se nalazi većina strana ove veb aplikacije definisanih indeksima **main**, **predmet**, **wall**, **wallfilter**, **docx**, **svojstva**, **termini**, **radnal**, **subjekti**, **kontakti**, **alertall**, **dialog**, **datedialog**.

Strana sa indeksom main predstavlja osnovni meni aplikacije:

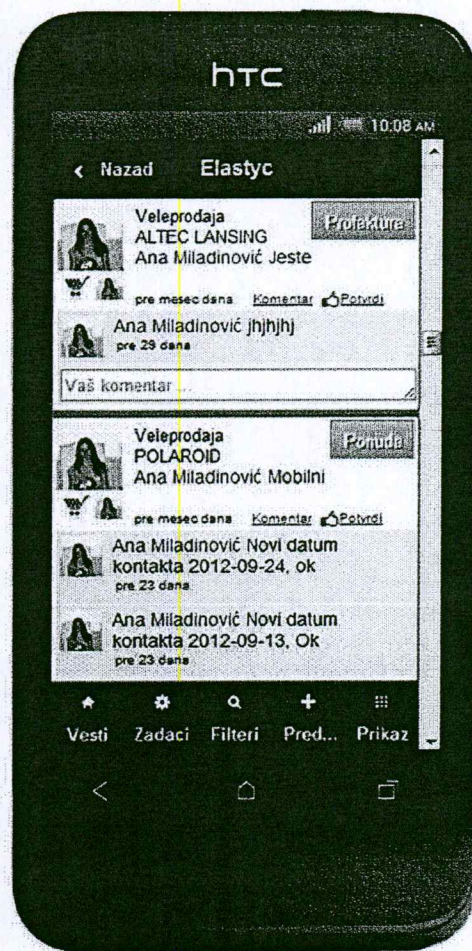


- Slika 9 -

Klikom na dugme Novi predmet, otvara se strana sa indeksom predmet (Slika 8). Ona daje korisniku mogućnost da definiše novi zahtev za određeno odeljenje i određenog klijenta. Nov predmet će odmah biti vidljiv na strani sa indeksom wall (Slika 9)



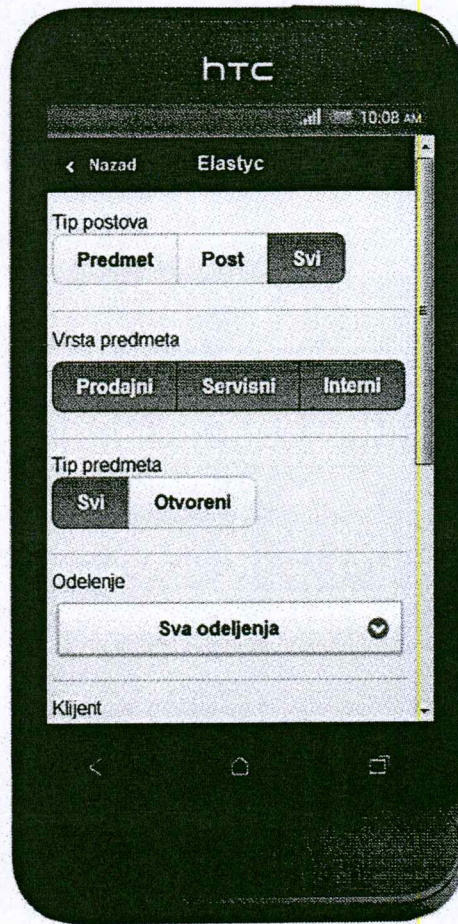
- Slika 10 -



- Slika 11 -

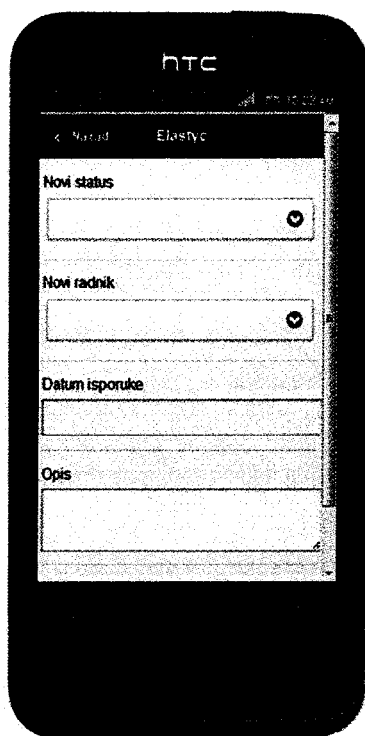
Do strane sa indeksom wall možemo doći iz glavnog menija, klikom na dugme *Pregled vesti*. Ova strana omogućuje korisniku sve ono što je već opisano kao funkcionalnost korporativne mreže. Po uzoru na Facebook, ovu stranu ćemo nazivati zidom (*engl. wall*). Klikom na dugme *Filteri* u podnožju zida, otvara se strana sa indeksom wallfilter (Slika 10) koja omogućuje pretragu i to po:

- tipu poruka (predmeti, poruke, moji predmeti, svi) – moji predmeti su predmeti za koje je ulogovani korisnik definisan kao osoba.
- vrsti predmeta (prodajni, servisni, interni) - za klijenta, unutar preduzeća, slobodna komunikacija
- tipu predmeta (otvoreni i zatvoreni)
- odeljenju - u okviru svakog preduzeća je obavezno definisati organizacionu strukturu koju karakterišu odeljenja i funkcije zaposlenih unutar njih,
- klijentu,
- odgovornoj osobi,
- projektu.



- Slika 12 -

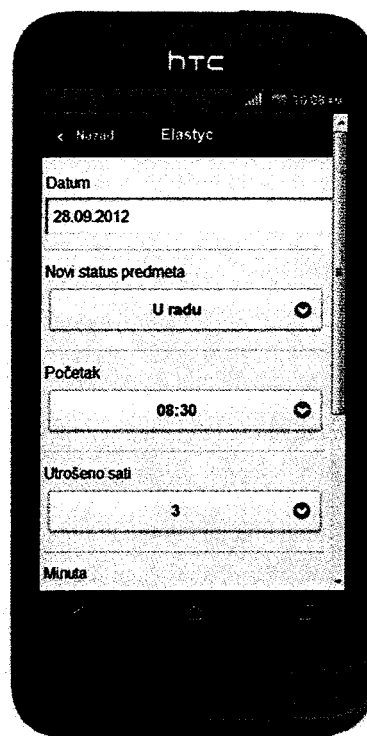
Novi predmet je moguće kreirati i klikom na dugme *Predmet* u podnožju zida. Klikom na zeleno dugme u desnom uglu svakog predmeta na zidu, otvara se strana sa indeksom docx koja sadrži tri dugmeta: *Promeni svojstva*, *Novi termin*, *Radni nalog*. Ova dugmad redom vode na strane sa indeksima *svojstva*, *termini*, *radnal*. Na strani sa indeksom *svojstva* (Slika 11) moguće je promeniti status datog predmeta, odgovornu osobu, datum isporuke i opis. Klikom na dugme *Novi termin*, korisnik može da definiše vreme nekog događaja vezanog za predmet (npr. sastanak) (Slika 12). Klikom na dugme *Radni nalog*, korisnik može da zabeleži svoje aktivnosti vezane za odabrani predmet (Slika 13).



- Slika 13 -



- Slika 14 -



- Slika 15 -

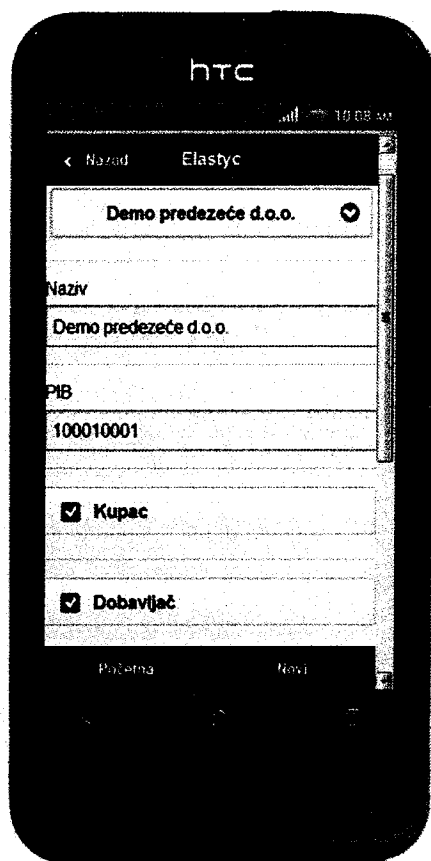
Zaglavlje zida je definisano u datoteci **header.asp** i uključeno je u sve strane aplikacije. U levom uglu zaglavlja se nalazi dugme *Nazad* tako da se u svakom trenutku možemo vratiti na prethodnu stranu.

Podnožje zida je definisano u datoteci **footerwall.asp**. Osim pomenutih, postoji i dugme *Prikaz* kojim se uklanjaju i prikazuju komentari predmeta na zidu. (Slika 14) Klikom na dugme *Vesti* u podnožju strane uklanjaju se svi filteri za pregled vesti. Predmete za koje je definisan kao odgovorno lice, korisnik može videti klikom na dugme *Zadaci* u podnožju stranice.

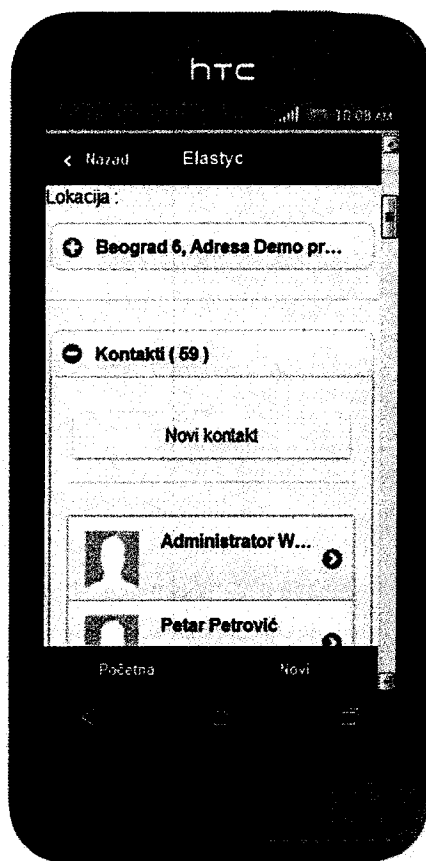


- Slika 16 -

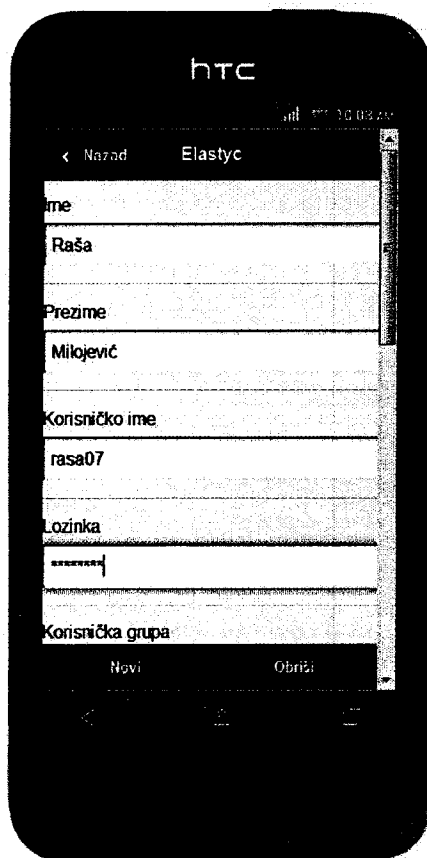
Klikom na *Subjeki* iz glavnog menija možemo da unesemo novog partnera ili da izmenimo postojeći (Slika 17 i Slika 18). Osim osnovnih podataka o partneru (naziv, država, posta, adresa, PIB, telefon, e-mail...) definiše se i tip partnera (u mobilnoj verziji su izdvojeni samo kupac i dobavljač, a u osnovnoj postoje još i banka, skladište, opština i odeljenje čime se definiše funkcija partnera kao korisnika Elastyc-a). Ukoliko je partner kupac i/ili dobavljač, možemo da definišemo rabat, broj dana za plaćanje, tip kupca i valutu plaćanja. Za svakog partnera moguće je dodati i informacije o osobama koje predstavljaju kontakte u poslovanju sa datim partnerom. Klik na dugme *Novi kontakt* vodi do strane sa indeksom kontakti (Slika 17) gde se unose osnovni podaci o novom kontaktu (ime, prezime, korisničko ime, lozinka, adresa, telefon, e-mail). Podatke o postojećem kontaktu možemo da vidimo i izmenimo klikom na kontakt iz liste svih kontakata partnera. Nakon unosa novog kontakta ili promenom korisničkog imena ili lozinke postojećem kontaktu, šalje se verifikacioni e-mail tom korisniku kako bi potvrdio svoj nalog.



- Slika 17 -



- Slika 18 -



- Slika 19 -

7.2 ASP

ASP (*Active Server Pages* – Stranice aktivne na serveru) je prva Microsoft tehnologija za kreiranje dinamičkih veb aplikacija koja kombinuje HTML i neki skript jezik, uglavnom VB script ili JavaScript. Funkcionalnost ASP-a se deli u sedam generičkih objekata i svaki upravlja svojim delom interakcije između veb servera i veb klijenta :

- Request
- Response
- Server
- Application
- Session
- ObjectContext
- ASPError

Prva dva objekta omogućuju razmenu informacija na relaciji klijent – server i obrnuto.

Objekat **Request** se koristi za zahtev od strane korisnika da vidi neku veb stranicu ili neku veb aplikaciju, a objekat **Response** vraća odgovor servera korisniku (npr. `Response.Write „Zdravo“`).

Preostalih pet su pozadinski objekti koji obogaćuju funkcionalnost ASP.

Objekat **Server** se koristi za predstavljanje veb servera. Obezbeđuje nekoliko najčešće upotrebljivanih funkcija u vezi sa onim šta veb server može da uradi, kao što je stvaranje novih objekata i podešavanje svojstava isteka vremena za skriptove.

Objekat **Application** se koristi za upravljanje sadržajem aplikacije.

Objekat **Session** predstavlja sesiju baziranu na kolačiću (*eng. cookie*) koja čuva informacije o korisniku od strane do strane.

Objekat **ObjectContext** upravlja transakcijama.

Objekat **ASPError** sadrži detalje o svim greškama koje stvara ASP skript ili sama datoteka `asp.dll`.

8. Zaključak

jQuery Mobile je novo i jednostavno okruženje za razvoj mobilnih veb aplikacija. Osnovno poznavanje HTML-a, JavaScript-a i CSS-a je dovoljno za rad u ovom okruženju. jQuery Mobile se nameće kao univerzalno rešenje za kreiranje veb aplikacija optimizovanih za rad na gotovo svim uređajima (telefonima, elektronskim čitačima, tablet i desktop računarima). Svaka jQuery Mobile aplikacija će raditi u mnogim pregledačima veba. Pristupačnost i jednostavnost su takođe dobre osobine. Ne treba zaboraviti i malu veličinu fajlova koje čine okruženje.

Podržan i finansiran od strane uticajnih kompanija (*Mozilla Corporation, Nokia, Blackberry, Palm...*), jQuery Mobile se brzo razvija i poboljšava. Iz tog razloga, i zbog svih dobrih osobina o kojima je već bilo reči, jQuery Mobile će verovatno postati još popularniji među programerima.

Kreirana mobilna aplikacija Elastyc, ne obuhvata sve mogućnosti Elastyc-a već samo one koje se najčešće koriste. Iz tog razloga, dalji razvoj aplikacije bi mogao da uključi i dodatne funkcionalnosti – pregled po projektima, prenošenje datoteka, govorne poruke... Pored toga, ideja za dalji razvoj je i da se kreira matična mobilna aplikacija kako bi se koristile i funkcionalnosti samih mobilnih uređaja. Za razvoj iste bi se koristio Phone Gap – okruženje za razvoj mobilnih aplikacija za sedam mobilnih platformi (iOS, Android, Blackberry, Windows Phone, Palm WebOS, Bada i Symbian), zasnovano na korišćenju HTML, CSS i JavaScript. (više o *PhoneGap*-u se može naći na <http://www.phonegap.com/>)

9. Literatura

- [1] Maximiliano Firtman: jQuery Mobile: Up an Running, O'Reilly Media
- [2] Jon Reid: jQuery Mobile, O'Reilly Media, 2011
- [3] Raymond Camden, Andy Matthews: jQuery Mobile Web Development Essentials, Packt Publishing , 2012
- [4] Giulio Bai: jQuery mobile First Look, Packt Publishing, 2011
- [5] Brad Broulik: Pro jQuery Mobile, Apress, 2011
- [6] www.jquerymobile.com
- [7] http://en.wikipedia.org/wiki/Progressive_enhancement
- [8] http://en.wikipedia.org/wiki/JQuery_Mobile