

STRUČNI RADOVI

PRIMJENA BOOTSTRAP CSS FRAMEWORKA U JSF WEB APLIKACIJAMA

APPLICATION OF BOOTSTRAP CSS FRAMEWORK IN JSF WEB APPLICATIONS

MUHAREM KOZIĆ

Univerzitet "Džemal Bijedić" Mostar,

SAMIR ALAGIĆ

JP Željeznice FBiH d.o.o. Sarajevo – Područje infrastrukture Bihać

AMEL DŽANIĆ

Univerzitet u Bihaću Tehnički fakultet Bihać

Sažetak: *Dostupnost sa bilo kojeg mjesta i u bilo koje vrijeme dovele su do naglog porasta i razvoja web aplikacija. Danas korisnici u sve većem broju za pregledavanje web aplikacija koriste svoje pametne telefone. Zbog toga je danas važno, da pored toga što aplikacije moraju biti sigurne, pouzdane i efikasne, također moraju biti i prilagodljive različitim vrstama uređaja sa kojih se pregledaju i koriste tj. one moraju biti lagane za korisničku upotrebu. Problem koji se postavlja pred programere jeste kako što brže i što funkcionalnije kreirati web aplikaciju koja će to sve ispuniti. Korištenje Jave i javinih frameworka kao što je JSF, rješava probleme pouzdanosti i efikasnosti i donekle prezentacije prema krajnjem korisniku, ali ne rješava problem prilagodljivosti niti bilo kakve druge, kao što su izgled i ponašanje web stranice. Za to je neophodno dobro poznavanje CSS i JavaScript-a. Jedan od najpopularnijih frameworka današnjice Bootstrap, je spoj CSS i JavaScript-a koji upravo rješava probleme prilagodljivosti i prezentacije, a programerima nudi bogatu paletu komponenti koje mogu koristiti u razvoju svojih web aplikacija. Spajanjem ovih frameworka, Bootstrapa i JSF, ispunjavaju se zahtjevi koji stoje pred web aplikacijama. U radu će se pokazati kako i na koji način spojiti i iskoristiti ova dva frameworka.*

Ključne riječi: *aplikacija, sigurnost, zaštita, prilagodljivost*

Abstract: *Availability from anywhere and at anytime has led to a sudden increase in the development of web applications. Today users are increasingly using smartphones to view web applications. Therefore in addition to having applications to be safe, reliable and efficient they must also be adaptable to the different types of devices that are being viewed and used, ie, they must be easy for use. Using Java and its frameworks, such as JSF, solves problems of reliability and efficiency, and upto certain level presentation to end-user, but does not solve*

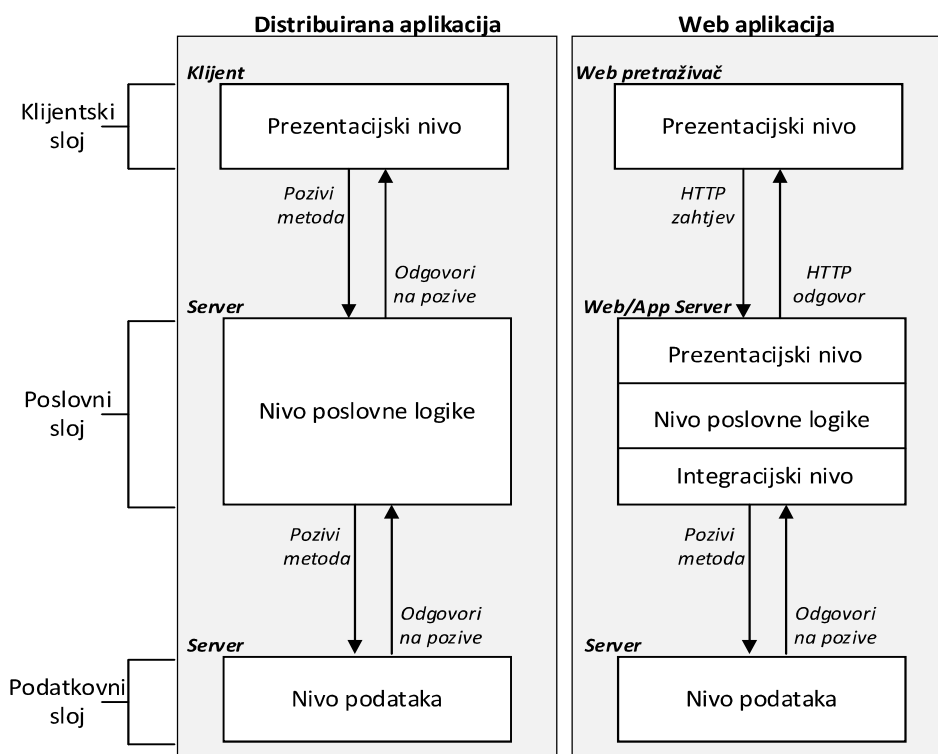
the problem of responsibility or any other such as the appearance and behavior of the website. This requires a good knowledge of CSS and JavaScript. Bootstrap is one of the most popular frameworks nowadays. It is a combination of CSS and JavaScript that is currently solving responsibility and presentation issues and provides developers with a rich set of components that can be used in the development of their web applications. By joining these frameworks Bootstrap and JSF the requirements for web applications are met. The paper shown how to connect and use of these two frameworks.

Key words: application, safety, protection, adaptability

FRONT-END I BACK-END

Za razvoj web aplikacija moguće je koristiti mnogo tehnologija i alata, odnosno frameworka. Razvoj web aplikacija sastoji se od front-end razvoja i back-end razvoja, na osnovu čega se i web tehnologije mogu podijeliti na front-end i back-end tehnologije. Svaka web aplikacija u suštini jeste distribuirana aplikacija. Današnje moderne distribuirane aplikacije su troslojne aplikacije (eng. Three-tier) sa sljedećim slojevima koji dijele izvršenje aplikacije: prezentacijski sloj, sloj aplikacijsko/poslovne logike i podatkovni sloj [1].

Slika 1. Troslojna arhitektura distribuiranog sistema i web aplikacije



Izvor: <https://www.tiobe.com/tiobe-index/>

Jedina bitna razlika između distribuirane aplikacije i web aplikacije jeste ta da se kod web aplikacije dio prezentacijskog sloja stavlja u poslovni sloj tj. sloj aplikacijske/poslovne logike i postoji još dodatni dio, integracijski sloj koji sve to povezuje, kao što je prikazano na slici 1.

Programski jezici široke rasprostranjenosti, kao što je Java, koriste se se za back-end razvoj web aplikacijazajedno sa sistemima za upravljanje bazama podataka (engl. Data Base Management System – DBMS). Back-end je dio kôda koji se pokreće na web serveru i koji je odgovoran za poslovnu logiku aplikacije, komunikaciju sa bazom podataka i samo izvođenje aplikacije. Front-end je dio kôda koji se izvršava na web pretraživaču, a rezultat tog kôda je korisnički interfejs koji omogućava interakciju klijenta i web aplikacije. Front-end-u pripada prezentacijski dio, što znači da front-end predstavlja sve što klijent može vidjeti kada koristi aplikaciju. U tom smislu moderni front-end mora biti odgovoran prema prilagodljivosti uređaja, tj. prilagodljivom web dizajnu (engl. Responsive Web Design – RWD) kako bi se omogućio jednostavan pristup stranicama web aplikacije preko različitih uređaja, a naročito za pristup preko pametnih telefona koji su postali sastavni dio života i čija je upotreba u stalnom porastu.

U radu će se objasniti Java programski jezik koji se može koristiti za back-end, JavaServer Faces – JSF kao Java framework za front-end razvoj na nivou servera, te Bootstrap front-end framework prezentacijskog dijela na nivou klijenta za ispunjenje ciljeva RWD-a. Podatkovni dio – DBMS neće biti uzet u razmatranje.

JAVA PROGRAMSKI JEZIK

Programski jezik Java je viši programski jezik čije se karakteristike najčešće opisuju s riječima: jednostavan, objektno orijentisan, mrežno efikasan, nezavisan o hardveru (platformi), dinamičan, siguran i efikasan [2]. Preko Java virtualne mašine (engl. Java Virtual Machine – JVM) Java je dostupna svakom hardverskom uređaju što je čini programskim jezikom opšte namjene. Java se primjenjuje u velikom broju domena, te je zbog toga široko rasprostanjena.

Prema podacima TIOBE indeksa programerske zajednice (engl. TIOBE Programming Community Index) Java programski jezik dugi niz godina, a i danas drži vodeću poziciju. TIOBE (engl. The Importance Of Being Earnest) indeks je pokazatelj popularnosti programskih jezika koji se ažurira jednom mjesečno. U tabeli 1. prikazan je TIOBE indeks za februar 2019. i poređenje sa istim periodom 2018. godine [3].

Tabela 1. TIOBE indeks za februar 2019

Februar 2019	Februar 2018	Promjena	Programski jezik	Procjena [%]
1	1		Java	15.876
2	2		C	12.424
3	4	↑	Python	7.574
4	3	↓	C++	7.444
5	6	↑	Visual Basic .NET	7.095
6	8	↑	JavaScript	2.848
7	5	↓	C#	2.846
8	7	↓	PHP	2.271
9	11	↑	SQL	1.900
10	20	↑↑	Objective-C	1.447
11	15	↑↑	Assembly language	1.377
12	19	↑↑	MATLAB	1.196
13	17	↑↑	Perl	1.102
14	9	↓↓	Delphi/Object Pascal	1.066
15	13	↓	R	1.043
16	10	↓↓	Ruby	1.037
17	12	↓↓	Visual Basic	0.991
18	18		Go	0.960
19	49	↑↑	Groovy	0.936
20	16	↓↓	Swift	0.918

Izvor: <https://www.tiobe.com/tiobe-index/>

Java kao jedan od najpopularnijih i najefikasnijih programskih jezika se može koristiti za back-end razvoj web aplikacija, gdje je zadužena za poslovnu logiku aplikacije i komunikaciju sa podacima. Velika prednost Jave je ogromna količina

open source kôda, zbog čega Java ima široku primjenu, te zauzima važnu ulogu u razvoju Android platforme i općenito web aplikacija. Zbog svega navedenog Java se danas nalazi na skoro svim dostupnim komunikacijskim uređajima.

JAVASERVER FACES – JSF

JavaServer Faces – JSF je komponentno orijentisan Java framework za prezentaciju, tj. izradu korisničkog interfejsa za web aplikacije. Što je najvažnije, JSF pojednostavljuje razvoj korisničkog interfejsa, koji je često jedan od težih i zahtjevnijih dijelova razvoja web aplikacija. Iako je moguće da se izradi korisnički interfejs pomoću temeljnih Java web tehnologija (kao što su Java servleti i JavaServer Pages – JSP) bez sveobuhvatanog okvira dizajniranog za razvoj web aplikacija, te osnovne tehnologije često mogu dovesti do različitih razvojnih problema i problema održavanja. JSF je jedan od najzastupljenijih Java frameworka koji pruža moćnu platformu za rješavanje uobičajenih problema koji se često pojavljuju u toku razvoja Java web aplikacija, kao što su validacija, konverzija, navigacija, templating i tokovi stranica.

Pružanje standardiziranog načina za rješavanje zajedničkih problema razvoja web aplikacija čini JSF odličnim frameworkom koji smanjuje vrijeme razvoja i održavanje web aplikacija. Ovo posebno važi kada je razvojni tim velik i distribuiran, što je uobičajeni scenario koji se susreće u kompanijama. Izrada korisničkih iskustava oko skupa standardizovanih JSF komponenti omogućuje pravi stepen prilagođavanja i izražavanja, ali također uspostavlja „zajedničku DNK“ kako bi aplikacija trebala izgledati, ponašati se i reagirati na različite implementacije [4].

JSF framework slijedi MVC (engl. Model-View-Controller) arhitekturu što JSF web aplikacije čini dosta fleksibilnijim i pogodnijim za rad, pošto je UI kôd (Pogled) odvojen od aplikacijskih podataka i poslovne logike (Model). JSF Faces Servlet (Kontroler) upravlja cijelom korisničkom interakcijom sa aplikacijom. On priprema JSF okruženje koje omogućuje stranicama pristup aplikacijskim podacima kao i zaštitu od neovlaštenog i neadekvatnog pristupa stranicama.

JSF nije jedini komponentno orijentisan web framework, ali je od verzije 2.0 dio standardnog Java poslovnog izdanja (engl. Java Enterprise Edition – Java EE). JSF je uključen u svaki Java EE aplikacijski server i može se jednostavno dodati na samostalni web kontejner kao što je Tomcat. JSF sadrži sav neophodan kôd za obradu događaja i organizaciju komponenti. Razvojni web programeri mogu da zanemare sve nepotrebne detalje i da se usredotoče na sam razvoj aplikacijske logike.

JSF ima ove dijelove [5]:

- Skup ugrađenih UI komponenti.
- Događajima vođen (engl. Event Driven) programski model.
- Model komponenti koji omogućava programerima serverske strane razvoj dodatnih komponenti.

JSF UI komponente korespondiraju sa Java managed bean-ovima, što u stvari predstavlja poveznicu između nivoa poslovne logike i prezentacijskog nivoa. Managed bean je Java bean kojem se može pristupiti sa JSF stranice. Managed bean mora imati definisano ime i obim. Najjednostavniji način specificiranja imena i obima managed bean-a je korištenjem atributa kao u sljedećem primjeru [5]:

Slika 2. Specificiranje imena Managed bean-a

```
@Named("user") // or @ManagedBean(name="user")
@SessionScoped
public class UserBean implements Serializable
```

David Geary, Cay Horstmann: Core JavaServer Faces, Third Edition, Prentice Hall, 2010.

Atribut name se može izostaviti u anotaciji ManagedBean. U tom slučaju, ime bean-a je izvedeno iz imena klase pretvaranjem prvog slova u malo slovo. Npr., ako se izostavi (name="user") ime bean-a postaje userBean.

Iz primjera vidimo da se managed bean imenuje kao user i da mu je obim na području sesije što znači da se varijabla name može dohvatiti (engl. Get) sa više različitih JSF stranica. Svaki korisnik koji koristi web aplikaciju dobiva svoju instancu bean objekta. Managedbean-ovi se koriste na sljedeći način: kada se ime managed bean-a pojavi na JSF stranici, JSF implementacija pozove objekt s tim imenom ili ga konstruiše ako još uvijek ne postoji u odgovarajućem obimu. U beanse spremaju svi podaci koji se pojavljuju na JSF stranici bilo da se radi o ulaznim ili izlaznim podacima. Znači, u JSF aplikacijama, managed bean-ovi se mogu koristiti za sve podatke koji trebaju biti dostupni na stranici. Bean-ovi su poveznica između korisničkog interfejsa i pozadine aplikacije [5].

Razvoj naprednih JSF aplikacija zahtijeva dobro razumijevanje biblioteka JSF tagova. Od JSF verzije 2.0, postoji šest biblioteka s više od 100 tagova, što je prikazano u tabeli 2.

Tabela 2. Biblioteke JSF tagova

Biblioteka	Identifikator imenskog prostora	Korišteni prefiks	Broja tagova
Osnovna	http://java.sun.com/jsf/core	f:	27
HTML	http://java.sun.com/jsf/html	h:	31
Facelets	http://java.sun.com/jsf/facelets	ui:	11
Kompozitne komponente	http://java.sun.com/jsf/composite	composite:	12
JSTL osnovna	http://java.sun.com/jsp/jstl/core	c:	7
JSTL funkcije	http://java.sun.com/jsp/jstl/functions	fn:	16

Izvor: David Geary, Cay Horstmann: Core JavaServer Faces, Third Edition, Prentice Hall, 2010.

Biblioteka osnovnih JSF tagova sadrži tagove koji su odgovorni za kategorizaciju raznih elemenata korisničkog interfejsa na formi, konverziju i validaciju podataka, upravljanje osluškivačima događaja (engl. Action Listener) za određenu komponentu, itd. Važno je napomenuti da osnovna biblioteka sadrži tagove koji su nezavisni od HTML renderiranja.

Jedan od najvažnijih aspekata dizajna JSF specifikacije je da se u potpunosti oslanja na postojeću Java EE web tehnologiju u njenoj osnovi. To znači da je JSF web aplikacija zapravo samo standardna Java EE web aplikacija s nekoliko specifičnih konfiguracija. Kada je Java EE web aplikacija ispravno konfigurisana za JSF, može se konstruirati pogled koristeći Facelets XHTML stranice. (Ranije verzije JSF-a prije 2.0 su naglašavale JSP kao deklarativni jezik stranice). Izrada JSF aplikacija pomoću XHTML-a vrši se korištenjem biblioteka Facelets tagova omogućenih sa JSF-om.

Facelets podržava veliki broj tagova koji se mogu grupisati u sljedeće kategorije [6]:

- Uključuje sadržaj iz drugih XHTML stranica (*ui:include*)
- Izrada stranica od template-a (*ui:composition*, *ui:decorate*, *ui:insert*, *ui:define*, *ui:param*)
- Kreiranje dodatnih komponenti bez pisanja Java kôda (*ui:component*, *ui:fragment*)
- Razne korisničke usluge (*ui:debug*, *ui:remove*, *ui:repeat*)

Kada pravimo JSF stranicu, dodajemo JSF tagove na XHTML stranici. XHTML stranica je jednostavno HTML stranica. Sve JSF stranice koriste ekstenziju .xhtml. Unutar taga potrebno je deklarirati imenske prostore [6]:

Slika 3. Zaglavlje JSF stranice sa deklariranjem imenskih prostora

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://xmlns.jcp.org/jsf/html"
xmlns:p="http://xmlns.jcp.org/jsf/passthrough"
xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
xmlns:f="http://xmlns.jcp.org/jsf/core"
xmlns:c="http://xmlns.jcp.org/jsp/jstl/core">
```

M.Kozić, S. Alagić, A. Džanić: Primjena Facelets template-a u izradi JSF web aplikacije, 11th

JSF stranica je slična je HTML formatu. Treba imati na umu sljedeće razlike [5]:

- JSF stranica mora biti pravilno XHTML formatirana. Za razliku od pretraživača, JSF implementacija ne traži sintaksne pogreške.
- Koriste se JSF tagovi *h:head*, *h:body* i *h:form* umjesto standardnih HTML tagova *head*, *body* i *form*.
- Umjesto korištenja poznatih *input* HTML tagova, koriste se *h:inputText*, *h:inputSecret* i *h:commandButton*.

BOOTSTRAP FRAMEWORK

CSS frameworki se koriste u izradi i optimizaciji web stranica omogućavajući programerima korištenje dodatnih opcija, ubrzanje procesa izrade i smanjenje zapremine kôda koji je napisan. CSS frameworki se obično sastoje od paketa sastavljenog od struktura datoteka i mapa standardiziranog kôda (HTML, CSS, JS dokumenti). Najpopularniji CSS framework za izradu web aplikacija je Bootstrap.

Razvili su ga 2011. godine Mark Otto i Jacob Thornton, web programeri i

dizajneri koji su radili za kompaniju Twitter i koji su za potrebe ove kompanije kreirali potpuno nov framework koji su podijelili sa cijelim svijetom. To je open source modularan CSS framework za razvoj web formi (interfejsa) i naprednih web komponenti. Bootstrap je zasnovan na CSS-u koji u sebi ima definisane standardne HTML elemente za tipografiju, forme, dugmad, tabele, navigaciju, modale i mnoge druge interfejs komponente, kao i dodatne JavaScript ekstenzije. Prednost korištenja Bootstrap frameworka je njegova odlična dokumentacija, tj. biblioteke koje sadrže veliku količinu gotovog kôda. Također je modularan, što olakšava njegovu nadogradnju i upotrebu sa različitim modulima koje razvijaju nezavisni web programeri.

Najvažnija karakteristika Bootstrap frameworka jeste kompatibilnost sa svim modernim web pretraživačima (Google Chrome, Firefox, Internet Explorer, Safari, Opera), kako na desktop računarima, tako i na mobilnim platformama. Sljedeća važna karakteristika jeste da Bootstrap podržava prilagodljivi web dizajn (RWD) kao bitan element pri razvoju web aplikacija u današnje vrijeme, gdje je potrebno razmišljati unaprijed i prilagoditi se najnovijim zahtjevima među kojima se već duži period nalazi i RWD. Prilagodljivi (responzivni) web dizajn omogućuje jednostavan pristup web stranicama aplikacija preko različitih uređaja.

Ne samo da je Bootstrap prilagodljiv, već je zasnovan na „Prvo mobilan“ (engl. Mobile First) strategiji. To znači da je prvenstveno dizajniran za mobilne uređaje, a zatim se skalira prema drugim uređajima (za razliku od drugih frameworka dizajniranih prvenstveno za desktop uređaje, pokušavajući ga zatim skalirati na mobilne uređaje) [7]. Ova karakteristika Bootstrapa je značajna zbog toga što su mobilni uređaji postali sastavni dio života i svakodnevnice. Vrijeme provedeno na mobilnim uređajima i dalje raste velikom brzinom. Oko dvije trećine (63%) korisnika pametnih (smart) telefona koriste svoj uređaj svakih 30 minuta ili manje, dok 22% koristi svoje telefone svakih pet minuta, pokazalo je istraživanje Interaktivnog biroa za oglašavanje (engl. Interactive Advertising Bureau – IAB) [8].

Za prilagodljivo oblikovanje web aplikacija Bootstrap koristi *grid* sistem ili sistem. Prilagođavanjem veličini, odnosno širini ekrana, web aplikacija može mijenjati vlastiti raspored elemenata kako bi oni bili pregledniji. Bootstrap sadrži prilagodljivi sistem rešetki – fluidnu mrežu koja se sastoji od 12 kolona. Fluidne mreže definiraju elemente web stranice u relativnim vrijednostima, a ne u apsolutnim kao što je dosad bilo. Medijski upit ispituje veličinu ekrana, te se prema tome stranica prikazuje u određenoj veličini i obliku. Podešavanjem minimalne i maksimalne veličine pretraživača i dimenzija ekrana postiže se usklađen prikaz web stranica na svim uređajima. Bootstrap *grid* sistem koristi niz kontejnera, redova i kolona za oblikovanje i usklađivanje sadržaja web stranice. Izgrađen je s fleksibilnim blokovima (engl. Flexbox) i potpuno je responzivan,

tj. prilagodljiv korisničkim uređajima. Kontejneri su osnovne komponente Bootstrap *grid* sistema. Redova može biti jedan ili više, a kolona do 12 na cijeloj stranici. Bootstrap *grid* sistem ima pet klasa [9]:

.col- (extra mali uređaji – širina ekrana manja od 576px),
 .col-sm- (mali uređaji – širina ekrana jednaka ili veća od 576px),
 .col-md- (srednji uređaji – širina ekrana jednaka ili veća od 768px),
 .col-lg- (veliki uređaji – širina ekrana jednaka ili veća od 992px),
 .col-xl- (extra veliki uređaji, ekrani sa većim dijagonalama – širina ekrana jed. ili veća od 1200px).

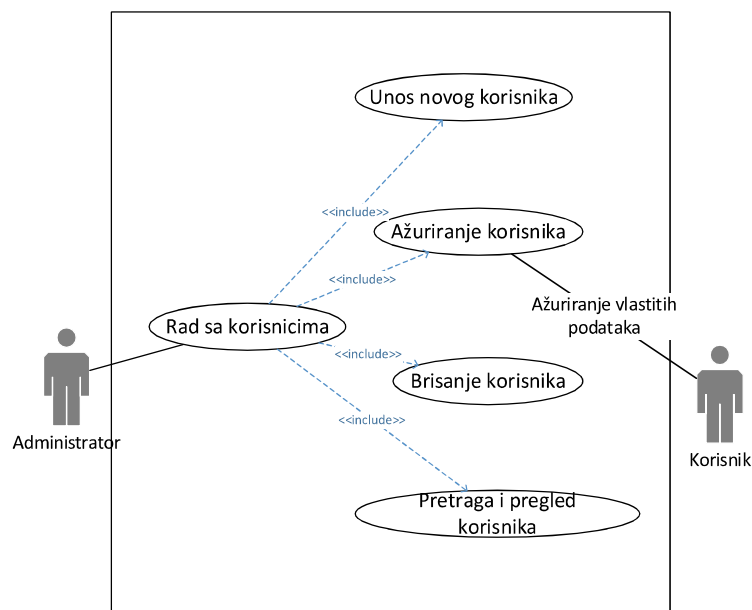
Navedene klase se mogu kombinirati kako bi se kreirali dinamičniji i fleksibilniji *layout*-i (organizacija izgleda web stranice). Kako se svaka klasa skalira, ako se želi postaviti ista širina, npr. Za klase sm i md, potrebno je samo navesti sm.

Bootstrap framework sadrži veliki broj komponenti, među kojima su: dugmad (engl. Buttons), kartice (engl. Cards), forme (engl. Forms), navigacijska traka (engl. Navigation Bar – Navbar), modalitd [10].

SPOJJSF I BOOTSTRAPFRAMEWORKA SA PRIMJEROM

Kao primjer za spajanje ovih frameworka poslužiti će mala aplikacija za logiranje korisnika na sistem. Pored logiranja, ova aplikacija mora da ima mogućnost i za unos, ažuriranje i brisanje korisnika, kao što je prikazano na slici 4.

Slika 4. Dijagram korištenja aplikacije



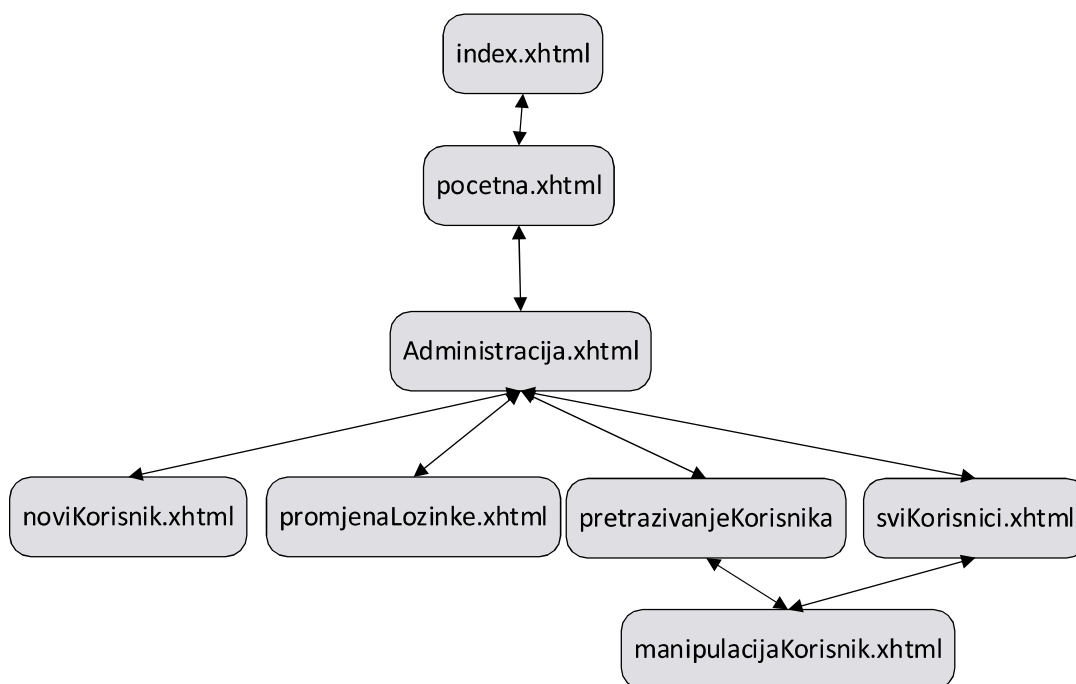
Shodno slici 4. proces projektovanja aplikacije će se dekomponovati na tri osnovna dijela:

1. Projektovanje nivoa podataka
2. Projektovanje poslovne, tj. aplikacijske logike
3. Projektovanje prezentacijskog nivoa, tj. korisničkog interfejsa

Budući da je težište ovoga rada na spoju ova dva velika frameworka, tada se neće razmatrati projektovanje nivoa podataka i poslovne tj. aplikacijske logike, nego samo projektovanje korisničkog interfejsa.

Korisnički interfejs će imati strukturu prikazanu na slici 5. gdje je vidljivo da je ovo struktura koju bi imao tip korisnika *Administrator*.

Slika 5. Struktura i navigacija web aplikacije za tip korisnika Administrator



Implementacija korisničkog interfejsa će se izvršiti u JSF frameworku koji zahtjeva XHTML format i uključivanjem Bootstrap frameworka. Kada se pravi JSF, tj. Facelets stranica, dodaju se JSF tagovi na XHTML stranici. Unutar taga potrebno je deklarirati imenske prostore [11].

Slika 6. Zaglavlje JSF stranice sa deklarisanjem imenskih prostora

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://xmlns.jcp.org/jsf/html"
xmlns:p="http://xmlns.jcp.org/jsf/passthrough"
xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
xmlns:f="http://xmlns.jcp.org/jsf/core"
xmlns:c="http://xmlns.jcp.org/jsp/jstl/core">
```

Ed Burns, Chris Schalk: Java Server Faces 2.0: The Complete Reference, McGraw-Hill, 2010.

Uključivanje Bootstrap okvira u JSF stranicu prikazano je na slici 7.

Slika 7. Uključivanje Bootstrap okvira u JSF stranicu

```
<h:head>
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<h:outputStylesheet library="resources" name="css/my.css" />
<h:outputStylesheet name="bootstrap/css/bootstrap.min.css"
library="resources"/>
<h:outputScript name="bootstrap/js/jquery-2.2.4.min.js"
library="resources"/>
<h:outputScript name="bootstrap/js/bootstrap.min.js" library="resources"/>
<title>DocSYSTEM</title>
</h:head>
```

Prvo se definiše *master* stranica. Ova stranica treba da definiše sve dijelove web stranice. U direktoriju *web* ovog projekta kreiraće se direktorij *template*, te će se tu smjestiti datoteka predložka *master.xhtml* [6]. Zaglavlje datoteke neće biti prikazano jer je isto kao na slikama 6. i 7.

Slika 8. Dio kôda master stranice gdje se definišu dijelovi stranice

```
<h:body style="background-color: white">
<ui:insert name="navigacija">
<ui:include src="/stranice/navigacijaTest_1.xhtml"/>
</ui:insert>
<ui:insert name="jumboNav">
```

```

<ui:include src="/stranice/jumboNav.xhtml"/>
</ui:insert>
<ui:insert name="greska">
<ui:include src="/stranice/greska.xhtml"/>
</ui:insert>
<div style="background-color: white">
<ui:insert name="tjelo"></ui:insert>
</div>
<ui:insert name="footer">
<ui:include src="/stranice/footer.xhtml"/>
</ui:insert>
</h:body>

```

Kao što se može vidjeti sa slike 8. prisutna su tri dijela kôda koja su statična, odnosno koja se mijenjaju i gdje se učitavaju uvijek iste stranice: *navigacija.xhtml*, *jumboNav.xhtml* i *footer.xhtml*. Samo u dijelu gdje je dato ime *tjelo* će se odigravati scenarij web aplikacije, tj. tu će se učitavati stranice koje nastaju u interakciji sa korisnikom.

Primjer korištenja Bootstrap frameworka sa JSF frameworkom je prikazan na slici 9.

Slika 9. Ispis kôda stranice novi Korisnik.xhtml

```

<ui:composition template="/predlozak/master.xhtml">
<ui:define name="tjelo">
<c:if test="#{LogIn.testRegistracije==true}">
<div class="container">
<h3 style="color: black">
<i class="fa fa-cogs" aria-hidden="true"></i> Administracija korisnika</h3>
<hr class="my-1" />
<div class="card sjena" >
<div class="card-header" >
<div class="mb-0">
<h4 class="text-primary"> Unos novog korisnika</h4>
<div class="text-muted"> Unos novog korisnika sistema </div>
</div>
</div>
<div class="card-block">
<h:form >
Ime :<h:inputText class="form-control" value="#{LogIn.noviKorisnik.ime}"
p:autofocus="true" p:required="true" />
Prezime :<h:inputText class="form-control"

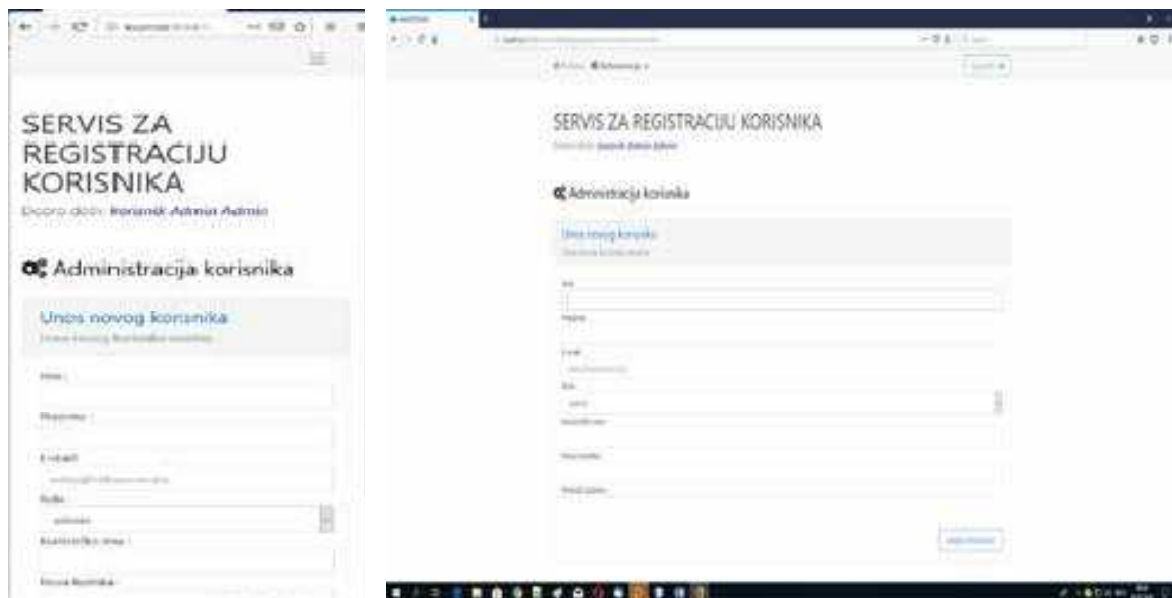
```

```

value="#{LogIn.noviKorisnik.prezime}"p:required="true" />
E-mail:<h:inputText label="email" class="form-control"
value="#{LogIn.noviKorisnik.emailp:placeholder="neko@nekiserver.ba"
p:pattern="[a-z0-9._%+~]+@[a-z0-9.-]+\.[a-z]{2,4}$"
p:required="true" />
Rola :<h:selectOneMenu class="form-control"
value="#{LogIn.noviKorisnik.rola}" >
<f:selectItem itemValue="admin" itemDescription="Administrator"/>
<f:selectItem itemValue="korisnik" itemDescription="Korisnik"/>
</h:selectOneMenu>
Korisničko ime :<h:inputText class="form-control"
value="#{LogIn.noviKorisnik.user}" p:required="true" />
Nova lozinka :<h:inputText class="form-control" value="#{LogIn.newPass}"
p:required="true" />
Potvrdi lozinku :<h:inputText class="form-control "
value="#{LogIn.confirmPass}" p:required="true" /><br/>
<h:commandButton id="btnNoviKorisnik" class="btn btn-outline-primary
float-xs-right" value="UNESI PODATKE!"
action="#{LogIn.unesiNovogKorisnika()}" />
</h:form>
</div>
</div>
</div>
</c:if>
<c:if test="#{LogIn.testRegistracije==false}"><ui:include src="/test/greska.
xhtml"/>
</c:if>
</ui:define>
</ui:composition>

```

Izvršavanjem ovog kôda dobijaju se web stranice prilagodljive ekranima koje korisnik koristi. Tako na slici 10 a) imamo sliku koja se dobija kao prikaz ove web stranice na mobitelu, te na slici 10 b) imamo prikaz koji se dobija na standardnom računarskom monitoru.



Slika 10. a) Prikaz na mobitelu b) prikaz na monitoru

Na ovaj način, korištenjem gotovih Bootstrapovih komponenti, dobila se web aplikacija sa, da tako kažemo, standardiziranim korisničkim interfejsom koji je prilagodljiv svim vrstama uređaja, a što je i bio cilj da programeri ne troše vrijeme na razvoj sopstvenog front-enda, kako bi se na taj način uveliko smanjilo vrijeme razvoja web aplikacije.

ZAKLJUČAK

Web aplikacije su svakim danom sve prisutnije i popularnije, iz razloga što su dostupne sa bilo kog računara i bilo kog mjesta, pa tako i sa mobilnih uređaja koji su već sada standardno opremljeni web pretraživačem. Problem koji se postavlja pred web programere jeste kako što brže i što funkcionalnije kreirati web aplikaciju koja će biti sigurna, efikasna i lagana za korisničku upotrebu. Sam proces razvoja web aplikacija se odvija dijelom u „pozadini“ kao back-end dio razvoja, a dijelom na „prednjoj strani“ kao front-end dio razvoja. Jedan od ključnih problema svakako je rješavanje prilagodljivog, tj. responzivnog web dizajna (RWD) kako bi se omogućio jednostavan pristup stranicama web aplikacije preko različitih uređaja, a naročito za pristup preko pametnih (smart) telefona koji su postali sastavni dio života i čija je upotreba u stalnom porastu. Korištenjem Java programskog jezika kao back-end-a, JavaServer Faces – JSF kao Java frameworka za front-end razvoj na nivou servera i Bootstrap CSS frameworka moguće je ubrzati razvoj web aplikacija. Prednost ovako razvijenih web aplikacija jeste mogućnost pristupa (učitavanja) sa bilo koje platforme. Bootstrap kao RWD framework, sa kolekcijom gotovih komponenti štedi vrijeme razvoja korisničkog interfejsa, što uveliko dovodi do podizanja kvalitete web aplikacije i podiže korisnički ugođaj.

LITERATURA

1. K. Qian, R. Allen, M. Gan, R. Brown: Java Web Development Illuminated, Jonesand Bartlett Publishers, Inc., 2007.
2. Herbert Schildt: Java 2: The Complete Reference, Fifth Edition, The McGraw Hill, 2005.
3. TIOBE Index for October 2017, <https://www.tiobe.com/tiobe-index/>
4. Hazem Saleh, Allan Lykke Christensen, Zubin Wadia: ProJSF and HTML: Building Rich Internet Components, Second Edition, Apress, 2013.
5. David Geary, Cay Horstmann: Core JavaServer Faces, Third Edition, Prentice Hall, 2010.
6. M.Kozić, S. Alagić, A. Džanić: Primjena Facelets template-a u izradi JSF web aplikacije, 11th International Scientific Conference on Production Engineering RIM 2017, Sarajevo, BiH, 2017.
7. Bootstrap 4 Tutorial, https://www.quackit.com/bootstrap/bootstrap_4/tutorial/what_is_bootstrap.cfm
8. Interactive Advertising Bureau: Always On – A Global Perspective of Mobile Consumer Experience, June 2017.
9. Bootstrap 4 Grid System, https://www.w3schools.com/bootstrap4/bootstrap_grid_system.asp, pristupano novembar – decembar 2018.
10. Bootstrap 4: Components, <http://getbootstrap.com/docs/4.0/components/>
11. Ed Burns, Chris Schalk: Java Server Faces 2.0: The Complete Reference, McGraw-Hill, 2010.