

**UPOTREBLJIVOST KORISNIČKOG INTERFEJSA APLIKACIJE RAZVIJENE IONIC OKRUŽENJEM****USABILITY OF THE USER INTERFACE OF AN APPLICATION DEVELOPED IN THE IONIC FRAMEWORK**Ivana Bugarski, *Fakultet tehničkih nauka, Novi Sad***Oblast – RAČUNARSTVO I AUTOMATIKA**

**Kratak sadržaj** – U okviru ovog rada analizirana je upotrebljivost korisničkog interfejsa aplikacije razvijene Ionic okruženjem. Uočene su razlike u performansama aplikacije u zavisnosti od programskog jezika korišćenog u Ionic okruženju pri razvijanju aplikacije.

**Ključne reči:** *Ionic, React, Angular, Vue*

**Abstract** – *This paper analyzes the usability of the user interface of an application developed in the Ionic framework. Differences in the application's performance were observed depending on the programming language used in the Ionic framework during the application's development.*

**Keywords:** *Ionic, React, Angular, Vue*

**1. UVOD**

Porast korišćenja mobilnih telefona među korisnicima izazvao je porast u razvoju mobilnih aplikacija. Ova tendencija je podstakla tražnju za alatima koji omogućavaju kreiranje aplikacija kompatibilnih sa svim platformama, sa ciljem unapređenja kvaliteta mobilnih aplikacija. Pored univerzalnosti, javila se potreba za alatom gde će programeri implementirati delove koda uz pomoć već poznatih radnih okvira. Prethodno navedeno olakšava proces implementacije mobilnih aplikacija.

Tema istraživanja fokusira se na evaluaciju upotrebljivosti korisničkog interfejsa razvijenog uz upotrebu Ionic okruženja. Cilj istraživanja je analizirati performanse i korisničko iskustvo interfejsa, istražiti tehničke aspekte implementacije, kao i pružiti detaljan pregled koraka u izradi istog.

*Ionic* okruženje spada među pet najzastupljenijih radnih okvira za razvoj multiplatformskih aplikacija. Najzastupljeniji okvir je *Flutter*, zatim sledi *ReactNative*, potom *Kotlin*, a nakon njega dolazi *Ionic*. Jedna od ključnih prednosti *Ionic* okruženja u odnosu na prethodno navedena okruženja je njegova sposobnost da koristi poznate veb tehnologije poput *HTML*, *CSS* i *JavaScript* biblioteke, što omogućava brzu adaptaciju programera na razvoj mobilnih aplikacija i olakšava razvoj multiplatformskih aplikacija.

**NAPOMENA:**

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Ivetić, red. prof.

**2. ARHITEKTURA IONIC OKRUŽENJA**

*Ionic* predstavlja otvoreni okvir za razvoj hibridnih mobilnih aplikacija primenom veb tehnologija poput *HTML*, *CSS* i *JavaScript* ili *TypeScript* biblioteke. Ovaj alat omogućuje stvaranje mobilnih aplikacija koje su interoperabilne na više platformi, uključujući *Android*, *iOS* i veb. Dizajniran je da radi i da se prikazuje na svim trenutnim mobilnim uređajima i platformama onako kako je implementiran.

Sadrži gotove komponente, tipografiju i osnovnu temu prilagođenu svakoj platformi. Iako se *Ionic* aplikacije baziraju na veb tehnologijama, pružaju korisnicima iskustvo slično nativnim aplikacijama.

To znači pristup hardverskim funkcijama uređaja, brzo reagovanje kao i glatke animacije. Podržava različite *JavaScript* okvire kao što su *Angular*, *React* i *Vue*, što pruža mogućnost odabira tehnologije od strane razvojnog tima aplikacije.

Kao što je prethodno navedeno, *Ionic* okruženje može koristiti različite radne okvire za implementaciju mobilnih aplikacija, uključujući *Angular*, *React* i *Vue*, što će biti detaljnije obrađeno u nastavku teksta.

Na Slici 1 prikazane su komponente koje učestvuju u izgradnji mobilnih i veb aplikacija. Prilikom korišćenja *Ionic* okruženja za razvoj mobilnih i veb aplikacija, neophodno je implementirati aplikaciju koristeći neku od frontend tehnologija u sklopu *Ionic* okruženja.

Takođe, potrebno je uključiti *Capacitor* koji omogućava pristup nativnim funkcionalnostima uređaja i dodatno doprinosi njegovoj efikasnosti i upotrebljivosti.



Slika 1. Prikaz komponenti za izgradnju mobilnih i veb aplikacija

## 2.1. Angular

*Angular* je *JavaScript* okvir otvorenog koda za izgradnju veb aplikacija. Razvijen je od strane *Google* kompanije i koristi se za razvoj robusnih i skalabilnih aplikacija. *Angular* se oslanja na komponentnu arhitekturu, što znači da se aplikacija sastoji od komponenata koje se mogu ponovo koristiti i lako održavati [1].

## 2.2. React

*React* je biblioteka *JavaScript* programskog jezika koji uz predstavljanje pomoću komponenti doprinosi fleksibilnosti i učinkovitosti. Koristi se pri izradi klijentske strane veb aplikacije. Razvijen je od strane *Meta* kompanije koristeći ga za razvoj svoje aplikacije *Facebook*. Kasnije je primenjen i na *Instagram* aplikaciju, nakon čega je u upotrebi mnogih programera. Veoma je pogodan za izradu mobilnih aplikacija kao i veb sajtova, zbog dinamičkog i interaktivnog prikaza korisnicima [2].

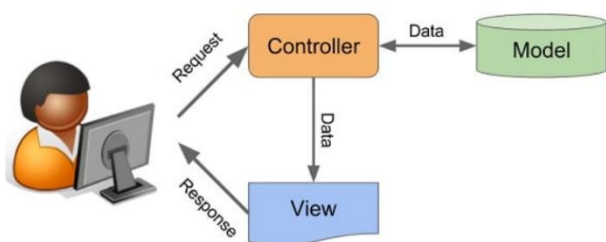
## 2.3. Vue

*Vue* je *JavaScript* okvir za izgradnju korisničkih interfejsa. Korisnički interfejs implementira uz pomoć *HTML*, *CSS* bibliotekama kao i koda implementiranog u *JavaScript* jeziku. Pruža deklarativni komponentni model programiranja sa ciljem da se dobije željeni korisnički interfejs [3].

## 2.4. Ionic

*Ionic* predstavlja moćno okruženje za razvoj hibridnih mobilnih aplikacija koji, u kombinaciji sa radnim okvirima kao što su *React*, *Angular* i *Vue*, omogućava efikasan i efektivan razvoj aplikacija. Bez obzira na odabrani radni okvir, komponente i alati *Ionic* okruženja pružaju potrebnu podršku za kreiranje aplikacija koje zadovoljavaju visoke standarde performansi i korisničkog iskustva. S obzirom na sve ove prednosti, *Ionic* se nameće kao logičan izbor za timove koji žele da razvijaju mobilne aplikacije uz minimalne troškove i maksimalnu efikasnost [4].

*Ionic* primenjuje *MVC* arhitekturu kroz *Angular* za formiranje jednostrane aplikacije koja je upotrebljiva za razne mobilne uređaje. Ovaj pristup razdvaja celu aplikaciju na tri dela, a to su *Model*, *View* i *Controller*. *Model* je odgovoran za održavanje podataka. *View* je odgovoran za prikazivanje podataka korisniku, odnosno deo koji se odnosi na *HTML* i *CSS*. *Controller* kontroliše interakcije između *Model* i *View* delova [5]. Na Slici 2 prikazana je arhitektura *MVC* modela.



Slika 2. Prikaz *MVC* arhitekture

## 3. IMPLEMENTACIJA

Aplikacija koja služi za ilustraciju je jednostavna aplikacija koja omogućava pregled liste korisnika kao i prikaz detalja o svakom korisniku. Takođe, pruža mogućnost ažuriranja podataka o korisniku.

Nakon konfiguracije radnog okruženja i instalacije potrebnih paketa, započinje se proces implementacije aplikacije.

Prethodno je naglašeno da je izgled aplikacije od izuzetne važnosti iz perspektive korisnika. Postoji mnogo biblioteka koje se koriste za uređivanje izgleda aplikacije, a koje se često koriste i u aplikacijama razvijenim isključivo u *React* programskom jeziku. Neke od tih biblioteka su *Material UI*, *Chakra UI* i druge. U implementiranoj aplikaciji korišćena je biblioteka *Chakra UI*, zbog svoje sposobnosti da olakša rad sa brendiranjem i temom aplikacije, kao i to što poseduje veliki broj već implementiranih komponenti. *Chakra UI* nudi različite mogućnosti za unapređenje elemenata aplikacije, što je čini pogodnom za kreiranje vizuelno privlačnih i funkcionalnih korisničkih interfejsa.

U kontekstu *Angular* programskog jezika, kod se može implementirati sa dodatnim koracima koji su karakteristični za ovaj okvir. Da bi se postiglo funkcionalno okruženje za rad aplikacije, neophodno je sprovesti određene konfiguracije, uključujući postavke verzija, rutiranje, podešavanje tema i druge aspekte. Ovi zahtevi se zadovoljavaju putem definisanja odgovarajućih parametara i postavki u više datoteka.

Struktura direktorijuma organizovana je tako da se u *src* direktorijumu nalaze datoteke koje obuhvataju konfiguraciju verzije, pokretanja, rutiranja i druge aspekte aplikacije.

U suprotnosti sa *Ionic* aplikacijom implementiranom uz *React*, kod implementacije sa *Angular* bibliotekom primetna je razlika u organizaciji komponenti. Konkretno, u *React* biblioteci se često sva tri dela *HTML*, *CSS* i funkcionalna logika integrišu unutar jedne komponente, dok su ovi aspekti kod *Angular* biblioteke često razdvojeni na tri odvojene komponente. Razdvajanje ovih elemenata na tri komponente može imati svoje prednosti, uključujući veću preglednost koda, poboljšanu čitljivost, kao i bržu identifikaciju željenih delova koda. Međutim, kada se svi ovi aspekti kombinuju unutar jedne komponente, kao što je slučaj u *React* biblioteci, to može doprineti bržim izmenama, kako u funkcionalnom delu tako i u stilskom.

Dodatno, u *React* biblioteci postoji prednost renderovanja koja omogućava brzo osvežavanje samo delova stranice koji su podložni promenama. Na primer, nakon izmena u kodu koji se odnosi na tabelu, samo će se tabela osvežiti i prikazaće se promene, što doprinosi bržem iterativnom razvoju aplikacije. Opisani princip renderovanja je posebno koristan zbog svoje efikasnosti u procesu implementacije aplikacija, obezbeđujući brže i intuitivnije iskustvo za razvojne timove.

Za precizno merenje performansi implementiranih aplikacija, korišćena je *performance* funkcija u *JavaScript* biblioteci. Ovaj objekat omogućava pristup različitim

metodama i svojstvima koja su ključna za analizu performansi veb aplikacija. *Performance* objekat pruža detaljne informacije o vremenu izvršavanja koda, kao i o performansama mrežnih zahteva i drugih relevantnih aspekata u radu aplikacije u pretraživaču. Ključne funkcije uključuju *performance.now()*, koja se koristi za tačno merenje vremena izvršavanja, kao i svojstva poput *performance.timing()* i *performance.memory()*, koja pružaju informacije o vremenskim tačkama učitavanja resursa i upotrebi memorije aplikacije.

#### 4. ANALIZA UPOTREBLJIVOSTI

U analizi *UI* iz korisnikovog ugla, fokus je na interakciji i doživljaju korisnika tokom korišćenja aplikacije.

Koristeći *Ionic* alat uz *Angular*, *React* ili *Vue*, korisnici mogu primetiti da je *UI* dizajniran sa naglaskom na mobilno iskustvo. Na primer, navigacija je obično jednostavna i intuitivna, sa jasno označenim elementima kao što su naslovi i dugmad. Dodatno, korisnici mogu uočiti brzinu interakcije, što može doprineti pozitivnom doživljaju. Takođe, korisnici mogu ocenjivati i faktore kao što su estetika, organizacija sadržaja, upotrebljivost ikona i intuitivnost navigacije. Dobro postavljeni elementi mogu olakšati korisnicima pronalaženje željenih funkcionalnosti, dok previše složen *UI* može izazvati konfuziju ili frustraciju.

Uzimajući u obzir sve ove faktore, analiza korisničkog interfejsa omogućava da se sagleda kako korisnici doživljavaju i interaguju sa aplikacijom, čime se dobija uvid u potencijalne oblasti za poboljšanje ili optimizaciju korisničkog iskustva.

##### 4.1. Demonstracija

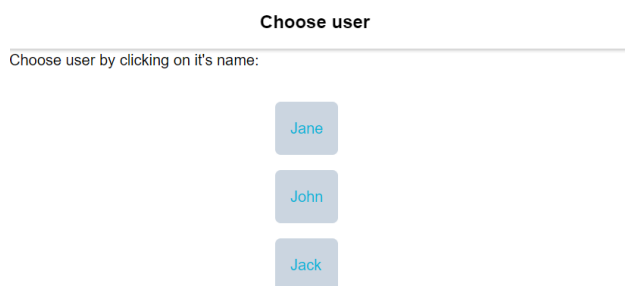
Demonstracija korisničkog interfejsa implementiranog u okruženju *Ionic* uz upotrebu *React*, *Angular* i *Vue* biblioteke predstavljena je na slikama ispod. Pošto je izgled za sve tri aplikacije isti, analiziraće se performanse i korisničko iskustvo sa stanovišta korisnika. Vizuelni elementi poput nijansi boja, veličina i jačine teksta su dosledno primenjeni u svim primerima, ključne razlike se pojavljuju u performansama i odzivu aplikacija tokom njihovog rada. *React*, *Angular* i *Vue* su svi moćni alati za izradu korisničkih interfejsa u okviru *Ionic* platforme, ali svaki od njih ima svoje prednosti i mane.

Prvo je predstavljena početna stranica sa prikazom liste korisnika, gde je svaki korisnik prikazan kao dugme. Klikom na to dugme prelazi se na stranicu sa detaljima o izabranom korisniku. Na stranici sa detaljima o korisniku, omogućeno je ažuriranje prikazanih detalja. Opisani način funkcionisanja korisničkog interfejsa je intuitivan i jednostavan za upotrebu. Korisnici sa minimalnim računarskim iskustvom mogu lako da zaključe da će se detalji odabranog korisnika prikazati nakon selektovanja željenog korisnika.

Dodatno, korisnički interfejs podstiče interakciju time što se kursor menja u oblik pokazivača prstom kada se pređe preko dugmeta sa imenom korisnika, što jasno sugerise da je potrebno pritisnuti dugme kako bi se dobile dodatne informacije. Ovaj pristup dizajnu korisničkog interfejsa naglašava jednostavnost i intuitivnost, omogućavajući

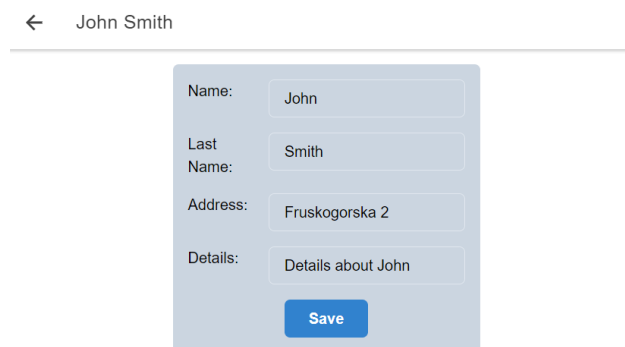
korisnicima da lako i brzo da pristupe potrebnim informacijama.

Na Slici 3 može se primetiti da je dizajn stranice *UserListPage* jednostavan. Stranica sadrži naslov, tekstualni opis i listu sa imenima korisnika. Tekstualni opis detaljno objašnjava šta treba korisnik da uradi kako bi video detalje o željenom korisniku prikazanog u listi. Lista korisnika prikazana je uz pomoć dugmadi, gde svako dugme ima istu nijansu pozadine, istu veličinu kao i isti stil slova kojima su imena korisnika predstavljena. Ukoliko bi se dugmad ili tekst razlikovali, stranica ne bi podržala konzistentnost i jednostavnost. Nije poželjno nemati boje i stil kako je to prikazano na prethodnim slikama, ali i preterana šarolikost može da dovede do nepreglednosti stranice i njenih delova.



Slika 3. Prikaz *UserListPage* stranice

Nakon što je korisnik selektovan, dobija se prelaz na stranicu sa detaljima o selektovanom korisniku. Slika 4 prikazuje stranicu *UserDetailPage* gde su prikazani detalji izabranog korisnika. Na osnovu prikazanog primer izgleda *UserDetailPage* stranice može da se zaključi da detalji prikazani na Slici 4, privlače pažnju i korisnika zadržavaju na stranici. Čine stranicu zanimljivijom i lakšom za gledanje kao i sam rad na njoj.



Slika 4. Prikaz *UserDetailPage* stranice

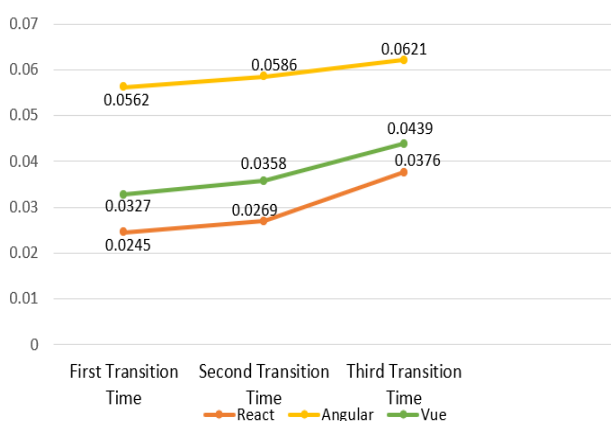
Nakon demonstracije aplikacija, može se zaključiti da izgled ostaje konzistentan bez obzira na to da li je korišćena *React*, *Angular* ili *Vue* biblioteka u *Ionic* okruženju.

Međutim, razlike se uočavaju u performansama sistema. Što se tiče brzine rada aplikacije, najbrža se pokazala aplikacija implementirana pomoću *React* biblioteke, čija se brzina nije menjala ni tokom procesa implementacije. Aplikacija implementirana *Angular* bibliotekom izdvaja se u odnosu na *React* i *Vue* po tome što ima sporije

inicijalno učitavanje zbog svoje kompleksnosti i obuhvatnosti. S druge strane, aplikacija implementirana pomoću *Vue* biblioteke zaostaje za *React* bibliotekom u pogledu brzine renderovanja i responzivnosti.

Nakon analize korišćenjem *performance* objekta, utvrđeno je da je vreme učitavanja *UserDetailPage* stranice nakon klika na dugme sa imenom korisnika bilo 0,0245 sekundi za *React*, 0,0562 sekunde za *Angular* i 0,0327 sekundi za *Vue* biblioteku. Nakon izmene podataka selektovanog korisnika vreme za koje se izvrši izmena podataka za *React* bilo je 0,0269 sekundi, za *Angular* 0,0586 i za *Vue* biblioteku bilo je 0,0358 sekundi. Vreme učitavanja *UserListPage* stranice nakon klika na dugme za vraćanje sa stranice *UserDetailPage* bilo je 0,0376 sekundi za *React*, 0,0621 sekundi za *Angular* i 0,0439 sekundi za *Vue* biblioteku. Na osnovu ovih rezultata može se zaključiti da je implementacija aplikacije u *React* biblioteci pokazala najbrže vreme učitavanja u poređenju sa *Angular* i *Vue* bibliotekama.

Na Slici 5 prikazani su podaci dobijeni korišćenjem *performance* objekta za merenje vremena prelaska između stranica za biblioteke *React*, *Angular* i *Vue*. Početne tačke na grafikonu predstavljaju vreme prelaska sa stranice *UserListPage* na stranicu *UserDetailPage*, dok krajnje tačke prikazuju vreme prelaska sa stranice *UserDetailPage* na stranicu *UserListPage*.



Slika 5. Prikaz podataka dobijenih korišćenjem *performance* objekta

*Ionic* se pokazao kao izuzetno učinkovit alat za izgradnju visokokvalitetnih aplikacija sa glatkim performansama i atraktivnim korisničkim interfejsima. Njegova integracija sa *Capacitor* bibliotekom, koja omogućava pristup nativnim funkcionalnostima uređaja, dodatno doprinosi njegovoj efikasnosti i upotrebljivosti. Ova fleksibilnost i sposobnost da pruži nativno iskustvo korisnicima, čini ga idealnim izborom za mnoge razvojne projekte. Upotreba *Ionic* okruženja ima smisla i sada i u budućnosti, jer kontinuirano unapređuje svoje mogućnosti i prati najnovije trendove u tehnologiji razvoja mobilnih aplikacija. Njegova popularnost je rezultat kombinacije fleksibilnosti, jednostavnosti upotrebe i moćnih funkcionalnosti, što ga izdvaja od ostalih alata na tržištu.

Kao rezultat toga, *Ionic* je popularan među programerima koji žele da kreiraju visoko performansne, multiplatformske aplikacije sa minimalnim naporom.

## 5. ZAKLJUČAK

Korišćenje *Ionic* okruženja kao alata za razvoj mobilnih aplikacija pruža brojne prednosti koje ga čine idealnim za savremene potrebe programera. Njegova sposobnost da omogući razvoj jedne aplikacije za više platformi, uključujući *iOS*, *Android* i veb, značajno smanjuje vreme i troškove razvoja, što je od ključnog značaja u brzorastućem svetu mobilnih tehnologija.

Nakon detaljne analize korisničkog interfejsa aplikacija razvijenih korišćenjem *React*, *Angular* i *Vue* biblioteka unutar *Ionic* okruženja, uočene su značajne razlike u performansama i brzini rada.

U zaključku, *Ionic* se izdvojio kao izuzetno korisno okruženje za razvoj multiplatformskih aplikacija, zahvaljujući svojoj fleksibilnosti, jednostavnosti upotrebe i moćnim funkcionalnostima. Njegova sposobnost da integriše *JavaScript* biblioteke i pristup nativnim funkcionalnostima uređaja čini ga idealnim izborom za mnoge razvojne projekte. Pored toga što ne uspeva da pruži odgovarajuću efikasnost za previše zahtevne aplikacije, *Ionic* pruža efikasan i ekonomičan način za razvoj visokokvalitetnih mobilnih aplikacija sa minimalnim naporom.

## 6. LITERATURA

- [1] Angular, <https://angular.dev/guide/components> (pristupljeno u maju 2024.)
- [2] React, <https://react.dev/learn> (pristupljeno u maju 2024.)
- [3] Vue, <https://vuejs.org/guide/introduction> (pristupljeno u maju 2024.)
- [4] Ionic, <https://ionic.io/blog/announcing-the-ionic-react-beta> (pristupljeno u junu 2024.)
- [5] Priyanka Chaudhary, „International Research Journal of Engineering and Technology (IRJET)“, Student, Department of computer science, ABESIT, Ghaziabad, Volume: 5, pp. 3182, May 2018.

### Kratka biografija:



**Ivana Bugarski** rođena je u Novom Sadu 1999. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Upotrebljivost korisničkog interfejsa aplikacije razvijene Ionic okruženjem odbranila je 2024. godine.  
kontakt: ivanabugarski19@gmail.com