NOTIS.IO - SUSTAV ZA MONITORING KLJUČNIH RIJEČI I EKSTRAKCIJU INFORMACIJA S INTERNETSKIH SJEDIŠTA

Selenić, Luka

Master's thesis / Specijalistički diplomski stručni

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **The Polytechnic of Rijeka / Veleučilište u Rijeci**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:125:839046

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Polytechnic of Rijeka Digital Repository - DR PolyRi

zir.nsk.hr

Image not found or type unknown

VELEUČILIŠTE U RIJECI

Luka Selenić

NOTIS.IO - SUSTAV ZA MONITORING KLJUČNIH RIJEČI I EKSTRAKCIJU INFORMACIJA S INTERNETSKIH SJEDIŠTA

(specijalistički završni rad)

Rijeka, 2016.

VELEUČILIŠTE U RIJECI

Poslovni odjel

Specijalistički diplomski stručni studij Informacijske tehnologije u poslovnim sustavima

NOTIS.IO - SUSTAV ZA MONITORING KLJUČNIH RIJEČI I EKSTRAKCIJU INFORMACIJA S INTERNETSKIH SJEDIŠTA

(specijalistički završni rad)

MENTOR

doc. dr. sc. Alen Jakupović, prof. v. š.

STUDENT

Luka Selenić

MBS: 2422000120/14

Rijeka, rujan 2016.

VELEUČILIŠTE U RIJECI

Poslovni odjel

Rijeka, 06.09.2016.

ZADATAK
za specijalistički završni rad

Pristupniku Luki Seleniću MBS: 2422000120/14

Studentu specijalističkog diplomskog stručnog studija informatike izdaje se zadatak specijalističkog završnog rada – tema specijalističkog završnog rada pod nazivom:

NOTIS.IO - SUSTAV ZA MONITORING KLJUČNIH RIJEČI I EKSTRAKCIJU INFORMACIJA S INTERNET SJEDIŠTA

Sadržaj zadatka: Provesti komparativnu analizu tri postojeća sustava monitoringa ključnih riječi i ekstrakcije informacija s internet sjedišta. Temeljem provedene analize identificirati zahtjeve koje treba zadovoljiti novi sustav. Provesti konceptualno modeliranje novoga sustava te prikazati scenarije i dijagrame uporabe. Izraditi dijagrame aktivnosti, model podataka i korisnička sučelja novoga sustava. Novi sustav opisati i praktično implementirati.

Rad obraditi sukladno odredbama Pravilnika o završnom radu Veleučilišta u Rijeci.

Zadano: 06.09.2016. **Predati do: 15.09.2016.**

Mentor:

(doc.dr.sc. Alen Jakupović)

Pročelnik odjela:

(mr.sc. Marino Golob)

Zadatak primio dana: 06.09.2016.

(Luka Selenić)

I Z J AV A

Izjavljujem da sam specijalistički završni rad pod naslovom **NOTIS.IO - SUSTAV ZA MONITORING KLJUČNIH RIJEČI I EKSTRAKCIJU INFORMACIJA S INTERNET SJEDIŠTA** izradio samostalno i pod nadzorom i uz stručnu pomoć mentora **doc. dr. sc. Alena Jakupovića**.

Luka Selenić



(potpis studenta)

SAŽETAK

Svrha i cilj ovog diplomskog rada je istražiti postoje li sustavi koji bi olakšali praćenje željenih ključnih riječi među informacijama i sadržajima koji se pojavljuju na raznim internetskim sjedištima. Nakon istraživanja želi se postaviti novo idejno rješenje koje bi ispravilo nedostatke istraženih sustava. U radu je opisan tijek istraživanja i njegovi rezultati. Prati se analiza i dizajn idejnog rješenja, odnosno idejnog sustava, te se opisuje njegov razvoj, koje se tehnologije koriste i u koju svrhu.

Ključne riječi*:* izvlačenje podataka, praćenje podataka, informacije, podaci, internet, sustav, notis

SADRŽAJ

1. UVOD 1
1.1. Problem i predmet istraživanja 1
1.2. Radna hipoteza 2
1.3. Svrha i cilj istraživanja 2
1.4. Znanstvene metode 2
1.5. Struktura rada 2
2. ISTRAŽIVANJE 4
2.1. Web scraping i media monitoring 4
2.2. Način provođenja istraživanja 4
2.3. Definiranje potreba korisnika 5
2.4. Odabir aplikacija za analizu 7
2.5. Mediatoolkit 8
2.5.1. Analiza 10
2.5.2. Zaključak 12
2.6. Import.io 12
2.6.1. Analiza 15
2.6.2. Zaključak 17
2.7. Mention.com 17
2.7.1. Analiza 20
2.7.2. Zaključak 21
2.8. Rezultati istraživanja 22
3. ANALIZA SUSTAVA 23
3.1. Konceptualni model 23

3.2. Opis sustava 25
3.2.1. Akteri 26
3.3. Use case dijagram 27
3.4. Opis scenarija 28
3.4.1. Registracija 28
3.4.2. Prijava 28
3.4.3. Upravljanje korisničkim podacima 29
3.4.4. Upravljanje praćenja ključnih riječi 29
3.4.5. Upravljanje praćenja internetskih stranica 32
3.4.6. Upravljanje praćenja RSS kanala 35
3.4.7. Upravljanje ključnim riječima 37
3.4.8. Upravljanje rezultatima izvlačenja s internetskih sjedišta 39
3.4.9. Upravljanje rezultatima izvlačenja s RSS kanala 40
3.4.10. Upravljanje korisnicima 40
3.4.11. Mobilna aplikacija - prijava 42
3.4.12. Mobilna aplikacija – Pregled rezultata izvlačenja s internetskih sjedišta 43
3.4.13. Mobilna aplikacija – Pregled rezultata izvlačenja s RSS sjedišta 43
3.4.14. Mobilna aplikacija – Upravljanje obavijestima 44
4. DIZAJN SUSTAVA 45
4.1. Relacijski model baze podataka 45
4.2. Dinamičko modeliranje sustava – dijagram aktivnosti 49
4.2.1. Registracija 49
4.2.2. Prijava 50
4.2.3. Uređivanje korisničkih podataka 51

4.2.4. Uređivanje praćenja ključnih riječi 52
4.2.5. Upravljanje praćenja internetskih stranica 57
4.2.6. Upravljanje praćenja RSS kanala 62
4.2.7. Upravljanje ključnim riječima 67
4.2.8. Upravljanje rezultatima izvlačenja s internetskih sjedišta 68
4.2.9. Upravljanje rezultatima izvlačenja s RSS kanala 69
4.2.10. Upravljanje korisnicima 70
4.2.11. Prijava korisnika preko mobilne aplikacije 71
4.2.12. Pregled rezultata izvlačenja s internetskih sjedišta preko mobilne aplikacije . 72
4.2.13. Pregled rezultata izvlačenja s RSS kanala preko mobilne aplikacije 73
4.2.14. Pregled obavijesti preko mobilne aplikacije 74
4.3. Mockup grafičkog sučelja sustava 75
4.3.1. Prijava 75
4.3.2. Registracija 75
4.3.3. Uređivanje korisničkih podataka 76
4.3.4. Upravljanje monitorima i rezultatima 76
4.3.5. Dodavanje, uređivanje monitora 77
4.3.6. Upravljanje ključnim riječima 77
4.3.7. Upravljanje korisnicima 78
4.3.8. Dodavanje, uređivanje korisnika 78
4.3.9. Mobilna aplikacija – izbornik 79
4.3.10. Mobilna aplikacija – Pregled rezultata monitora 79
4.3.11. Mobilna aplikacija – Uređivanje rezultata 80
5. OPIS SUSTAVA 81

5.1. Struktura sustava 81
5.2. Tehnologije 82
5.3. Razvoja okolina 83
5.4. Administracijsko sučelje 84
5.4.1. Migracije 86
5.4.2. Modeli 86
5.4.3. Kontroleri 87
5.4.4. Pogledi 88
5.5. Izgled administracijskog sučelja 89
5.6. Notifikacijski server 92
5.7. Aplikacija za ekstrakciju i provjeru ključnih riječi 93
5.7.1. KeywordMonitorRunner 94
5.7.2. WebMonitorRunner 96
5.7.3. RSSMonitorRunner 97
5.8. Pomoćni server 98
5.9. Mobilna aplikacija 100
5.9.1. Servisi 101
5.9.2. Kontroleri 101
5.9.3. Nox App Player 103
6. ZAKLJUČAK 106
7. POPIS LITERATURE 107
8. POPIS KORIŠTENIH KRATICA 108
9. POPIS SLIKA 109
10. POPIS TABLICA 112

1. UVOD

Na početku svakog istraživačkog rada potrebno je definirati što se i iz kojeg razloga ide istraživati. Potrebno je definirati problem, odnosno predmet istraživanja gdje se ukratko definira i objašnjava motiv istraživanja, a svako istraživanje izvode se koristeći neke od znanstvenih metoda. Definira se hipoteza prije istraživanja, ili više njih, odnosno subjektivno mišljenje i pretpostavka o odabranom predmetu istraživanja, a istraživanjem će se pokazati je li hipoteza točna ili netočna. Navodi se svrha i cilj istraživanja gdje se opisuje što se ovim istraživanjem želi postići i čemu bi ona pridonijela.

1.1. Problem i predmet istraživanja

Informacija je oduvijek imala visoku vrijednost i krucijalna je u gotovo svakoj profesiji, poslu ili zanimanju, a informacija dobiva još više na vrijednosti ako je pravovremena i točna. Internet je u 21. stoljeću postao izvor informacija broj jedan (Internet most popular information source: poll, 2009) zbog velike brzine objava, jednostavnosti pristupa, distribuciji i pronalasku informacija. Svatko ima određene i različite interese, stoga u informatici problem ne predstavlja samo točnost informacije, već i pravovremeni pronalazak željenih informacija. Problem točnosti teško je rješiv bez ljudske intervencije, no pravovremeni pronalazak željenih informacija u informatici bi bio rješiv automatizacijom. Automatizacija se u informatici stvara izradom aplikacija koje će imati određenu programsku logiku putem koje će se potencijalnom korisniku omogućiti da dobije željenu informaciju u što kraćem roku. Potencijalan korisnik treba znati odakle crpiti, odnosno gdje bi eventualno mogao pronaći željene informacije, a to znanje stječe iskustvom. Potencijalan korisnik korištenjem određenih internetskih servisa i sjedišta dobiva povjerenje autora informacija ukoliko su mu informacije koje se na njima dijele bile korisne, čime on stvara svoj izvor informacija. Ovisno o zanimanju, korisniku je dostava informacije također važna, jer ukoliko zbog svog zanimanja nema previše vremena tražiti željene informacije samostalno, to za njega mora učiniti netko drugi. Razvojem tehnologije, pametni telefoni u današnje vrijeme imaju mogućnost spajanja na širokopojasni, superbrzi Internet, a koristi ga sve više korisnika na svojim pametnim telefonima.

1.2. Radna hipoteza

Pretpostavka je da se može izgraditi sustav koji će s korisnički definiranih internetskih sjedišta moći izdvajati željene informacije, prezentirati ih korisniku te ga obavijestiti ukoliko se te informacije poklapaju s korisnički definiranim postavkama.

1.3. Svrha i cilj istraživanja

Cilj ovog istraživanja je istražiti postoji li gotov sustav koji će korisniku omogućiti da s željenih internetskih sjedišta može izdvajati željene informacije te koji će mu te informacije prikazati na što jednostavniji način. Također, želi se vidjeti sadrži li taj sustav mogućnost obavještavanja korisnika ukoliko te izdvojene informacije sadrže ključne riječi koje definira sam korisnik. Želja za takvim sustavom proizlazi iz potrebe za dobivanjem željenih informacija u što kraćem roku bez intervencije korisnika s tim da on bude obaviješten ako sustav pronađe korisniku relevantne informacije, gdje god da korisnik bio. To bi značilo da informacije korisnik treba dobivati i kada nije uz računalo, odnosno da te informacije dobiva na svoj pametni telefon. Cilj ovog rada je ponuditi idejno rješenje (u nastavku „idejni sustav“) za navedeni problem ukoliko istraženi sustavi ne sadrže neke od traženih mogućnosti. U tom slučaju, sustav će biti izrađen putem programskih jezika za rad u „oblaku“, jer na taj način korisniku se omogućava da što prije dođe do željenih informacija. Svrha je pronalazak ili izgradnja sustava koji će na što jednostavniji i brži način korisnicima omogućiti dolazak do željenih informacija.

1.4. Znanstvene metode

U radu će biti korištene metode analize kod istraživanja postojećih tehnologija, sinteze kod prikazivanja analize sustava, kao i deskripcije koja će se koristiti u prezentiranju rezultata istraživanja i prezentiranju ponuđenog idejnog rješenja, odnosno sustava.

1.5. Struktura rada

U prvom poglavlju rada opisuje se proces istraživanja postojećih sustava i tehnologija koje se dotiču teme ovog rada. Opisuje se što se istražuje i na koji način, kako se došlo do predmeta istraživanja. Opisuju se predmeti istraživanja, prezentira se njihova analiza i zaključak,

dok se na kraju prezentiraju rezultati istraživanja. Drugo poglavlje rada predstavlja analizu idejnog sustava prolaskom kroz njegov konceptualni model, dizajn idejnog sustava prikazom relacijskog modela baze podataka, dijagrama aktivnosti i dizajnom korisničkog sučelja. Rad sadrži i generalan opis svakog dijela idejnog sustava.

2. ISTRAŽIVANJE

U ovom poglavlju istraživačkog rada kreće se s istraživanjem već postojećih rješenja koje nude funkcionalnosti kako bi se dokazala hipoteza. Stoga, potrebno je prvo definirati što će se analizirati kod postojećih tehnologija, odnosno sustava. Prije istraživanja potrebno je upoznati s informatičkim izrazima koji se koriste za pronalazak i izvlačenje informacija: *web scraping* i *web monitoring*.

2.1. Web scraping i media monitoring

Engleski izraz „*web scraping*“ sastoji se od riječi „struganje“ i „mreža“, a služi kao naziv za automatiziranu aktivnost izvlačenja određenog dijela informacija s internetskih sjedišta koje su potrebne korisniku. Izvodi se putem posebno izrađenih aplikacija koje samostalno algoritmima izdvajaju dijelove teksta na internetskim sjedištima te ih prezentiraju korisniku ili spremaju u bazu podataka. (Web scraping, 06.07.2016.)

Engleski izraz „*media monitoring*“ sastoji se od riječi „medij“ i „praćenje“, a služi kao naziv za automatizirane aktivnosti praćenja aktivnosti objava sadržaja na različitim medijima, forumima i društvenim mrežama te njihovo izvlačenje i prezentiranje korisniku putem posebno izrađenih aplikacija. (Social Media Monitoring, 06.07.2016.)

U slučaju ovog rada, koriste se obje aktivnosti.

2.2. Način provođenja istraživanja

Istraživanje se vodilo putem Interneta koristeći tražilicu Google. Pošto se radi o istraživanju servisa o praćenju internetskih sjedišta i servisa za ekstrakciju podataka ključne riječi za pretragu su bile *web* (hrv. Internet), *monitor* (hrv. praćenje), *scraping* (hrv. struganje) i *extract* (hrv. izvlačenje, ekstrahiranje) te kombinacija tih ključnih riječi: *web monitor*, *web monitoring*, *web extractor*. Kod rezultata najčešće su se pojavljivali razni blogovi i članci, no Google je ponudio i sama internet sjedišta potencijalnih kandidata za predmet istraživanja, odnosno Internet aplikacija.

Posjetom tih Internet aplikacija radi se uvid u njegove mogućnosti (engl. *Features*) i odabiru sa samo one aplikacije koje se poklapaju s definiranim potrebama korisnika.

2.3. Definiranje potreba korisnika

Prije samog istraživanja, potrebno je odrediti kriterije, odnosno mogućnosti koje aplikacije moraju imati, a kriteriji se definiraju po potrebama korisnika. Pošto ovakve aplikacije imaju široku namjenu, teško je odrediti potrebe svakog korisnika ponaosob, stoga će se u kriterije uvažavati funkcionalnosti koje općenito imaju pomoćni karakter kod svakog korisnika aplikacije.

Informacija kao takva bitna je u mnogim profesijama, kao na primjer u novinarstvu. Osobe koje vode vlastite internetske portale, ali i velike novinarske kuće u stalnoj su potrazi za informacijama koje bi bile zanimljive njihovim čitateljima. Logički, kako bi bili konkurentniji bitna im je pravovremenost informacije, odnosno da određenu informaciju imaju prije svoje konkurencije. Isto vrijedi i za npr. ljude koji se bave nekretninama - praćenje cijena na tržištu, tj. pregled ponuda nekretnina na oglasima krucijalno je za uspjeh u ovom zanimanju. Slijedi i potreba praćenja vlastitog brenda, gdje tvrke mogu pratiti gdje se i koliko učestalo spominje njihov brend i u kojem kontekstu.

Danas se mnogo razgovora vodi preko Interneta pa tako se spominju i brendovi i tvrtke. Za mnoge to ima veliko značenje jer na taj način mogu pratiti svoju reputaciju na internetskim sjedištima, društvenim mrežama poput *Facebooka* i *Twittera*, ali i internetskih servisa poput *Youtubea*.

Stoga, svatko tko bi htio pratiti informacije na internetu teži slijedećim stavkama: pravovremenost, dostupnost, mobilnost, obaviještenost, relevantnost.

- **Pravovremenost** – korisnik može dobiti informaciju što je prije moguće kako bi bio korak ispred svoje konkurencije.

- **Dostupnost** – korisnik može dobiti informaciju na određeno mjesto, odnosno da mu se sve željene informacije agregiraju na istu lokaciju za skladištenje svih informacija.

- **Mobilnost** – korisnik može svojim informacijama pristupiti s više različitih uređaja – pametni telefoni, tableti, stolna i prijenosna računala.

- **Obaviještenost** – korisnik može automatski biti obaviješten kada se pronađe određena informacija.

- **Relevatnost** – korisnik može sam definirati izvore informacija i što želi u njima tražiti

Iz navedenih stavki može se dobiti nekoliko kriterija koje će se analizirati u potencijalnim aplikacijama:

- Pretraga informacija odvija se u određenim intervalima. Korisniku nije važno koliki su intervali, već mu je bitno da željenu informaciju dobije što prije (**Intervalna pretraga**).

- Radi jednostavnosti pregleda dobivenih informacija, one bi trebale biti agregirane na jednu lokaciju, odnosno aplikaciju, preko koje korisnik jednostavno može imati uvid u njih uz mogućnost pretrage pronađenih informacija te označavanje onih informacija koje je pročitao (**Agregirane informacije**).

- Korisnik mora imati mogućnost samostalno dodavati izvore iz kojih da crpi informacije putem određenog sučelja. Ukoliko su te informacije poveznice na druge informacije, tada bi korisnik trebao imati mogućnost odlaska na te druge informacije (**Dodavanje izvora**).

- Korisnik mora imati mogućnost samostalno dodavati ključne riječi koje želi pretraživati među informacijama koje dobiva te mogućnost administriranja istih (**Dodavanje ključnih riječi**).

- Mogućnost integracije i podrške za postojeće tehnologije za distribuciju informacija (RSS) bi bile poželjne. Korisnik bi tada trebao imati mogućnost pregleda informacija dobivenih putem tih tehnologija te mogućnost pretrage željenih ključnih riječi u njima (**Podržava tehnologije za distribuciju informacija**).

- Korisnik bi trebao imati mogućnost izvlačenja informacija iz onih izvora koje nemaju tehnologije za distribuciju informacija (RSS). Trebao bi imati mogućnost definiranja koje dijelove izvora informacija bi htio izvlačiti s kojih stranica. Svaka izvučena informacija bi mu trebala dolaziti na predviđeno mjesto, odnosno aplikaciju, gdje se agregira, uz mogućnost

označavanja pročitanih, tj. iskorištenih informacija (**Izvlačenje informacija s posebnih internet sjedišta**).

- Korisnik bi trebao imati mogućnost pretraživanja željenih ključnih riječi na željenim internetskim sjedištima (**Pretraživanje ključnih riječi**).

- Korisnik bi kod svakog pronalaska ključne riječi trebao dobivati obavijesti na različite uređaje. Poželjna je podrška za računala, tablete i mobilne telefone radi što veće pokrivenosti. (**Obavijesti**)

- Poželjna je mogućnost za podršku istovremenog rada više korisnika nad aplikacijom te da se sve funkcionalnosti dijele među njima, čak i postavke (**Višekorisnička aplikacija**).

Navedene funkcionalnosti biti će prikazane u tablici, koja će biti osjenčana zelenom bojom ako aplikacija podržava određenu funkcionalnost ili crvenom ako ne podržava.

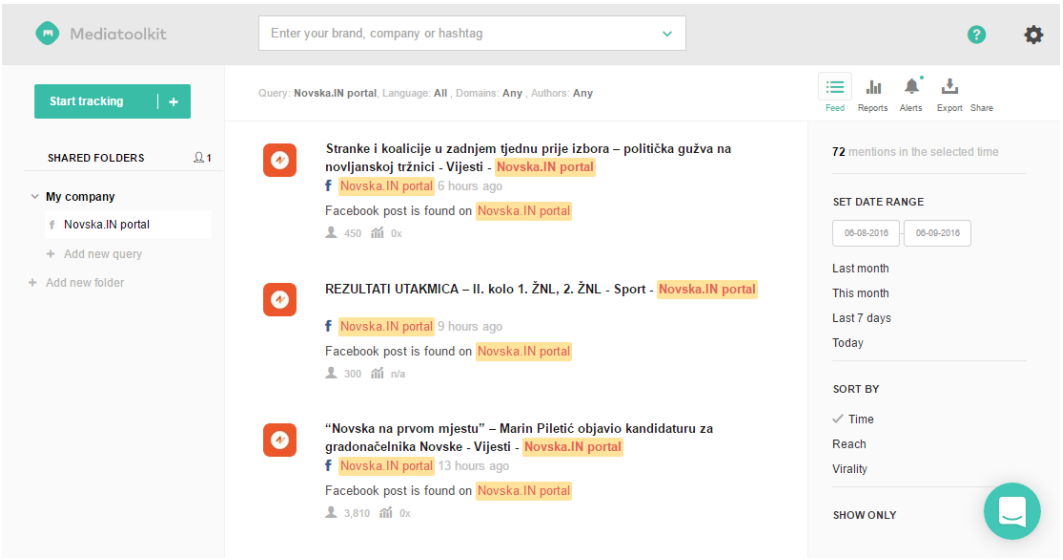
2.4. Odabir aplikacija za analizu

Odabir aplikacija za analizu vrši se na temelju istraživanja najpopularnijih alata, a analizirati će se tri aplikacije. Najpopularniji alati određuju se na uzorku od 10 internetskih sjedišta koje ocjenjuju ili rangiraju aplikacije koje služe za izvlačenje podataka ili praćenjem aktivnosti na internetskim sjedištima, a onaj koji se najviše spomenut odabran je za analizu.

2.5. Mediatoolkit

Poveznica: https://www.mediatoolkit.com (06.09.2016.)

Slika 1. – Sučelje Mediatoolkit aplikacije



(Izvor: Autor)

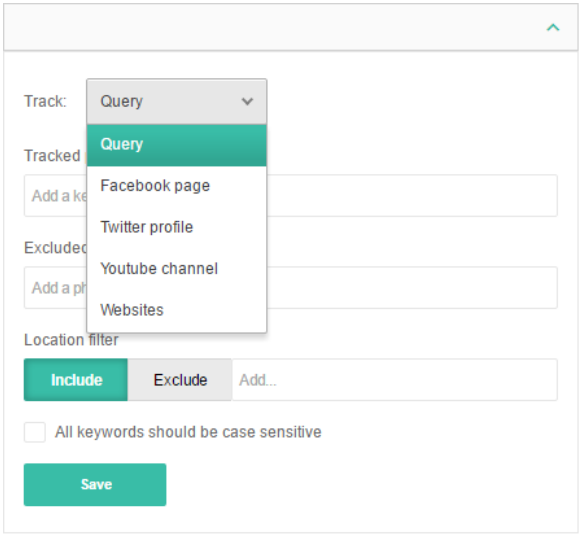
Prva aplikacija za analizu je aplikacija koja se bavi *media monitoringom*, domaći proizvod je tvrtke *Degordian*. Usluga koju nudi ova aplikacije je praćenje nekoliko milijuna internetskih sjedišta i društvenih mreža u realnom vremenu. Omogućava filtriranje i prikazivanje samo relevantnih informacija koje se u aplikaciji nazivaju „pojmovi" (engl. *mentions*), a to je omogućeno putem naprednih upita za pretraživanje. Korisnici alata mogu saznati najvažnije nove vijesti brzim pregledom, mogu pratiti i analizirati medijsku prisutnost konkurencije i analizirati reputaciju koju brend ili tvrtka korisnika ima na internetu.

Osim što nudi i mogućnosti praćenja sadržaja, nudi i analitičke mogućnosti pa tako korisnici mogu otkriti trendove, utjecaje, informacije koje bi ih navele na poboljšanje rada ili ih upozorile na potencijalnu krizu. Tim korisnici imaju mogućnost pronalaska najutjecajnijih medija ili ljudi u svojoj grani zanimanja, stupiti u kontakt s njima ili pak pronaći uzrok i rješavati krizne situacije vezane za njihov brand ili posao.

Od izvora ova aplikacija nudi nekoliko milijuna internetskih sjedišta, društvene mreže poput *Facebooka, Twittera, Youtubea*, raznih blogova i foruma, no nudi i mogućnost dodavanja vlastitih izvora. Svaki izvučeni podaci ostaju spremljeni u arhivi, a korisnik ima mogućnost dobivanja obavijesti o pronalasku relevantnih informacija.

Sadrži bogat set alata za obradu nađenih podataka koji se mogu pronaći, tj. filtrirati koristeći *boolean* operatore za pretraživanje (OR, AND itd.) te varijabilne filtere. Rezultati se prikazuju u realnom vremenu – sve u jednom pogledu, odnosno na jednom mjestu, a nudi se i nekoliko opcija za sortiranje rezultata. Prilikom unosa alat već nudi sučelje za stvaranje upita i pretrage informacija nad arhivom, *Facebook*, *Twitter*, *Youtube* te korisnički definiranim stranicama.

Slika 2. – Sučelje za stvaranje upita



(Izvor: Autor)

Obavještavanje o pronađenim podacima moguće je putem e-maila u nekoliko intervala (dnevni i tjedni pregled u definirano vrijeme) te u realnom vremenu uz podršku za *Android* i *iOS* uređaje.

Svi podaci mogu biti izvedeni u izvješća (*Microsoft Excel* i PDF).

Nudi se mogućnost besplatnog, probnog korištenja aplikacije u trajanju od 14 dana, izradu 100 istovremenih upita i pretragu do 50000 pronađenih relevantnih informacija. Korisnici se u svakom trenutku mogu kupiti godišnju ili mjesečnu pretplatu koju mogu krojiti po svojim potrebama.

Registracija je brza, jednostavna i besplatna. Aplikaciju krasi brzo i jednostavno za korištenje sučelje, gdje je sve „na dohvat ruke“. Upiti su brzi, nasumce odabrani upiti vrlo brzo daju rezultate. Sučelje je responzivno pa se može koristiti i preko mobilnih uređaja.

2.5.1. Analiza

Intervalna pretraga	**Agregirane informacije**	**Dodavanje izvora**	**Dodavanje ključnih riječi**	**Pretraživanje ključnih riječi**
Izvlačenje informacija s posebnih internet sjedišta		**Podržava tehnologije za distribuciju informacija**	**Obavijesti**	**Višekorisnička aplikacija**

Tablica 1. – Analizirane funkcionalnosti Mediatoolkit aplikacije

Intervalna pretraga – aplikacija intervalno pretražuje unesene upite, no nigdje nije specificirano u kolikim intervalima se upiti osvježavaju. Ovim je korisnička potreba za pravovremenosti dobivanja informacija u potpunosti zadovoljena.

Agregirane informacije – aplikacija sve upite i informacije koje vraćaju uredno prikazuje u jednom prozoru jer su pohranjene u bazu podataka, a korisnici mobilnih uređaja putem posebnih aplikacija te upite također jednostavno mogu vidjeti. Pošto se rezultati dodatno mogu filtrirati ovim je korisnička potreba za dostupnosti potpuno zadovoljena, kao i potreba za mobilnosti.

Dodavanje izvora – aplikacija nudi mogućnost korisniku da dodaje vlastite izvore prilikom stvaranja upita, iako aplikacija ključne riječi također traži i u arhivi drugih internetskih sjedišta koje su dodali drugi korisnici, čime se poboljšava pronalazak relevantnih podataka za korisnika. Ovim je zadovoljena korisnička potreba za relevantnosti podataka.

Dodavanje ključnih riječi – aplikacija nudi mogućnost korisniku da dodaje ključne riječi, odnosno pojmove koje želi tražiti. Pojmovi se dodaju vrlo jednostavno, a korisnik može unijeti i pojmove koje želi izbjeći u rezultatima upita. Ovim je dodatno zadovoljena korisnička potreba za relevantnosti podataka.

Pretraživanje ključnih riječi – dodana ključna riječ odmah se pretražuje u arhivi aplikacije, a s filtrima se dodatno namješta ishod rezultata. Ovim je dodatno zadovoljena korisnička potreba za relevantnosti podataka. Problem jedino predstavlja što unos ključne riječi nad određenim internetskim sjedištem i dalje pretražuje čitavu arhivu, ne samo navedeno internetsko sjedište.

Izvlačenje informacija s posebnih internet sjedišta – aplikacija posjeduje bogatu arhivu već dodanih internetskih sjedišta, no korisnik ima mogućnost dodavanja novih Internet sjedišta. Jednom dodani, aplikacija će svojim algoritmima prepoznati naslove na dodanoj stranici te će ih ponuditi kao rezultate upita. Nakon tog, korisnik može tražiti ključne riječi u tim naslovima. Ovim je dodatno zadovoljena korisnička potreba za relevantnosti podataka.

Podržava tehnologije za distribuciju informacija – dodavanjem RSS kanala u formu za unos upita aplikacija ju doda u arhivu, no ne daje rezultate, stoga ne podržava čitanje RSS kanala. Ovim na žalost korisnik gubi na širini izvora iz koje može tražiti željene podatke, odnosno djelomično gubi dio relevantih podataka.

Obavijesti – aplikacija omogućava obaviještavanje korisnika o pronađenim upitima, odnosno pojmovima. Korisnici aplikacije na internetu obaviješteni su u realnom vremenu s osjenčanim imenom upita u izborniku. Korisnici mobilne aplikacije putem „*push*“ notifikacija dobivaju obavijesti u realnom vremenu kada je pronađen određen upit, tj. pojam. Može se

odabrati i dnevne ili tjedne obavijesti putem e-maila. Ovim je zadovoljena potreba korisnika za mobilnosti, pravovremenosti i obaviještenosti.

Višekorisnička aplikacija – jednom kada se korisnik registrira on ima mogućnost pozvati druge osobe u aplikaciju te im dodijeliti određenu ulogu (administrator, analitičar, urednik ili samo gledatelj). Na taj način omogućeno je više korisnika da istovremeno koriste već unesene upite i pojmove.

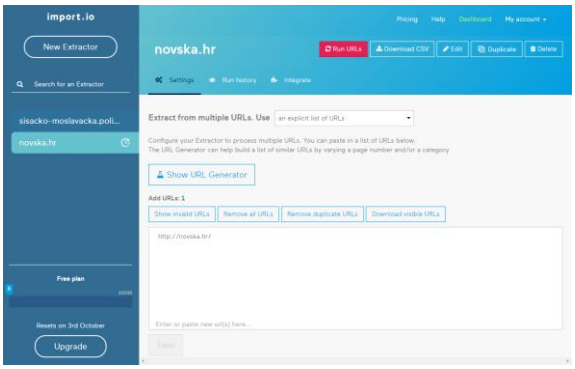
2.5.2. Zaključak

Odlična aplikacije domaće proizvodnje koja zadovoljava sve potrebe za praćenjem pojmova i ključnih riječi na internetskim sjedištima. Jednostavnost korištenja, velik broj sjedišta koje se već prate, kao i brzina dobivanja rezultata samo su neke od mogućnosti koje krase ovu aplikaciju. Praćenje trendova od strane razvojnog tima omogućava praćenje popularnih društvenih mreža i internetskih servisa čime se dodatno proširuje spektar relevantnih rezultata. Podrška za tehnologije poput RSS-a i čitanje JSON izvora bile bi odličan dodatak već postojećim alatima. Stoga ova aplikacija se može koristiti i za izvlačenje podataka, kao i za praćenje podataka.

2.6. Import.io

Poveznica: https://import.io (06.09.2016.)

Slika 3. – Sučelje Import.io aplikacije



(Izvor: Autor)

Druga istražena aplikacija prilikom pretraživanja potencijalnih predmeta za analizu pokazala se najzastupljenijom, gdje je na uzorku od 10 internetskih sjedišta bila spomenuta na svim sjedištima. Aplikacija je orijentirana izvlačenju podataka s internetskih sjedišta i njihovo prikazivanje korisniku. Analizom se ustvrdilo da aplikacija služi za kreiranje API-a, odnosno programskih izvora podataka različitog formata na temelju algoritamske obrade drugih podataka s internetskih sjedišta. To bi značilo da korisnik može izvući podatke iz određenih izvora, oblikovati ih po svojoj želji te napraviti pristup prema tim obrađenim podacima u različitim formatima, tako da ih mogu čitati druge aplikacije. Obrađeni podaci tada se mogu koristiti za analize, vizualizacije podataka, korištenje u računalnim ili mobilnim aplikacijama te znanstvenim istraživanjima.

Za kreiranje pristupa nije potrebno znanje programiranja, već se sve bazira na principu odabira mišem.

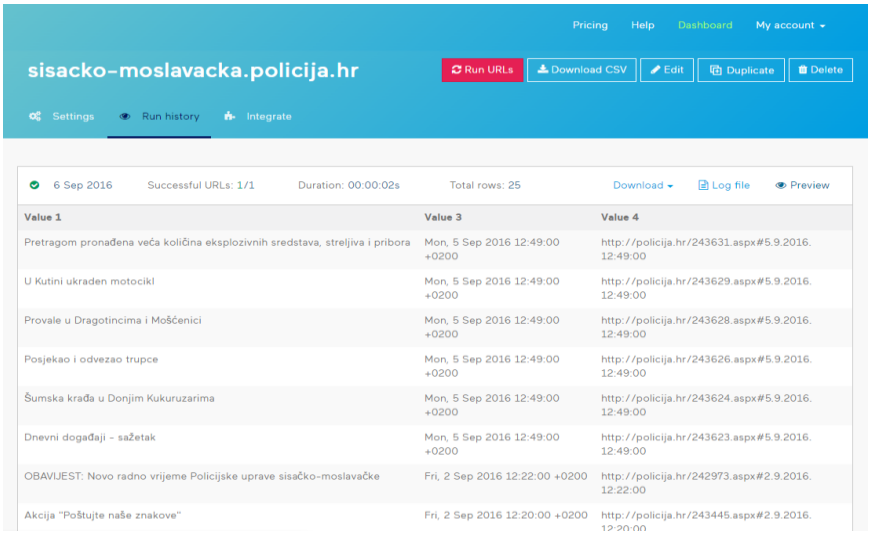
Aplikacija podržava internetska sjedišta koja su potpuno bazirana na Javascript programskom jeziku, sadrži mogućnosti tempiranja ekstrakcija te obavijesti prilikom pronalaska promjena u praćenim podacima.

Registracija je brza, jednostavna i besplatna, a aplikacija se besplatno može koristiti do 10000 upita na mjesec. Ukoliko se korisnik odluči za više, pretplatom na mjesečnoj razini može dobiti 50,000, 100,000 ili 1,000,000 upita na mjesec, uključujući i potpunu programsku podršku. Naravno, nudi se i mogućnost kreiranja vlastitog paketa usluga po posebnim cijenama. Jedan upit jednak je jednom izvlačenju podataka s nekog internetskog sjedišta.

Sučelje aplikacije je moderno i intuitivno. Kreiranje novog izvlačenja podataka je u prvom planu te je napravljeno u stilu „čarobnjaka“. Unosom internetskog sjedišta odakle se želi izvlačiti podatke aplikacija samostalno pokušava odrediti relevantne podatke u čemu je vrlo dobra. Ukoliko to ne može izvršiti automatski, korisniku je ponuđeno željeno internetsko sjedište gdje onda on jednostavnim klikom miša odabire dijelove sjedišta koji ga zanimaju. Jednom kada aplikacija pronađe željene informacije, korisnik ih dobiva u obliku tablice. Dobra funkcionalnost je što aplikacija prepoznaje i slike te i njih uključi u samu tablicu rezultata. Također, korisnik ima

mogućnost dodavanja novih, vlastitih stupaca u tablicu gdje opet ima mogućnost odabira izvora podataka klikom miša na željenom internetskom sjedištu.

Slika 4. – Izgled tablice izvučenih podataka u Import.io



(Izvor: Autor)

Pri završetku odabira podataka, korisniku je predstavljen novoizrađeni sustav za izvlačenje željenih podataka. On tada ima mogućnost kreiranja novog sustava za izvlačenje ili prilagodbe postojećeg. Ima mogućnost pokretanja ponovnog izvlačenja podataka, pregled povijesti izvlačenja ili pak integracije sustava u druge sustave (API). Povijest izvlačenja korisniku omogućava preuzimanje tadašnjih rezultata u određenom formatu (JSON, CSV) ili strukturirani pregled tih podataka. Posjeduje i moćnu integraciju novokreiranog sustava u druge sustave pa tako nudi poveznice na izvor podataka (API) koji se uživo generira, izvoz u uslugu **Google Sheets**, generiranje CSV dokumenata ili pak posebno napravljenog RSS kanala za novokreirani sustav.

Aplikacija nudi izradu više sustava za izvlačenje podataka, koje se jednostavno odabiru i pretražuju s lijeve strane sučelja aplikacije.

Bitno je napomenuti da besplatna verzija aplikacije ne nudi mogućnost postavke intervala pretraživanja promjena. Iako na internetskom sjedištu aplikacije navode mogućnost obavijesti

korisnika o promjenama u rezultatima izvlačenja, postavke obavijesti korisnika u sučelju aplikacije nisu navedene, kao ni u bazi znanja i njihovom sjedištu za podršku.

2.6.1. Analiza

<table>
<tr><th>Intervalna pretraga</th><th>Agregirane informacije</th><th>Dodavanje izvora</th><th>Dodavanje ključnih riječi</th><th>Pretraživanje ključnih riječi</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="2">Izvlačenje informacija s posebnih internet sjedišta</td><td>Podržava tehnologije za distribuciju informacija</td><td>Obavijesti</td><td>Višekorisnička aplikacija</td></tr>
<tr><td colspan="2"></td><td></td><td></td><td></td></tr>
</table>

Tablica 2. – Analizirane funkcionalnosti Import.io aplikacije

Intervalna pretraga – aplikacija, odnosno svaki kreirani sustav za izvlačenje u korisnički specificiranim intervalima izvlači podatke iz definiranih izvora. Korisnici besplatne verzije aplikacije nemaju mogućnost definiranja intervala kojim žele da se odabrani sustav osvježava. Ovim je korisnička potreba za pravovremenosti dobivanja informacija u potpunosti zadovoljena.

Agregirane informacije – aplikacija izvučene podatke s internetskih sjedišta uredno prikazuje u jednom prozoru u obliku tablice. Ovim je korisnička potreba za dostupnosti potpuno zadovoljena.

Dodavanje izvora – aplikacija nudi mogućnost korisniku da dodaje vlastite izvore prilikom stvaranja upita, što je ujedno i jedini način korištenja ove aplikacije. Ovim je zadovoljena korisnička potreba za relevantnosti podataka.

Dodavanje ključnih riječi – aplikacija ne nudi mogućnost dodavanja ključnih riječi koje korisnik želi pretraživati u rezultatima izvlačenja. Nažalost, ovim će korisnik gubiti dio potrebne obaviještenosti, ali i relevantnosti podataka.

Pretraživanje ključnih riječi – aplikacija ne nudi mogućnost pretraživanja ključnih riječi u rezultatima izvlačenja. Nažalost, ovim će korisnik gubiti dio potrebne obaviještenosti, ali i relevantnosti podataka.

Izvlačenje informacija s posebnih internet sjedišta – aplikacija nudi mogućnost kreiranja više sustava za izvlačenje informacija s korisnički definiranih sjedišta u obliku „čarobnjaka“. Korisnik urese poveznicu na internetsko sjedište iz koje želi izvlačiti podatke, nakon čega aplikacija sama prepozna sadržaje ili sama ponudi korisniku da ih odabere. Odabirom sadržaja aplikacija sve izvučene podatke prikazuje u tablici. Ovim je dodatno zadovoljena korisnička potreba za relevantnosti podataka.

Podržava tehnologije za distribuciju informacija – dodavanjem RSS kanala kao izvora za izvlačenje podataka aplikacija prepozna da se radi o RSS kanalu te uspješno izdvoji sve unose i strukturira ih u tablicu, uključujući datume, slike i autore. Ovim je dodatno zadovoljena korisnička potreba za relevantnosti podataka.

Obavijesti – Internetsko sjedište ove aplikacije u opisu navodi mogućnosti slanja obavijesti korisnicima o promjenama sadržaja u izvučenim podacima, no korisničko sučelje u besplatnoj verziji aplikacije nigdje ne navodi obavijesti, kao ni baza znanja i sjedište za pomoć aplikacije, a nije navedena ni mobilna verzija aplikacije. Ovime korisnik gubi dio mobilnosti, ali i na obaviještenosti. Mobilnost može dobiti korištenjem generiranih izvora podataka (API) kod svakog napravljenog sustavaza izvlačenje podataka.

Višekorisnička aplikacija – aplikacija zahtjeva registraciju za korištenje, čime je višekorisnička, no korisnici ne mogu koristiti, odnosno dijeliti i uređivati napravljene sustave za izvlačenje podataka preko istog sučelja. No, korisnik koji kreira neki sustav za izvlačenje podataka dobiva izvor na njega kojeg može podijeliti s ostalima, iako to razvojni tim aplikacije ne preporuča.

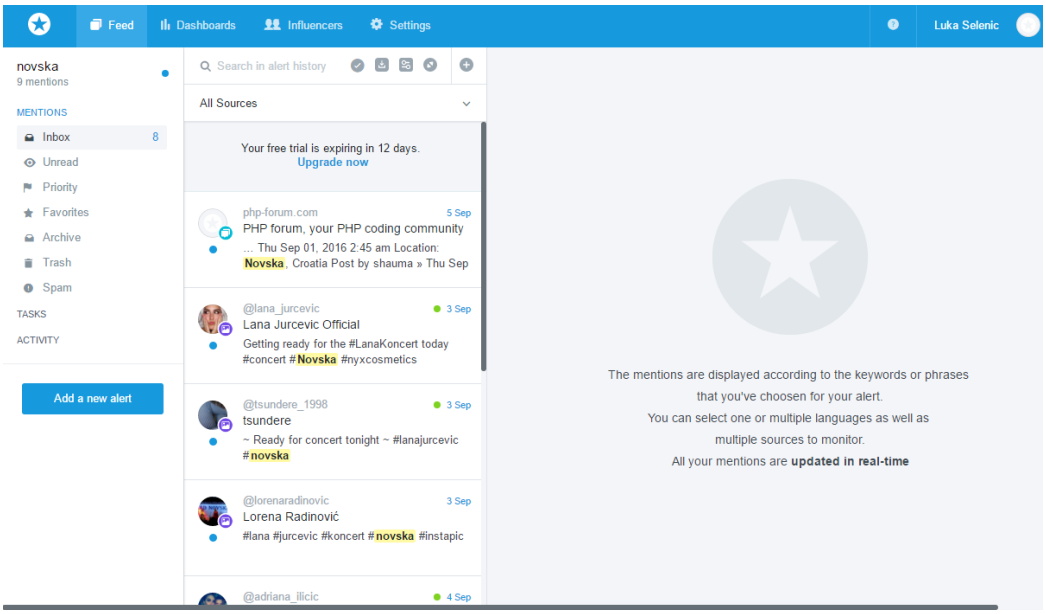
2.6.2. Zaključak

Import.io je aplikacija koja je namijenjena izvlačenju podataka s internetskih sjedišta kako bi se koristili za daljnju obradu. Besplatna verzija se ne može koristiti kao aplikacija za praćenje promjena informacija. Dobra strana aplikacije je njezina skalabilnost, jer se može kreirati neograničen broj sustava za izvlačenje podataka, a svaki od njih omogućava praćenje rezultata u realnom vremenu putem kreiranih izvora u nekoliko različitih formata. Prednost mu je i jednostavnost „čarobnjaka“ za dodavanje novih sustava za izvlačenje podataka koji dobro automatski prepoznaje sadržaje i pravilno ih strukturira u tablice. Mogućnost dodavanja vlastitih stupaca u rezultate tablice dobra je mogućnost radi proširivanja informiranosti.

2.7.Mention.com

Poveznica: https://mention.com (06.09.2016.)

Slika 5. – Sučelje Mention aplikacije



(Izvor: Autor)

Treći i posljednji alat u analizi bio je najzastupljeniji na **Google** tražilici preko plaćenih reklama. Unosom definiranih ključnih riječi **Google Ads** je kao početne rezultate pokazivao upravo ovu aplikaciju.

Aplikacija je, kao i **Mediatoolkit**, orjentirana *media monitoringu*. Analizom aplikacije pokazalo se da aplikacija pažnje prilaže društvenim mrežama. Nude funkcionalnost gdje korisnici kreirani račun na **Mentionu** mogu povezati s računima na drušvenim mrežama **Facebook** i **Twitter**, tako da čim apliakcija pronađe relevantnu informaciju na tim društvenim mrežama korisnici odmah mogu stupiti u kontakt s onima koji su tu relevantnu informaciju i pokrenuli. Aplikacija ima slične karakteristike kao i **Mediatoolkit** jer se za praćenje informacija kreiraju upiti, a svaki upit može se dodatno modificirati filterima koji se baziraju na izvoru informacija, jeziku, datumu ili ključnim riječima.

Razvojni tim ovu aplikaciju prezentira kao savršenu za marketinške agencije jer putem nje mogu pratiti kampanje klijenata, pronaći potencijalne nove klijente te generirati određene izvještaje i poslati ih klijentima. Uz agencije, spominju i fakultete i škole, koji preko **Mentiona** mogu pratiti kampuse, događanja, studente i djelatnike ovih ustanova.

Registracija je besplatna, minimalna, brza i jednostavna, a nudi se besplatan probni rok od 14 dana s ograničenim funkcionalnostima. Ono što **Mediatookit** zove „*upitom*“, **Mentions** zove „*upozorenje*“ (engl. *alert*). Upozorenja se kreiraju na intuitivan način u obliku „čarobnjaka“, gdje korisnik može odabrati kreiranje upozorenja za vlastitu tvrtku ili brand, za konkurenciju, „sve ostalo“ te *boolean* upozorenja koja su rezervirana samo za plaćenu verziju aplikacije. Odabirom jedne od opcija otvara se „čarobnjak“ koji korisnika vodi kroz kreiranje upozorenja po mjeri. Svako dodavanje upozorenja sastoji se od dodavanja glavnog internetskog sjedišta predmeta upozorenja, a aplikacija već ima veliku bazu predefiniranih internetskih sjedišta. Stoga npr. ako se želi pratiti brand „**Mercedes**“, tada korisnik ima mogućnost odabira brenda čime se automatski popuni poveznica do glavnog internetskog sjedišta za **Mercedes**, njihova **Twitter** stranica, **Facebook** stranica i ostale. Nakon odabira predmeta upozorenja korisnik definira predefinirane izvore gdje želi da aplikacija prati **Mercedes** (što je odmah ujedni i ključna riječ). Nudi mu se

izbor društvenih mreža, novinskih portala, blogova, video servisa, foruma, servisa za fotografija te ostalih internteskih sjedišta, ali i izbor na kojim jezicima da prati željenu ključnu riječ.

Ispis rezultata je brz i u realnom je vremenu. Korisničko sučelje je slično kao i kod modernih aplikacija za upravljanje e-mailovima. Ono što je unikatno kod ove aplikacije je to što se svaki pronađeni rezultat, bilo to **Facebook** ili **Twitter** objava, internetsko sjedište, može otvoriti unutar aplikacije, nakon čega aplikacija na otvorenom rezultatu osjenča ključnu riječ preko koje je našao taj rezultat.

Ono što je također dobra funkcionalnost ove aplikacije je što analizira svaki rezultat, te ukoliko on sadrži psovke naznači taj rezultat s crvenik kružićem, konotirajući da je to loš znak za korisnika i ono što prati, time mu davajući na znanje da je netko nezadovoljan s onim što prati. Identična stvar je i s rezultatima koji u sebi sadrže pozitivne riječi. Svaki rezultat se može dodatno pretražiti, ali i filtrirati po kategorijama: svi izvori, **Facebook**, **Twitter**, vijesti, blogovi, video servisi, forumi, servisi za slike i ostala internetska sjedišta. Jednom kreirana „upozorenja“ mogu se urediti i prilagoditi, npr. dodati nove ključne riječi za pretragu, urediti ime „upozorenja“ i slično.

Osim rezultata „upozorenja“, aplikacija nudi i statističke preglede generirane na temelju rezultata. Preko njih korisnik može vidjeti koliko je za određeno „upozorenje“ pronađeno rezultata, koliko ih je pozitivno, povijest pronalaženja, u postocima je izraženo s kojih izvora je za to „upozorenje“ najviše pronađenih rezultata i ostali statistički podaci koji bi bili od koristi korisniku. Među statistikom vode se podaci od kojih osoba, grupa ili stranica na društvenim mrežama su spominjane ključne riječi unesene od strane korisnika.

2.7.1. Analiza

Intervalna pretraga	Agregirane informacije	Dodavanje izvora	Dodavanje ključnih riječi	Pretraživanje ključnih riječi
Izvlačenje informacija s posebnih internet sjedišta		Podržava tehnologije za distribuciju informacija	Obavijesti	Višekorisnička aplikacija

Tablica 3. – Analizirane funkcionalnosti Mention aplikacije

Intervalna pretraga – aplikacija, odnosno svako kreirano „upozorenje“ u intervalima traži ključne riječi koje su postavljenje. Korisnik ne može definirati intervale, no rezultati se pojave čim aplikacija pronađe traženu ključnu riječ u realnom vremenu. Ovim je korisnička potreba za pravovremenosti dobivanja informacija u potpunosti zadovoljena.

Agregirane informacije – aplikacija izvučene podatke s internetskih sjedišta uredno prikazuje u jednom prozoru u obliku popisa. Svaki od rezultata se može označiti „pročitanim“ radi preglednosti i organizacije. Također, korisnici mobilne aplikacije sve rezultate dobivaju i na mobilne uređaje. Ovim je korisnička potreba za dostupnosti, ali i mobilnosti, potpuno zadovoljena.

Dodavanje izvora – aplikacija nudi mogućnost korisniku da dodaje vlastite izvore prilikom stvaranja „upozorenja“. No, iako se u aplikaciju doda prioritetno internetsko sjedište za pretragu, aplikacija i dalje traga po ostalim sjedištima. Ovim je zadovoljena korisnička potreba za relevantnosti podataka.

Dodavanje ključnih riječi – aplikacija nudi dodavanje ključnih riječi prilikom kreiranja „upozorenja“, ili naknadno dodavanje prilikom uređivanja postojećih. Ovim je dodatno zadovoljena korisnička potreba za relevantnosti podataka.

Pretraživanje ključnih riječi – dodana ključna riječ odmah se pretražuje u arhivi aplikacije, a s filtrima se dodatno namješta ishod rezultata. Ovim je dodatno zadovoljena korisnička potreba za relevantnosti podataka. Problem jedino predstavlja što unos ključne riječi nad određenim internetskim sjedištem i dalje pretražuje čitavu arhivu, ne samo navedeno internetsko sjedište.

Izvlačenje informacija s posebnih internet sjedišta – iako aplikacija nudi dodavanje vlastitog izvora, prilikom dobivanja rezultata u istraživanju aplikacija nije pokazala rezultate s dodanog internetskog sjedišta, već je pretraživala arhivu za ključne riječi s različitih internetskih sjedišta. Ovim se djelomično gubi mogućnost zadovoljenja korisničke potrebe za relevantnosti podataka jer korisnik ovisi o već napravljenoj arhivi, odnosno bazi podataka, a ne čitanju sadržaja s dodanih internetskih sjedišta.

Podržava tehnologije za distribuciju informacija – dodavanjem RSS kanala kao izvora za izvlačenje podataka aplikacija ne prepozna da se radi o RSS kanalu. Identično je kao i s dodavanjem vlastitih internetskih sjedišta – aplikacija ne prepozna kao izvor novih podataka. Ovim se djelomično gubi mogućnost zadovoljenja korisničke potrebe za relevantnosti podataka jer korisnik ovisi o već napravljenoj arhivi, odnosno bazi podataka, a ne čitanju sadržaja s RSS kanala.

Obavijesti – aplikacija prilikom svakog novog pronalaska ključnih riječi korisnika može obavijestiti putem e-maila, Google Chrome notifikacije, *push* notifikacije na Android i iOS uređaje te SMS porukom. Ovim je zadovoljena potreba korisnika za mobilnosti, pravovremenosti i obaviještenosti.

Višekorisnička aplikacija – korisnici imaju mogućnost dodavanja drugih korisnika i dodjeljivanja uloga tim novim korisnicima. U ovoj aplikaciji se višekorisnički rad naziva kao timski rad, gdje svi korisnici dijele dodana „upozorenja“.

2.7.2. Zaključak

Vrlo moćna aplikacija za praćenje ključnih riječi zbog velikog obujma već indeksiranih i agregiranih internetskih sjedišta, čime aplikacija za unesene ključne riječi može vrlo brzo davati

relevantne rezultate. Odlična ideja je mogućnost prikazivanja rezultata i otvaranja lokacije gdje je pronađena te osjenčavanje ključnih riječi preko kojih je aplikacija došla do tog rezultata. Negativna strana je što ne daje mogućnost određivanja, tj. limitiranja „upozorenja“ da pretražuje samo određeno internetsko sjedište za ključne riječi. Iako je pozitivno da pretražuje što širi spektar izvora, dobar dodatak bi bio filtar za jednu domenu.

2.8. Rezultati istraživanja

Rezultati istraživanja	Mediatoolkit	Import.io	Mention
Intervalna pretraga			
Agregirane informacije			
Dodavanje izvora			
Dodavanje ključnih riječi			
Pretraživanje ključnih riječi			
Izvlačenje informacija s posebnih internet sjedišta			
Podržava tehnologije za distribuciju informacija			
Obavijesti			
Višekorisnička aplikacija			

Tablica 4. – Rezultati istraživanja funkcionalnosti aplikacija

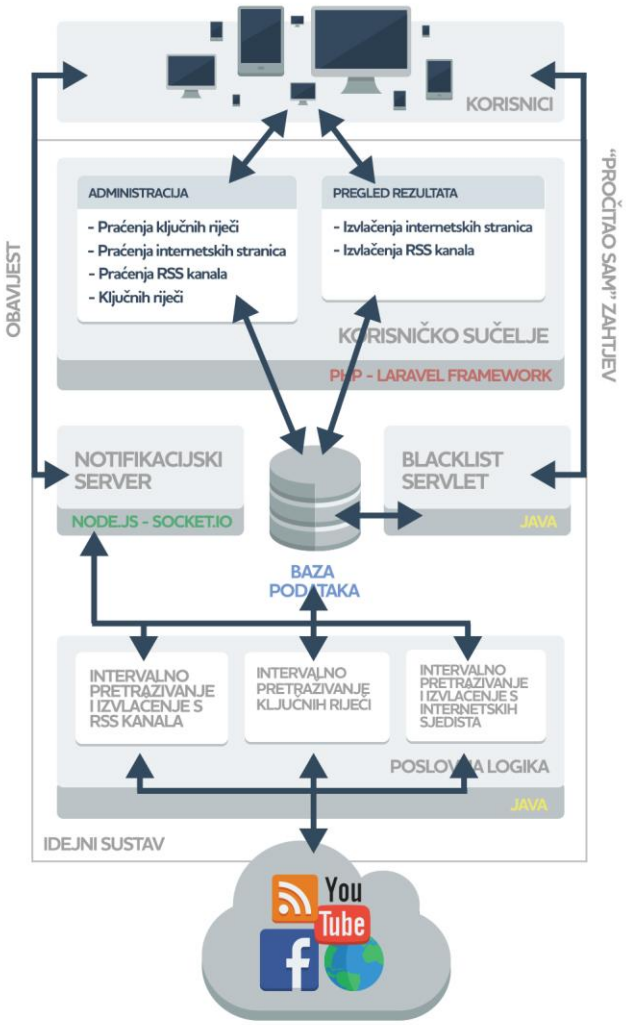
Na temelju rezultata se može izraditi konceptualni model i idealno rješenje za novi sustav koji će trebati riješiti navedene nedostatke analiziranih aplikacija.

3. ANALIZA SUSTAVA

3.1. Konceptualni model

Kako bi se što jednostavnije napravio novi idejni sustav, potrebno je napraviti konceptualni model na temelju kojeg će se vidjeti čitav sustav i veze među djelovima sustava. Novi idejni sustav obavlja aktivnosti *web scrapinga* i *media monitoringa*.

Slika 6. – Konceptualni model idejnog sustava



(Izvor: Autor)

Na temelju slike 6. vidljivo je da idejni sustav kreće od **korisnika.** Čitav idejni sustav nalazi se u oblaku, odnosno na internetu. Korisnik putem određenog uređaja pristupa korisničkom sučelju putem kojeg može vršiti određene radnje: administritrati praćenje ključnih riječi, internetskih stranica ili RSS kanala te administrirati ključne riječi. Administracija podrazumjeva dodavanje, uređivanje ili brisanje određenih unosa. Također, korisnik putem sučelja može pregledavati rezultate iz aktivnosti izvlačenja sadržaja s internteskih stranica, odnosno sjedišta te RSS kanala.

Korisničko sučelje je povezano s **bazom podataka** koja sprema korisnike, ključne riječi, postavke pretraživanja i izvlačenja, rezultate izvlačenja i slično.

Intervalno pretraživanje i izvlačenje **poslovna je logika** idejnog sustava. U određenom intervalu pokreću se određene aktivnosti koje uzimaju određene podatke iz baze podataka, obavljaju određenu aktivnost preuzimanja sadržaja s interneta (oblaka) te nakon što su izvršene nove podatke spremaju u istu bazu podataka.

Pri pronalaženju traženih postavki, djelovi poslovne logike šalju **obavijest** s rezultatima pretrage u **notifikacijski server** na kojeg su korisnici preko svojih uređaja već spojeni. **Notifikacijski server** tada tu **obavijest** prosljeđuje svim korisnicima na **notifikacijskom serveru**.

Korisnici mogu pročitati obavijest i vidjeti koje rezultate je sustav pronašao. Mogu se odlučiti označiti neki od rezultata da su pročitani ili nepročitani, čim se šalje zahtjev ***blacklist*** **servletu** (hrv. servlet za crnu listu), koji taj rezultat označava pročitanim ili nepročitanim u bazi podataka, a korisnik dobiva povratnu informaciju.

3.2. Opis sustava

Cilj sustava je omogućiti korisnicima da budu i ostanu informirani kada se određena informacija pojavi na internetskim sjedištima na kojima korisnici misle da bi se ona mogla pojaviti. Također, cilj sustava je da na jednom mjestu skuplja i agregira informacije s raznih izvora – Internet sjedišta, društvenih mreža, RSS kanala.

Sustav bi trebao omogućiti stalno, odnosno intervalno praćenje korisnički definiranih ključnih riječi na određenim internetskim sjedištima te u slučaju da tu ključnu riječ trebao bi biti obaviješten. Da bi se to omogućilo, sustav mora biti stalno dostupan što znači da bi trebao biti u oblaku, odnosno na internetu. Na taj način, procesi se nesmetano mogu odvijati, a korisnik/korisnici sustava mogu biti obaviješteni na svoje mobilne uređaje.

Sustav bi također trebao omogućiti izvlačenje podataka s internetskih sjedišta na način da korisnik odredi internetsko sjedište te sustavu da primjer sadržaja kojeg želi izvlačiti. Ova funkcionalnost bi trebala korisniku omogućiti da izvlači podatke koji su promjenivi, npr. naslovi na novinskim portalima, naslovi oglasa u internetskim oglasnicima ili bilo koji drugi oblik sadržaja koji je informativan, a onda te izvučene informacije pružiti korisniku. Pošto sustav samostalno ne može odrediti sadržaj, korisnik bi trebao sustavu dati primjer sadržaja koji se trenutno nalazi na traženom internetskom sjedištu, a sustav bi trebao određenim algoritmom onda prepoznati željeni sadržaj. Korisnik bi također trebao imati mogućnost praćenja ključnih riječi i u sadržaju koji sustav izvuče sa internetskih sjedišta te o pronalasku biti obaviješten.

Pošto već postoje određene tehnologije za distribuciju informacija, kao što je na primjer RSS, sustav bi trebao podržavati neke od tih tehnologija. Stoga, korisnik bi trebao imati mogućnost pretplate na željeni RSS kanal, iz kojeg bi sustav povlačio sadržaj i uredno ga prikazao korisniku. Korisnik bi i ovdje također trebao imati mogućnost praćenja ključnih riječi te biti na vrijeme obaviješten o pronalasku podudarnosti.

Sustav bi trebao omogućiti rad više korisnika, stoga radi sigurnosti treba biti mogućnost registracije i prijave u sustav.

Sve izvučene informacije korisnik bi trebao imati mogućnost označiti kao pročitane, tako da sustav prestane s obaviještavanjem o već pregledanim rezultatima, a i radi preglednosti.

Zbog bolje obaviještenosti o pronađenim ključnim riječima, korisnik bi preko određene mobilne aplikacije trebao imati uvid u sve rezultate izvlačenja informacija, ali trebao bi biti putem nje obaviješten kada sustav pronađe neku ključnu riječ.

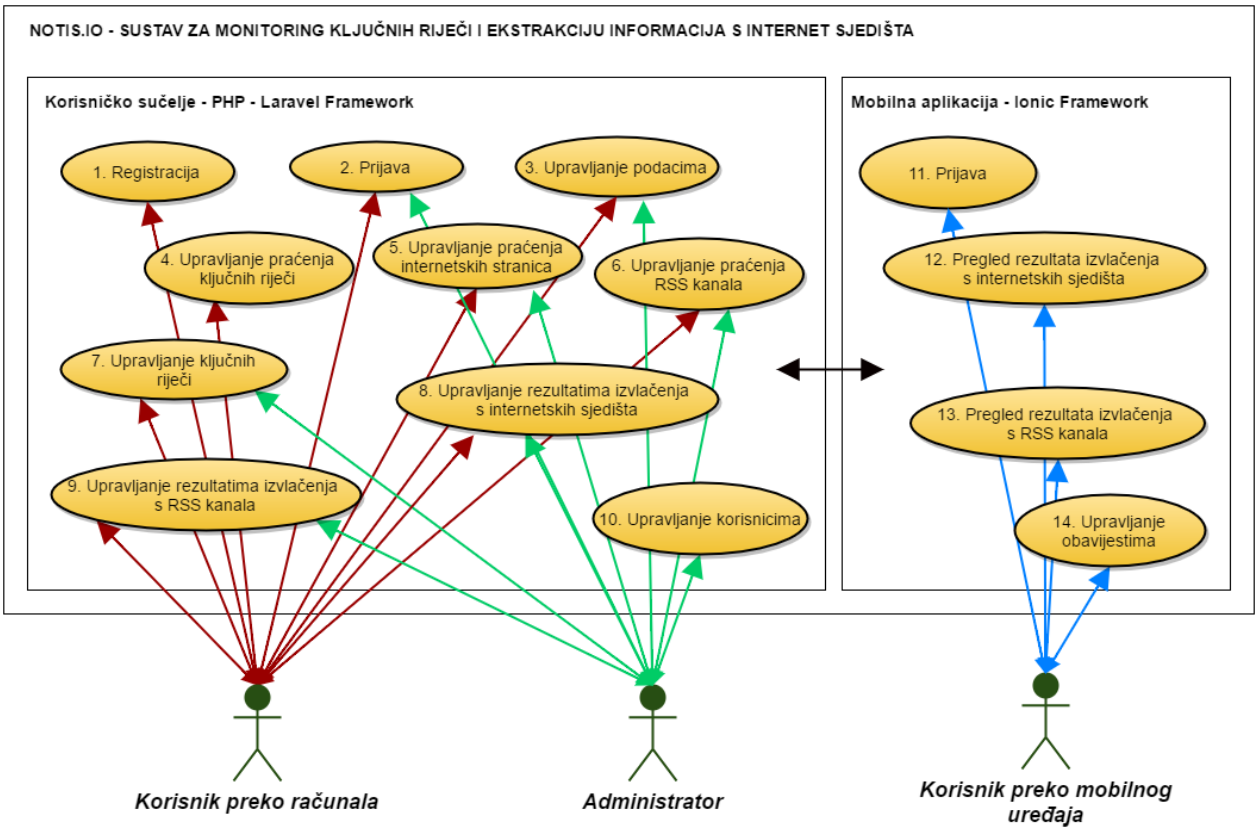
3.2.1. Akteri

1. **Administrator** – upravlja korisnicima – dodaje, uređuje i briše korisnike/administratore
2. **Korisnik preko računala**:
    - Ima mogućnost registracije
    - Ima mogućnost prijave u sustav
    - Upravlja praćenjem ključnih riječi – dodaje, uređuje i briše monitore za praćenje ključnih riječi
    - Upravlja praćenjem internetskih sjedišta – dodaje, uređuje i briše monitore za praćenje internetskih sjedišta
    - Upravlja praćenjem RSS kanala – dodaje, uređuje i briše monitore za praćenje RSS kanala
    - Upravlja ključnim riječima – dodaje, uređuje i briše ključne riječi u sustavu
    - Upravlja rezultatima izvlačenja s internetskih sjedišta – pregledava rezultate i označava ih pročitanim ili nepročitanim
    - Upravlja rezultatima izvlačenja s RSS kanala – pregledava rezultate i označava ih pročitanim ili nepročitanim
3. **Korisnik preko mobilnog uređaja**:
    - Prijavljuje se u sustav prije prvog korištenja sustava
    - Pregledava rezultate izvlačenja s internetskih sjedišta
    - Pregledava rezultate izvlačenja s RSS kanala
    - Upravlja obavijestima – pregledava obavijesti o pronađenim ključnim riječima, označava obavijest pročitanom

3.3. Use case dijagram

Use case dijagram prikazuje svaku radnju koju određeni korisnik može izvoditi u sustavu.

Slika 7. – Use case dijagram idejnog sustava



(Izvor: Autor)

Bitno je spomenuti da svi korisnici moraju biti prijavljeni u sustav da ga koriste, osim za proces „Registracija“ i „Prijava“, a korisnik mobilne aplikacije mora biti registriran u sustav kako bi koristio aplikaciju te onda prijavljen procesom „Prijava“.

3.4. Opis scenarija

3.4.1. Registracija

Identifikacijski sažetak:

Korisnik se registrira u sustav kako bi se mogao prijaviti u njega, pošto je prijava obavezna za korištenje sustava. Sustav prilikom registracije od korisnika traži unos e-maila, lozinke, imena i prezimena te kontakt broj. Nakon registracije, korisnik mora potvrditi svoju registraciju putem poveznice poslane na uneseni e-mail.

Akteri: Korisnik računala

Tijek događanja:

1. Korisniku se prikazuje forma za prijavu u sustav
2. Korisnik pritisne tipku „Registracija“
3. Korisniku se prikazuje forma za registraciju
4. Korisnik unosi e-mail, lozinku, ime, prezime i kontakt broj
5. Korisnik pritisne tipku „Registriraj se“
6. Korisnik u dobivenom e-mailu od sustava pritisne poveznicu unutar e-maila
7. Poveznica korisnika vodi na stranicu za potvrdu
8. Korisniku se prikazuje forma za prijavu u sustav

3.4.2. Prijava

Identifikacijski sažetak:

Korisnik (u nastavku i administrator) prijavljuje se kako bi mogao koristiti sustav. Sustav prilikom prijave od korisnika traži unos e-maila i lozinke za prijavu.

Akteri: Korisnik računala, administrator

Tijek događanja:

1. Korisniku se prikazuje forma za prijavu u sustav
2. Korisnik unosi e-mail i lozinku
3. Sustav provjerava unesene podatke
4. Ukoliko su podaci ispravni, korisnik se uspješno prijavljuje u sustav
5. Ukoliko su podaci neispravni, korisniku se ponovno prikazuje forma za prijavu u sustav s obavijesti o neuspješnoj prijavi.

3.4.3. Upravljanje korisničkim podacima

Identifikacijski sažetak:

Korisnik (u nastavku i administrator) nakon prijave dobiva prozor u kojem može vidjeti svoje podatke i urediti ih. Na istom prozoru dobiva i generirani QR kod kojeg mora skenirati kako bi koristio mobilnu aplikaciju. On sadrži unikatan API kod od korisnika.

Akteri: Korisnik računala, administrator

Tijek događanja:

1. Korisnik se prijavljuje u sustav
2. Korisniku se prikazuje forma s njegovim podacima i QR kod
3. Korisnik prepravlja neki od unesenih podataka
4. Korisniku pritisne na tipku „Spremi promjene“

3.4.4. Upravljanje praćenja ključnih riječi

Identifikacijski sažetak:

Korisnik (u nastavku i administrator) nakon prijave klikom na tipku „Riječi“ u navigaciji ima mogućnost pregleda, pretraživanja, dodavanja, brisanja i uređivanja monitora ključnih riječi.

Akteri: Korisnik računala, administrator

Tijek događanja:

Uređivanje monitora

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Riječi“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za ključne riječi
4. Korisnik odabire željeni monitor
5. Korisniku se prikazuje novi prozor s popunjenom formom i podacima od odabranog monitora: Ime monitora, URL stranice, ključne riječi
6. Korisnik mijenja podatke
7. Korisnik pritisne tipku „Spremi“

Dodavanje monitora

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Riječi“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za ključne riječi
4. Korisnik pritisne tipku sa simbolom „+“ ispod popisa monitora
5. Korisniku se prikazuje novi prozor s formom za unos podataka: Ime monitora, URL stranice, ključne riječi
6. Korisnik unosi podatke
7. Korisnik pritisne tipku „Dodaj“
8. Korisnik je preusmjeren na popis napravljenih monitora

Brisanje monitora u isto vrijeme

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Riječi“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za ključne riječi
4. Korisnik pritisne prazna polja pored imena monitora koje želi obrisati
5. Korisnik pritisne tipku sa simbolom „–„ ispod popisa monitora
6. Korisniku se prikazuje prozor za potvrdu brisanja monitora
7. Korisnik može pritisnuti tipku „Odustani“ za odustajanje od brisanja ili „Briši“ za potvrdu brisanja
8. Korisnik je preusmjeren na popis napravljenih monitora s porukom da su monitori obrisani

Brisanje jednog monitora

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Riječi“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za ključne riječi
4. Korisnik pritisne na ime monitora kojeg želi obrisati
5. Korisniku se prikazuje novi prozor s formom i popunjenim podacima odabranog monitora
6. Korisnik pritisne tipku „Briši“
7. Korisnik je preusmjeren na popis napravljenih monitora s porukom da je monitor obrisan

Pretraživanje monitora

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Riječi“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za ključne riječi
4. Korisnik unosi ime monitora kojeg želi tražiti u prazno polje „Pretraživanje“
5. Sustav korisniku ispiše monitore koji odgovaraju njegovom upitu

3.4.5. Upravljanje praćenja internetskih stranica

Identifikacijski sažetak:

Korisnik (u nastavku i administrator) nakon prijave klikom na tipku „Internet stranice“ u navigaciji ima mogućnost pregleda, pretraživanja, dodavanja, brisanja i uređivanja monitora za praćenje internetskih stranica.

Akteri: Korisnik računala, administrator

Tijek događanja:

Uređivanje monitora

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Internet stranice“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za internet stranice
4. Korisnik prelaskom miša preko monitora pritisne na simbol u obliku olovke
5. Korisniku se prikazuje prozor s popunjenom formom i podacima od odabranog monitora: Ime monitora, URL stranice, vrsta pretrage, tekst i pretraga ključnih riječi u rezultatima
6. Korisnik mijenja podatke

7. Korisnik pritisne tipku „Uredi“
8. U slučaju da želi odustati od promjena pritisne tipku „Odustani“

Dodavanje monitora

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Internet stranice“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za internetske stranice
4. Korisnik pritisne tipku sa simbolom „+“ ispod popisa monitora
5. Korisniku se prikazuje novi prozor s formom za unos podataka: Ime monitora, URL stranice, vrsta pretrage, tekst i pretraga ključnih riječi u rezultatima
6. Korisnik unosi podatke
7. Korisnik pritisne tipku „Pronađi“
8. Korisniku se pokaže novi prozor za izdvajanje naslova
9. Korisnik unosi primjer naslova ili teksta koji predstavlja sadržaj kojeg želi izvlačiti
10. Korisnik pritisne tipku „Pronađi“
11. Sustav algoritmima pretražuje relevantan sadržaj koji odgovara unesenom tekstu
12. Korisniku sustav izlistava pronađene sadržaje
13. Ako korisniku ne pašu naslovi pritisne na tipku „Koristi naprednu pretragu“ pa na tipku „Pronađi“
14. Korisniku sustav izlistava nove pronađene sadržaje
15. Korisnik pritisne tipku „Prihvati“
16. Korisnik pritisne tipku „Dodaj“
17. Korisnik je preusmjeren na popis napravljenih monitora

Brisanje monitora u isto vrijeme

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Internet stranice“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za internetske stranice
4. Korisnik pritisne prazna polja pored imena monitora koje želi obrisati
5. Korisnik pritisne tipku sa simbolom „–„ ispod popisa monitora
6. Korisniku se prikazuje prozor za potvrdu brisanja monitora
7. Korisnik može pritisnuti tipku „Odustani“ za odustajanje od brisanja ili „Briši“ za potvrdu brisanja
8. Korisnik je preusmjeren na popis napravljenih monitora s porukom da su monitori obrisani

Brisanje jednog monitora

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Internet stranice“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za internetske stranice
4. Korisnik pritisne na ime monitora kojeg želi obrisati
5. Korisniku se prikazuje novi prozor s formom i popunjenim podacima odabranog monitora
6. Korisnik pritisne tipku „Briši“
7. Korisnik je preusmjeren na popis napravljenih monitora s porukom da je monitor obrisan

Pretraživanje monitora

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Internet stranice“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za internetske stranice
4. Korisnik unosi ime monitora kojeg želi tražiti u prazno polje „Pretraživanje“
5. Sustav korisniku ispiše monitore koji odgovaraju njegovom upitu

3.4.6. Upravljanje praćenja RSS kanala

Identifikacijski sažetak:

Korisnik (u nastavku i administrator) nakon prijave klikom na tipku „RSS“ u navigaciji ima mogućnost pregleda, pretraživanja, dodavanja, brisanja i uređivanja monitora za praćenje RSS kanala.

Akteri: Korisnik računala, administrator

Tijek događanja:

Uređivanje monitora

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „RSS“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za RSS kanale
4. Korisnik prelaskom miša preko monitora pritisne na simbol u obliku olovke
5. Korisniku se prikazuje prozor s popunjenom formom i podacima od odabranog monitora: Ime monitora, URL kanala, pretraga ključnih riječi u rezultatima
6. Korisnik mijenja podatke

7. Korisnik pritisne tipku „Uredi“
8. U slučaju da želi odustati od promjena pritisne tipku „Odustani“

Dodavanje monitora

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „RSS“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za RSS kanale
4. Korisnik pritisne tipku sa simbolom „+“ ispod popisa monitora
5. Korisniku se prikazuje novi prozor s formom za unos podataka: Ime monitora, URL kanala, pretraga ključnih riječi u rezultatima
6. Korisnik unosi podatke
7. Korisnik pritisne tipku „Dodaj“
8. Korisnik je preusmjeren na popis napravljenih monitora

Brisanje monitora u isto vrijeme

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „RSS“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za RSS kanale
4. Korisnik pritisne prazna polja pored imena monitora koje želi obrisati
5. Korisnik pritisne tipku sa simbolom „–„ ispod popisa monitora
6. Korisniku se prikazuje prozor za potvrdu brisanja monitora
7. Korisnik može pritisnuti tipku „Odustani“ za odustajanje od brisanja ili „Briši“ za potvrdu brisanja
8. Korisnik je preusmjeren na popis napravljenih monitora s porukom da su monitori obrisani

Brisanje jednog monitora

1. Korisnik se prijavljuje u sustav

2. Korisnik odabire tipku „RSS“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za RSS kanale
4. Korisnik pritisne na ime monitora kojeg želi obrisati
5. Korisniku se prikazuje novi prozor s formom i popunjenim podacima odabranog monitora
6. Korisnik pritisne tipku „Briši“
7. Korisnik je preusmjeren na popis napravljenih monitora s porukom da je monitor obrisan

Pretraživanje monitora

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „RSS“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za RSS kanale
4. Korisnik unosi ime monitora kojeg želi tražiti u prazno polje „Pretraživanje“
5. Sustav korisniku ispiše monitore koji odgovaraju njegovom upitu

3.4.7. Upravljanje ključnim riječima

Identifikacijski sažetak:

Korisnik (u nastavku i administrator) nakon prijave klikom na tipku „Ključne riječi“ u navigaciji ima mogućnost pregleda, pretraživanja, dodavanja i brisanja ključnih riječi za čitav sustav.

Akteri: Korisnik računala, administrator

Tijek događanja:

Dodavanje ključnih riječi

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Ključne riječi“ u navigaciji, kategorija „Postavke“
3. Korisniku se prikazuje prozor s popisom napravljenih ključnih riječi u obliku tablice
4. Korisnik unese željenu ključnu riječ u prazno polje koje sadrži simbol „+“
5. Korisnik pritisne tipku „Dodaj ...“
6. Korisnik je preusmjeren na popis napravljenih ključnih riječi

Brisanje ključnih riječi

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Ključne riječi“ u navigaciji, kategorija „Postavke“
3. Korisniku se prikazuje prozor s popisom napravljenih ključnih riječi u obliku tablice
4. Korisnik pritisne na tipku u stupcu „Radnja“ u retku ključne riječi koju želi obrisati
5. Korisnik je preusmjeren na popis napravljenih ključnih riječi s porukom da je ključna riječ obrisana

Pretraživanje ključnih riječi

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Ključne riječi“ u navigaciji, kategorija „Postavke“
3. Korisniku se prikazuje prozor s popisom napravljenih ključnih riječi u obliku tablice

4. Korisnik se klikom miša fokusira na polje sa simbolom „povećala“ u naslovu tablice
5. Korisnik unosi tekst
6. Sustav korisniku ispiše ključne riječi koji odgovaraju njegovom upitu

3.4.8. Upravljanje rezultatima izvlačenja s internetskih sjedišta

Identifikacijski sažetak:

Korisnik (u nastavku i administrator) nakon prijave klikom na tipku „Internetske stranice“ te odabirom nekog od izlistanih, prethodno unesenih monitora ima mogućnost pregleda rezultata koje je taj monitor kroz svoj proces ostvario.

Akteri: Korisnik računala, administrator

Tijek događanja:

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „Internet stranice“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za internetske stranice
4. Korisnik pritisne na ime monitora od kojeg želi pregledati rezultate
5. Sustav korisniku ispiše rezultate u novom prozoru
6. Korisnik klikne na tipku sa simbolom „globusa“ ukoliko je izvučeni rezultat imao poveznicu na nekakav sadržaj
7. Korisnik klikne na tipku sa simbolom „kvačice“ je pročitao određeni rezultat ili ga želi označiti nepročitanim

3.4.9. Upravljanje rezultatima izvlačenja s RSS kanala

Identifikacijski sažetak:

Korisnik (u nastavku i administrator) nakon prijave klikom na tipku „RSS“ te odabirom nekog od izlistanih, prethodno unesenih monitora ima mogućnost pregleda rezultata koje je taj monitor kroz svoj proces ostvario.

Akteri: Korisnik računala, administrator

Tijek događanja:

1. Korisnik se prijavljuje u sustav
2. Korisnik odabire tipku „RSS“ u navigaciji
3. Korisniku se prikazuje prozor s popisom napravljenih monitora za RSS kanale
4. Korisnik pritisne na ime monitora od kojeg želi pregledati rezultate
5. Sustav korisniku ispiše rezultate u novom prozoru
6. Korisnik klikne na tipku sa simbolom „globusa“ ukoliko želi posjetiti poveznicu određenog rezultata
7. Korisnik klikne na tipku sa simbolom „kvačice“ je pročitao određeni rezultat ili ga želi označiti nepročitanim

3.4.10. Upravljanje korisnicima

Identifikacijski sažetak:

Administrator nakon prijave klikom na tipku „Korisnici“ u kategoriji „Administracija“ u navigaciji ima mogućnost pregleda, pretraživanja, dodavanja i brisanja korisnika.

Akteri: Administrator

Tijek događanja:

Dodavanje korisnika

1. Administrator se prijavljuje u sustav
2. Administrator odabire tipku „Korisnici“ u navigaciji, kategorija „Administracija“
3. Administratoru se prikazuje prozor s popisom svih korisnika u obliku tablice
4. Administrator pritisne tipku „Dodaj korisnika …“
5. Administratoru se prikaže novi prozor s formom za unos podataka o korisniku: E-mail, ime, prezime, kontakt broj, uloga
6. Administrator unosi podatke
7. Administrator pritisne tipku „Dodaj“ ako želi potvrditi podatke ili „Odustani“ ako ne želi dodati novog korisnika

Uređivanje korisnika

1. Administrator se prijavljuje u sustav
2. Administrator odabire tipku „Korisnici“ u navigaciji, kategorija „Administracija“
3. Administratoru se prikazuje prozor s popisom svih korisnika u obliku tablice
4. Administrator pritisne tipku sa simbolom „olovke“ u stupcu „Radnja“ u retku korisnika
5. Administratoru se prikaže novi prozor s popunjenom formom za unos podataka o korisniku: E-mail, ime, prezime, kontakt broj, uloga
6. Administrator prepravlja podatke
7. Administrator pritisne tipku „Uredi“ ako želi potvrditi podatke ili „Odustani“ ako ne želi urediti podatke odabranog korisnika

Pretraživanje korisnika

1. Administrator se prijavljuje u sustav
2. Administrator odabire tipku „Korisnici“ u navigaciji, kategorija „Administracija“
3. Administratoru se prikazuje prozor s popisom svih korisnika u obliku tablice
4. Administrator se mišem pozicionira u prazno polje sa simbolom „povećala“
5. Administrator unosi neki od podataka korisnika, dio e-maila, imena ili prezimena
6. Sustav administratoru prikazuje korisnike koji odgovaraju unesenom podatku

Brisanje korisnika

1. Administrator se prijavljuje u sustav
2. Administrator odabire tipku „Korisnici“ u navigaciji, kategorija „Administracija“
3. Administratoru se prikazuje prozor s popisom svih korisnika u obliku tablice
4. Administrator pritisne tipku sa simbolom „križića“ u stupcu „Radnja“ u retku korisnika kojeg želi obrisati
5. Sustav administratora preusmjeri na popis svih korisnika s porukom da je korisnik obrisan

3.4.11. Mobilna aplikacija - prijava

Identifikacijski sažetak:

Korisnik mobilne aplikacije prije prvog korištenja mora se prijaviti u sustav na način da skenira generirani QR kod koji se nalazi uz njegove podatke na administracijskoj aplikaciji. Skenirani kod se sprema u mobilnu aplikaciju tako da može koristiti funkcionalnosti sustava.

Akteri: Korisnik mobilne aplikacije

Tijek događanja:

1. Korisnik se prijavljuje u sustav
2. Korisnik pokreće mobilnu aplikaciju
3. Korisniku se ponudi skeniranje QR koda
4. Korisnik skenira QR kod
5. Mobilna aplikacija prihvaća korisnikov kod i sprema ga u memoriju

3.4.12. Mobilna aplikacija – Pregled rezultata izvlačenja s internetskih sjedišta

Identifikacijski sažetak:

Korisnik mobilne aplikacije može pregledati rezultate svih procesa izvlačenja s internetskih sjedišta – po monitoru. Korisnik mora biti spojen na internet.

Akteri: Korisnik mobilne aplikacije

Tijek događanja:

1. Korisnik pokreće mobilnu aplikaciju
2. Korisnik u izborniku odabire tipku „Internetske stranice“
3. Korisnik odabire željeni monitor
4. Korisniku aplikacija ispiše sve rezultate procesa izvlačenja za taj monitor
5. Korisnik klikne na željeni rezultat
6. Aplikacija korisniku otvara poveznicu na željeni rezultat

3.4.13. Mobilna aplikacija – Pregled rezultata izvlačenja s RSS sjedišta

Identifikacijski sažetak:

Korisnik mobilne aplikacije može pregledati rezultate svih procesa izvlačenja s RSS kanala – po monitoru. Korisnik mora biti spojen na internet.

Akteri: Korisnik mobilne aplikacije

Tijek događanja:

1. Korisnik pokreće mobilnu aplikaciju
2. Korisnik u izborniku odabire tipku „RSS“
3. Korisnik odabire željeni monitor
4. Korisniku aplikacija ispiše sve rezultate procesa izvlačenja za taj monitor
5. Korisnik klikne na željeni rezultat
6. Aplikacija korisniku otvara poveznicu na željeni rezultat

3.4.14. Mobilna aplikacija – Upravljanje obavijestima

Identifikacijski sažetak:

Korisnik mobilne aplikacije može primati obavijesti kada je sustav, odnosno neki od kreiranih monitora pronašao traženu ključnu riječ među rezultatima izvlačenja ili praćenja. Korisnik mora biti spojen na internet.

Akteri: Korisnik mobilne aplikacije

Tijek događanja:

1. Korisnik pokreće mobilnu aplikaciju
2. Aplikacija ostaje pokrenuta u memoriji mobilnog uređaja
3. Kad sustav pronađe ključnu riječ, korisnik dobiva obavijest
4. Korisnik u izborniku odabire tipku „Notifikacije“
5. Korisnik odabire obavijest
6. Korisnik može označiti obavijest kao pročitanu ili pritisnuti tipku „Odi na stranicu“ ako obavijest sadrži poveznicu

4. DIZAJN SUSTAVA

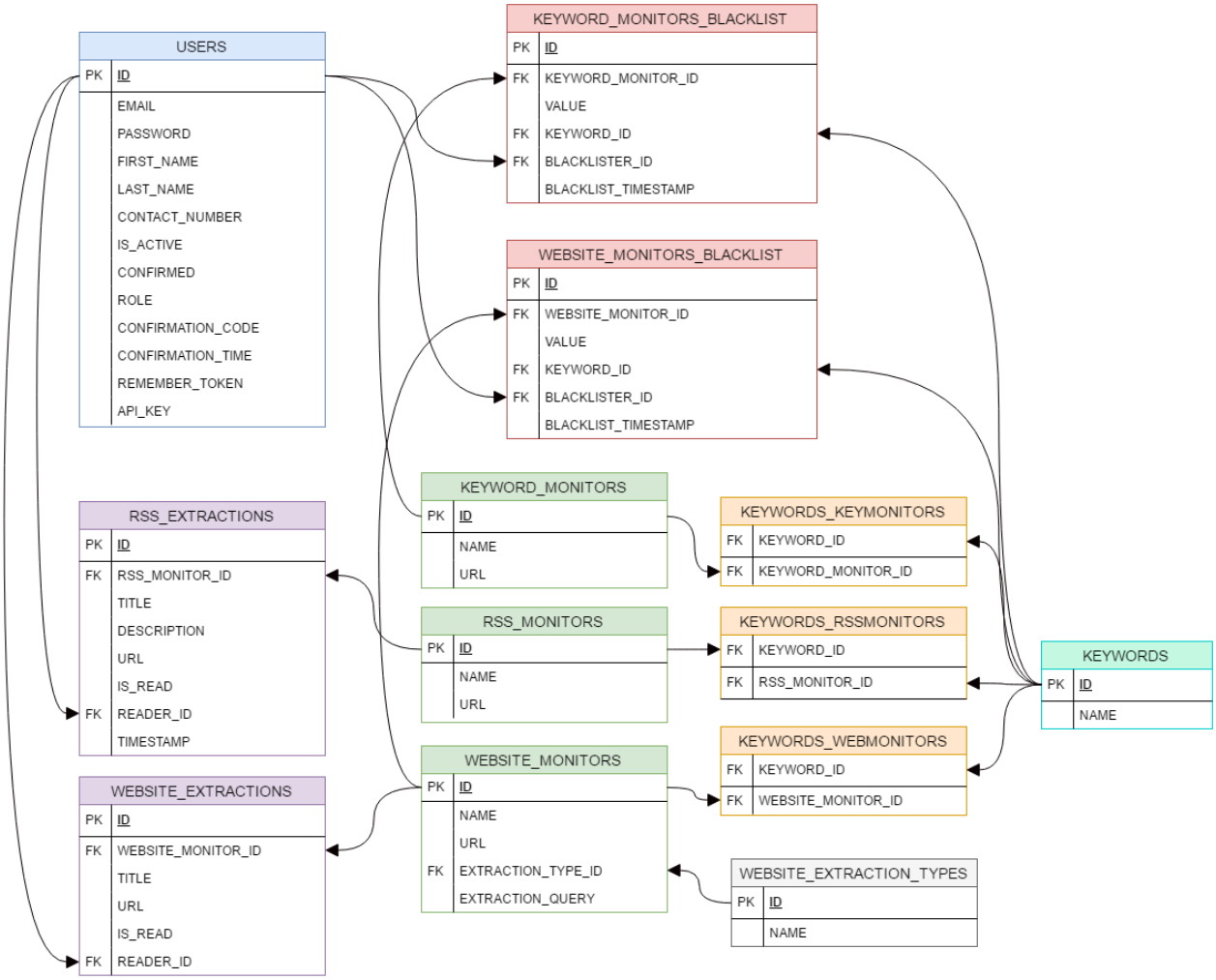
4.1. Relacijski model baze podataka

Tablice modela baze podataka:

- **USERS** – **ID (PK)**, EMAIL, PASSWORD, FIRST_NAME, LAST_NAME, CONTACT_NUMBER, IS_ACTIVE, CONFIRMED, ROLE, CONFIRMATION_CODE, CONFIRMATION_TIME, REMEMBER_TOKEN, API_KEY
- **RSS_EXTRACTIONS** – **ID (PK)**, *RSS_MONITOR_ID* (FK), TITLE, DESCRIPTION, URL, IS_READ, *READER_ID* (FK), TIMESTAMP
- **WEBSITE_EXTRACTIONS** – **ID (PK)**, *WEBSITE_MONITOR_ID* (FK), TITLE, URL, IS_READ, *READER_ID* (FK)
- **KEYWORD_MONITORS** – **ID (PK)**, NAME, URL
- **RSS_MONITORS** – **ID (PK)**, NAME, URL
- **WEBSITE_MONITORS** – **ID (PK)**, NAME, URL, *EXTRACTION_TYPE_ID* (FK), EXTRACTION_QUERY
- **WEBSITE_EXTRACTION_TYPES** – **ID (PK)**, NAME
- **KEYWORDS** – **ID (PK)**, NAME
- **KEYWORD_MONITORS_BLACKLIST** – **ID (PK)**, *KEYWORD_MONITOR_ID* (FK), VALUE, *KEYWORD_ID* (FK), *BLACKLISTER_ID* (FK), BLACKLIST_TIMESTAMP
- **WEBSITE_MONITORS_BLACKLIST** – **ID (PK)**, *WEBSITE_MONITOR_ID* (FK), VALUE, *KEYWORD_ID* (FK), *BLACKLISTER_ID* (FK), BLACKLIST_TIMESTAMP
- **KEYWORDS_KEYMONITORS** – *KEYWORD_ID* (FK), *KEYWORD_MONITOR_ID* (FK)
- **KEYWORDS_RSSMONITORS** – *RSS_MONITOR_ID* (FK), *KEYWORD_ID* (FK)

- **KEYWORDS_WEBMONITORS** – *KEYWORD_ID* (FK), *WEBSITE_MONITOR_ID* (FK)

Slika 8. – Relacijski model baze podataka idejnog sustava



(Izvor: Autor)

RB	Tablica 1	Tablica 2	Veza	Opis
1	USERS	RSS_EXTRACTIONS	1:M	Svaki korisnik može označiti više izvučenih podataka RSS kanala
2	RSS_EXTRACTIONS	USERS	M:1	Više izvučenih podataka RSS kanala može biti označeno od istog korisnika
3	USERS	WEBSITE_EXTRACTIONS	1:M	Svaki korisnik može označiti više izvučenih podataka internetskih sjedišta
4	WEBSITE_EXTRACTIONS	USERS	M:1	Više izvučenih podataka internetskih sjedišta može biti označeno od istog korisnika
5	RSS_MONITORS	RSS_EXTRACTIONS	1:M	Svaki RSS monitor može pronaći više rezultata
6	RSS_EXTRACTIONS	RSS_MONITORS	M:1	Više rezultata može pripadati jednom RSS monitoru
7	WEBSITE_MONITORS	WEBSITE_EXTRACTIONS	1:M	Svaki monitor internetskih sjedišta može pronaći više rezultata
8	WEBSITE_EXTRACTIONS	WEBSITE_MONITORS	M:1	Više rezultata može pripadati jednom monitoru internetskih sjedišta
9	WEBSITE_EXTRACTION_TYPES	WEBSITE_MONITORS	1:M	Jedna vrsta načina izvlačenja podataka može pripadati više monitora internetskih stranica
10	WEBSITE_MONITORS	WEBSITE_EXTRACTION_TYPES	M:1	Više monitora internetskih stranica može pripadati jednoj vrsti načina izvlačenja podataka
11	KEYWORDS_MONITORS	KEYWORD_MONITORS_BLACKLIST	1:M	Jedan monitor ključnih riječi može imati više unosa na crnoj listi (pronađenih) ključnih riječi
12	KEYWORD_MONITORS_BLACKLIST	KEYWORDS_MONITORS	M:1	Više unosa na crnoj listi (pronađenih) ključnih riječi može pripadati jednom monitoru ključnih riječi
13	WEBSITE_MONITORS	WEBSITE_MONITORS_BLACKLIST	1:M	Jedan monitor internetskih sjedišta može imati više unosa na crnoj listi (pronađenih) rezultata izvlačenja s

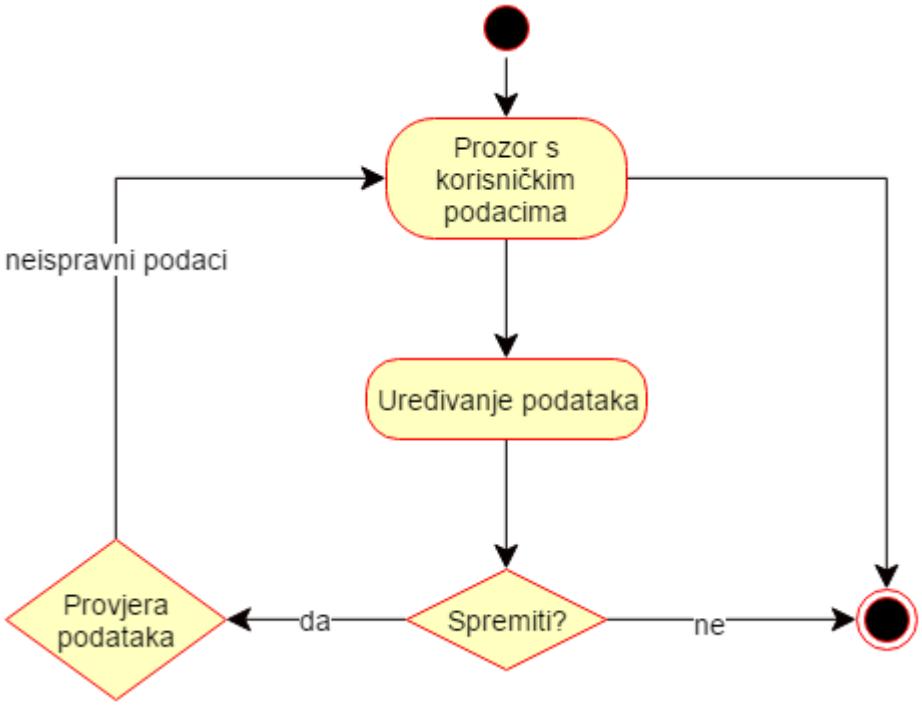
				internetskih sjedišta
14	WEBSITE_MONITORS_BLACKLIST	WEBSITE_MONITORS	M:1	Više unosa na crnoj listi (pronađenih) rezultata izvlačenja s internetskih sjedišta može pripadati jednom monitoru internetskih sjedišta
15	USERS	KEYWORD_MONITORS_BLACKLIST	1:M	Jedan korisnik može dodati više ključnih riječi na crnu listu (pronađenih) ključnih riječi
16	KEYWORD_MONITORS_BLACKLIST	USERS	M:1	Više unosa na crnoj listi (pronađeni) rezultata ključnih riječi može biti dodano od jednog korisnika
17	USERS	WEBSITE_MONITORS_BLACKLIST	1:M	Jedan korisnik može dodati više pronađenih rezultata izvlačenja s internetskih sjedišta na crnu listu
18	WEBSITE_MONITORS_BLACKLIST	USERS	M:1	Više unosa na crnoj listi (pronađenih) rezultata izvučenih s internetskih sjedišta može biti dodano od jednog korisnika

Tablica 5. – Opis veza relacijskog modela idejnog rješenja

4.2. Dinamičko modeliranje sustava – dijagram aktivnosti

4.2.1. Registracija

Slika 8. – Dijagram aktivnosti - Registracija

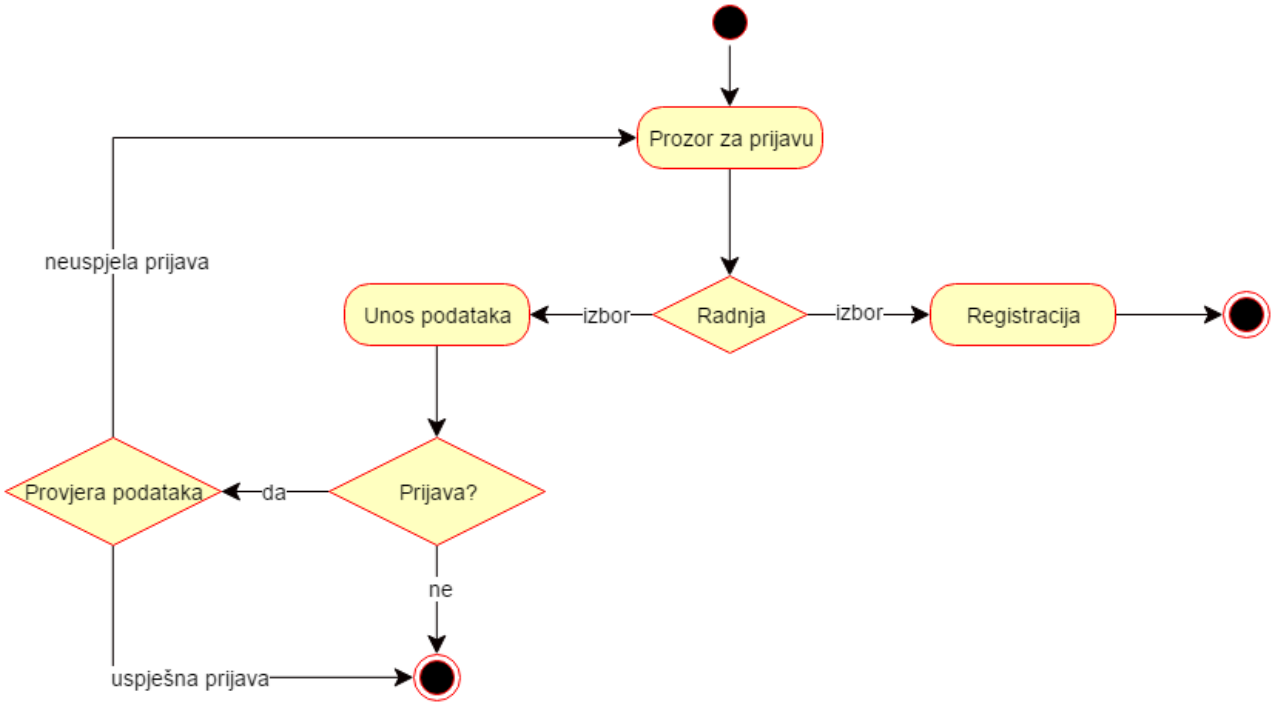


(Izvor: Autor)

Slika 8. prikazuje dijagram aktivnosti za registraciju korisnika. Prilikom dolaska na prozor za registraciju korisnik ima izbor za povratak na prijavu ili mogućnost unosa podataka. Nakon unosa može odustati od registracije za povratak na prijavu ili potvrditi registraciju. Sustav tada provjerava podatke te korisnika vraća na prozor za registraciju ukoliko su podaci netočni ili vodi korisnika dalje u sustav.

4.2.2. Prijava

Slika 9. – Dijagram aktivnosti - Prijava

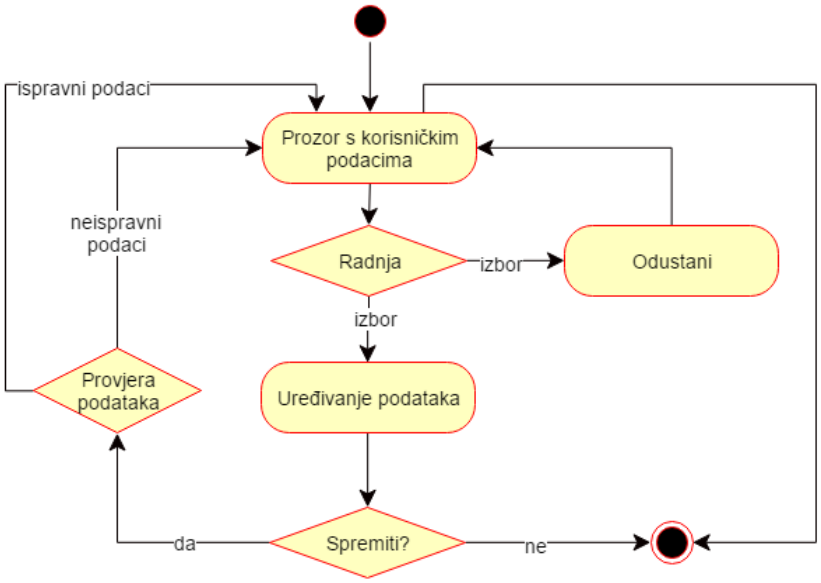


(Izvor: Autor)

Slika 9. prikazuje dijagram aktivnosti za prijavu korisnika. Prilikom dolaska korisnika na prozor za prijavu on ima mogućnost odlaska na registraciju ili na unos svojih podataka za prijavu. Nakon unosa može potvrditi prijavu ili odustati od prijave. Ukoliko prihvati prijavu sustav provjerava podatke pa ako je prijava uspješna korisnik odlazi dalje u sustav. Ako prijava nije uspjela, sustav korisnika preusmjerava na prozor za prijavu.

4.2.3. Uređivanje korisničkih podataka

Slika 10. – Dijagram aktivnosti – Uređivanje korisničkih podataka

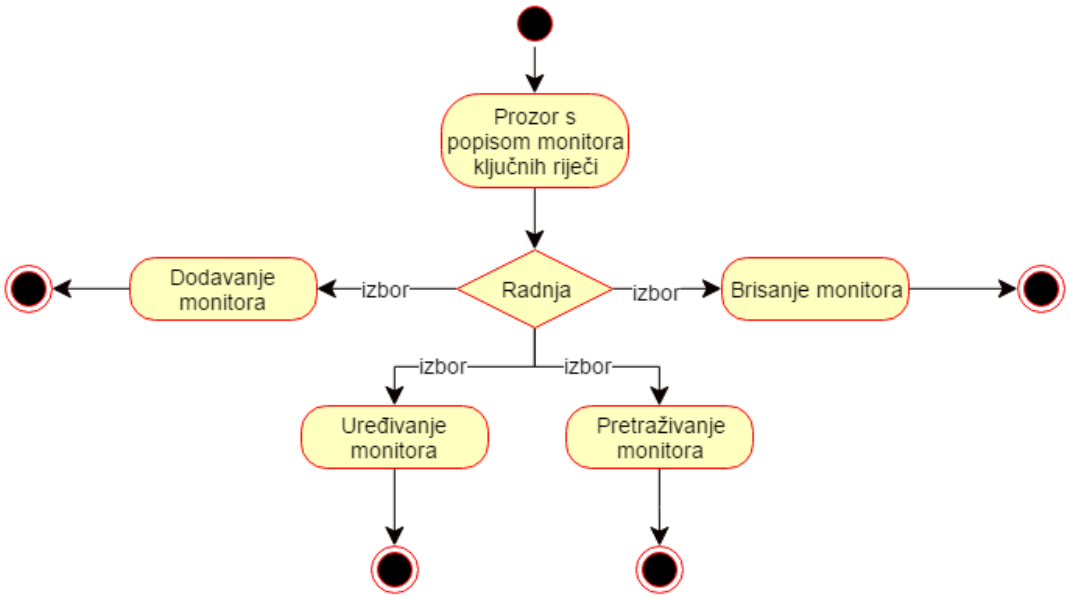


(Izvor: Autor)

Slika 10. prikazuje dijagram aktivnosti za uređivanje korisničkih podataka. Prilikom dolaska korisnika na prozor s korisničkim podacima korisnik ima mogućnost uređivanja podataka ili odustajanja. Nakon uređivanja podataka korisnik ima mogućnost spremanja. Ukoliko se za njega odluči, sustav provjerava podatke te će biti preusmjeren na prozor s korisničim podacima, no s različitim porukama.

4.2.4. Uređivanje praćenja ključnih riječi

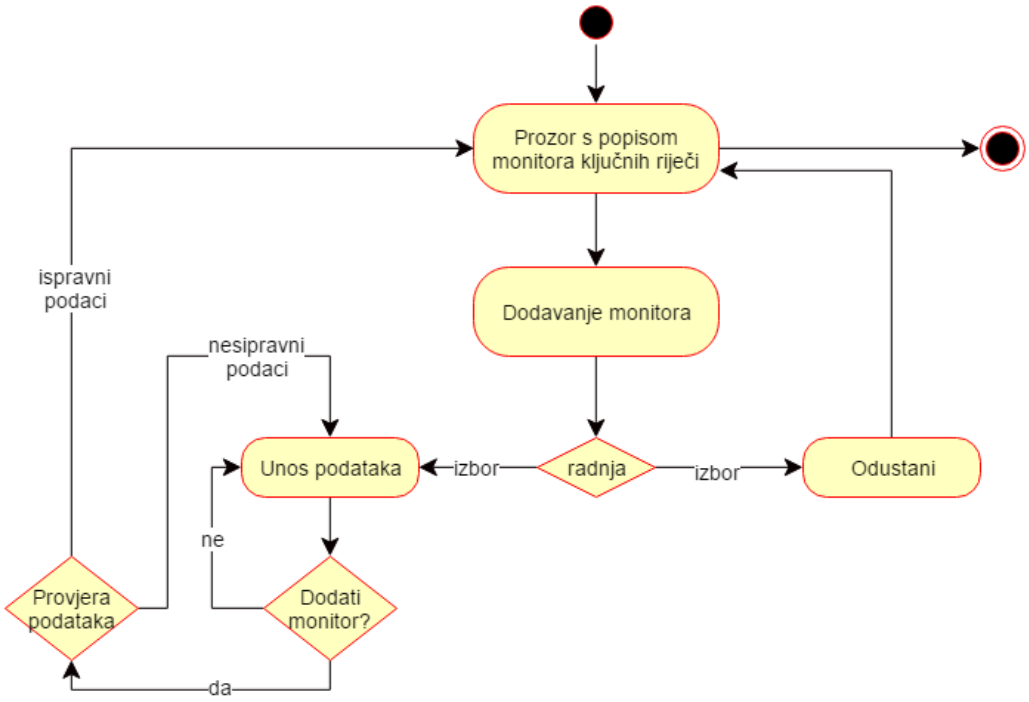
Slika 11. – Dijagram aktivnosti – Uređivanje praćenja ključnih riječi



(Izvor: Autor)

Slika 11. prikazuje dijagram aktivnosti za uređivanje praćenja ključnih riječi. Prilikom dolaska korisnika na prozor s popisom monitora ključnih riječi ima mogućnost dodavanja, uređivanja, pretraživanja ili brisanja monitora.

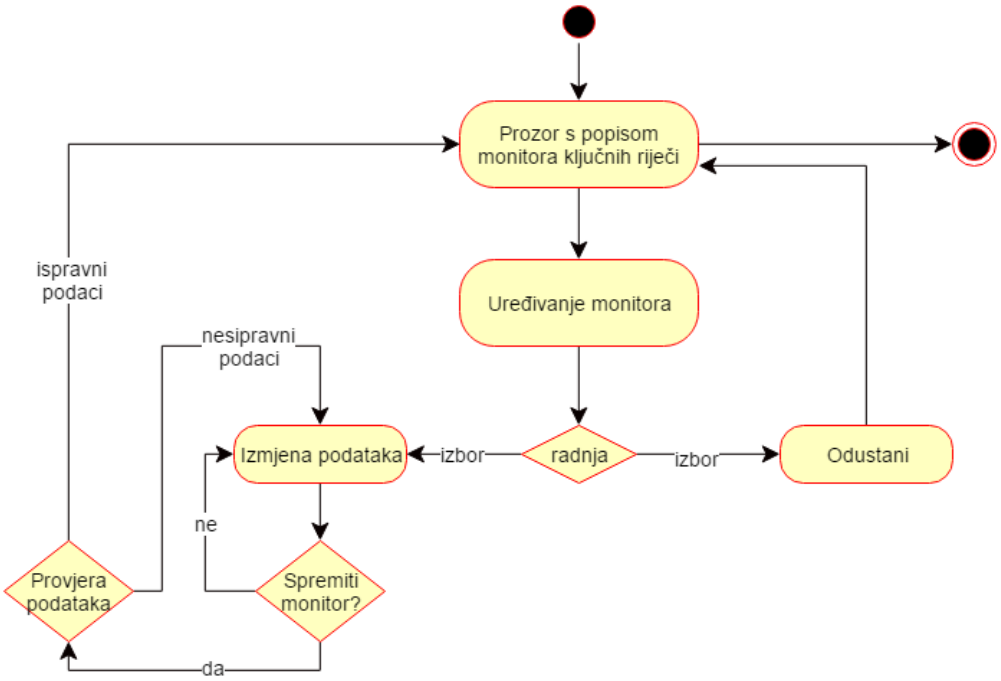
Slika 12. – Dijagram aktivnosti – Dodavanje monitora za praćenje ključnih riječi



(Izvor: Autor)

Slika 12. prikazuje dijagram aktivnosti za dodavanje monitora za praćenje ključnih riječi. Prilikom dolaska korisnika na prozor s popisom monitora ključnih riječi ima mogućnost dodavanja monitora. Može odustati od dodavanja novog monitora ili unositi podatke. Ukoliko se odluči dodati novi monitor sustav će provjeravati podatke, dok će u suprotnom biti vraćen na unos podataka. Ukoliko je provjera podataka uspješna sustav će korisnika prebaciti na prozor s popisom monitora ključnih riječi. U suprotnom vratiti će korisnika na unos podataka.

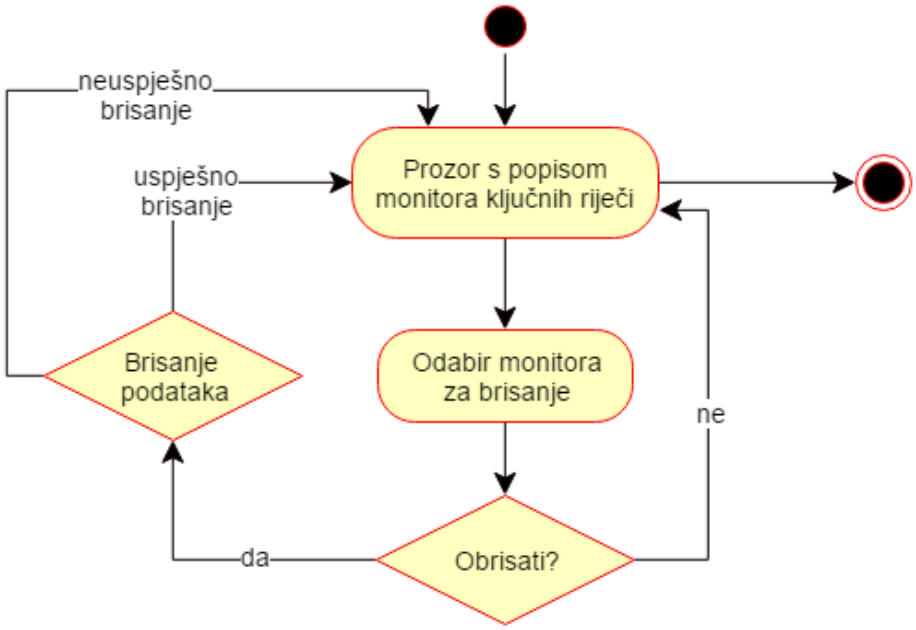
Slika 13. – Dijagram aktivnosti – Uređivanje monitora za praćenje ključnih riječi



(Izvor: Autor)

Slika 13. prikazuje dijagram aktivnosti za uređivanje monitora za praćenje ključnih riječi. Prilikom dolaska korisnika na prozor s popisom monitora ključnih riječi ima mogućnost uređivanja monitora. Može odustati od uređivanja monitora ili može mijenjati podatke. Ukoliko se odluči spremiti promjene sustav će provjeravati podatke, dok će u suprotnom biti vraćen na izmjenu podataka. Ukoliko je provjera podataka uspješna sustav će korisnika prebaciti na prozor s popisom monitora ključnih riječi. U suprotnom vratiti će korisnika na unos podataka.

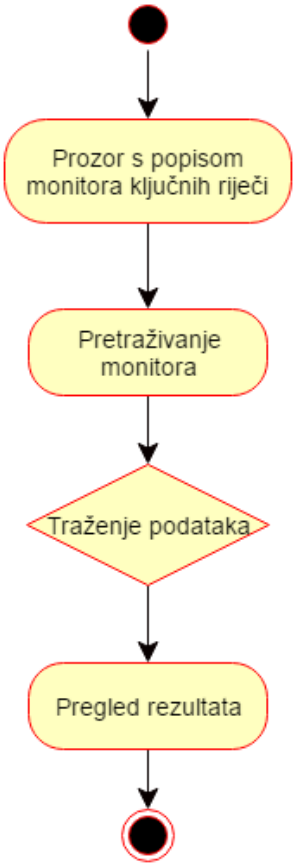
Slika 14. – Dijagram aktivnosti – Brisanje monitora za praćenje ključnih riječi



(Izvor: Autor)

Slika 14. prikazuje dijagram aktivnosti za brisanje monitora za praćenje ključnih riječi. Prilikom dolaska korisnika na prozor s popisom monitora ključnih riječi ima mogućnost brisanja monitora. Može odustati od brisanja monitora ili može označiti željene monitore za brisanje. Ukoliko ih se odluči obrisati sustav će brisati podatke te preusmjeriti korisnika na prozor s popisom monitora ključnih riječi.

Slika 15. – Dijagram aktivnosti – Pretraživanje monitora za praćenje ključnih riječi

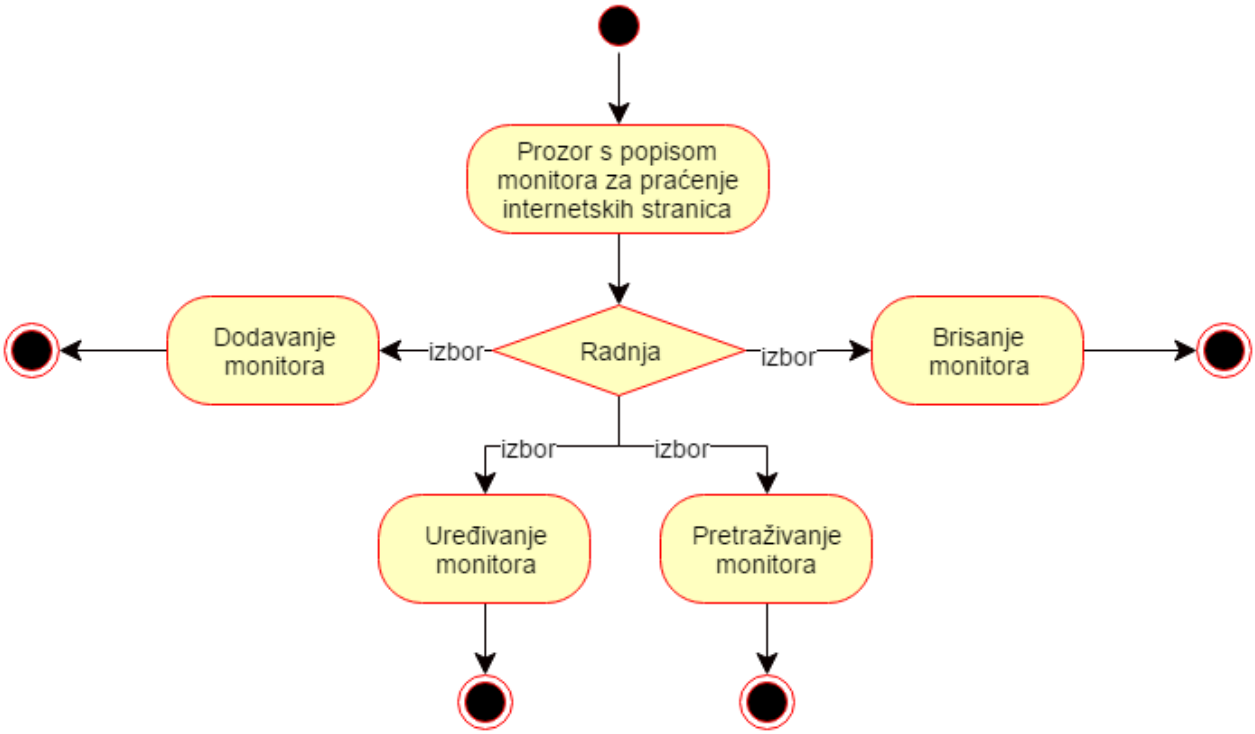


(Izvor: Autor)

Slika 15. prikazuje dijagram aktivnosti za pretraživanje monitora za praćenje ključnih riječi. Prilikom dolaska korisnika na prozor s popisom monitora ključnih riječi ima mogućnost pretraživanja tog popisa. Unosom teksta sustav traži podatke koji odgovaraju unesenom sadržaju. Korisnik nakon tog procesa ima mogućnost pregleda podataka.

4.2.5. Upravljanje praćenja internetskih stranica

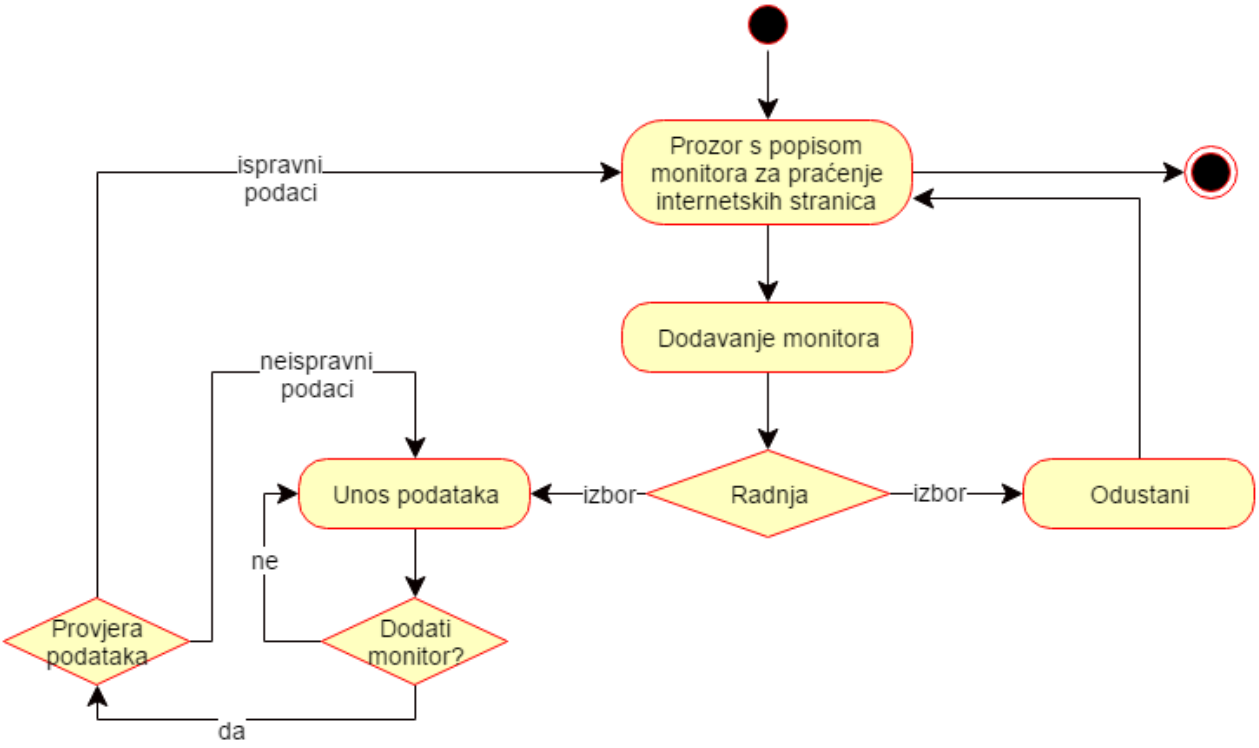
Slika 16. – Dijagram aktivnosti – Upravljanje monitora za praćenje internetskih stranica



(Izvor: Autor)

Slika 11. prikazuje dijagram aktivnosti za uređivanje praćenja internetskih stranica. Prilikom dolaska korisnika na prozor s popisom monitora internetskih stranica ima mogućnost dodavanja, uređivanja, pretraživanja ili brisanja monitora.

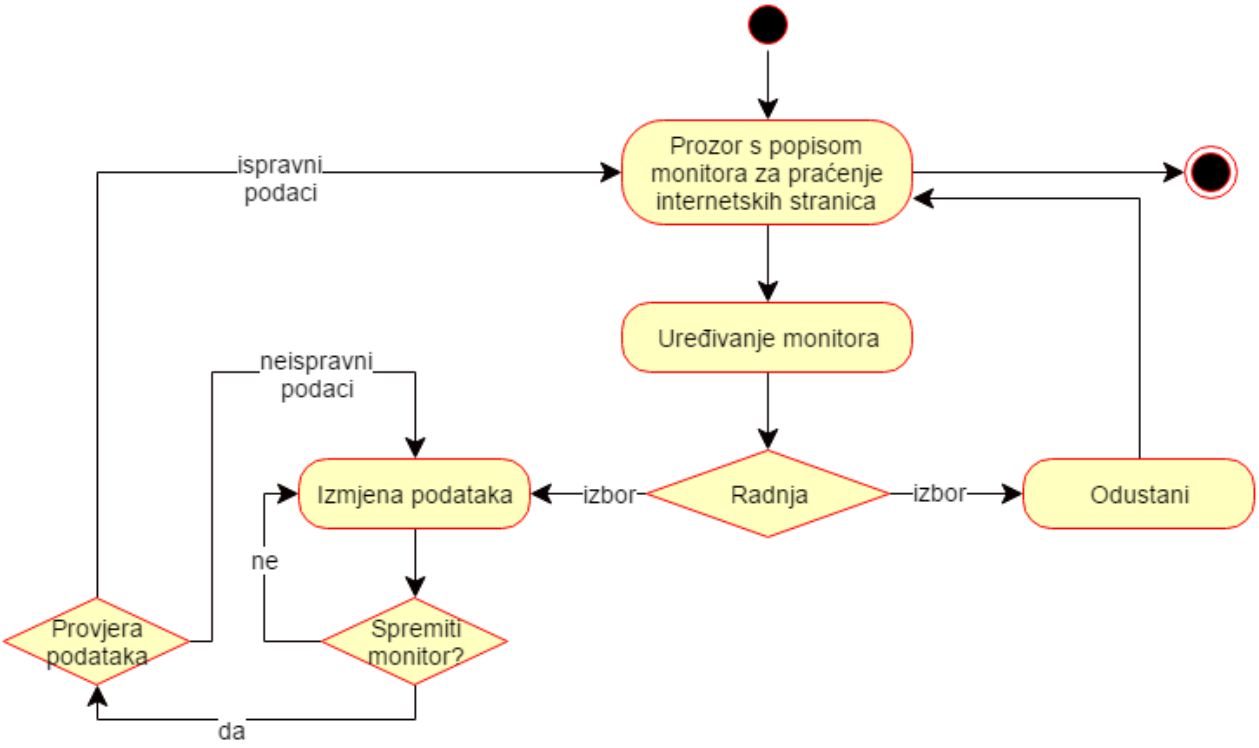
Slika 17. – Dijagram aktivnosti – Dodavanje monitora za praćenje internetskih stranica



(Izvor: Autor)

Slika 17. prikazuje dijagram aktivnosti za dodavanje monitora za praćenje internetskih stranica. Prilikom dolaska korisnika na prozor s popisom monitora internetskih stranica ima mogućnost dodavanja monitora. Može odustati od dodavanja novog monitora ili unositi podatke. Ukoliko se odluči dodati novi monitor sustav će provjeravati podatke, dok će u suprotnom biti vraćen na unos podataka. Ukoliko je provjera podataka uspješna sustav će korisnika prebaciti na prozor s popisom monitora ključnih riječi. U suprotnom vratiti će korisnika na unos podataka.

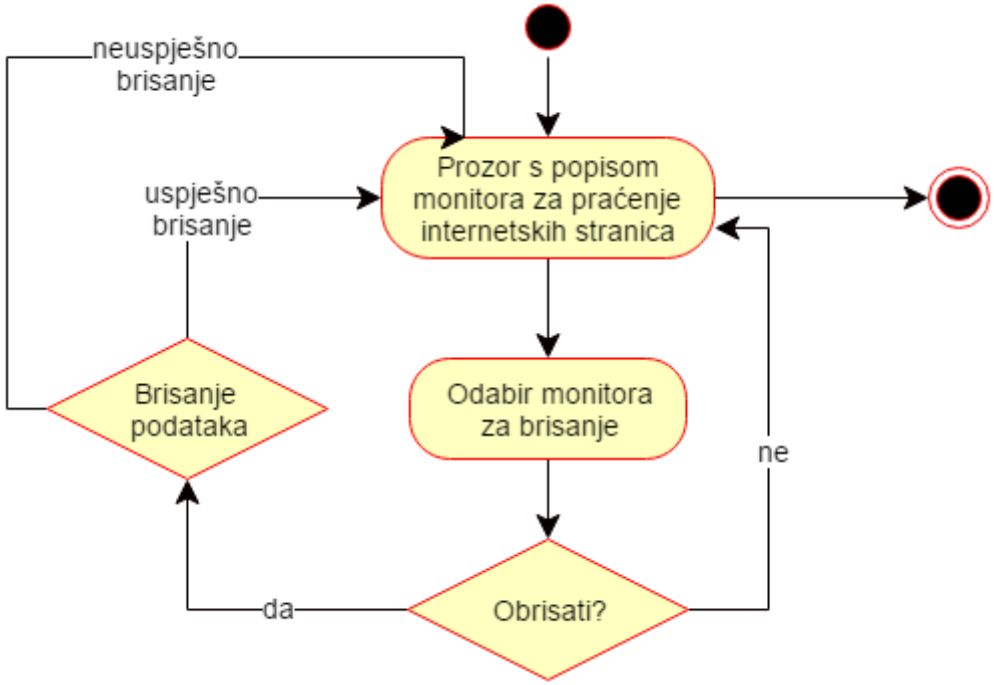
Slika 18. – Dijagram aktivnosti – Uređivanje monitora za praćenje internetskih stranica



(Izvor: Autor)

Slika 18. prikazuje dijagram aktivnosti za uređivanje monitora za praćenje internetskih stranica. Prilikom dolaska korisnika na prozor s popisom monitora internetskih stranica ima mogućnost uređivanja monitora. Može odustati od uređivanja monitora ili može mijenjati podatke. Ukoliko se odluči spremiti promjene sustav će provjeravati podatke, dok će u suprotnom biti vraćen na izmjenu podataka. Ukoliko je provjera podataka uspješna sustav će korisnika prebaciti na prozor s popisom monitora internetskih stranica. U suprotnom vratiti će korisnika na unos podataka.

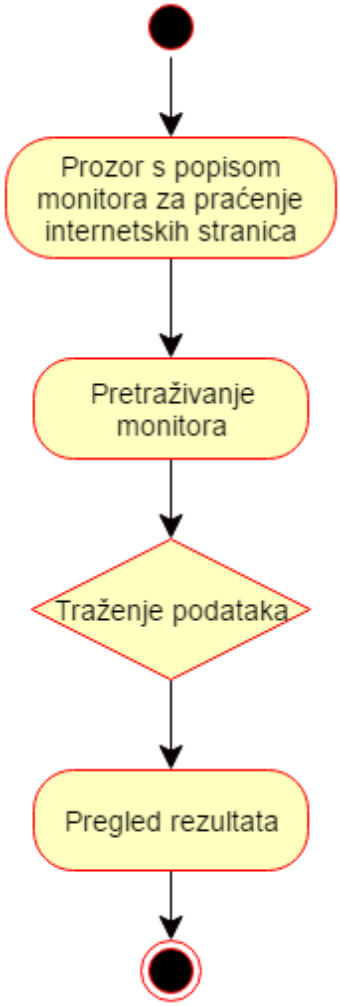
Slika 19. – Dijagram aktivnosti – Brisanje monitora za praćenje internetskih stranica



(Izvor: Autor)

Slika 19. prikazuje dijagram aktivnosti za brisanje monitora za praćenje internetskih stranica. Prilikom dolaska korisnika na prozor s popisom monitora internetskih stranica ima mogućnost brisanja monitora. Može odustati od brisanja monitora ili može označiti željene monitore za brisanje. Ukoliko ih se odluči obrisati sustav će brisati podatke te preusmjeriti korisnika na prozor s popisom monitora internetskih stranica.

Slika 20. – Dijagram aktivnosti – Pretraživanje monitora za praćenje internetskih stranica

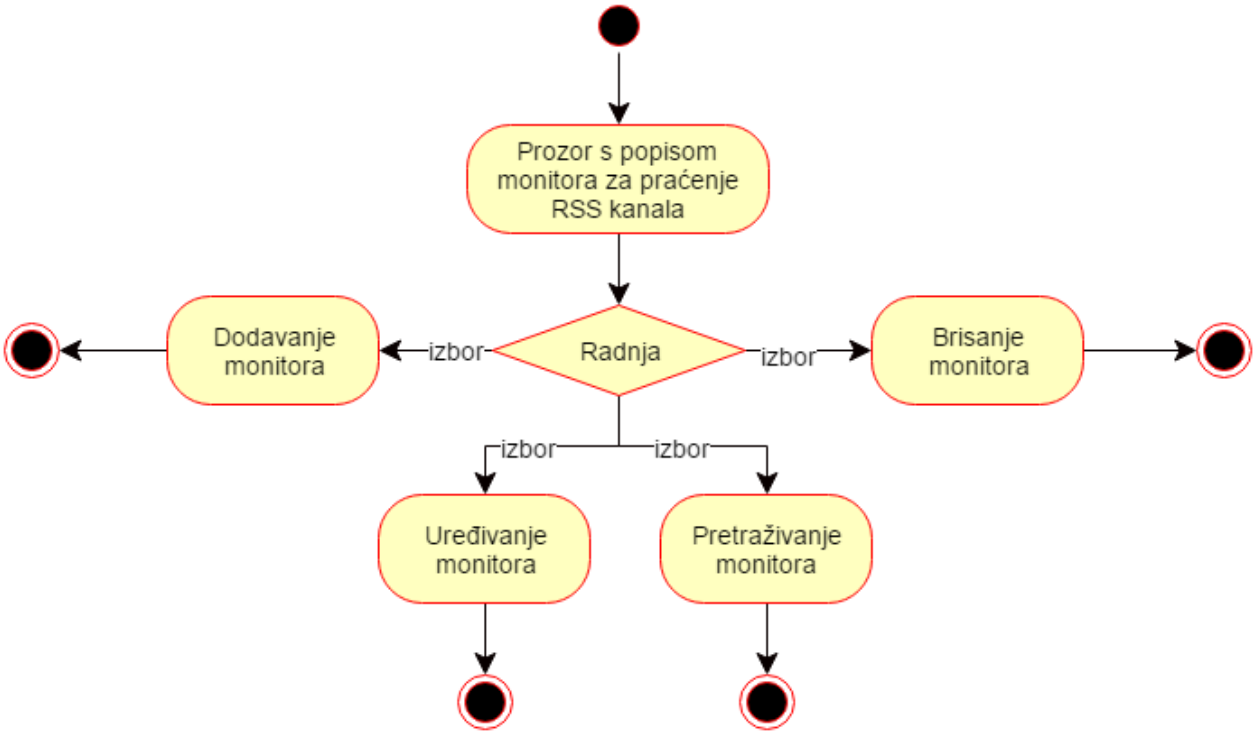


(Izvor: Autor)

Slika 20. prikazuje dijagram aktivnosti za pretraživanje monitora za praćenje internetskih stranica. Prilikom dolaska korisnika na prozor s popisom monitora internetskih stranica ima mogućnost pretraživanja tog popisa. Unosom teksta sustav traži podatke koji odgovaraju unesenom sadržaju. Korisnik nakon tog procesa ima mogućnost pregleda podataka.

4.2.6. Upravljanje praćenja RSS kanala

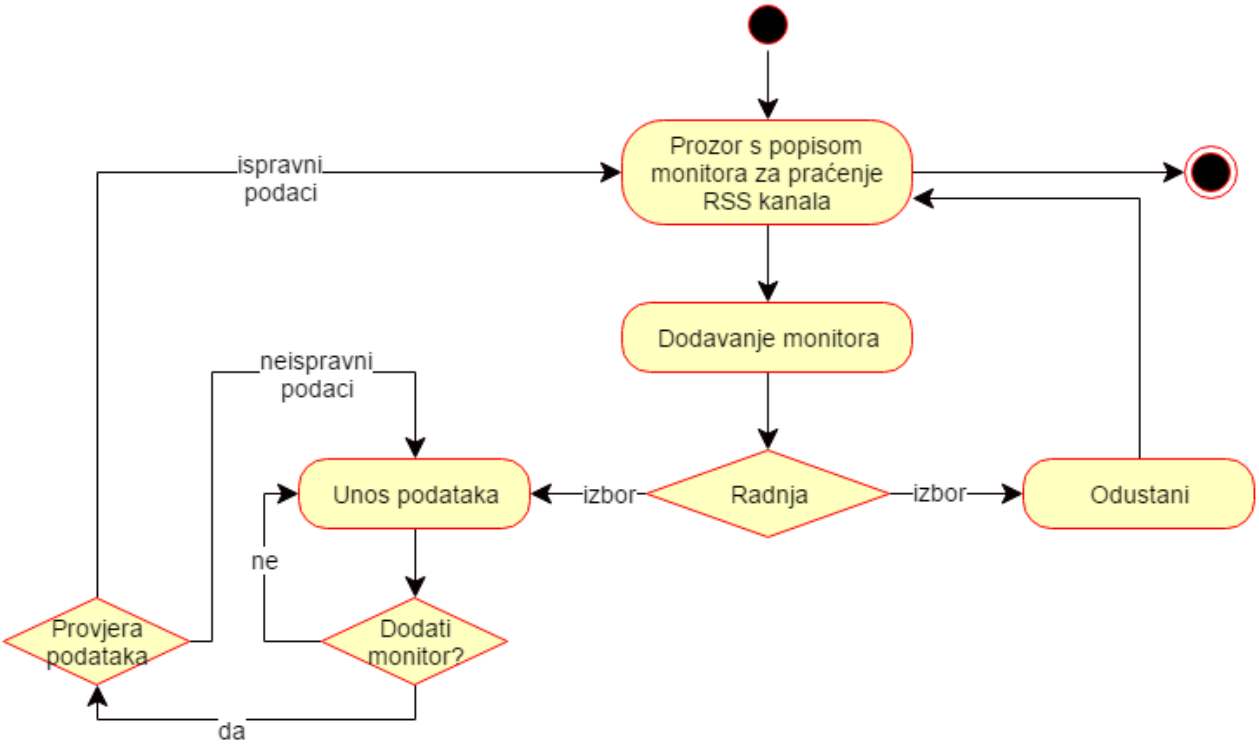
Slika 21. – Dijagram aktivnosti – Upravljanje monitora za praćenje RSS kanala



(Izvor: Autor)

Slika 21. prikazuje dijagram aktivnosti za uređivanje praćenja RSS kanala. Prilikom dolaska korisnika na prozor s popisom monitora RSS kanala ima mogućnost dodavanja, uređivanja, pretraživanja ili brisanja monitora.

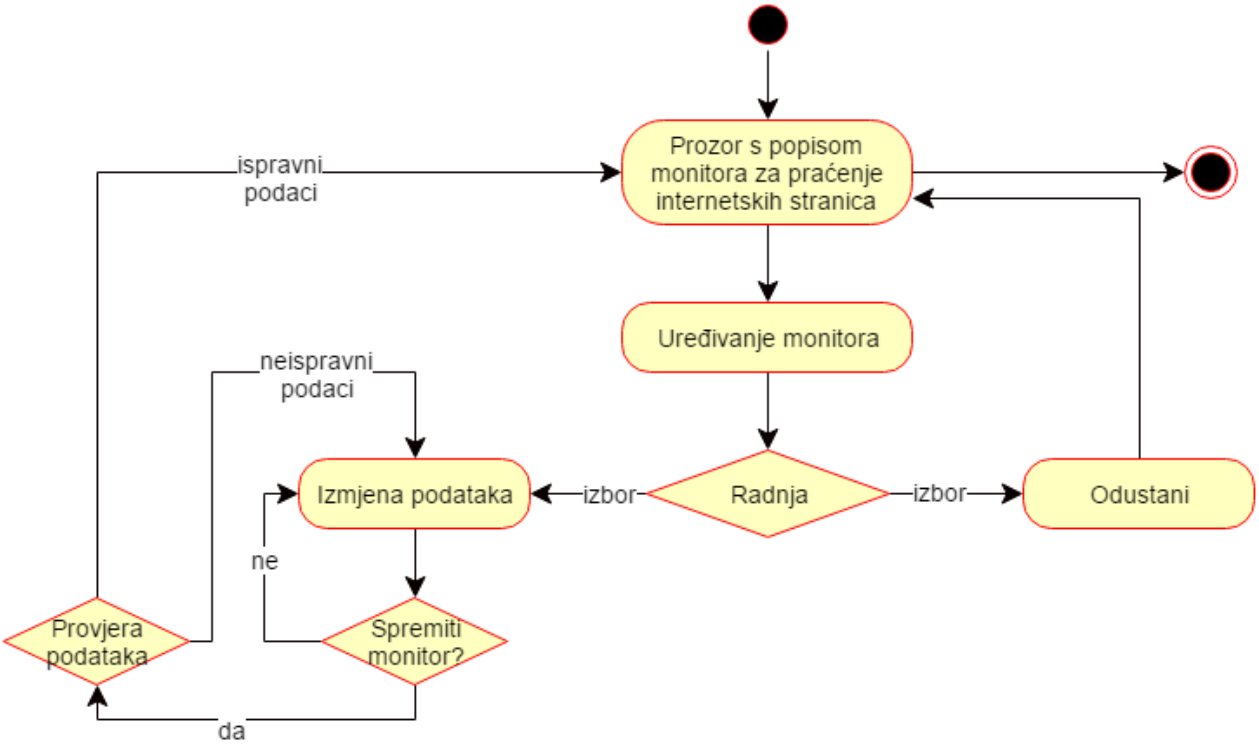
Slika 22. – Dijagram aktivnosti – Dodavanje monitora za praćenje RSS kanala



(Izvor: Autor)

Slika 22. prikazuje dijagram aktivnosti za dodavanje monitora za praćenje RSS kanala. Prilikom dolaska korisnika na prozor s popisom monitora RSS kanala ima mogućnost dodavanja monitora. Može odustati od dodavanja novog monitora ili unositi podatke. Ukoliko se odluči dodati novi monitor sustav će provjeravati podatke, dok će u suprotnom biti vraćen na unos podataka. Ukoliko je provjera podataka uspješna sustav će korisnika prebaciti na prozor s popisom monitora RSS kanala. U suprotnom vratiti će korisnika na unos podataka.

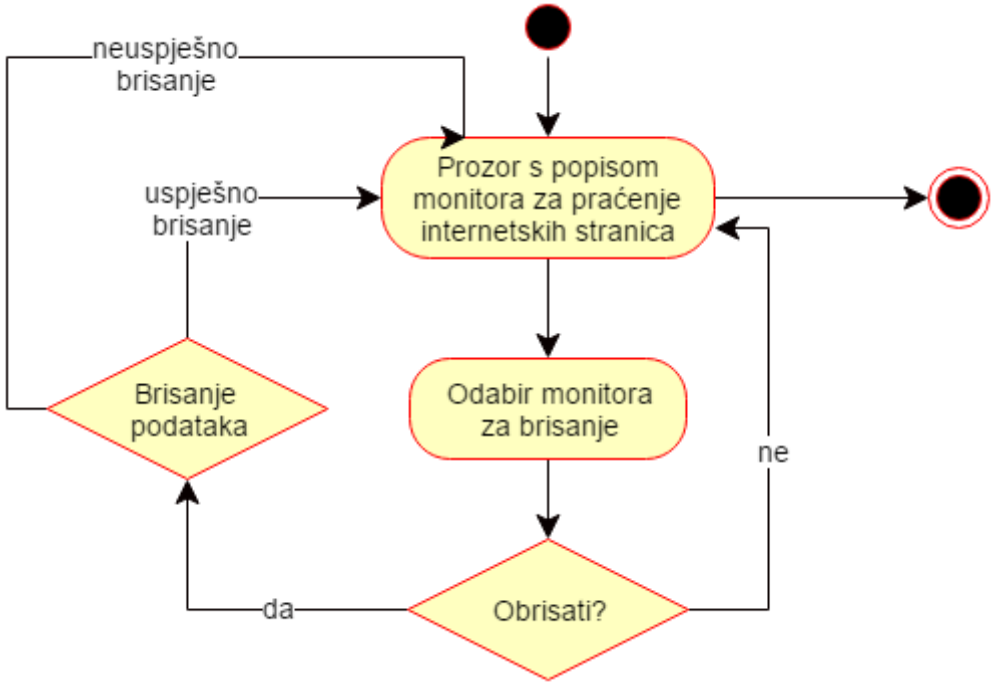
Slika 23. – Dijagram aktivnosti – Uređivanje monitora za praćenje RSS kanala



(Izvor: Autor)

Slika 23. prikazuje dijagram aktivnosti za uređivanje monitora za praćenje RSS kanala. Prilikom dolaska korisnika na prozor s popisom monitora RSS kanala ima mogućnost uređivanja monitora. Može odustati od uređivanja monitora ili može mijenjati podatke. Ukoliko se odluči spremiti promjene sustav će provjeravati podatke, dok će u suprotnom biti vraćen na izmjenu podataka. Ukoliko je provjera podataka uspješna sustav će korisnika prebaciti na prozor s popisom monitora RSS kanala. U suprotnom vratiti će korisnika na unos podataka.

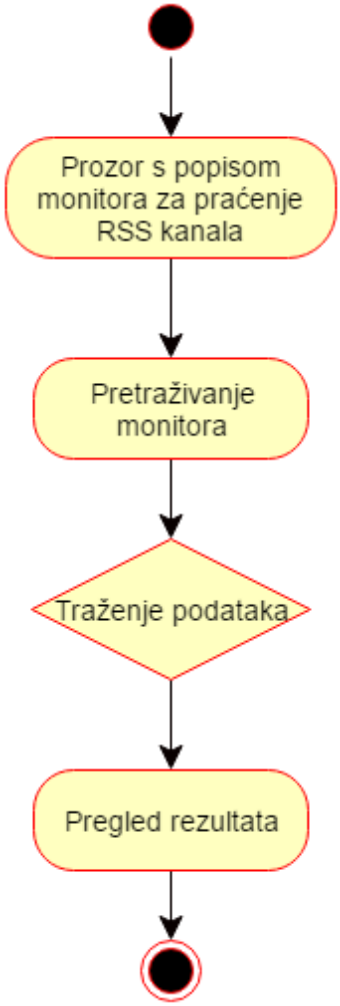
Slika 24. – Dijagram aktivnosti – Brisanje monitora za praćenje RSS kanala



(Izvor: Autor)

Slika 24. prikazuje dijagram aktivnosti za brisanje monitora za praćenje RSS kanala Prilikom dolaska korisnika na prozor s popisom monitora RSS kanala ima mogućnost brisanja monitora. Može odustati od brisanja monitora ili može označiti željene monitore za brisanje. Ukoliko ih se odluči obrisati sustav će brisati podatke te preusmjeriti korisnika na prozor s popisom monitora RSS kanala.

Slika 25. – Dijagram aktivnosti – Pretraživanje monitora za praćenje RSS kanala

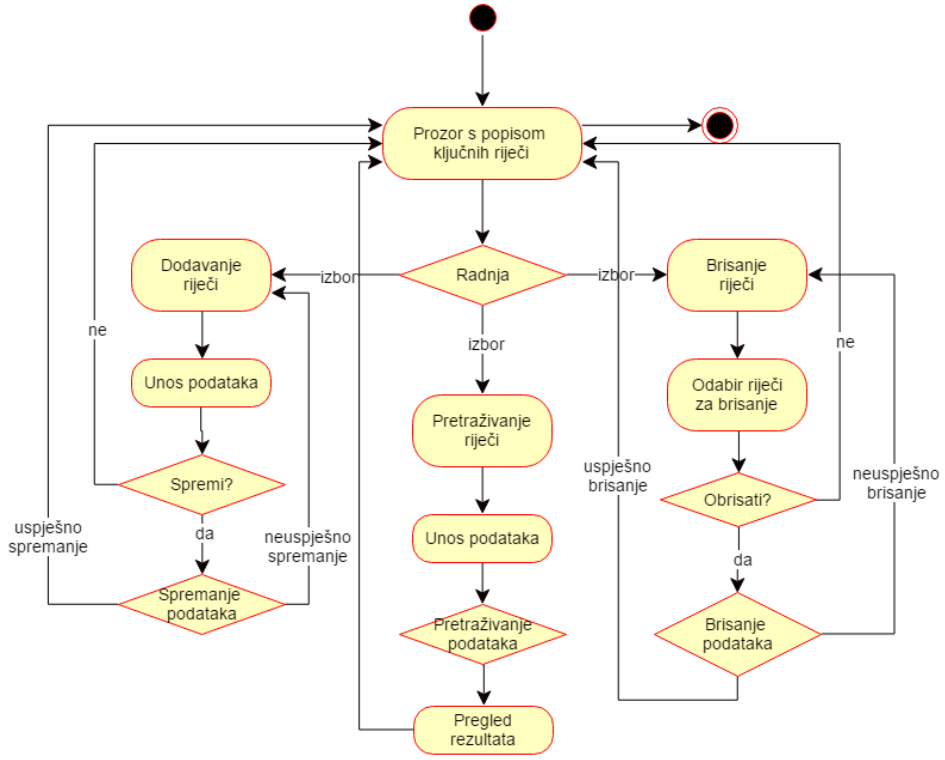


(Izvor: Autor)

Slika 25. prikazuje dijagram aktivnosti za pretraživanje monitora za praćenje RSS kanala. Prilikom dolaska korisnika na prozor s popisom monitora RSS kanala ima mogućnost pretraživanja tog popisa. Unosom teksta sustav traži podatke koji odgovaraju unesenom sadržaju. Korisnik nakon tog procesa ima mogućnost pregleda podataka.

4.2.7. Upravljanje ključnim riječima

Slika 26. – Dijagram aktivnosti – Upravljanje ključnim riječima

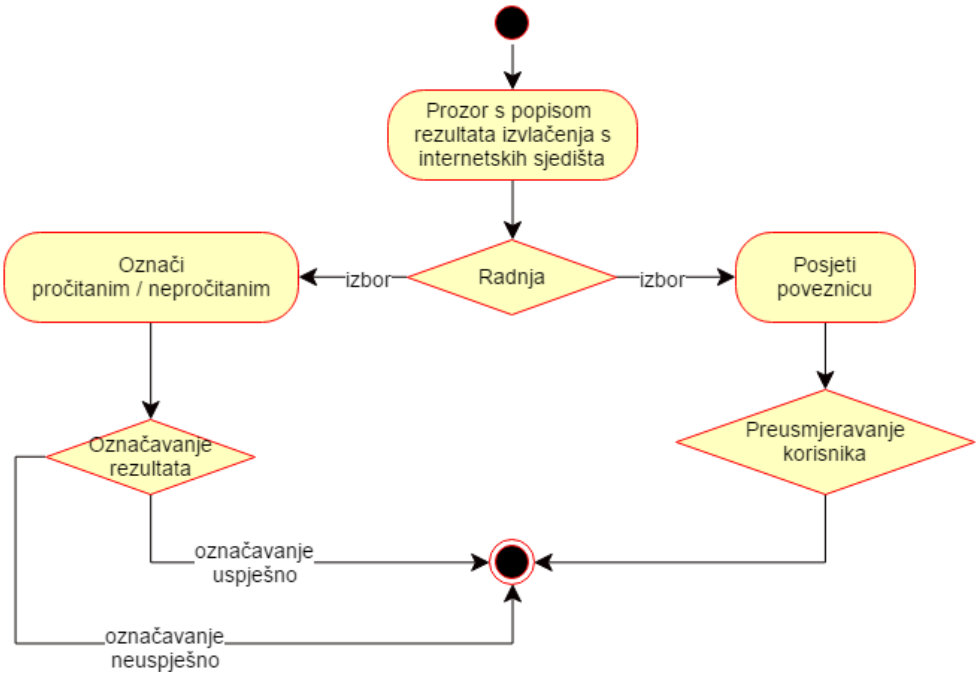


(Izvor: Autor)

Slika 26. prikazuje dijagram aktivnosti za upravljanje ključnim riječima. Korisnik kod dolaska na prozor s popisom ključnih riječi ima mogućnost dodavanja, pretraživanja ili brisanja ključnih riječi. Kod dodavanja i brisanja riječi korisnik ima mogućnost odustajanja ili izvršavanja radnji, a svaka radnja se obavlja u sustavu. Uspješno izvršene radnje vraćaju korisnika na prozor s popisom ključnih riječi, dok neuspjeli vraćaju na prozore za dodavanje ili brisanje riječi.

4.2.8. Upravljanje rezultatima izvlačenja s internetskih sjedišta

Slika 27. – Dijagram aktivnosti – Upravljanje rezultatima izvlačenja s internetskih sjedišta

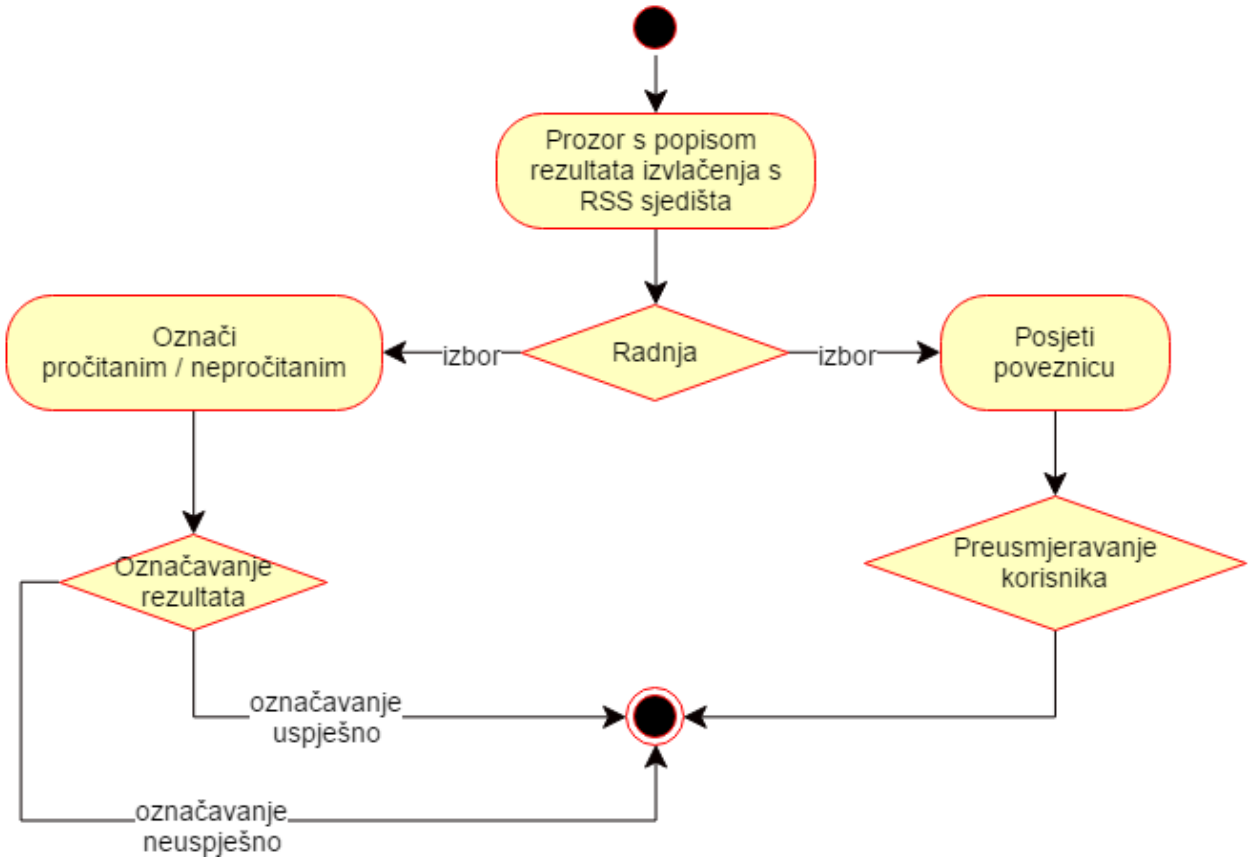


(Izvor: Autor)

Slika 27. prikazuje dijagram aktivnosti za upravljanje rezultatima izvlačenja s internetskih sjedišta. Korisnik kod dolaska na prozor s popisom rezultata izvlačenja s internetskih sjedišta ima mogućnost označiti određeni rezultat kao pročitanim ili nepročitanim te posjetiti poveznicu koju je sustav prikupio prilikom povlačenja podataka.

4.2.9. Upravljanje rezultatima izvlačenja s RSS kanala

Slika 28. – Dijagram aktivnosti – Upravljanje rezultatima izvlačenja s RSS kanala

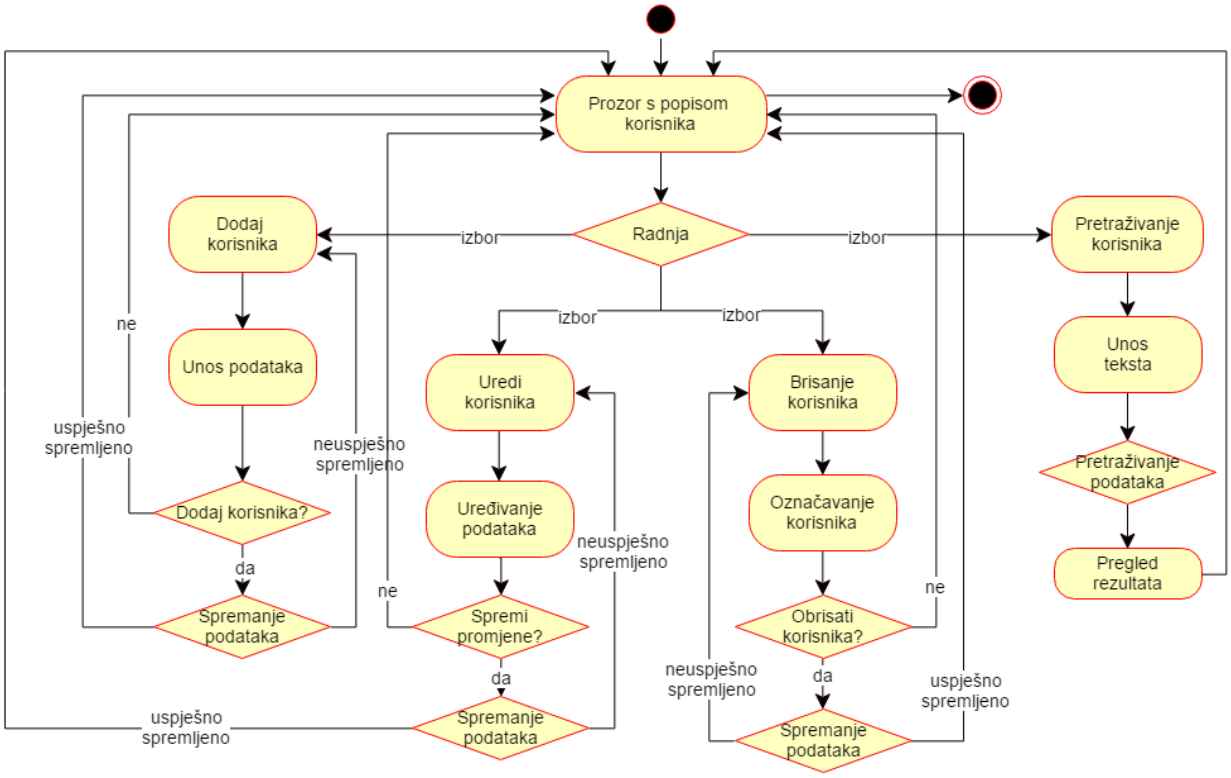


(Izvor: Autor)

Slika 28. prikazuje dijagram aktivnosti za upravljanje rezultatima izvlačenja s RSS kanala. Korisnik kod dolaska na prozor s popisom rezultata izvlačenja s RSS kanala ima mogućnost označiti određeni rezultat kao pročitanim ili nepročitanim te posjetiti poveznicu koju je sustav prikupio prilikom povlačenja podataka.

4.2.10. Upravljanje korisnicima

Slika 29. – Dijagram aktivnosti – Upravljanje korisnicima

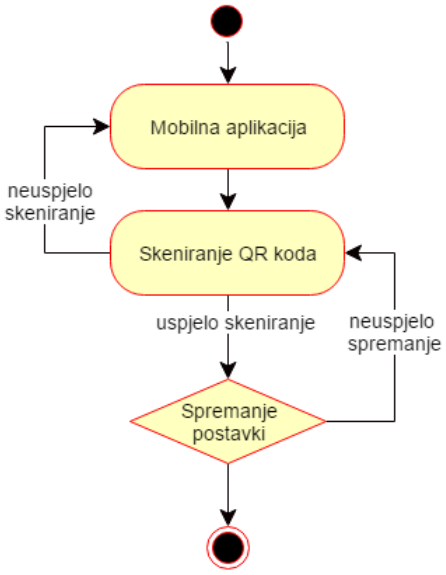


(Izvor: Autor)

Slika 29. prikazuje dijagram aktivnosti za upravljanje korisnicima. Administrator kod dolaska na prozor s popisom korisnika ima mogućnost dodati, urediti, brisati ili pretraživati korisnike. Dodavanje, uređivanje i brisanje zahtjeva administratorov pristanak, nakon čega slijedi spremanje podataka, no administrator se može odlučiti odustati od navedenih radnji.

4.2.11. Prijava korisnika preko mobilne aplikacije

Slika 30. – Dijagram aktivnosti – Prijava korisnika preko mobilne aplikacije

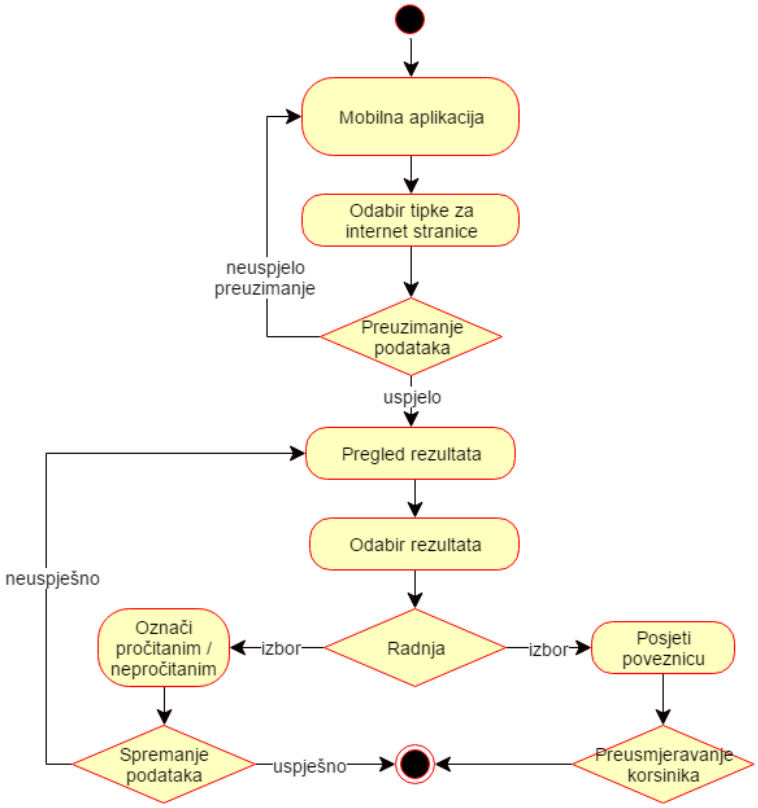


(Izvor: Autor)

Slika 30. prikazuje dijagram aktivnosti za prijavu korisnika preko mobilne aplikacije. Korisnik kod prvog pokretanja aplikacije skenira QR kod iz sučelja internet aplikacije idejnog rješenja nakon čega se kod za korištenje mobilne aplikacije sprema na mobilni uređaj.

4.2.12. Pregled rezultata izvlačenja s internetskih sjedišta preko mobilne aplikacije

Slika 31. – Dijagram aktivnosti – Pregled rezultata izvlačenja s internetskih sjedišta preko mobilne aplikacije

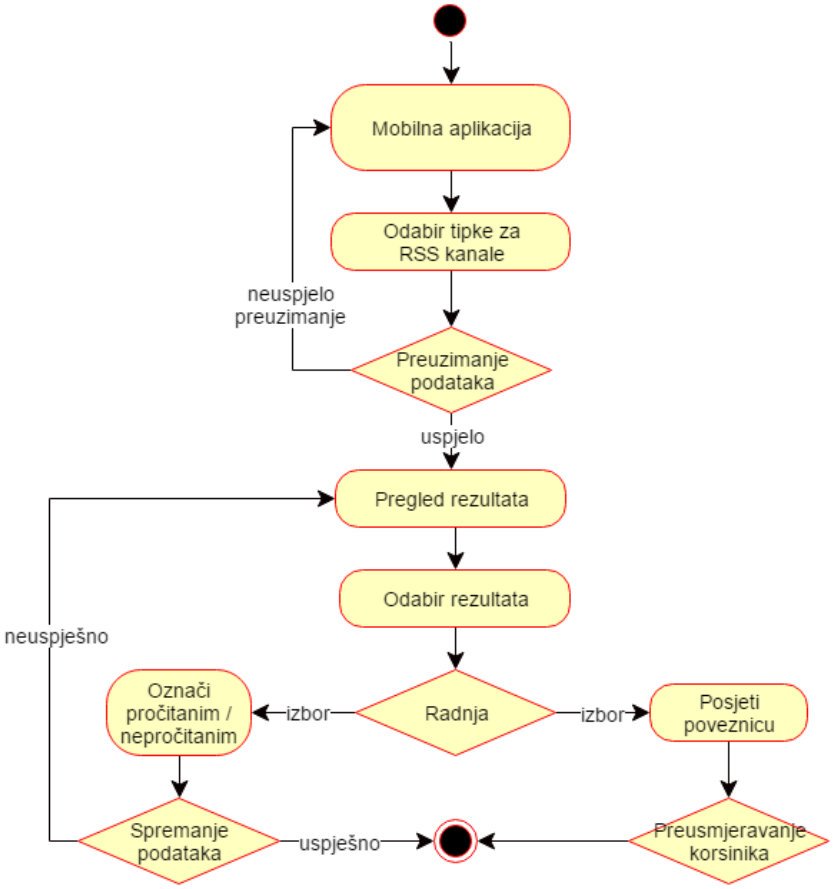


(Izvor: Autor)

Slika 31. prikazuje dijagram aktivnosti za pregled rezultata izvlačenja s internetskih sjedišta preko mobilne aplikacije. Korisnik preko navigacije dolazi do popisa rezultata izvlačenja, nakon čega odabire rezultat te ga može označiti pročitanim ili posjetiti njegovu poveznicu.

4.2.13. Pregled rezultata izvlačenja s RSS kanala preko mobilne aplikacije

Slika 32. – Dijagram aktivnosti – Pregled rezultata izvlačenja s RSS kanala preko mobilne aplikacije

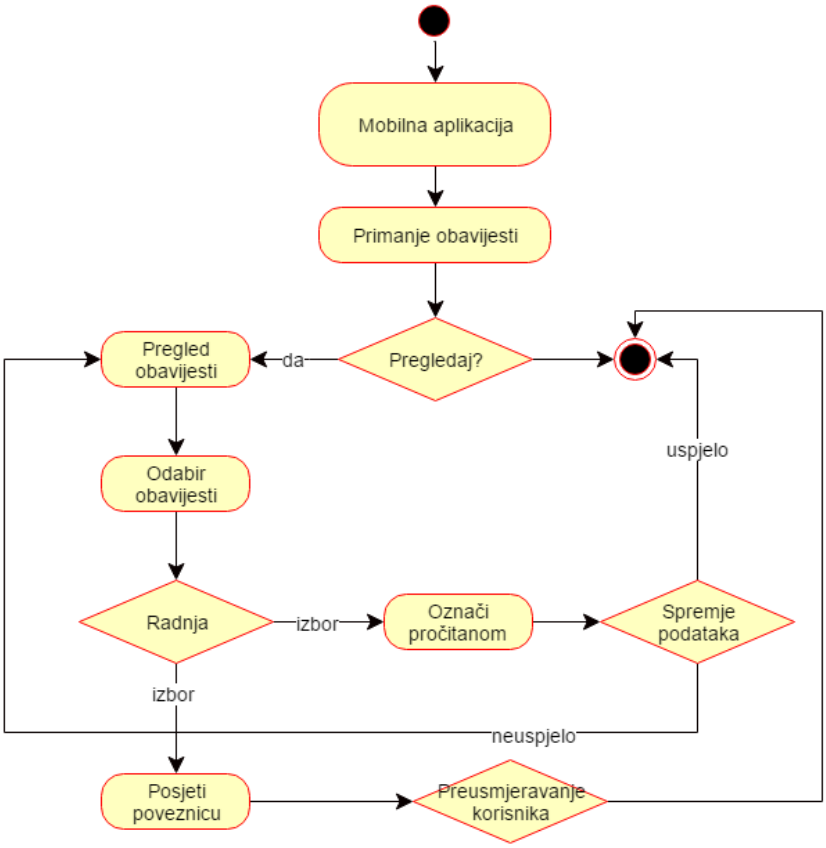


(Izvor: Autor)

Slika 32. prikazuje dijagram aktivnosti za pregled rezultata izvlačenja s RSS kanala preko mobilne aplikacije. Korisnik preko navigacije dolazi do popisa rezultata izvlačenja, nakon čega odabire rezultat te ga može označiti pročitanim ili posjetiti njegovu poveznicu.

4.2.14. Pregled obavijesti preko mobilne aplikacije

Slika 33. – Dijagram aktivnosti – Pregled obavijesti preko mobilne aplikacije



(Izvor: Autor)

Slika 33. prikazuje dijagram aktivnosti za pregled obavijesti preko mobilne aplikacije. Korisnik dobiva obavijest pa ima mogućnost pregleda i odabira obavijesti, nakon čega ju može označiti pročitanom ili posjetiti njenu poveznicu.

4.3. Mockup grafičkog sučelja sustava

4.3.1. Prijava

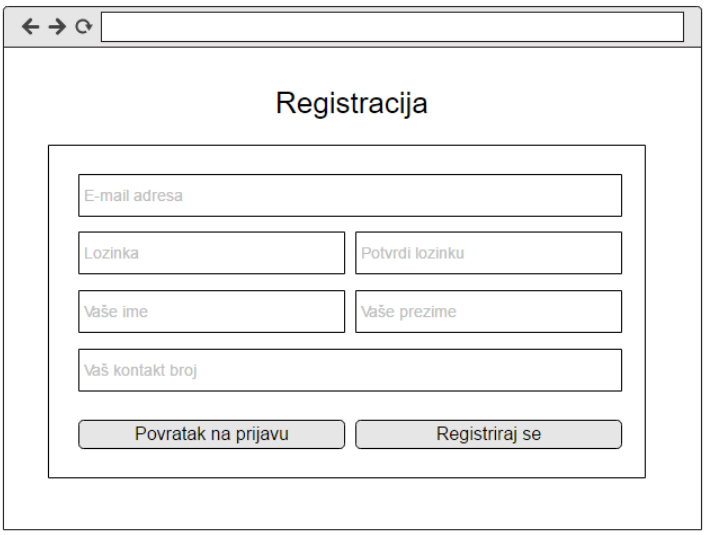
Slika 34. – Mockup – Prozor za prijavu



(Izvor: Autor)

4.3.2. Registracija

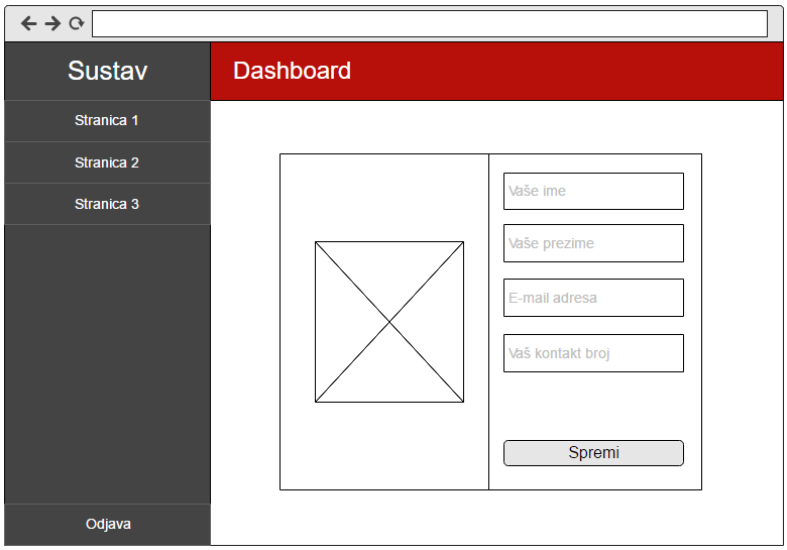
Slika 35. – Mockup – Prozor za registraciju



(Izvor: Autor)

4.3.3. Uređivanje korisničkih podataka

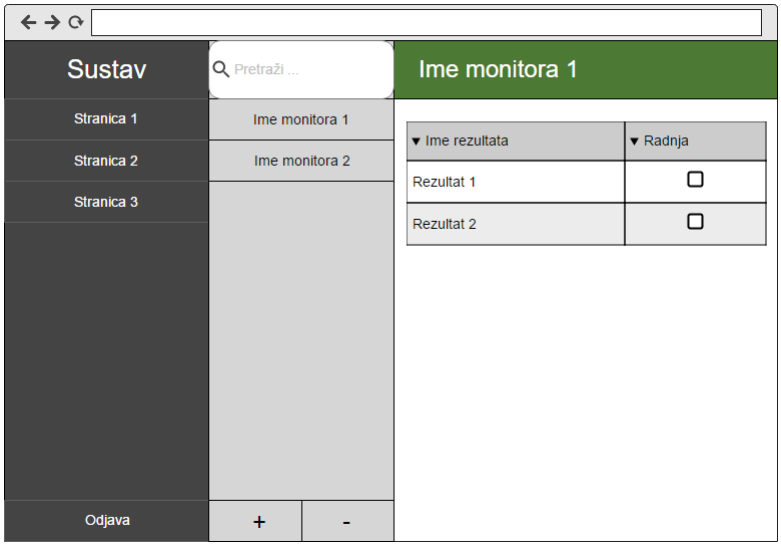
Slika 36. – Mockup – Prozor za uređivanje korisničkih podataka



(Izvor: Autor)

4.3.4. Upravljanje monitorima i rezultatima

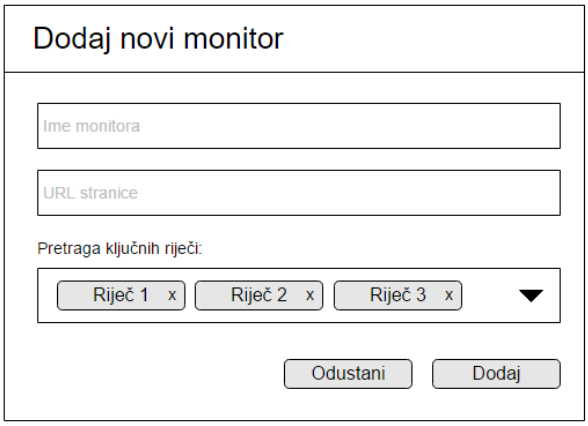
Slika 37. – Mockup – Prozor za pregled monitora i rezultata



(Izvor: Autor)

4.3.5. Dodavanje, uređivanje monitora

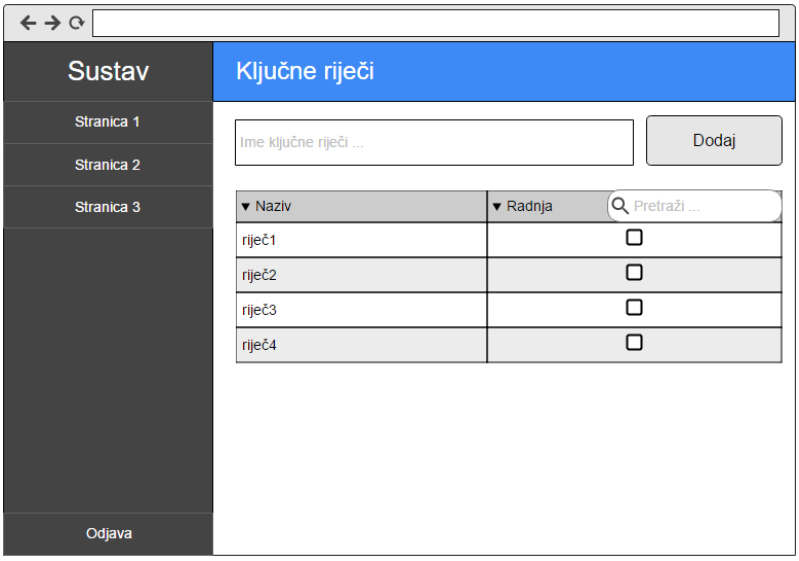
Slika 38. – Mockup – Prozor za dodavanje ili uređivanje monitora



(Izvor: Autor)

4.3.6. Upravljanje ključnim riječima

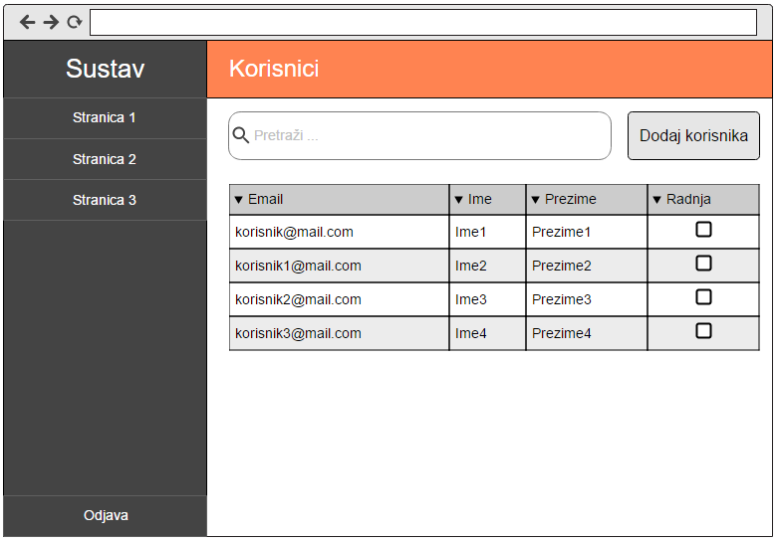
Slika 39. – Mockup – Prozor za upravljanje ključnim riječima



(Izvor: Autor)

4.3.7. Upravljanje korisnicima

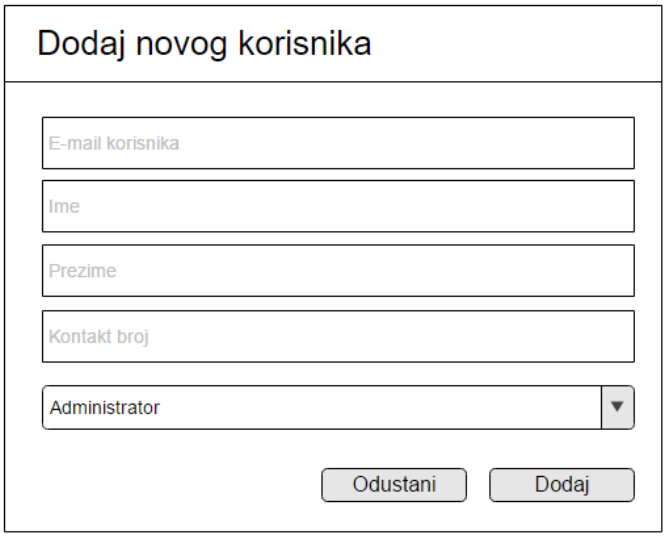
Slika 40. – Mockup – Prozor za upravljanje korisnicima



(Izvor: Autor)

4.3.8. Dodavanje, uređivanje korisnika

Slika 41. – Mockup – Prozor za dodavanje ili uređivanje korisnika



(Izvor: Autor)

4.3.9. Mobilna aplikacija – izbornik

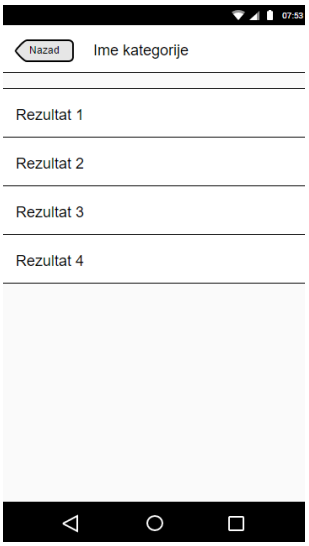
Slika 42. – Mockup – Mobilna aplikacija - Izbornik mobilne aplikacije



(Izvor: Autor)

4.3.10. Mobilna aplikacija – Pregled rezultata monitora

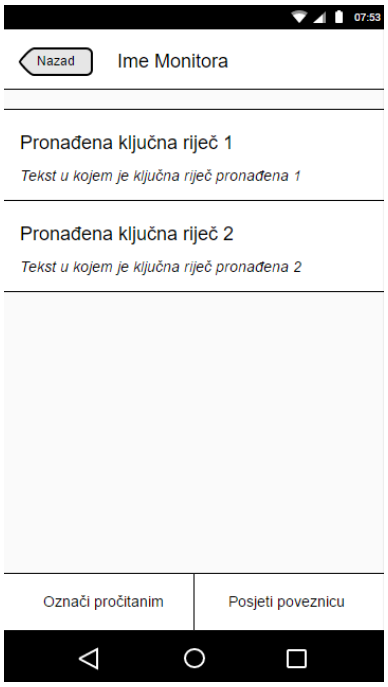
Slika 43. – Mockup – Mobilna aplikacija - Pregled rezultata monitora



(Izvor: Autor)

4.3.11. Mobilna aplikacija – Uređivanje rezultata

Slika 44. – Mockup – Mobilna aplikacija - Pregled uređivanja rezultata



(Izvor: Autor)

5. OPIS SUSTAVA

Ovaj sustav nosi ime „**Notis**“, lat. riječ za obavijest.

Slika 45. – Logotip sustava Notis



(Izvor: Autor)

5.1. Struktura sustava

Konceptualnim modelom predstavljen je cjelokupni sustav i odnosi među dijelovima sustava. Na temelju njega mogu se odrediti logički dijelovi sustava koje je potrebno napraviti, odnosno isprogramirati. Tako se sustav dijeli na:

1. **Administracijsko sučelje / aplikacija** – dio sustava namijenjen za administraciju korisnicima sustava, pojedinim postavkama sustava te kreiranju i administraciji sustava za praćenje informacija na internetu
2. **Notifikacijski (obavještajni) server** – dio sustava namijenjen upravljanju distribucije obavijesti svim korisnicima sustava
3. **Aplikacija za ekstrakciju i provjeru ključnih riječi** – dio sustava namijenjen obavljanju poslovne logike, odnosno sustav koji će intervalno putem određenih algoritama provjeravati kreirane monitore
4. **Baza podataka** – dio sustava koji služi za pohranu postavki, korisnika sustava, monitore te pohranu rezultata pretraživanja i izvlačenja od strane svakog pojedinog monitora
5. **Pomoćni server** – dio sustava koji služi za dodatnu poslovnu logiku

6. **Mobilna aplikacija** – dio sustava za primanje i uređivanje obavijesti te pregled rezultata praćenja i izvlačenja monitora

5.2. Tehnologije

Struktura sustava je poznata, stoga je potrebno odrediti putem kojih tehnologija će biti svaki od njih biti izrađivani. Kod svake aplikacije potrebna je jednostavnost održavanja i razvoja te performanse kojima će aplikacija odrađivati neku poslovnu logiku. **Administracijsko sučelje** nije opterećenje za sustav, stoga ono može biti izrađivano i u PHP programskom jeziku. Razlog odabira PHP programskog jezika je zbog *frameworka* u kojem će ovaj dio sustava biti izrađen – **Laravel Framework**. **Laravel** je popularan PHP *framework* za razvoj internetskih aplikacija, koji daje veliki naglasak i važnost u jednostavnosti izrade sustava i brzini izrade. Baziran je na MVC arhitekturi, što za idejni sustav uvelike odgovara. Na taj način sustav postaje skalabilniji jer se preko **Laravela** mogu napraviti API pristupi za izvršavanje određenih radnji za vanjske aplikacije. Tako primjerice mobilna aplikacija može dobivati rezultate rada monitora u JSON obliku jednostavnim pozivom na određeni URI od administratorske aplikacije. (Laravel PHP Framework, 06.09.2016.) Također, **Laravel** sadrži i moćan sustav za objektno-relacijsko mapiranje „**Eloquent**“, koji na temelju kreiranih modela sustava radi upite na bazu podataka bez da programer piše te upite te ih omogućava sustavu na korištenje u obliku objekata, što uvelike čini razvoj objektno-orijentiranim i puno čišćim. (Eloquent: Getting Started, 06.09.2016.)

Obavještavanje korisnika mora se odvijati u realnom vremenu, što znači da odmah čim npr. sustav pronađe nekakvu ključnu riječ, on tu obavijest mora poslati svim korisnicima sustava. Kako bi se to što jednostavnije napravilo u realnom vremenu, sustav za obavještavanje koristiti će *Websocket* tehnologiju, koja omogućava razmjenu informacija u realnom vremenu putem interneta. Za kreiranje *Websocket* servera koristiti će se **Node.js** i **Socket.IO**, jer je kreiranje tog servera s navedenim tehnologijama vrlo brzo i jednostavno, a i vrlo je skalabilno. (About Node.js, 06.09.2016.)

Sustav za pretraživanje i izvlačenje podataka s interneta mora biti vrlo brzo, a pošto sustav može sadržavati na tisuće monitora, koji svoje algoritme moraju pokretati paralelno jedne s drugima, stabilnost i performanse programskog jezika koji ih izvodi mora biti vrlo visok. Iz tog

razloga, za ovaj dio sustava koristiti će se **Java** programski jezik, koji se koristi i u izradi kompleksnih, *enterprise* aplikacija. Radi se o aplikaciji bez grafičkog sučelja, a neće se raditi ni kao *servlet* da bi se omogućile maksimalne performanse.

Pomoćni server također će biti rađen u **Java** programskom jeziku radi iskorištavanja programskog koda sustava za pretraživanje i izvlačenje podataka, ali i radi performansi. Radi se o *servlet* aplikaciji jer će se na njega morati raditi pozivi vanjskih aplikacija, kao npr. mobilne aplikacije za potrebe „pročitao sam obavijest“ naredbi.

Mobilna aplikacija biti će izrađivana putem **Ionic Frameworka**. Radi se o besplatnoj aplikaciji koja omogućava kreiranje mobilnih aplikacija na temelju *HTML*, *CSS* i *Javascript* tehnologija. Nudi već gotove komponente za razvoj mobilnih aplikacija koje su kompatibilne s *Android*, *iOS* i *Windows Phone* operativnim sustavima. Zbog navedenih mogućnosti, razvoj mobilne aplikacije će biti jednostavniji i brži nego razvoj samo za jedan operativni sustav. (Ionic Documentation Overview, 06.09.2016.)

Baza podataka će biti bazirana na **MySQL** sustavu.

Dizajn i izgled sustava u potpunosti je podržan i kreiran putem **Semantic UI**.

5.3. Razvoja okolina

Definirane su tehnologije pomoću kojih će biti kreiran sustav, stoga se trebaju odrediti i alati s kojima će biti izrađivani. Za razvoj *Java* aplikacija korišten je **IntelliJ IDEA 2016**, za razvoj PHP aplikacija **PhpStorm 2016**, za Javascript **Webstorm 2016**. Svi navedeni alati su IDE alati koji su posebno prilagođeni za razvoj aplikacija u određenom programskom jeziku pa nude mogućnosti koje olakšavaju razvoj aplikacija.

Pošto se koristi **Laravel**, razvojni tim tog *frameworka* nudi i posebno prilagođenu razvojnu okolinu za razvoj *Laravel* aplikacija – „**Homestead“**. Radi se o virtualnom uređaju baziranom na **Ubuntu Linux** koji u sebi sadrži već instalirane tehnologije potrebne za razvoj *Laravel* aplikacija, od kojih su, između ostalih, *Nginx*, PHP, *MySQL*, *Git*, *Mailcatcher* i ostale. *Homestead* tako služi kao zatvorena radna i razvojna okolina unutar razvojne okoline, a sve

promjene nad *Homesteadom* ostaju spremljene unutar njega pa je time ta okolina i prijenosna. Za njegov rad potrebno je imati i aplikaciju koja može stvarati i pokretati virtualne uređaje na **Windows 10 Home** operativnom sustavu, a za to će se koristiti **VirtualBox 5.1**. *Homestead* je zatvorena okolina, ali postaje otvorena koristeći Vagrant, aplikaciju koja spaja VirtualBox s *Homesteadom* te radi datotečnu povezanost s glavnim operativnim sustavom (*Windows 10*). Na taj način sve PHP skripte mogu biti spremljene i uređivane od strane aplikacija na glavnom operativnom sustavu, a koje u isto vrijeme čita i *Homestead*. Rezultat je da glavni operativni sustav na sebi ne mora imati instalirane instance PHP, *MySQL* i ostalih, već ih može koristiti direktno iz *Homesteada*.

Za rad nad bazom podataka koristi se **HeidiSQL** aplikacija.

Aplikacije su razvijane na računalu s procesorom *Intel i5 6500*, 16GB RAM memorije i pokretane sa SSD diska.

Za internet preglednik koristi se **Google Chrome** v53.

5.4. Administracijsko sučelje

Razvoj sustava počinje s administracijskim sučeljem. Pošto je relacijski model baze podataka realiziran, a *Laravel* nudi mogućnost kreiranja tzv. „migracija“, skripti u kojem se u obliku PHP koda kreiraju tablice i veze između njih, potrebno je na taj način kreirati fizičku bazu podataka. *Laravel* posjeduje „**Artisan**“, pomoćnu PHP skriptu koja služi kao naredbeni redak za rad s *Laravelom*. Putem nje mogu se kreirati modeli, migracije, kontroleri (pošto je MVC arhitektura), pogledi i ostalo, ali može se i kreirati početna verzija sustava.

PhpStorm aplikacija nudi mogućnost spajanja na određeni *Vagrant* uređaj, u ovom slučaju *Homestead*. Na taj način s njim se može upravljati (pokretati, gasiti i resetirati) direktno iz *Phpstorm* aplikacije.

Da bi preuzeli *Laravel*, verziju 5.2, potrebno je se spojiti na *Homestead* putem SSH pristupa. Svi potrebni podaci za spajanje mogu se pronaći u jako kvalitetnoj i opširnoj

dokumentaciji za *Laravel* i *Homestead*. Dokumentacija u detalje prolazi kroz prilagođavanje okoline za rad i prvo pokretanje *Homesteada*. (Laravel Homestead, 06.09.2016.)

Prilikom postavki sustava, određena je IP adresa na **192.168.10.10**, što je lokalna IP adresa *Homesteada*, na kojoj će biti i pogonjena *Laravel* aplikacija. U *hosts* datoteci glavnog operativnog sustava postavlja se ista IP adresa na domenu „*notis.dev*“, čime će se omogućiti, upisivanjem te iste domene u preglednik, odlazak i učitavanje *Laravel* aplikacije. (Laravel Homestead, 06.09.2016.)

Velika prednost korištenja *Homesteada* je također ta što on u sebi ima već instaliran **Composer**, sustav putem kojeg se na jednostavan način može upravljati gotovim skriptama o kojima može zavisiti neki dio *Notisa*. Primjerice, putem *Composera* se i instalira *Laravel* putem naredbe „*composer global require "laravel/installer"*“ u SSH naredbenom retku. *Laravel* nudi mogućnost „brzog starta“, tj. da on samostalno napravi čitavu strukturu potrebnih datoteka i postavki za novu aplikaciju. To se čini naredbom „*laravel new notis*“.

Slijedi postavljanje „*.env*“ datoteke, koja Laravelu služi kao izvor statičkih postavki gdje se postavlja:

```
APP_URL=http://notis.dev
APP_NAME=Notis.IO
DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=33060
DB_DATABASE=notis-dev
DB_USERNAME=homestead
DB_PASSWORD=secret
```

Postavljaju se podaci za spajanje *Laravela* na bazu podataka, koju stvara *Homestead* prilikom prvog pokretanja.

Generiranje QR koda omogućeno je dodatkom *Simple QrCode*. (todo)

5.4.1. Migracije

Migracije se kreiraju putem *Artisana* naredbom „*php artisan make:migration ime_migracije*“, koje se kreiraju u mapu „*database/migrations*“. Jedna migracija može predstavljati jednu ili više tablica, ovisno o tome kako se ona napravi. Čitav relacijski model baze podatka mora se kreirati u migracijama te pokrenuti kako bi *Artisan* te tablice napravio u postavljenoj bazi podataka. To se čini naredbom „*php artisan migrate*“. (Database: Migrations, 14.07.2016.)

5.4.2. Modeli

MVC arhitektura, na kojoj se bazira *Laravel* i na temelju kojeg radi, bazira se na modelima-kontrolerima-pogledima. Modeli u *Laravelu* služe kao objekti u koje će biti „spremani“ podaci, najviše prilikom dobivanja upita iz baze podataka putem prethodno spomenutog *Eloquenta*, ali sadrže i poslovnu logiku nad modelima.

Potrebno je kreirati modele za svaki entitet u bazi podataka. Tako će tablica „*USERS*“ imati model „*User*“, „*KEYWORDS*“ će imati „*Keyword*“, „*KEYWORD_MONITOR*“ će imati „*KeywordMonitor*“ itd. Ime svakog modela po standardu piše se u jednini. Unutar svakog modela potrebno je definirati ime tablice (*$table*), primarni ključ (*$primaryKey*) te varijable koje se mogu spremati u bazu podataka (*$fillable*). Tako će recimo *Keyword* model imati varijablu „*table*“ postavljenu na „*KEYWORDS*“, varijablu „*primaryKey*“ na „*ID*“ te „*fillable*“ varijablu na niz s vrijednosti „*NAME*“.

Modeli se kreiraju također putem *Artisana*: „*php artisan make:model ImeModela*“.

Kreirani modeli:

- Keyword.php
- KeywordMonitor.php
- KeywordsKeymonitors.php
- KeywordsRSSmonitors.php
- KeywordsWebmonitors.php

- RSSExtraction.php
- RSSMonitor.php
- User.php
- WebsiteExtraction.php
- WebsiteExtractionType.php
- WebsiteMonitor.php

5.4.3. Kontroleri

Kontroleri omogućuju povezivanje modela s pogledima, tj. preko njih se određuje izvođenje određene poslovne logike na temelju ulaznih podataka. *Laravel* razlikuje nekoliko vrsta ulaznih podataka, koje su ujedno i imena metoda u svakom kontroleru, na temelju kojih će se odrađivati određena logika: (HTTP Controllers, 14.07.2016.)

1. **Index –** koristi se za pregled određenog pogleda, bez ulaznih podataka
2. **Create** – koristi se za pogled koji sadrži formu za unos podataka
3. **Store –** koristi se za pokretanje logike dodavanja novog unosa u bazu podataka
4. **Edit** – koristi se za pogled koji sadrži formu s popunjenim, već unesenim podacima
5. **Update** – koristi se za pokretanje logike spremanja uređenih u bazu podataka
6. **Destroy** – koristi se za pokretanje logike brisanja podataka

Stvoreni kontroleri za *Notis* (svaki od njih organiziran je u svoje mape):

1. **Auth\AuthController.php** – za potrebe prijave i registracije
2. **Dashboard\DashboardController.php** – za potrebe upravljanjem korisničkih podataka
3. **Keywords\KeywordsController.php** – za potrebe upravljanja ključnim riječima
4. Monitors\Keywords\KeywordMonitorController.php – za potrebe upravljanja monitorima za praćenje ključnih riječi
5. **RSS\RSSMonitorController.php** – za potrebe upravljanja praćenja RSS kanala

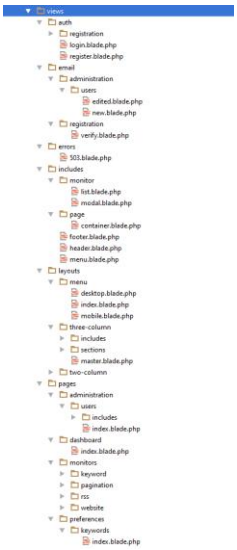
6. **Websites\WebsiteMonitorController.php** – za potrebe upravljanja praćenja monitora internetskih stranica
7. **Users\UsersController.php** – za potrebe upravljanja korisnicima

Svi kontroleri su povezani putem datoteke „*routes.php*“, u kojoj se definira strutkura URL-a putem kojeg se dolazi do određenog kontrolera i njegove metode. Tako npr. „*Route::get('/preferences/keywords', ['as' => 'KeywordsIndex', 'uses' => 'Keywords\KeywordsController@index']);*“ određuje da kada korisnik posjeti „*notis.dev/preferences/keywords*“ otvoriti će mu se pogled koji je određen u metodi „*index*“ u datoteci „*KeywordsController.php*“.

5.4.4. Pogledi

Pogledi, (engl. *views*), koriste se kako bi na određeni način prikazali korisniku podatke iz baze podataka ili mu omogućili da radi određene radnje nad sustavom. *Laravel* za poglede koristi svoj sustav za izradu predložaka (engl. *template*) zvan „*Blade*“. Koristeći Blade, može se kreirati modularan i iskoristi kod za izradu pogleda putem nasljeđivanja ili proširivanja postojećeg HTML i PHP koda. Svaki pogled rađen je prema smjernicama za izradu pogleda u *Laravelu*.

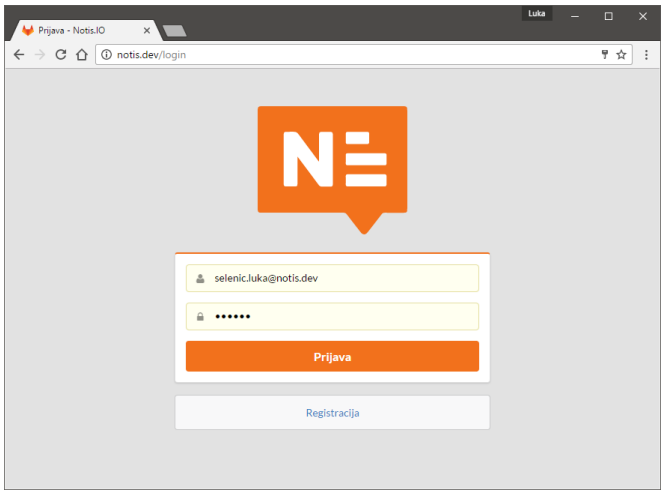
Slika 46. – Struktura pogleda u sustavu Notis



(Izvor: Autor)

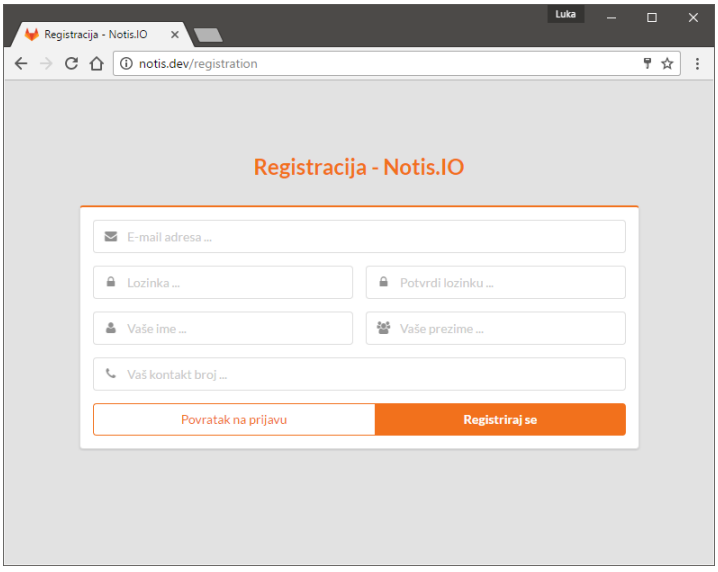
5.5. Izgled administracijskog sučelja

Slika 47. – Prijava u sustav



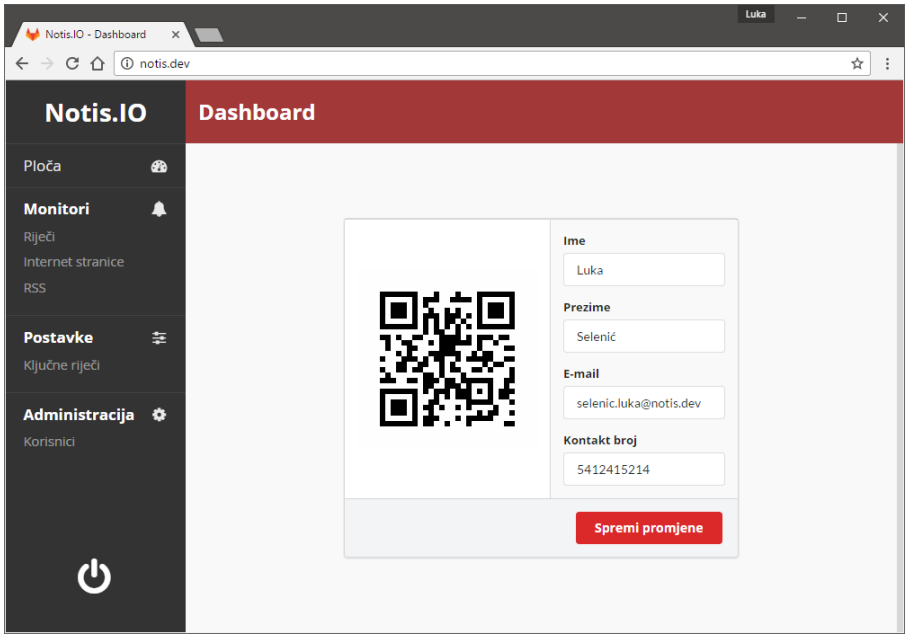
(Izvor: Autor)

Slika 47. – Prijava u sustav



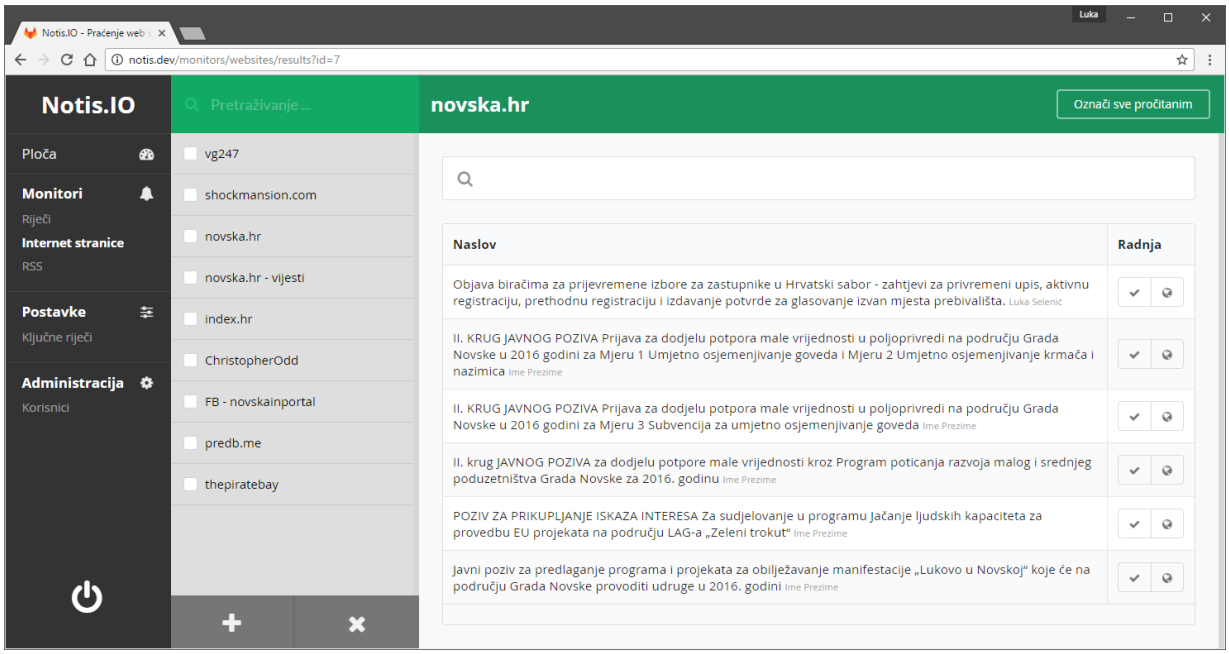
(Izvor: Autor)

Slika 47. – Pogled na upravljanje korisničkim podacima



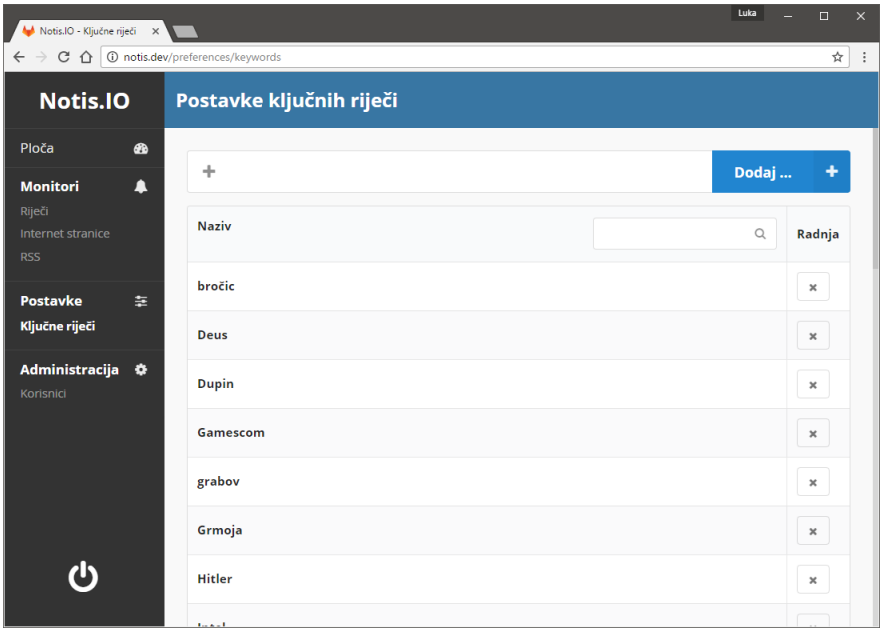
(Izvor: Autor)

Slika 47. – Pogled na pregled rezultata monitora i njihovo upravljanje



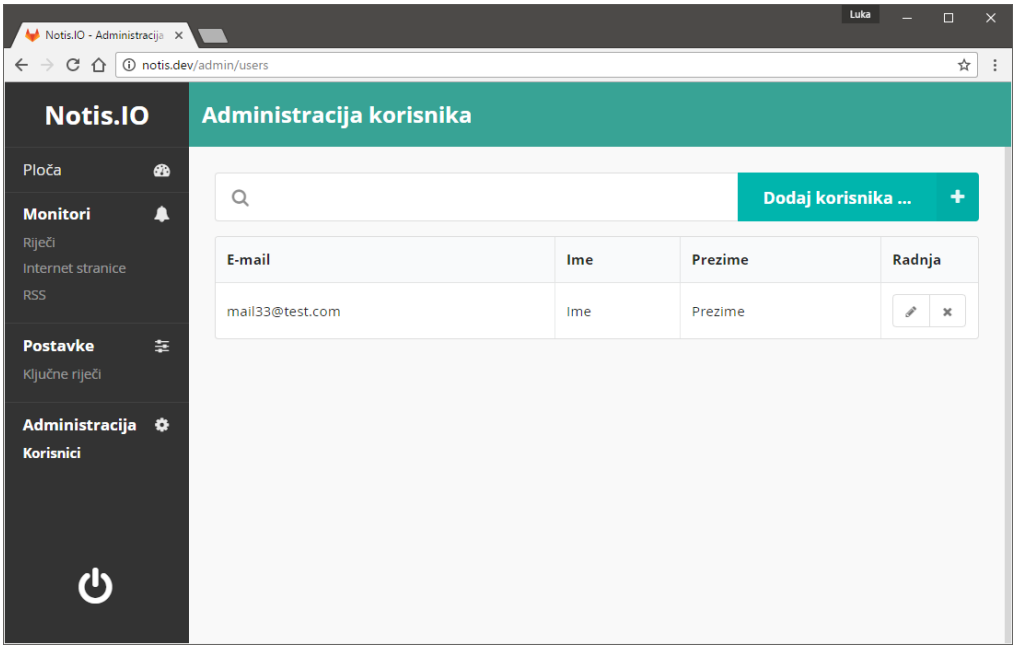
(Izvor: Autor)

Slika 48. – Pogled na postavke ključnih riječi



(Izvor: Autor)

Slika 49. – Pogled na pogled za administraciju korisnika



(Izvor: Autor)

5.6. Notifikacijski server

Notifikacijski server pogonjen je putem *Node.js* – sustav koji omogućava pokretanje *Javascript* aplikacija na serveru, odnosno izradu samostalni *Javascript* aplikacija bez potrebe za njihovim pokretanjem u nekom internet pregledniku (npr. *Google Chrome*). Pošto sustav koristi *WebSockets* tehnologiju za distribuciju informacija, uz njega korišten je i *Socket.IO* paket. Radi se o *frameworku* koji nudi jednostavne metode za izradu *WebSockets* kanala i servera.

Postavljanje je vrlo brzo i jednostavno. Definira se da server radi na portu 7777. Svatko tko posjeti taj server predstavlja jednu konekciju i on se stavlja u niz za konekcije „*connections*“. Putem Socket.IO definira se da server prati sve konekcije te kada primi poruku „*send alert*“ (hrv. šalji upozorenje), što znači da je od aplikacije za ekstrakciju i provjeru ključnih riječi primio tu poruku skupa s nekom obavijesti te da svim konekcijama (korisnicima) mora proslijediti tu obavijest. Na taj način se postiže obavještavanje svih spojenih korisnika na ovaj server u realnom vremenu. (Socket.IO - How to use, 25.07.2016.)

Slika 50. – Kod notifikacijskog servera

```
var express = require('express');
var app = express();
var server = require('http').createServer(app);
var io = require('socket.io').listen(server);

users=[];
connections=[];

server.listen(7777);

io.sockets.on('connection', function(socket) {

  connections.push(socket);

  // Odlazak sa servera
  socket.on('disconnect', function(data) {
    connections.splice(connections.indexOf(socket), 1);
  });

  // Šalji poruku
  socket.on('send alert', function(data) {
    io.sockets.emit('new alert', JSON.parse(data));
  });
```

(Izvor: Autor)

5.7. Aplikacija za ekstrakciju i provjeru ključnih riječi

Ova aplikacija ima za cilj intervalno provjeravati definirane internetske stranice za ključne riječi te izvlačiti podatke iz njih na temelju određenih pravila. Prema konceptu, aplikacija će u određenom intervalu pokretati provjeru za tri vrste monitora: za ključne riječi, internetske stranice te RSS kanale. To bi značilo da mora postojati tri različita algoritma koji će to obavljati.

Prvo, definirane su klase za svaki algoritam - *KeywordMonitorRunner*, *RSSMonitorRunner* te *WebMonitorRunner*. Glavna metoda mora pokretati svaku od njih u zasebnim intervalima. To je postignuto kombinacijom *ScheduledExecutorService*, *Executor* i *Runnable* klasa. *ScheduledExecutorService* ima mogućnost pokretanja svake *Runnable* klase putem *Executora* u intervalu kojeg mu se definira. Svaki algoritam pokretati će se u intervalu od 10 minuta, s razmakom od jedne minute između svakog algoritma. (Jenkov, 2014)

Slika 51. – Kod glavne metode za aplikaciju za ekstrakciju i provjeru ključnih riječi

```
public class Main
{
    public static void main(String args[]) throws Exception
    {
        ScheduledExecutorService webExecutor = Executors.newSingleThreadScheduledExecutor();
        ScheduledExecutorService keywordExecutor = Executors.newSingleThreadScheduledExecutor();
        ScheduledExecutorService rssExecutor = Executors.newSingleThreadScheduledExecutor();

        Runnable periodicWebMonitorCheck = new WebMonitorRunner();
        Runnable periodicKeywordMonitorCheck = new KeywordMonitorRunner();
        Runnable periodicRSSMonitorCheck = new RSSMonitorRunner();

        webExecutor.scheduleAtFixedRate(periodicWebMonitorCheck, 0, 10, TimeUnit.MINUTES);
        keywordExecutor.scheduleAtFixedRate(periodicKeywordMonitorCheck, 1, 10, TimeUnit.MINUTES);
        rssExecutor.scheduleAtFixedRate(periodicRSSMonitorCheck, 2, 10, TimeUnit.MINUTES);
    }
}
```

(Izvor: Autor)

Kod svakog algoritma bitno je omogućiti da se provjere vrše paralelno jedna s drugom, što znači da se mora koristiti višedretveni rad. Algoritam mora preuzeti sve korisnički izrađene monitore te na temelju njih mora odrađivati svoju svrhu. Za lakše upravljanje podacima, odnosno

za što jednostavniji rad s objektima, napravio se model za svaki entitet iz baze koji se koristi u algoritmima:

- ExtractionType.java
- FoundKeyword.java
- KeyMonitorReport.java
- Keyword.java
- KeywordMonitor.java
- KeywordMonitorsBlacklist.java
- Monitor.java
- RSSExtraction.java
- RSSMonitor.java
- RSSMonitorReport.java
- UserAgent.java
- WebMonitorReport.java
- WebsiteExtraction.java
- WebsiteMonitor.java
- WebsiteMonitorsBlacklist.java

Potreban je i pristup bazi pa se za objektno-relacijsko mapiranje koristiti „*DBUtils*“ od *Apache-a*, a sve metode pristupa i rada nad bazom podataka izrađene su i DAO klase za svaki algoritam: *KeywordsDAO, KeywordMonitorDAO, RSSMonitorDAO, WebsiteMonitorDAO*.

5.7.1. KeywordMonitorRunner

Koristi se za provjeru ključnih riječi na definiranim internetskim stranicama. Cilj je u korisniku čitljivom tekstu pronaći riječ koja odgovara korisnički definiranim ključnim riječima. Ukoliko je pronađena, korisnik bi trebao dobiti obavijest o tome.

Može postojati jako velik broj monitora, a potrebno analizirati što veći broj njih. To se postiglo koristeći *ExecutorService, Executors, Callable* i *Future* klase. Svaki korisnički definirani monitor iz baze podataka biti će tretiran kao „zadatak“, proslijeđen klasi *TaskAsCallable*, koji

sadrži algoritam za pretragu te je taj monitor dodan na zasebnu listu „*tasks*“ (*TaskAsCallable task = new TaskAsCallable(monitor);*) Slijedi potrebno pokretane i instanciranje što više paralelnih dretvi koje će odrađivati svoj posao, svaka za svoj monitor. Kod “*ExecutorService workers = Executors.newFixedThreadPool(50)*“ definira da će biti pokretano do 50 paralelnih zadataka, dok će ostatak biti pokretan kako se koji od ovih prethodnih završi.

Slika 52. – Kod za pokretanje višedretvene provjere ključnih riječi

```
KeyMonitorReport report = new KeyMonitorReport();
for (Callable<Monitor> task : tasks) {
    Future<Monitor> future = workers.submit(task);
    report.getMonitors().add(future.get());
}
workers.shutdown();
```

(Izvor: Autor)

Algoritam u *TaskAsCallable* funkcionira tako da pročita koji je URL definiran u za trenutni monitor koji obrađuje. Pošto se radi o provjerama HTML koda, kako bi to bilo što jednostavnije, koristi se „*Jsoup“* (todo) dodatak. Radi se o *frameworku* koji ima mogućnost čitanja HTML koda, pretvaranju HTML koda i njegovih elemenata u *Javi* čitljive objekte. *Jsoup* nudi povezivanje na URL, preuzimanje i čitanje koda sa tog sjedišta i njegovo procesiranje, tako da se kroz čitav kod može prolaziti preko programskog koda. (Lokesh, 2015)

Jednom kada je HTML kod sjedišta preuzet, tada ide čišćenje svih razmaka i novih redaka, iz baze se preuzimaju tekstovi koje je korisnik prethodno označio kao pročitane (*KeywordMonitorsBlacklist*) tako da bude siguran kako će dobivati samo nove, dosad neobjavljene podatke te se ti tekstovi uklanjaju iz koda.

Slijedi jednostavna pretraga svih korisnikovih ključnih riječi unutar preuzetog teksta. Kako bi se pružili točniji pronalasci, bitno je da su ključne riječi i sav tekst sa stranica prebačeni u mala slova. Ukoliko je pronađena ključna riječ, ona se dodaje u posebnu klasu „*Monitor“*, koju vraća svaka dretva u glavnu dretvu.

Kada se sve radnje dovrše, aplikacija provjerava postoji li barem jedna pronađena ključna riječ pa ako je, spajati će se na notifikacijski server putem „*Socket.IO“* dodatka za *Javu*. Kako bi se notifikacijskom serveru poslalo više pronađenih ključnih riječi, tj. sve pronađene ključne riječi skupa s monitorom u kojem je pronađena, one trebaju biti konvertirane u JSON. Za to se koristi „*GSON*“ dodatak za Javu, koji pretvara čitav popis *Monitor* klasa (koje u sebi sadrže ključne riječi) te ih pretvara u JSON strukturu. Tako će obavijest biti čitljiva i u mobilnoj aplikaciji, jer se u njoj koristi *Javascript*, koji nativno može čitati JSON.

5.7.2. WebMonitorRunner

Koristi se za izvlačenje podataka s korisnički definiranih internetskih sjedišta i provjeru ključnih riječi u rezultatima izvlačenja. Cilj je da putem CSS ili *XPath* upita aplikacija pronađe i izdvoji korisniku relevantan tekst te da odmah među tim tekstom pronađe ključne riječi.

Proces se od *KeywordMonitorRunner* razlikuje samo po algoritmu, dok je pozivanje algoritama potpuno isto.

Postupak izvlačenja radi se putem CSS upita (npr. „*.imePrveKlase div .imeDrugeKlase*“) pronašao bi sve elemente koji sadrže klasu „*imeDrugeKlase*“, a nalaze se unutar elementa „*div*“ koji se nalazi unutar elementa koji sadrži klasu „*imePrveKlase*“. CSS upit korisnik može dodati prilikom stvaranja monitora ili ga može pronaći putem posebnog algoritma koji će biti objašnjen dalje u radu.

Algoritam iz baze preuzima tekstove koji su pročitani od strane korisnika kako bi ih izdvojio iz pretrage koja slijedi, tako da korisnik dobiva samo najnovije sadržaje. Putem *Jsoup* apliakcija se spaja na URL koji je korisnik definirao i preuzima HTML kod. Kad je Jsoup pretvorio HTML kod u sebi čitljiv kod, tada on može izvesti CSS upit nad tim kodom pomoću metode „*select*“, koja će vratiti objekte vrste „*Element*“, koji u sebi sadrže nekakav tekst. Pošto mogu biti razne strukture HTML-a, a korisniku bi koristila i poveznica (ako je ima) na sadržaj koji je pronađen, algoritam će polaziti od elementa kojeg je dobio putem CSS upita prema roditeljskim elementima te provjeravati jesu li oni „a“ element, odnosno element koji sadrži poveznicu na sadržaj. Stati će onda kada pronađe taj element. Tada će izdvojiti njegovu

poveznicu i spremiti je u već spomenuti *Monitor* element kojeg vraća svaka dretva u glavnu dretvu.

Svaki novi pronađeni sadržaj aplikacija će spremati u bazu podataka za prikaz u mobilnoj i administracijskoj aplikaciji.

Slijedi standardna provjera je li pronađen novi sadržaj pa ako je aplikacija kreira popis pronađenih *Monitor* objekata te ih pretvara u JSON i šalje obavijest svim korisnicima putem notifikacijskog servera.

5.7.3. RSSMonitorRunner

Ostaje algoritam za čitanje RSS kanala. Cilj je pročitati korisnički definiran RSS kanal te sav sadržaj omogućiti korisniku na pregled.

Kao *KeywordMonitorRunner* i *WebMonitorRunner* proces se jedino razlikuje po algoritmu.

Pošto se radi o RSS tehnologiji, koristi se „*Rome“* dodatak za *Javu*. Radi se o *frameworku* koji nudi set alata za rad s RSS kanalima. Omogućava spajanje na njih i čitanje njihovog sadržaja.

Kao i prethodni algoritmi, ovaj isto preuzima već pročitane naslove od strane korisnika te će ih izdvojiti iz konačnog rezultata pretraživanja ukoliko ih pronađe.

Slika 53. – Kod za čitanje sadržaja RSS kanala

```
List<RSSExtraction> potentialNewExtractions = new ArrayList<~>();

reader = new XmlReader(url);
SyndFeed feed = new SyndFeedInput().build(reader);

potentialExtractionsLoop:
for (Iterator i = feed.getEntries().iterator(); i.hasNext();) {
    SyndEntry entry = (SyndEntry) i.next();

    RSSExtraction rssExtraction = new RSSExtraction();

    for (String alreadyFoundTitle : alreadyFoundTitles) {
        if (entry.getTitle().equals(alreadyFoundTitle)) {
            continue potentialExtractionsLoop;
        }
    }

    rssExtraction.setTITLE(entry.getTitle());

    if (entry.getDescription().getValue().length() > 0) {
        rssExtraction.setDESCRIPTION(entry.getDescription().getValue());
        rssExtraction.setDESCRIPTION(Jsoup.parse(rssExtraction.getDESCRIPTION()).text());
    }
    else
    {
        rssExtraction.setDESCRIPTION(null);
    }

    rssExtraction.setURL(entry.getLink());

    potentialNewExtractions.add(rssExtraction);
}
```

(Izvor: Autor)

5.8. Pomoćni server

Pomoćni server služi kako bi omogućio iskorištavanje prethodno napravljenih algoritama u aplikaciji za ekstrakciju i provjeru ključnih riječi od strane vanjskih aplikacija, npr. mobilne aplikacije. Radi se o *servlet* aplikaciji pisane u *Java* programskom jeziku.

Putem *IntelliJ IDEA* alata napravljen je novi projekt za izradu internetskih aplikacija, koji je odmah kreirao sve potrebno za pokretanje i rad. Za pokretanje ove aplikacije koristi se *Tomcat* od Apache-a.

Iako je proširiv, i može se omogućiti za razne aktivnosti, trenutno služi za dvije svrhe: algoritam za generiranje CSS upita i omogućavanje korisniku mobilne aplikacije označavanja pročitanih ključnih riječi.

Svi upiti koji dolaze na ovaj server provjeravaju ima li zahtjev u sebi i API ključ korisnika preko kojeg se dokazuje pristup iz aplikacije bez da se korisnik prijavi u sustav.

Algoritam za generiranje CSS upita napravljen je radi *Jsoup-a*, koji se također koristi u aplikaciji za ekstrakciju i provjeru ključnih riječi. Da bi se ubrzao razvoj samog sustava, iskorišteni su dijelovi koda te aplikacije u svrhu kreiranja ovog algoritma. On se bazira na tome da *Jsoup* od određenog internetskog sjedišta preuzme HTML kod te unutar njega traži element koji sadrži tekst jednak onome koji je korisnik unio kao primjer sadržaja kojeg želi izdvojiti. Jednom kada pronađe taj tekst, algoritam će prepoznati u kojem se HTML elementu nalazi te će njega koristiti kao polazište u generiranju CSS upita. Algoritam tada putuje na roditelj element od polazišnog elementa, uzima njegovo ime i klase te ih dodaje u posebnu varijablu, pa onda odlazi na njegov roditelj element. To se ponavlja skroz dok algoritam ne dođe do „*body*“ elementa u HTML kodu, čime završava generiranje CSS upita. Dobivena varijabla, odnosno CSS upit, uvijek će izdvajati sve sadržaje koji su relevantni onom primjeru kojeg je korisnik unio i na taj način će sustav izdvajati željene sadržaje za korisnika. Dodatne pogodnosti nosi bolje performanse prilikom izvođenja generiranja koda od npr. PHP programskog jezika.

Iskorištavanje postojećeg koda razlog i zašto se ovaj server koristi za označavanje pročitanih ključnih riječi od strane korisnika mobilne aplikacije.

Bitno je napomenuti da je radi međusobne komunikacije između ovog servera i klijentskih aplikacija potrebno omogućiti CORS – međusobno dijeljenje resursa. (CORS Filter, 05.08.2016.)

Slika 54. – Kod za omogućavanje CORS-a

```
public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain chain) throws IOException, ServletException {
    HttpServletRequest request = (HttpServletRequest) servletRequest;
    HttpServletResponse resp = (HttpServletResponse) servletResponse;
    resp.addHeader("Access-Control-Allow-Origin","*");
    resp.addHeader("Access-Control-Allow-Methods","GET,POST");
    resp.addHeader("Access-Control-Allow-Headers","Origin, X-Requested-With, Content-Type, Accept");


    if ( request.getMethod().equals("OPTIONS") ) {
        resp.setStatus(HttpServletResponse.SC_OK);
        return;
    }
    chain.doFilter(request, servletResponse);
}
```

(Izvor: Autor)

5.9. Mobilna aplikacija

Posljednji dio sustava je mobilna aplikacija, koju koriste korisnici kako bi dobivali obavijesti o pronađenim ključnim riječima, ali i kako bi pregledavali pronađene rezultate praćenja sadržaja. Korisnik pregledava rezultate za monitore internetskih stranica i RSS kanala.

Mobilna aplikacija izrađena je putem *Ionic Frameworka*, verzija 1.3, aplikacija koja omogućava razvoj mobilnih aplikacija za *Android, iOS* i *Windows Phone* operativne sustave koristeći *Javascript*, HTML i CSS. Ova aplikacija bazira se na Angular.JS *frameworku,* koji je inače poznat alat za kreiranje jednostraničnih internet sjedišta i aplikacija.

Kada se instalira na razvojnu okolinu *Ionic* ima mogućnost kreiranja predloška iz kojeg kreće razvoj mobilne aplikacije putem naredbenog retka. Putem naredbenog retka se odvija i proces izgradnje aplikacije u iskoristivu aplikaciju. Tijekom razvoja, moguće je putem naredbenog retka kreirati lokalni server na kojem će u realnom vremenu biti prikazana potpuno funkcionalna aplikacija. Nativne mogućnosti mobilnih uređaja (kamera, senzori, komunikacija s operativnim sustavom itd.) omogućena je putem *ngCordova*, koji se instalira u *Ionic* također putem naredbenog retka.

Pošto korisnik mora potvrditi da je dio sustava, mora skenirati QR kod u administracijskoj aplikaciji. *ngCordova* sadrži veliku bazu dodataka, tako sadrži i „*BarcodeScanner*“ dodatak koji ima mogućnost skeniranja i prepoznavanja raznih grafičkih kodova putem instalirane kamere na mobilnom uređaju. Nakon skeniranja koda, on se sprema u memoriju aplikacije za što je zadužena HTML5 tehnologija „*local storage*“. (Raboy, 2014)

Aplikacija mora biti pokrenuta u svakom trenutku, čak i kad nije u fokusu, a to se omogućilo korištenjem *Cordova* dodatka „*BackgroundMode*“.

Aplikacija mora omogućiti nativne notifikacije na mobilnom uređaju, što se omogućilo korištenjem *Cordova* dodatka „*CordovaLocalNotification*“.

Struktura ove mobilne aplikacije je vrlo slična *Laravel* aplikaciji. Radi se o MVC arhitekturi, tako da i ona sadrži modele, kontrolere i poglede. U ovom slučaju modeli se ne koriste, zamjenjuju ih tzv. „*servisi*“, a koriste se i kontroleri i pogledi.

5.9.1. Servisi

Servisi u ovoj aplikaciji služe kao objekti koji u sebi sadrže metode rada nad podacima relevantnim tom servisu. Oni imaju ulogu modela. Na primjer, svaki monitor ima svoj servis, koji će kao objekt pohranjivati varijable, imena ili identifikacijskog broja iz baze podataka te „*get*“ i „*set*“ metode za svaki od varijabli. (Services, 15.08.2016.)

Kreirani servisi:

1. **Monitor** – spremanje vrijednosti za monitore i njihovo korištenje u pogledima
2. **WebmonitorNotification –** spremanje vrijednosti za notifikacije sustava za pretraživanje i praćenje internetskih stranica i njihovo korištenje u pogledima
3. **RssmonitorNotification -** spremanje vrijednosti za notifikacije sustava za pretraživanje i praćenje RSS kanala i njihovo korištenje u pogledima
4. **NotificationDetails –** spremanje vrijednosti primljene notifikacije za njegov prikaz na pogledu
5. **SocketService –** spajanje na notifikacijski server

5.9.2. Kontroleri

Kontroleri u ovoj aplikaciji skuže kao kanali putem kojih će pogledi dobivati mogućnost korištenja poslovne logike koji se nalaze u servisima, tj. povezuje servise s pogledima.

Svaki kontroler povezan je s određenim pogledom putem tzv. „ruta“. Rute definiraju koje će se pogledi prikazati korisniku kada se pritisne nekakva tipka (najčešće u navigaciji koju je kreirao sam Ionic prilikom kreiranja projekta) te definiranje koji će pogledi koristiti određeni kontroler. (Understanding Controllers, 15.08.2016.)

Slika 55. – Primjer koda postavljenih ruta u mobilnoj aplikaciji



```
.state('izbornik.landing', {
  url: '/landing',
  cache: false,
  views: {
    'side-menu21': {
      templateUrl: 'templates/page1.html',
      controller: 'landingController'
    }
  }
})

.state('izbornik.internetStranice', {
  url: '/websites',
  cache: false,
  views: {
    'side-menu21': {
      templateUrl: 'templates/internetStranice.html',
      controller: 'internetStraniceCtrl'
    }
  }
})

.state('izbornik.internetStranica', {
  url: '/websites/view/:id',
  cache: false,
  views: {
    'side-menu21': {
      templateUrl: 'templates/internetStranica.html',
      controller: 'internetStranicaCtrl'
    }
  }
})
```

(Izvor: Autor)

1. **MainController** – prvi kontroler koji se pokrene prilikom pokretanja aplikacije. U njemu se postavljaju globalne metode koje koristi čitava aplikacija poput otvaranja poveznica u internet pregledniku mobilnog uređaja, postavke notifikacija i pozadinskog rada aplikacije te spajanja na notifikacijski server.
2. **landingController** – nema funkciju, no može se koristiti za postavljanje određenih metoda ili servisa koji bi se koristili u polaznom pogledu (prvom koji se pokaže) kada se pokrene aplikacija
3. **internetStraniceCtrl** – definira preuzimanje svih monitora za izvlačenje podataka s internetskih sjedišta putem API-a administracijske aplikacije te njihovo postavljanje u odgovarajući servis

4. **internetStranicaCtrl** – definira preuzimanje rezultata iz određenog monitora za izvlačenje podataka s internetskih sjedišta putem API-a administracijske aplikacije te njihovo postavljanje u odgovarajući servis
5. **rSSCtrl** – definira preuzimanje svih monitora za izvlačenje podataka s RSS kanala putem API-a administracijske aplikacije te njihovo postavljanje u odgovarajući servis
6. **rSSViewCtrl** – definira preuzimanje rezultata iz određenog monitora za izvlačenje podataka s RSS kanala putem API-a administracijske aplikacije te njihovo postavljanje u odgovarajući servis
7. **notificationCtrl** – agregira sve dobivene notifikacije (ključne riječi, internteske stranice i RSS kanali) na jedan pogled radi njihovog prikaza u pogledu
8. **keymonitorNotificationViewCtrl** – sadrži metodu koja omogućava korisniku da označi obavijest o ključnoj riječi pročitanom koristeći pomoćni server
9. **webmonitorNotificationViewCtrl** – sadrži metodu koja omogućava korisniku da označi obavijest o pronađenim ključnim riječima monitora internetskih stranica koristeći administracijsko sučelje
10. **rssmonitorNotificationViewCtrl** - sadrži metodu koja omogućava korisniku da označi obavijest o pronađenim ključnim riječima monitora RSS kanala koristeći administracijsko sučelje

5.9.3. Nox App Player

Kako bi se aplikacija prilikom razvoja i testirala na što jednostavniji i brži način, potrebno je bilo koristiti emulator operativnog sustava mobilnog uređaja. Za testiranje na *Android* operativnim sustavima, sustav za kojeg je najviše i namijenjena ova aplikacija, koristila se aplikacija „*Nox App Player*“. Radi se o posebno dizajniranom i prepravljenom emulatoru *Android* operativnih sustava (verzija 4.4.2 „*Kit Kat*“).

Jednom kada se aplikacija želi testirati, naredbom „*ionic build android*“ u naredbenom retku na razvojnoj okolini pokreće se proces izrade aplikacije za *Android* operativni sustav.

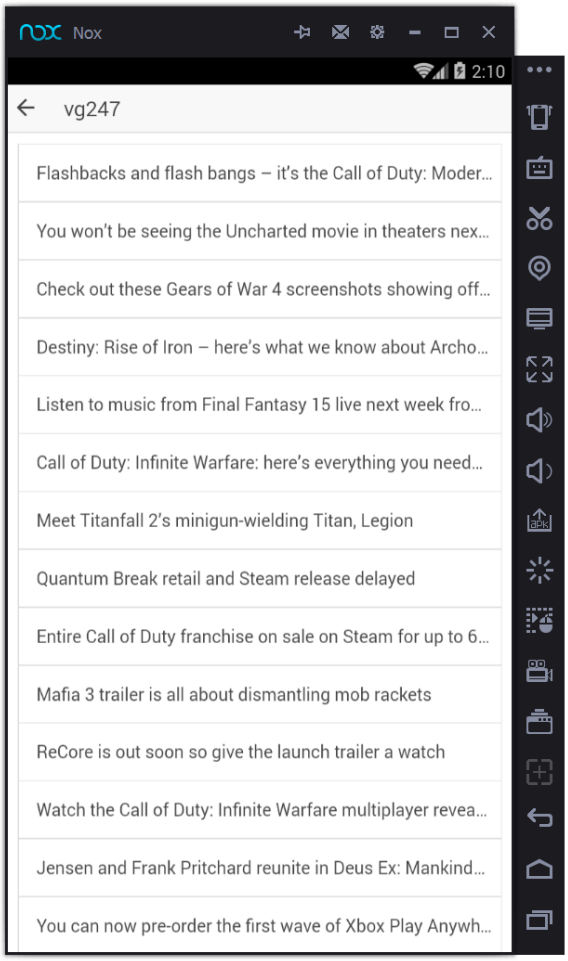
Rezultat je datoteka s ekstenzijom „apk“, koja se može učitati putem *Nox App Player-a*, čime će se pokrenuti instalacija mobilne aplikacije. Jednom instalirana, može se pokrenuti.

Slika 56. – Početni ekran nakon pokretanja aplikacije



(Izvor: Autor)

Slika 57. – Početni ekran nakon pokretanja aplikacije



(Izvor: Autor)

6. ZAKLJUČAK

Ovim istraživačkim radom i realizacijom idejnog sustava *Notis* dokazana je radna hipoteza da se može izgraditi sustav koji će s korisnički definiranih internetskih sjedišta moći izdvajati željene informacije, prezentirati ih korisniku te ga obavijestiti ukoliko se te informacije poklapaju s korisnički definiranim postavkama. Također, *Notis* je sustav, odnosno aplikacija koja sada pokriva nedostatke aplikacija koje su bile predmet istraživanja. Notis ima mogućnosti intervalnog pregleda podataka, svi pronađeni podaci su agregirani na jedno mjesto, korisnik ima mogućnosti dodavati vlastite izvore podataka kao i ključnih riječi, ima mogućnost pretraživanja ključnih riječi. Omogućeno je izvlačenje informacija s posebnih internet sjedišta, sustav podržava tehnologije za distribuciju informacija, šalje svim korisnicima sustava obavijesti te je omogućeno korištenje sustava za više korisnika u isto vrijeme.

Tijekom istraživanja najviše poteškoća bilo je kod ograničenja aplikacija koje su bile predmetima istraživanja, no uz pomoć opisa funkcionalnosti tih aplikacija i njihovih sustava za pomoć korisnicima doznalo se podržava li pojedina aplikacija pojedinu, traženu funkcionalnost.

Tijekom izrade idejnog sustava najzahtjevnije je bilo pronaći način na koji izraditi višedretveni sustav koji će što brže pretraživati internetske stranice u potrazi za ključnim riječima i izvlačenjem podataka. Programski jezik Java srećom svojim performansama i mogućnostima to pruža na vrlo jednostavan i funkcionalan način.

7. POPIS LITERATURE

1. About Node.js, https://nodejs.org/en/about/ (7.7.2016.)
2. Artisan Console, https://laravel.com/docs/5.2/artisan (14.7.2016.)
3. CORS Filter, https://github.com/eBay/cors-filter (5.8.2016.)
4. Database: Migrations, https://laravel.com/docs/5.2/migrations (14.7.2016.)
5. Eloquent: Getting Started, https://laravel.com/docs/5.2/eloquent (7.7.2016.)
6. HTTP Controllers, https://laravel.com/docs/5.2/controllers (14.7.2016.)
7. Internet most popular information source: poll (2009), http://www.reuters.com/article/us-media-internet-life-idUSTRE55G4XA20090618 (6.7.2016.)
8. Ionic Documentation Overview, http://ionicframework.com/docs/overview/ (7.7.2016.)
9. Jenkov, J. (2014), ScheduledExecutorService, http://tutorials.jenkov.com/java-util-concurrent/scheduledexecutorservice.html (29.7.2016.)
10. Laravel Homestead, https://laravel.com/docs/5.2/homestead (7.7.2016.)
11. Laravel PHP Framework, https://github.com/laravel/laravel (7.7.2016.)
12. Lokesh, G. (2015), Jsoup Tutorial and Examples, http://howtodoinjava.com/jsoup/complete-jsoup-tutorial/ (3.8.2016.)
13. Raboy, N. (2014), https://www.thepolyglotdeveloper.com/2014/09/implement-barcode-scanner-using-ionic-framework/ (11.8.2016.)
14. Services, https://docs.angularjs.org/guide/services (15. 8. 2016.)
15. Social Media Monitoring, https://www.techopedia.com/definition/29592/social-media-monitoring (6.7.2016.)
16. Socket.IO - How to use, http://socket.io/docs/ (25.7.2016.)
17. Understanding Controllers, https://docs.angularjs.org/guide/controller (15.8.2016.)
18. Views, https://laravel.com/docs/5.2/views (19.7.2016.)
19. Web Scraping, https://www.techopedia.com/definition/5212/web-scraping (06.7.2016.)

8. POPIS KORIŠTENIH KRATICA

PDF - Portable Document Format

RSS - Rich Site Summary

JSON - JavaScript Object Notation

API - Application programming interface

CSV - Comma-separated values

SMS - Short Message Service

MVC - Model–view–controller

IDE - Integrated development environment

RAM - Random-access memory

PHP - Hypertext Preprocessor

SSD - Solid-state disk

IP – Internet protocol

HTML - Hyper Text Markup Language

DAO - Data access object

CSS - Cascading Style Sheets

9. POPIS SLIKA

Slika 1. – Sučelje Mediatoolkit aplikacije 8

Slika 2. – Sučelje za stvaranje upita 9

Slika 3. – Sučelje Import.io aplikacije 12

Slika 4. – Izgled tablice izvučenih podataka u Import.io 14

Slika 5. – Sučelje Mention aplikacije 17

Slika 6. – Konceptualni model idejnog sustava 23

Slika 7. – Use case dijagram idejnog sustava 27

Slika 8. – Relacijski model baze podataka idejnog sustava 46

Slika 8. – Dijagram aktivnosti - Registracija 49

Slika 9. – Dijagram aktivnosti - Prijava 50

Slika 10. – Dijagram aktivnosti – Uređivanje korisničkih podataka 51

Slika 11. – Dijagram aktivnosti – Uređivanje praćenja ključnih riječi 52

Slika 12. – Dijagram aktivnosti – Dodavanje monitora za praćenje ključnih riječi 53

Slika 13. – Dijagram aktivnosti – Uređivanje monitora za praćenje ključnih riječi 54

Slika 14. – Dijagram aktivnosti – Brisanje monitora za praćenje ključnih riječi 55

Slika 15. – Dijagram aktivnosti – Pretraživanje monitora za praćenje ključnih riječi 56

Slika 16. – Dijagram aktivnosti – Upravljanje monitora za praćenje internetskih stranica 57

Slika 17. – Dijagram aktivnosti – Dodavanje monitora za praćenje internetskih stranica 58

Slika 18. – Dijagram aktivnosti – Uređivanje monitora za praćenje internetskih stranica 59

Slika 19. – Dijagram aktivnosti – Brisanje monitora za praćenje internetskih stranica 60

Slika 20. – Dijagram aktivnosti – Pretraživanje monitora za praćenje internetskih stranica 61

Slika 21. – Dijagram aktivnosti – Upravljanje monitora za praćenje RSS kanala 62

Slika 23. – Dijagram aktivnosti – Uređivanje monitora za praćenje RSS kanala 64

Slika 24. – Dijagram aktivnosti – Brisanje monitora za praćenje RSS kanala 65

Slika 25. – Dijagram aktivnosti – Pretraživanje monitora za praćenje RSS kanala 66

Slika 26. – Dijagram aktivnosti – Upravljanje ključnim riječima 67

Slika 27. – Dijagram aktivnosti – Upravljanje rezultatima izvlačenja s internetskih sjedišta 68

Slika 28. – Dijagram aktivnosti – Upravljanje rezultatima izvlačenja s RSS kanala 69

Slika 29. – Dijagram aktivnosti – Upravljanje korisnicima 70

Slika 30. – Dijagram aktivnosti – Prijava korisnika preko mobilne aplikacije 71

Slika 31. – Dijagram aktivnosti – Pregled rezultata izvlačenja s internetskih sjedišta preko mobilne aplikacije 72

Slika 32. – Dijagram aktivnosti – Pregled rezultata izvlačenja s RSS kanala preko mobilne aplikacije 73

Slika 33. – Dijagram aktivnosti – Pregled obavijesti preko mobilne aplikacije 74

Slika 34. – Mockup – Prozor za prijavu 75

Slika 35. – Mockup – Prozor za registraciju 75

Slika 36. – Mockup – Prozor za uređivanje korisničkih podataka 76

Slika 37. – Mockup – Prozor za pregled monitora i rezultata 76

Slika 38. – Mockup – Prozor za dodavanje ili uređivanje monitora 77

Slika 39. – Mockup – Prozor za upravljanje ključnim riječima 77

Slika 40. – Mockup – Prozor za upravljanje korisnicima 78

Slika 41. – Mockup – Prozor za dodavanje ili uređivanje korisnika 78

Slika 42. – Mockup – Mobilna aplikacija - Izbornik mobilne aplikacije 79

Slika 43. – Mockup – Mobilna aplikacija - Pregled rezultata monitora 79

Slika 44. – Mockup – Mobilna aplikacija - Pregled uređivanja rezultata 80

Slika 45. – Logotip sustava Notis 81

Slika 46. – Struktura pogleda u sustavu Notis 88

Slika 47. – Prijava u sustav 89
Slika 47. – Prijava u sustav 89
Slika 47. – Pogled na upravljanje korisničkim podacima 90
Slika 47. – Pogled na pregled rezultata monitora i njihovo upravljanje 90
Slika 48. – Pogled na postavke ključnih riječi 91
Slika 49. – Pogled na pogled za administraciju korisnika 91
Slika 50. – Kod notifikacijskog servera 92
Slika 51. – Kod glavne metode za aplikaciju za ekstrakciju i provjeru ključnih riječi 93
Slika 52. – Kod za pokretanje višedretvene provjere ključnih riječi 95
Slika 53. – Kod za čitanje sadržaja RSS kanala 98
Slika 54. – Kod za omogućavanje CORS-a 99
Slika 55. – Primjer koda postavljenih ruta u mobilnoj aplikaciji 102
Slika 56. – Početni ekran nakon pokretanja aplikacije 104
Slika 57. – Početni ekran nakon pokretanja aplikacije 105

10. POPIS TABLICA

Tablica 1. – Analizirane funkcionalnosti Mediatoolkit aplikacije 10

Tablica 2. – Analizirane funkcionalnosti Import.io aplikacije 15

Tablica 3. – Analizirane funkcionalnosti Mention aplikacije..... 20

Tablica 4. – Rezultati istraživanja funkcionalnosti aplikacija..... 22

Tablica 5. – Opis veza relacijskog modela idejnog rješenja..... 48