

SOLID RADNI OKVIR**SOLID FRAMEWORK**Ivana Marković, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu su predstavljene teorijske osnove Solid radnog okvira. Implementirana je aplikacija za čuvanje bilješki kao studija slučaja. U razvoju aplikacije korišten je React okvir za razvoj korisničkog interfejsa.

Ključne reči: Solid, veb aplikacije, decentralizovani sistem, pod

Abstract – This paper presents the theoretical foundations of the Solid framework. An application for keeping notes is implemented as case study. The React framework was used for user interface development.

Keywords: Solid, web applications, decentralized system, pod

1. UVOD

Ovaj rad za cilj ima predstavljanje Solid (*eng. Social Linked Data*) radnog okvira, odnosno decentralizovanu platformu za veb aplikacije.

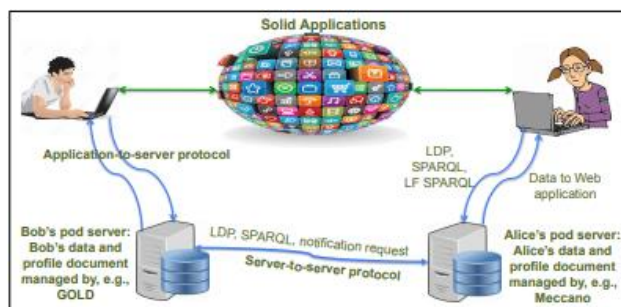
Unutar Solid radnog okvira, korisnikovim podacima se upravlja nezavisno od aplikacije koja ih kreira i koristi. Podaci korisnika se čuvaju u ličnom skladištu podataka dostupnom na vebu (Solid pod). Solid omogućava korisnicima da posjeduju više različitih Solid pod-ova različitih dobavljača pod-ova, dok je istovremeno omogućeno lako prebacivanje sa jednog dobavljača na drugi. Solid protokoli se zasnivaju na postojećim W3C preporukama za čitanje, pisanje i kontrolu pristupa sadržaju na korisnikovom pod-u.

U Solid arhitekturi, aplikacije mogu da pristupaju podacima, koji se nalaze u vlasništvu korisnika ili korisnik ima mogućnost pristupa podacima bez obzira na njihovu lokaciju na vebu. Takođe, korisnici mogu da kontrolišu pristup svojim podacima i imaju mogućnost prebacivanja između aplikacija u bilo kom trenutku. Ovo su neki od najznačajnijih koncepata Solid radnog okvira, čiji će detaljniji opis funkcionisanja biti prikazan u nastavku rada. Na slici 1. je predstavljen uopšten prikaz funkcionisanja Solid radnog okvira.

Prikazana su dva korisnika Solid aplikacija, Alice i Bob, koji koriste različite pod servere. Iako korisnici koriste različite pod servere, oba korisnika mogu koristiti iste aplikacije za pristup i održavanje svojih podataka.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Stevan Gostojić, vanr. prof.



Slika 1. Uopšten prikaz Solid radnog okvira [1]

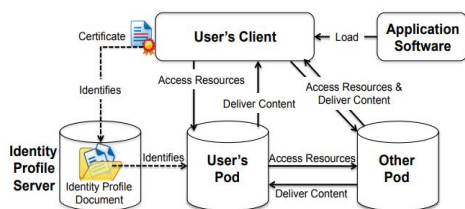
2. SOLID

Solid se zasniva na RDF i tehnologiji semantičkog veba pružajući nezavisnost podataka i omogućavajući jednostavan, ali moćan mehanizam upravljanja podacima. U okviru Solid platforme, svaki korisnik čuva svoje podatke unutar skladišnog prostora dostupnom na mreži, koji se naziva lična mrežna baza podataka (Solid pod). Solid pod-ovi su servisi za skladištenje podataka dostupni na vebu, koji mogu biti dostupni na ličnim serverima samih korisnika ili na javnim serverima dobavljača pod-ova, koji su slični trenutnim dobavljačima skladišta podataka čuvanim na cloud-u (npr. Dropbox).

Korisnici mogu da imaju više od jednog pod-a. Korisnik može da bira između različitih dobavljača pod-ova, pošto Solid aplikacije mogu da rade sa bilo kojim pod serverom, bez obzira na njegovu lokaciju ili usluge dobavljača. Različiti dobavljači pod-ova mogu ponuditi različite stepene privatnosti i pouzdanosti (npr. garancije dostupnosti i kašnjenja) ili pravnu zaštitu (npr. pravni okvir specifičan za zemlju porijekla pod-a).

Solid aplikacije su implementirane kao klijentske veb ili mobilne aplikacije, koje čitaju i zapisuju podatke direktno unutar pod-a. Solid omogućava lak razvoj i korištenje aplikacija, jer su podaci aplikacija uvijek dostupni na vebu, u skladu sa mehanizmom za upravljanje kontrolom pristupa. Aplikacija može objediniti podatke iz različitih izvora na internetu, pristupajući pod-u korisnika koji je pokrenuo aplikaciju i pod-vima koji pripadaju drugim korisnicima. Solid radni okvir omogućava da više aplikacija koriste iste podatke na pod-u. Korisnici mogu birati između različitih aplikacija, koje pružaju sličnu funkcionalnost. Ukoliko se korisnik prebaci na novu aplikaciju, ta aplikacija može pristupiti postojećim podacima korisnika budući da su po dizajnu sistema aplikacije odvojene od podataka koje koriste.

Na slici br.2 prikazana je arhitektura Solid radnog okvira.



Slika 2. Arhitektura Solid radnog okvira [2]

U Solid radnom okviru se podacima upravlja po REST principima, kako je definisano LDP (Linked Data Platform) preporukama [3]. Nove stavke podataka kreiraju se u kontejneru (koji bi se mogli nazvati kolekcijom ili direktorijumom), slanjem podataka na URL kontejnera sa HTTP POST zahtjevom ili izdavanjem HTTP PUT zahtjeva unutar njegovog URL prostora. Pojedinačne stavke podataka se ažuriraju pomoću HTTP PUT i HTTP PATCH zahtjeva, dok se podaci uklanjaju sa korisnikovog pod-a slanjem HTTP DELETE zahtjeva. Uz pomoć HTTP GET zahtjeva se dobija kolekcija podataka, koji se nalaze unutar kontejnera korisnikovog pod-a.

Podaci aplikacije se u Solid-u čuvaju u dokumentima, koji su identifikovani od strane jedinstvenog identifikatora resursa (URI). Solid razlikuje strukturirane podatke predstavljene pomoću okvira za opis resursa (*Resource Description Framework - RDF*) i nestrukturirane podatke bilo koje vrste (npr. video snimci, slike, veb stranice). Ovo omogućuje strukturiranim podacima da budu parsirani i serijalizovani u različitim sintaksama poput Turtle, JSON ili RDFa. RDF slijedi REST principe u kojima resursi imaju pojedinačne URI-je.

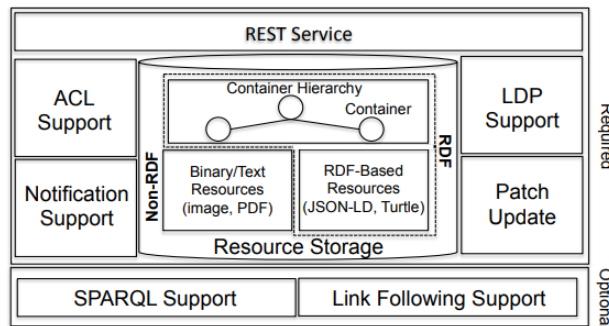
Solid aplikacije ne zahtijevaju potvrdu identiteta kao što to radi većina uobičajenih aplikacija, jer korisnik ne treba da potvrdi identitet kod dobavljača aplikacije. Umjesto toga, Solid aplikacije bi mogle da iznude „lažne“ zahtjeve za potvrdu identiteta kako bi pribavile WebID korisnika iz sertifikata klijenta. Iz tog razloga, autentifikacija se izvodi samo između klijenta (browser-a) i korisnikovog pod-a, gdje su podaci smješteni. Ovakav način autentifikacije uklanja potrebu korisnika da unosi WebID prilikom procesa prijavljivanja.

Da bi Solid funkcionisao na zaista decentralizovan način, neophodan je identitetski prostor u kojem korisnici mogu lako da upravljaju i proširuju sopstveni identitet i akreditive. Postojeći protokoli identifikacije, kao što su OAuth [4] i OpenID Connect [5] nude određeni stepen decentralizacije, ali se ipak ne uklapaju u strukturu podataka profila zasnovanih na RDF-u, što otežava proširivanje informacija o profilu dodatnim korisničkim atributima. Solid koristi WebID pri implementaciji globalne arhitekture upravljanja identitetom, zasnovanom na konceptu decentralizovanog dobavljača identiteta, što u kombinaciji sa WebID-TLS [6] omogućava jednokratnu prijavu na vebu.

Protokol WebID-TLS je decentralizovani protokol za potvrdu identiteta, koji omogućava sigurnu i efikasnu potvrdu identiteta na vebu. Protokol omogućava korisnicima da se autentifikuju na bilo kom sajtu, jednostavnim odabirom jednog od predloženih korisnikovih sertifikata. Sertifikate može kreirati bilo koja veb stranica za svoje korisnike. Za razliku od klasične autentifikacije sertifikata korisnika,

koja se oslanja na Public Key Infrastructure (PKI), WebID-TLS ne zahtijeva potpisivanje sertifikata od strane pouzdanog autoriteta za izdavanje sertifikata. Razlog tome je što WebID-TLS koristi korisnikov sertifikat kao sredstvo za obavljanje autentifikacije javnim ključem.

Na slici 3 predstavljena je arhitektura pod-a. U okviru pod-a je moguće čuvati RDF i ne-RDF resurse. Pod serveri podržavaju LDP, patch resursa, kontrolu pristupa, live ažuriranje i opciono SPARQL.



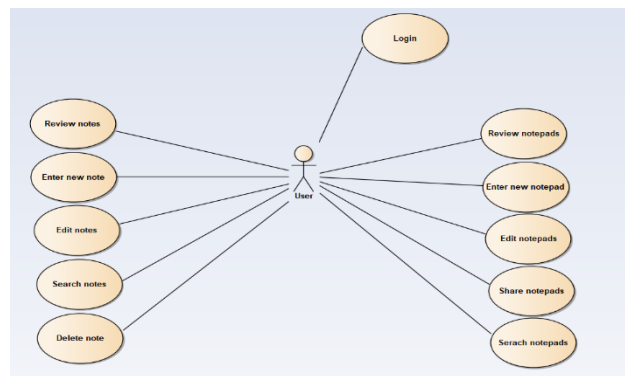
Slika 3. Arhitektura pod-a [2]

3. SPECIFIKACIJA

U ovom odeljku detaljno je opisana specifikacija aplikacije za čuvanje bilješki, koja je u ovom radu korištena kao studija slučaja. Predstavljene su funkcionalni zahtjevi koje je bilo neophodno implementirati.

3.1. Funkcionalni zahtjevi

U ovom odjeljku su opisane funkcionalnosti koje je neophodno podržati u razvoju aplikacije. Na slici 4 su predstavljene funkcionalni zahtjevi UML dijagramom slučajeva korištenja.

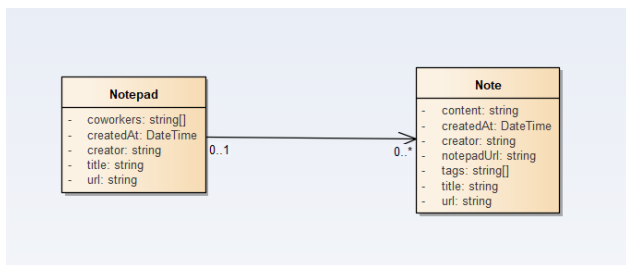


Slika 4. UML dijagram slučajeva korištenja Notepad aplikacije

Jedan od osnovnih zahtjeva, koje je neophodno podržati jeste prijava na sistem. U okviru ove aplikacije, prijava je nešto drugačija u odnosu na standardnu prijavu veb aplikacija. Prijava obuhvata unos WebID-ija korisnika, odnosno URL-a na kome se mogu pronaći informacije o korisniku i kako bi sama aplikacija znala odakle da čita i piše podatke. Neophodno je omogućiti korisniku prikaz svih bilježnica i njihovih bilješki, dodavanje novih, kao i editovanje i brisanje postojećih bilješki i bilježnica. Takođe, korisnik ima mogućnost da svoju bilježnicu sa drugim korisnicima, odnosno da omogući drugim korisnicima pristup njegovom pod-u i mogućnost da drugi korisnici izvršavaju izmjene nad datoj bilježnici.

3.2. Model podataka

U okviru aplikacije se razlikuju dva tipa entiteta, jedan je bilježnica, a drugi je bilješka. Na slici 5 je predstavljen model podataka.



Slika 5. Model podataka

4. IMPLEMENTACIJA

U ovom odeljku je prikazan detaljan opis implementacije Notepad aplikacije, čija je specifikacija prikazana u prethodnom poglavlju. Prilikom implementacije aplikacije za čuvanje bilješki, nazvane Notepad, korišten je React okvir za realizaciju korisničkog interfejsa aplikacije.

Pri realizaciji konekcije ka Solid-u, korištena je biblioteka „solid/react“ iz koje je uključena *LoginButton* klasa koja definiše izgled pop-up prozora u kojem je neophodno unijeti korisnikov WebID, radi dalje autentifikacije i prijave na sistem. Nakon unosa WebID-ija korisnik je u obavezi da unese kredencijale definisane prikom kreiranja pod-a, kako bi sistem dozvolio interakciju sa korisnikovim pod-om. Nakon unosa korisničkog imena i lozinke, neophodno je izvršiti autentifikaciju korisnika. Za autentifikaciju korisnika u ovom projektu je korišćena „solid-auth-client“ biblioteka.

Prilikom realizacije postupka prijave na sistem se može uočiti koncept decentralizacije Solid radnog okvira. U toku implementacije Notepad aplikacije, nije bilo potreba za razvojem sopstvenog mehanizma autentifikacije korisnika, nego su iskorištene dostupne Solid biblioteke za tu svrhu.

Ukoliko korisnik prvi put pokreće aplikaciju, nakon uspješne autentifikacije i prijave na sistem, neophodno je da dozvoli aplikaciji da pristupa njegovom pod-u. Na taj način je obezbjeđeno da korisnik ima potpunu kontrolu nad svojim pod-om, imajući mogućnost da definiše koje akcije aplikacija može izvršavati nad njegovim pod-om.

U okviru ovog projekta je definisana klasa *SolidBackend*, čija je osnovna namjena pristup i izvršavanje svih akcija nad korisnikovim pod-om. Biblioteka „rdflib“ je iskorištena za rad sa podacima čuvanim u okviru Solid okvira. Biblioteka „rdflib“ predstavlja opšti skup alata za upravljanje podacima u RDF formatu.

Omogućava skladištenje podataka, njihovo parsiranje i serijalizaciju u različite formate, ali takođe može da prati i promjene podataka koje dolaze iz aplikacije ili sa servera. Postoji nekoliko klasa „rdflib“ biblioteke, koje su korišćeni u okviru *SolidBackend* klase, i njihovo značenje je navedeno u nastavku:

- *Store* – predstavlja strukturu podataka za čuvanje podataka u obliku grafa i omogućava izvršavanje upita nad podacima

- *Fetcher* – pomoćna klasa koji se koristi za povezivanje na veb, učitavanje podataka kao i njihovo čuvanje nazad
- *UpdateManager* – veoma moćna i korisna klasa koja omogućava slanje malih promjena serveru kako bi se podaci ispravili dok korisnik mijenja podatke u realnom vremenu i
- *Graph* – predstavlja bazu podataka za semantički veb

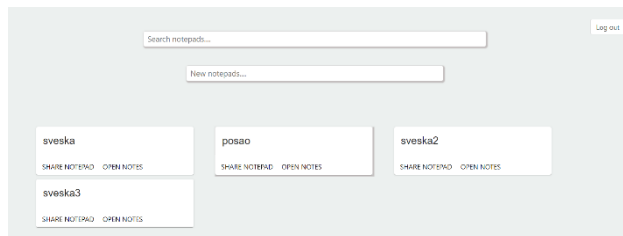
Prilikom implementacije funkcionalnosti dijeljenja bilježnica između korisnika neophodno je vršiti i izmjene nad .acl fajlovima bilježnica. U svrhu pristupa .acl fajlovima i njihovom editovanju iskorištena je biblioteka „solid-acl-utils“, to je JavaScript biblioteka za rad sa .acl fajlovima, koja omogućava dodavanje, mijenjanje i uklanjanje dozvola za pristup datotekama i pod-ovima korisnika.

5. DEMONSTRACIJA

U ovom odjeljku je predstavljen rad sa Notepad aplikacijom, s tim da nije stavljen akcenat na načinu implementacije funkcionalnih zahtijeva, nego na prikazu njihove realizacije, kao i na prikazu promjena nad korisnikovim pod-om.

Prilikom demonstracije rada aplikacije su korištena dva Solid pod naloga, jedan sa WebID-ijem <https://testic.inrupt.net/>, a drugi sa <https://ivanam96.inrupt.net/> WebID-ijem. Unutar oba profila korisnika je definisana *friends* kategorija, u čijem sadržaju se nalazi WebID drugog naloga.

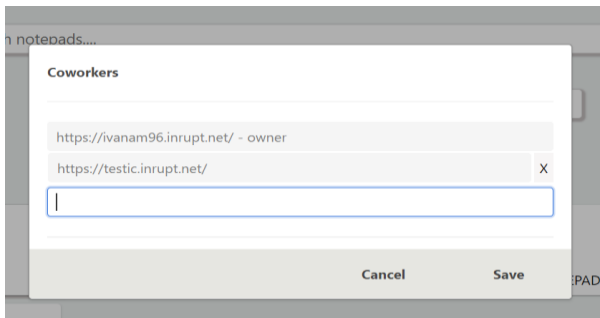
Demonstracija rada aplikacije počinje logovanjem korisnika WebID-ija <https://testic.inrupt.net/>. Na slici 6 je prikazan ekran korisnika nakon logovanja.



Slika 6. Prikaz ekrana nakon logovanja

Sve bilježnice i bilješke korisnika se čuvaju u *public/solidapp/* folderu, u odvojenim folderima za bilježnice notepads, a bilješke u folderu notes.

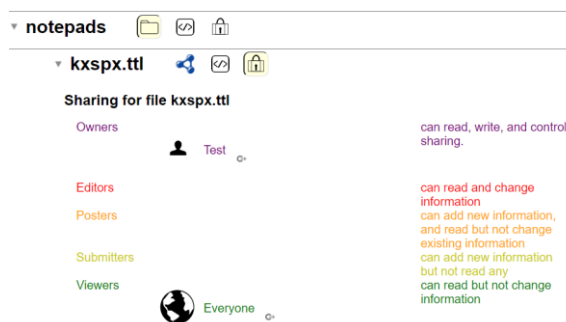
Korisnik u svakom trenutku može provjeriti i pregledati podatke sačuvane na svom pod-u, logovanjem na svoj profil unutar pod dobavljača (u ovom slučaju *inrupt.net*). Proširivanjem notepads i notes foldera su prikazani .ttl fajlovi sa nasumično kreiranim nazivima, koji u sebi sadrže informacije o svim bilježnicama i bilješkama korisnika. U toku demonstracije rada aplikacije je izvršeno dodavanje nove bilježnice nad kojom su kasnije izvršene određene izmjene. Korisniku je omogućeno editovanje svih polja unutar bilježnice, međutim kao posebna funkcionalnost izdvojeno je dijeljenje bilježnica između različitih korisnika. Na slici 7 je predstavljen pop-up prozor unutar kog se nalaze elementi korisničkog interfejsa za datu funkcionalnost.



Slika 7. Prikaz prozora za editovanje liste saradnika

Korisnik ima mogućnost dodavanja WebID-ija novog saradnika, kao i brisanje WebID saradnika iz liste klikom na dugme X, koje se nalazi u istom redu kao i dati WebID saradnika. Nakon unosa željenog WebID-ija novog saradnika i klika na taster *Enter*, WebID saradnika se dodaje u listu saradnika i prikaz se osvježava sa novom vrijednošću liste. Međutim, kako bi izmjene nad datom listom bile sačuvane i projektovane nad korisnikovom pod-u, neophodno je da korisnik klikne na dugme *Save*. Ukoliko korisnik želi odustati od unijetih izmjena ili jednostavno želi izaći iz datog prozora dovoljan je klik na dugme *Cancel*.

Takođe, je prikazan i rad sa `.acl` fajlovima bilježnica. Dati fajlovi definišu pravo pristupa podacima svakog pojedinačnog resursa na korisnikovom pod-u. Na slici 8 je dat prikazan izgled `.acl` fajla bilježnice.



Slika 8. Prikaz `.acl` fajla bilježnice

6. ZAKLJUČAK

U ovom radu su predstavljene teorijske osnove Solid radnog okvira, decentralizovane platforme za razvoj veb aplikacija. Pored teorijskog opisa, predstavljen je način implementacije Notepod aplikacije za čuvanje bilješki. Ova aplikacije je poslužila kao studija slučaja pri analizi Solid radnog okvira. Cilj ovog rada je predstavljanje osnovnih teorijskih aspekata Solid-a, kao i njihova kasnija primjena.

Notepod aplikacija predstavlja jednostavnu aplikaciju za unos bilješki čiji model podataka je prost. Ali bez obzira na to, kroz implementaciju date aplikacije je uspješno implementirano dosta bitnih mogućnosti Solid radnog okvira. Obezbijedena je decentralizovana autentifikacija korisnika aplikacije, kao jednog od osnovnih načela Solid-a.

Jedan od najbitnijih funkcionalnih zahtjeva Notepod aplikacije jeste mogućnost dijeljenja bilježnica između korisnika. Kroz implementaciju ovog funkcionalnog zahtjeva prikazan je i objašnjen način funkcionisanja Solid-a prilikom definisanja prava pristupa kako korisnikovom pod-u, tako i pojedinačnim fajlovima sačuvanim u korisničkom skladištu.

Na internetu kakav je danas, korisnik otprema svoje podatke svakom servisu koji koristi. Svi podaci koje korisnik postavi na svoj *Facebook* nalog se čuvaju u jednom silosu, svi podaci koje postavi na *Twitter* se čuvaju u drugom silosu itd.. Koncept Solid-a je drugačiji. Umjesto otpremanja podataka na udaljene servere, nego korisnik dozvoljava pristup podacima koji se nalaze unutar jednog od korisnikovih pod-ova servisu sa kojim želi ostvariti interakciju. Sveobuhvatni cilj je eliminisanje uticaja velikih korporacija koje imaju nad korisničkim podacima i negativnih posljedica koje iz toga proizilaze i da se korisnicima pruži konačna odlika o tome ko pristupa njihovim podacima. Međutim, u praksi postoji nekoliko prepreka koje se moraju prevazići da bi se to postiglo i one zahtijevaju velike strukturne promjene u načinu svakodnevnog korištenja interneta.

7. LITERATURA

- [1] E. Mansour, A. Vlad Samba, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulnaga and T. Berners-Lee, „A demonstration of the Solid platform for Social Web applications“ <http://emansour.com/publications/paper/www16-solid-essam.pdf>. Novembar, 2020.
- [2] E. Mansour, A. Vlad Samba, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulnaga and T. Berners-Lee, „A Platform for Decentralized Social Applications Based on Linked Data“, http://emansour.com/research/lusail/solid_protocols.pdf. Novembar, 2020.
- [3] A. Malhotra, J. Arwe, and S. Speicher. „Linked Data Platform Specification. W3C Recommendation“, <http://www.w3.org/TR/ldp/>. Novembar, 2020.
- [4] D. Hardt. The OAuth 2.0 authorization framework.
- [5] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore. Openid connect core 1.0. The OpenID Foundation.
- [6] T. Inkster, H. Story, and B. Harbulot., WebID-TLS Specification. „<http://www.w3.org/2005/Incubator/webid/spec/tls/>“. Novembar, 2020.

Kratka biografija:



Ivana Marković je rođena 02.08.1996. godine u Bijeljini. Školske 2015/2016. godine upisuje Fakultet tehničkih nauka u Novom Sadu, smjer „Primenjeno softversko inženjerstvo“. Diplomirala 2019. godine. Upisuje master studije na istom fakultetu, smjer „Računarstvo I automatika“. kontakt: markovic.ivana996@gmail.com