

Završno izvješće

Projektni zadatak: Android aplikacija za OpenStack

Sažetak

Rad prati izradu mobilne aplikacije koja omogućuje korisniku pregled i upravljenje instancama servera na OpenStacku.

Student: Tina Škrtić
tskrtic@riteh.hr

Sadržaj

Sadržaj.....	2
Uvod.....	3
Motivacija.....	3
Kratki opis funkcionalnosti.....	3
Primjer primjene.....	4
Radni okvir upravljačke aplikacije.....	4
OpenStackClient.....	4
Android sučelje	5
Tijek komunikacije između komponenti aplikacije	7
Primjeri komunikacije s OpenStack API servisima	11
Primjer autentikacije.....	11
Primjer zahtjeva za listom projekata	12
Primjer autentikacije s autorizacijom za odabrani projekt	12
Primjer dohvaćanja liste instanci za odabrani projekt.....	14
Primjer pauziranja instance	15
Moguća proširenja aplikacije	16
Usvojena znanja i vještine.....	16
Implementacija	17
Rasprava.....	22
Zaključak	23
Kratki pregled literature.....	23
Literatura	24

Uvod

Računarstvo u oblaku donosi brojne prednosti, kako administratorima i programerima, tako i krajnjim korisnicima usluga u oblaku. Poboljšava agilnost organizacija povećavanjem fleksibilnosti dodavanja ili proširivanja tehnološke infrastrukture. Smanjuju se troškovi omogućavanjem plaćanja samo za usluge i resurse koji se koriste, za razliku od kupovanja hardvera ili iznajmljivanja virtualnog stroja koji se ne koristi cijelo vrijeme. Održavanje je lakše zato što se aplikacijama može pristupiti s raznih lokacija. Povećana je produktivnost jer veliki broj korisnika može istovremeno raditi na istim dokumentima, za razliku od, na primjer, slanja e-mailom. Skalabilnost je omogućena u gotovo stvarnom vremenu putem dinamičkog rezerviranja resursa.

OpenStack je skup softverskih alata za izradu i upravljanje platformama računarstva u oblaku, kao i kolekcija softverskih projekata koji nude *Infrastructure as a Service* (IAAS) rješenje kroz skup međusobno povezanih servisa. Rješenje poput kreiranja virtualnih servera u oblaku je lako izvedivo korištenjem OpenStack okruženja, a postojeće web sučelje omogućuje jednostavnu administraciju resursa.

Dostupnost mobilnih mreža u gotovo svakoj urbanoj lokaciji i šire, te rastuća upotreba mobilnih uređaja povlači potrebu za mobilnim aplikacijama koje bi prednosti korištenja usluga u oblaku proširile s desktop računala i web preglednika na aplikacije namijenjene mobilnim uređajima.

Motivacija

S obzirom na užurbani tempo života, bilo bi dobro korisniku omogućiti pristup virtualnim resursima putem android mobilnog uređaja, kako bi u bilo kojoj situaciji mogao lako provjeriti stanje svojih virtualnih servera, te na njima izvršiti potrebne administrativne akcije.

Trenutni nedostatak aplikacija sličnih funkcija motivirao me je, temeljem ovog rada, na izradu Android aplikacije za Openstack, koja je jednostavna za korištenje i omogućava korisniku nekoliko osnovnih akcija na virtualnim serverima.

Ciljani korisnici su svi oni koji žele vidjeti stanje svojih virtualnih servera na mobitelu i imati mogućnost izvršavanja osnovnih administrativnih akcija na serveru.

Kratki opis funkcionalnosti

Aplikacija omogućuje korisniku pregled i upravljanje instancama servera na OpenStacku.

Osnovne funkcionalnosti:

- Login
- Pregled liste servera
- Filtriranje liste servera
- Pretraživanje liste servera
- Pregled podataka o serveru
- Operacije na serveru: pauza, start, stop, reboot

Aplikacija nakon pokretanja prvo prikazuje Login ekran, te lokalno pohranjuje vjerodajnice korisnika kako bi se mogle ponovo iskoristiti kad istekne token.

Korištenjem Identity API-ja, dobije se token koji se koristi u budućim zahtjevima. Aplikacija koristi Compute API za izvršenje akcija.

Aplikacija je napisana u Java jeziku i namijenjena Android uređajima. Svaka od opcija prikazana je na ekranu kao gumb koji korisnik može dodirnuti za izvršenje akcije. Nakon izvršenja akcije, aplikacija će prikazati ekran s ispisanim rezultatom.

Primjer primjene

Korisnik je dobio svoj account na OpenStack okruženju i kreirao je nekoliko instanci servera. Korisnik nije uvijek u prilici sjesti za stolno računalo kako bi provjerio stanje svojih instanci. Mobilna aplikacija omogućuje korisniku pregled i osnovne upravljačke akcije na instancama s njegovog mobilnog uređaja.

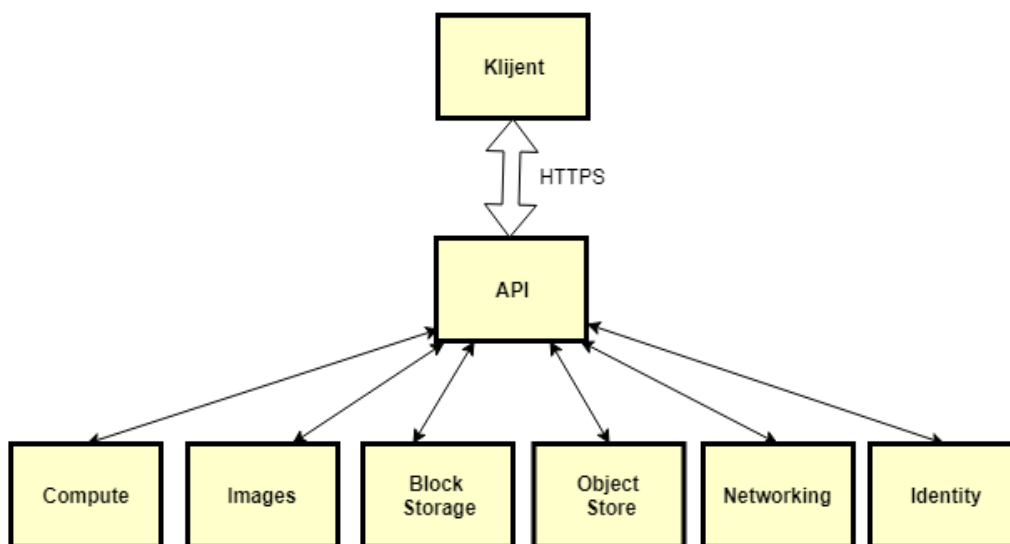
Korisnik se logira u aplikaciju koristeći svoje korisničko ime i lozinku, kao i na web sučelju. Nakon toga, ako ima više od jednog projekta, prikaže se izbornik s projektima. Korisnik može izabrati bilo koji od projekata. Sljedeći ekran prikazuje listu instanci servera koje pripadaju odabranom projektu. Ako ima veliki broj instanci, postoji mogućnost filtriranja prikazanih instanci prema unesenom tekstu. Korisnik dodirom instance može vidjeti dijagnostičke informacije o instanci, te se prikazuje novi izbornik sa akcijama kao što su pauziranje, zaustavljanje, reboot ili pokretanje instance.

Radni okvir upravljačke aplikacije

Aplikacija se sastoji od dvije glavne komponente: klijenta koji komunicira s OpenStack API-jem i Android sučelja. Klijent, kojeg smo nazvali OpenStackClient, je dodan u Android projekt kao vanjska biblioteka. Metode OpenStackClient biblioteke ne ovise direktno o implementaciji sučelja iz kojeg se pozivaju, te ih je moguće iskoristiti i u drugačijoj aplikaciji, na primjer, desktop aplikaciji slične primjene.

OpenStackClient

OpenStack nudi pristup servisima za upravljanje resursima putem web API-ja. Aplikacija koja konzumira OpenStack RESTful servise mora kreirati URL adrese, te, ovisno o zahtjevu, zadati vrijednosti zaglavlja HTTP zahtjeva i kreirati tijelo zahtjeva. Nakon toga, ista klijentska aplikacija treba pročitati odgovor i pretvoriti tijelo odgovora u objekte ili variable razumljive ostatku aplikacije. Slika 1 prikazuje model komunikacije klijentske aplikacije i OpenStack servisa.



SLIKA 1: OPENSTACK API

Aplikacija koristi Identity API za autentikaciju korisnika i Compute API za prikaz i upravljanje instancama. Pristup RESTful API servisima ostvaren je korištenjem Apache HttpClient paketa u Java jeziku.

Identity API metode koje aplikacija koristi:

POST /v3/auth/token – za autentikaciju korisnika

GET /v3/auth/projects – za dohvaćanje liste projekata

Compute API metode koje aplikacija koristi:

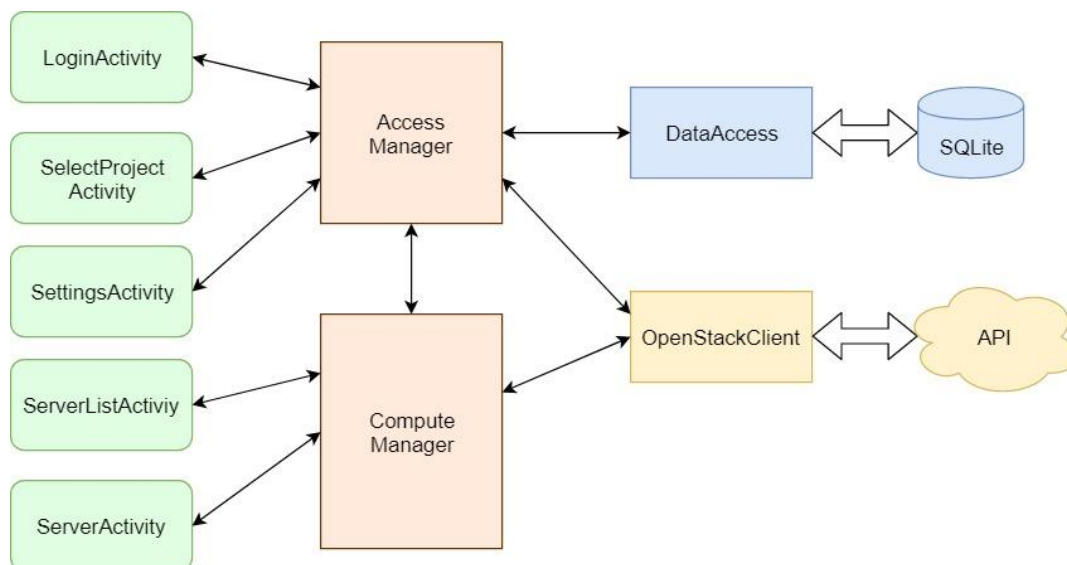
GET /servers – za dohvaćanje liste servera

GET /servers/{serverId} – za pregled podataka o serveru

POST /servers/{serverId}/action – u tijelu zahtjeva zadana je jedna od akcija, na primjer: pause, reboot, start, stop, createImage

Android sučelje

Android aplikacija koristi Activity klase za prikaz sučelja pojedinog ekrana. Kako Activity klase ne bi morale biti zadužene za zadatke osim prikaza sadržaja, klase ComputeManager i AccessManager podatke dobivene iz OpenStackClienta ili baze podataka pretvara u oblik koji odgovara prikazu u aktivnosti. Prilikom komunikacije s OpenStackClient klasom, AccessManager će, ako je potrebno osvježiti token, dohvatiti vjerodajnice i poslati ih OpenStackClientu u odgovarajućem formatu. Operacije dohvaćanja informacija se izvršavaju na pozadinskom procesu, tako da je sučelje aplikacije responzivno dok traje izvršavanje zahtjeva. Skica arhitekture Android aplikacije prikazana je na slici 2.



SLIKA 2: MODEL APLIKACIJE

Aplikacija koristi internu SQLite bazu podataka za pohranu korisnikovih vjerodajnica, te ostalih postavki vezanih za aplikaciju, kao što je URL OpenStack servera. Metode koje upravljaju spremanjem i dohvaćanjem podataka iz baze su u posebnoj klasi, nazvanoj DataAccess.

Izrada Android aplikacije

Aplikacija će sadržavati sljedeće aktivnosti:

- LoginActivity - autentikacija korisnika. Ako u bazi već postoji korisničko ime i lozinka, polja za unos će biti ispunjena postojećim podacima i, ukoliko ih ne želi zamijeniti, korisnik može samo pritisnuti Login gumb za nastavak.
- SelectProjectActivity - izbor projekta ako postoji više od jednog.
- ServerListActivity - izbornik koji sadrži listu servera.
- ServerActivity - izbornik koji se pojavi nakon odabira servera i prikazuje podržane akcije na serveru.
- SettingsActivity - na svakom ekranu postoji "Settings" gumb, koji omogućuje mijenjanje postavki aplikacije, kao što je URL Openstack servera, i sadrži opciju za odjavljivanje korisnika.

Izgled (*layout*) aktivnosti je definiran pomoću Androidovog XML riječnika, koji se sastoji od niza ugnježđenih elemenata, slično kao i HTML web stranice.

Aplikacija će korištenjem *toast* obavijesti prikazati status odgovora na pojedini zahtjev, na primjer, 200 OK ako je akcija bila uspješna. U slučaju greške u autentikaciji, aplikacija će se, osim prikazivanja obavijesti, automatski prebaciti na Login aktivnost. Ako se dogodi bilo koja druga greška, bilo u komunikaciji s API-jem, bazom, ili u nekoj drugoj komponenti, informacija će biti prikazana korisniku, ali aplikacija treba moći nastaviti s radom.

Za izradu aplikacije koristi se Android Studio. Aplikacija će se tijekom razvoja testirati na emulatoru, a zatim na stvarnom Android pametnom telefonu. Za vrijeme razvoja aplikacije koristi se DevStack okruženje na lokalnoj mreži, što omogućuje provjeru uspješne komunikacije OpenStackClient komponente i OpenStack API-ja.

Kako je aplikacija podijeljena na module, moguće je izradu svakog modula tretirati kao poseban zadatak. OpenStackClient se može napisati neovisno od ostataka aplikacije. Za izradu AccessManager klase nije potrebno imati gotovo korisničko sučelje, a DataAccess klasa, koja pristupa bazi, može se privremeno zamijeniti klasom sa statičnim varijablama ili implementacijom koja čuva podatke u konfiguracijskoj datoteci. Podjela aplikacije na više modula također olakšava testiranje pojedinih komponenti putem unit test metoda. Na primjer, komponentu kao što je DataAccess ili ComputeManager lako je zamijeniti "mock" klasom, što omogućuje izradu unit testova za metode pojedinih Activity klasa.

Tijek komunikacije između komponenti aplikacije

Komunikacija se odvija između tri glavne komponente: Android sučelja i OpenStackClienta na Android uređaju, te OpenStack API-ja na udaljenom serveru dostupnom putem lokalne mreže ili Interneta. Jednostavni model razmjene poruka u aplikaciji prikazan je na slici 3.

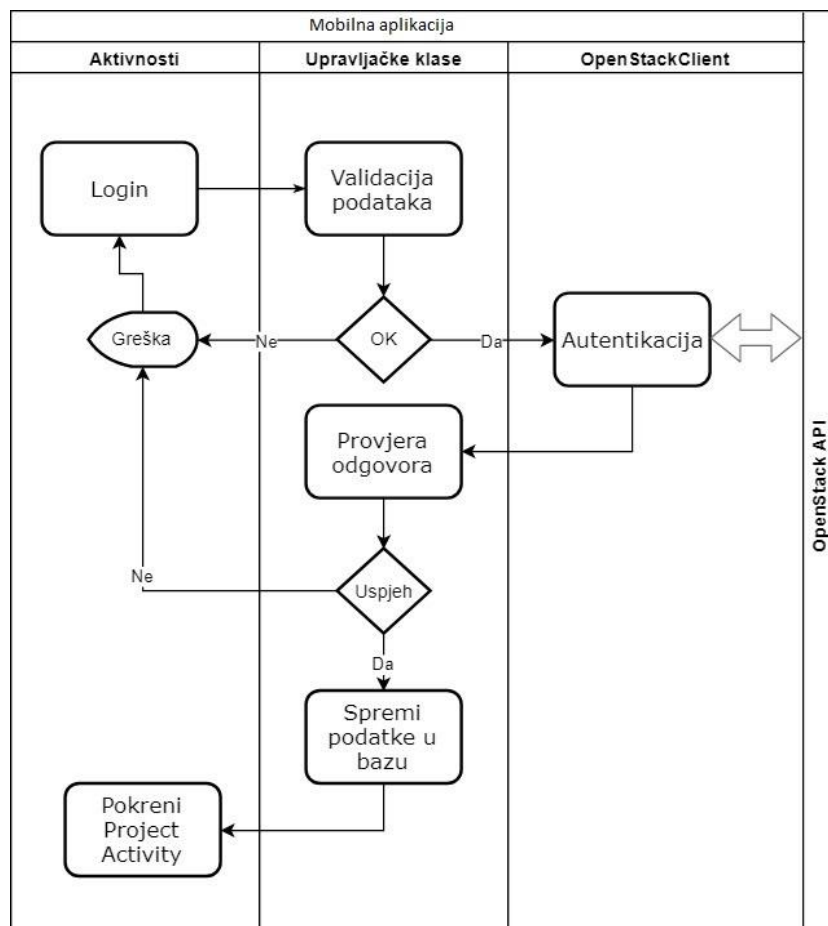


SLIKA 3: MODEL OSNOVNIH KOMPONENTI U KOMUNIKACIJI

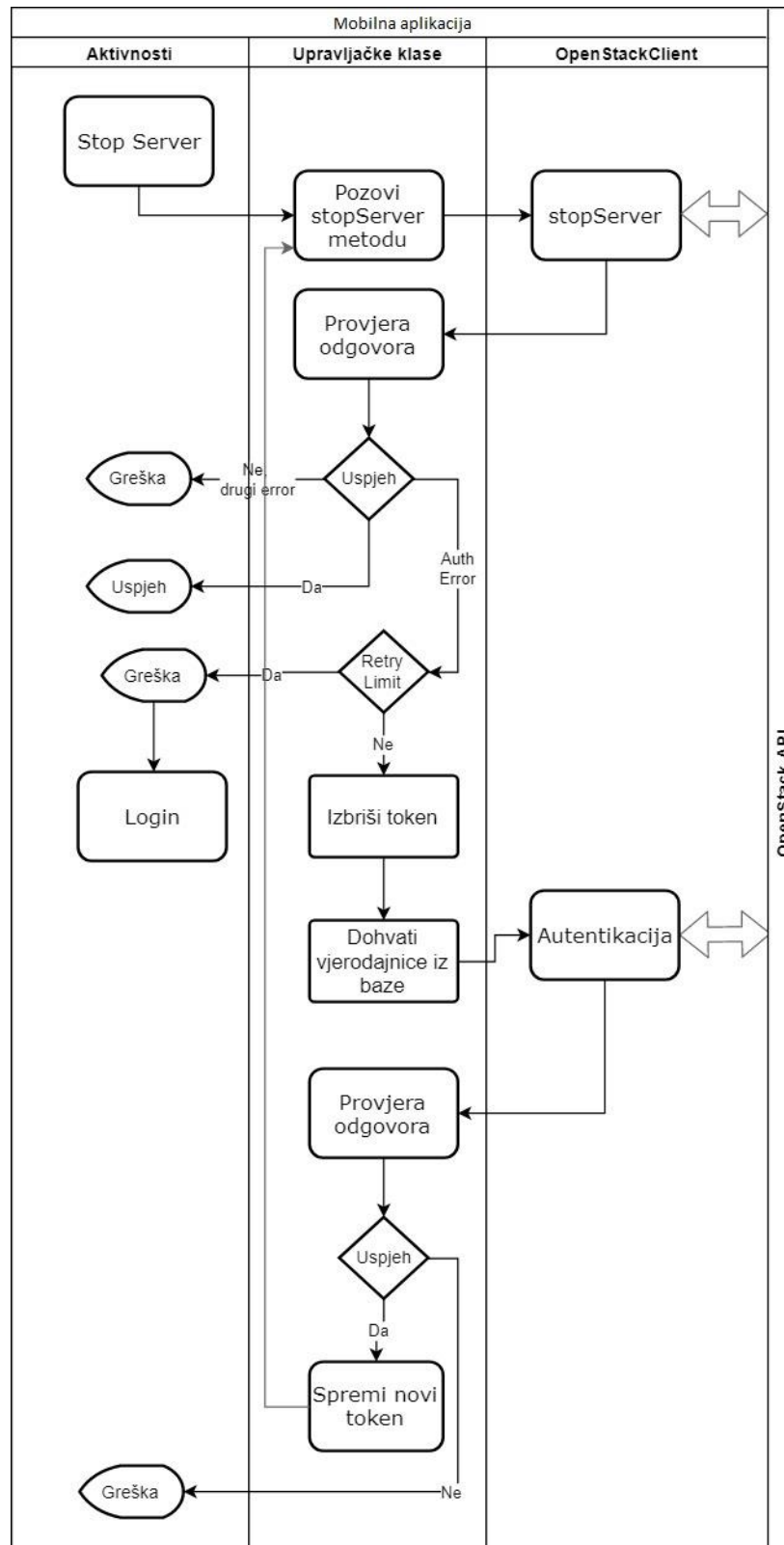
Prva akcija koju aplikacija mora izvršiti kako bi mogla uspješno komunicirati sa OpenStack servisima je autentikacija korisnika i dohvaćanje tokena (slika 4). Zatim je, u slučaju uspješne autentikacije, moguće dohvatiti listu projekata kojima korisnik ima pristup. Svaki od projekata može sadržavati više instanci, koje su prikazane na korisničkom sučelju odabirom pojedinog projekta. Kako se u svakom zahtjevu nakon autentikacije šalje autorizacijski token, moguće je da token istekne tijekom korištenja aplikacije. U tom slučaju, aplikacija će automatski poslati zahtjev za novim tokenom (slika 5). Specifični sadržaj zahtjeva za tokenom ovisi o tome traži li se token za pristup listi projekata ili token za pristup instancama odabranog projekta.

Aplikacija reagira na grešku u komunikaciji tako da ispiše poruku o grešci u obavijesti koja se može prikazati na bilo kojoj aktivnosti. Neke od mogućih poruka o grešci dobivene od OpenStack API-ja su: badRequest(400), unauthorized(401), forbidden(403), itemNotFound(404) ili conflict(409). U slučaju jednog od dokumentiranih odgovora koji sadrži grešku ili u slučaju nepredviđene greške u radu aplikacije, korisnik, uz prikaz punog teksta greške, ima mogućnost pokušati nastaviti s korištenjem aplikacije ili se vratiti na početni Login ekran. Iznimno, greška u autentikaciji ili nemogućnosti dohvaćanja tokena tretira se tako da se osim prikazivanja poruke, aplikacija automatski vrati na Login ekran.

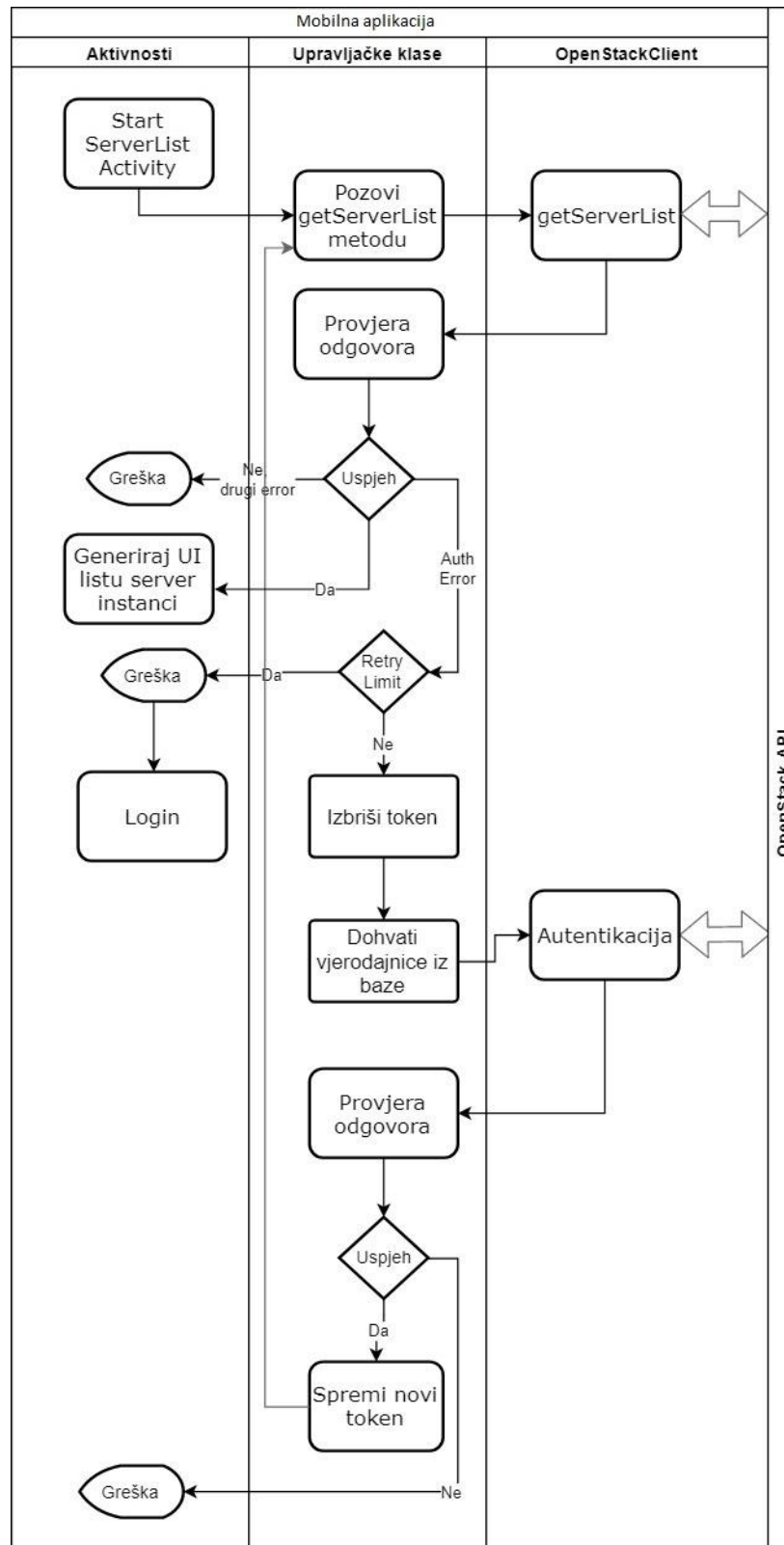
Dijagram toka za primjer zaustavljanja instance prikazan je na slici 5. Analogno bi izgledao dijagram toka za bilo koju od ostalih podržanih akcija. Slika 6 prikazuje sličan dijagram toka za prikaz liste instanci.



SLIKA 4: DIJAGRAM TOKA – AUTENTIKACIJA I PRIJELAZ NA SLJEDEĆU AKTIVNOST



SLIKA 5: DIJAGRAM TOKA – ZAUSTAVLJANJE INSTANCE



SLIKA 6: DIJAGRAM TOKA – POKRETANJE AKTIVNOSTI SERVERLIST

Primjeri komunikacije s OpenStack API servisima

Primjer autentikacije

Šaljemo POST zahtjev na URI v3/auth/token.

URI: `http://192.168.122.168/identity/v3/auth/tokens`

Tijelo zahtjeva:

```
{
  "auth": {
    "identity": {
      "methods": ["password"],
      "password": {
        "user": {
          "auth": {
            "identity": {
              "name": "admin",
              "domain": { "id": "default" },
              "password": "secret"
            }
          }
        }
      }
    }
  }
}
```

Zaglavlje odgovora:

```
Date = Mon, 09 Apr 2018 01:58:22 GMT
Server = Apache/2.4.25 (Debian)
X-Subject-Token = gAAAAABa...
Vary = X-Auth-Token
Content-Type = application/json
Content-Length = 312
x-openstack-request-id = req-aa064cc7-38e7-4e1c-8e18-1c7991980ad4
Connection = close
```

Tijelo odgovora:

```
{
  "token": {
    "issued_at": "2018-04-09T01:58:22.000000Z",
    "audit_ids": [
      "PeNXUOneTbGYyX5xM-Pzsg"
    ],
    "methods": [
      "password"
    ],
    "expires_at": "2018-04-09T02:58:22.000000Z",
    "user": {
      "password_expires_at": null,
      "domain": {
```

```

        "id": "default",
        "name": "Default"
    },
    "id": "0bd5d0b530dd4e809e17ab4ed6223805",
    "name": "admin"
}
}
}

```

Autorizacijski token kopiramo iz zaglavlja kao vrijednost ključa “X-Subject-Token”.

Primjer zahtjeva za listom projekata

GET <http://192.168.122.168/identity/v3/auth/projects>

Zaglavlje zahtjeva:

X-Auth-Token: gAAAAABa... (kopirano iz zaglavlja prethodnog odgovora)

Tijelo odgovora (lista projekata je skraćena):

```

{
  "links": {
    "self": "http://192.168.122.168/identity/v3/auth/projects",
    "previous": null,
    "next": null
  },
  "projects": [
    ...
    {
      "is_domain": false,
      "description": "",
      "links": {
        "self": "http://192.168.122.168/identity/v3/projects/ebae99749b094bb48b5d2a04743de410"
      },
      "tags": [
        ...
      ],
      "enabled": true,
      "id": "ebae99749b094bb48b5d2a04743de410",
      "parent_id": "default",
      "domain_id": "default",
      "name": "demo"
    }
  ]
}

```

Primjer autentikacije s autorizacijom za odabrani projekt

POST <http://192.168.122.168/identity/v3/auth/tokens>

Tijelo zahtjeva:

```
{
  "auth": {
    "identity": {
      "methods": ["password"],
      "password": {
        "user": {
          "name": "admin",
          "domain": { "id": "default" },
          "password": "secret"
        }
      }
    },
    "scope": {
      "project": {
        "name": "demo",
        "domain": { "id": "default" }
      }
    }
  }
}
```

Zaglavlje odgovora:

```
Date = Mon, 09 Apr 2018 03:52:42 GMT
Server = Apache/2.4.25 (Debian)
X-Subject-Token = gAAAAABayuOLl1QBZ6n...
Content-Type = application/json
Content-Length = 3396
x-openstack-request-id = req-f6157146-2acd-4f2d-b821-9403de881266
Connection = close
```

Tijelo odgovora (skraćeno):

```
{
  "token":{
    "is_domain":false,
    "methods":[
      "password"
    ],
    "roles":[
      {
        "id":"98a6ca7cb1af4404865b0dd0db49bb33",
        "name":"admin"
      }
    ],
    "expires_at":"2018-04-09T04:52:43.000000Z",
    "project":{
      "domain":{
        "id":"default",
        "name":"Default"
      },

```

```

        "id":"ebae99749b094bb48b5d2a04743de410",
        "name":"demo"
    },
    "catalog":[
        {
            "endpoints":[
                {
                    "region_id":"RegionOne",
                    "url":"http://192.168.122.168/compute/v2.1",
                    "region":"RegionOne",
                    "interface":"public",
                    "id":"da27767e6359453691bbeba0cff11b48"
                }
            ],
            "type":"compute",
            "id":"01894b766a464a23a1b028917ec17a33",
            "name":"nova"
        },
        ...
    ],
    "user":{
        "password_expires_at":null,
        "domain":{
            "id":"default",
            "name":"Default"
        },
        "id":"0bd5d0b530dd4e809e17ab4ed6223805",
        "name":"admin"
    },
    "audit_ids":[
        "o0HMOZUQSUiJprJ6waB82Q"
    ],
    "issued_at":"2018-04-09T03:52:43.000000Z"
}

```

Primjer dohvaćanja liste instanci za odabrani projekt

GET

<http://192.168.122.168/compute/v2.1/ebae99749b094bb48b5d2a04743de410/servers>

Zaglavlje zahtjeva:

X-Auth-Token: gAAAAABayuOLl1QBZ6n... (kopirano iz zaglavlja prethodnog odgovora)

Tijelo odgovora:

```

{
  "servers":[
    {
      "id":"d76bb0a1-b8e7-4393-95bd-fe8a1110e7fa",
      "links":[
        {

```

```

        "href":"http://192.168.122.168/compute/v2.1/ebae99749b094bb48b5d2a04743de410/servers/d76bb0a1-b8e7-4393-95bd-fe8a1110e7fa",
        "rel":"self"
    },
    {
        "href":"http://192.168.122.168/compute/ebae99749b094bb48b5d2a04743de410/servers/d76bb0a1-b8e7-4393-95bd-fe8a1110e7fa",
        "rel":"bookmark"
    }
],
"name":"testInstance"
}
]
}

```

Primjer pauziranja instance

POST

http://192.168.122.168/compute/v2.1/ebae99749b094bb48b5d2a04743de410/servers/d76bb0a1-b8e7-4393-95bd-fe8a1110e7fa/action

Zaglavlje zahtjeva:

X-Auth-Token: gAAAAABayuOLllQBZ6n...

Tijelo zahtjeva:

```

{
    "pause": null
}

```

Status uspješnog odgovora:

HTTP/1.1 202 Accepted

Tijelo uspješnog odgovora je prazno.

Status odgovora u slučaju da je instance već pauzirana:

HTTP/1.1 409 Conflict

Tijelo odgovora u slučaju da je instance već pauzirana:

```

{
    "conflictingRequest":{
        "message":"Cannot 'pause' instance d76bb0a1-b8e7-4393-95bd-fe8a1110e7fa while it is in vm_state paused",
        "code":409
    }
}

```

Moguća proširenja aplikacije

Aplikacija omogućuje korisniku komunikaciju s OpenStack API servisima, te implementira nekoliko jednostavnih akcija koje nudi Compute API. Implementacija autentikacije i pristupa OpenStack API-ju otvara vrata korištenju bilo koje od dostupnih metoda na Compute i drugim javno dostupnim RESTful servisima. Neki od primjera su:

- akcije vezane za upravljanje sigurnosnim kopijama (eng. backup) podataka,
- upravljanje *block storage*-om,
- promjena veličine diska na serveru (resize),
- upravljanje migracijom servera,
- upravljanje *load balancer*-ima

Aplikacija je dizajnirana da podržava jednog korisnika i jedan OpenStack server, no, lako bi se u bazu moglo spremati više redaka koji sadrže korisnikove vjerodajnice i URL OpenStack servera, te ponuditi padajući izbornik za brzi izbor bilo kojeg prethodno korištenog OpenStack servera ili korisničkog računa.

Android aplikacije imaju mogućnost prikazivanja obavijesti na *lock* ekranu. Aplikacija bi mogla periodički dohvaćati informacije o instancama sa servera, te, u slučaju neke specifične informacije, prikazati obavijest koju će korisnik vidjeti sljedeći put kad pogleda svoj mobilni uređaj, bez potrebe za čestim otvaranjem same aplikacije.

Ukoliko OpenStack server koji želimo koristiti nije javno dostupan, ali postoji javno dostupan server koji može pristupiti API servisima OpenStack servera, što bi mogla biti neka instanca kreirana na OpenStack serveru, tada se aplikacija može proširiti serverskom *proxy* aplikacijom koja će preusmjeravati zahtjeve sa servera koji ima javnu IP adresu na OpenStack server.

Usvojena znanja i vještine

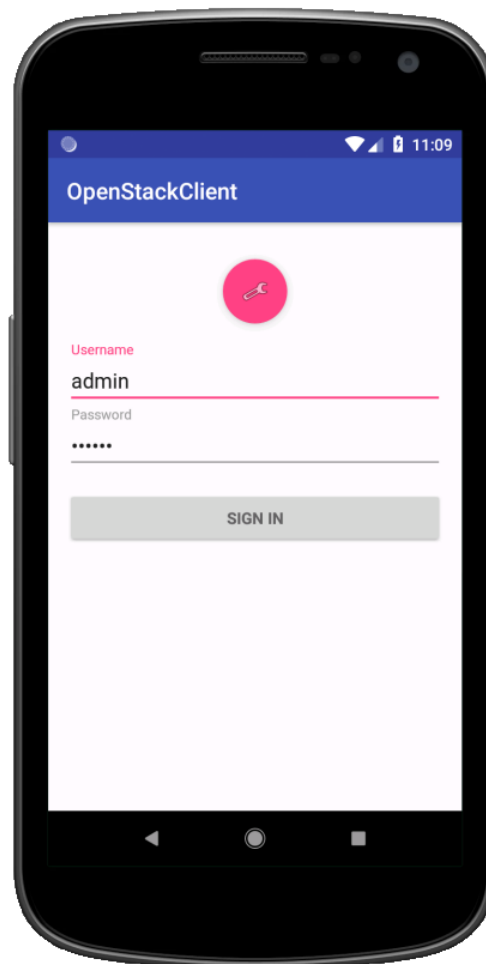
Tijekom dosadašnjeg rada na aplikaciji usvojila sam sljedeća znanja:

- Instalacija DevStack okruženja
- Kreiranje instanci na OpenStacku pomoću Horizon web sučelja
- Pristup OpenStack servisima putem REST sučelja
 - Autentikacija korištenjem Identity servisa
 - Upravljanje instancama putem Compute servisa
- Slanje HTTP zahtjeva u Javi pomoću Apache HttpComponents paketa
- Kreiranje Android aplikacije s Login ekranom
 - Upisivanje korisničkog imena i lozinke
 - Prijelaz na novi ekran nakon autentikacije
 - Pohrana podataka u SQLite bazu
 - Korištenje ListView elementa za dinamičko generiranje izbornika

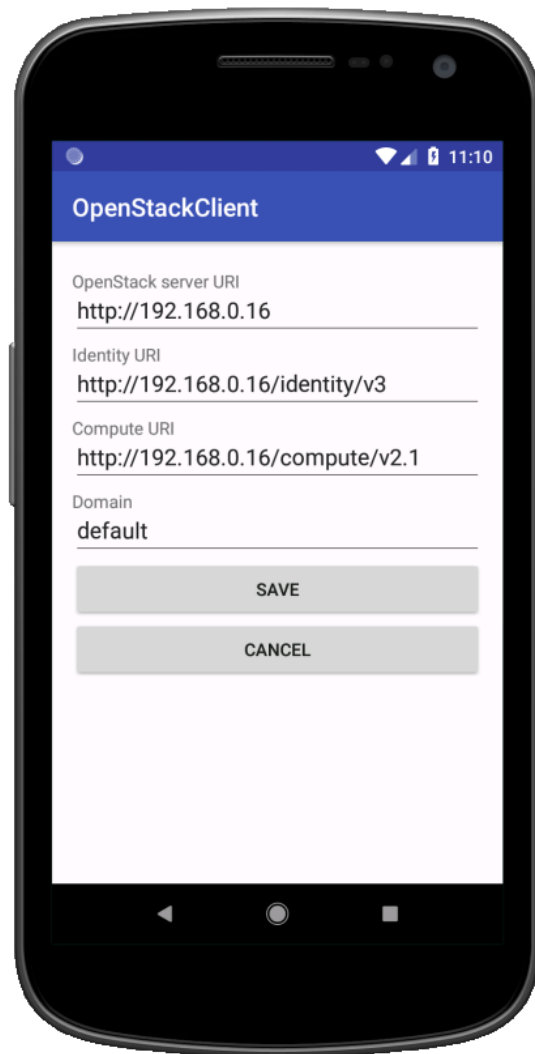
- Intent klasa, pomoću koje prenosimo informacije iz jedne aktivnosti u drugu

Implementacija

Aplikacija je napisana u Anroid Studio sučelju. Pri izradi aplikacije korišteno je lokalno DevStack okruženje i udaljeni OpenStack server koji pruža CloudVPS posužitelj. Kako CloudVPS koristi drugačije putanje do API servisa Compute i Identity, na Settings ekranu dodana su polja u koja je moguće unijeti specifične putanje do servisa. Ako su polja prazna, bit će ignorirana i aplikacija će koristiti putanje identične onima na DevStack-u. Login ekran i postavke prikazane su na slikama 7 i 8.

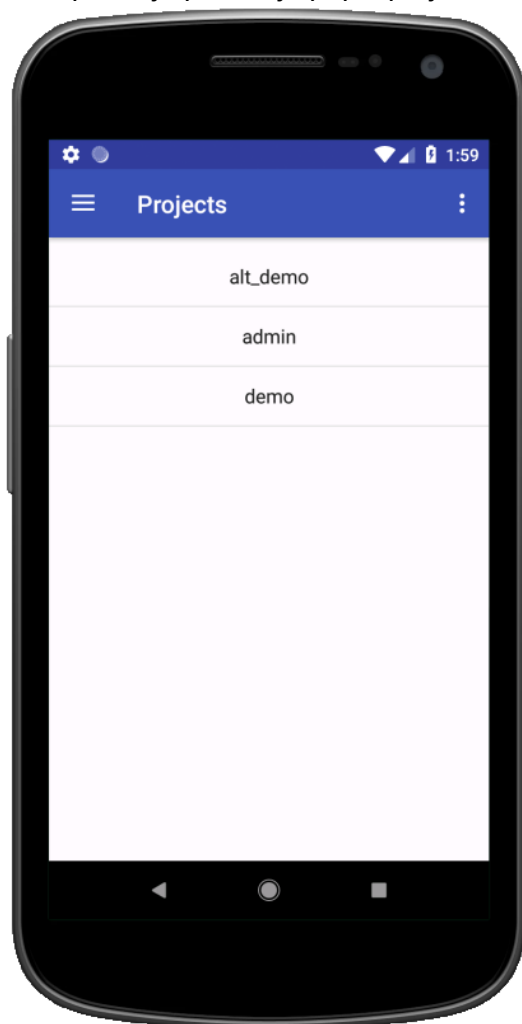


SLIKA 7: LOGIN



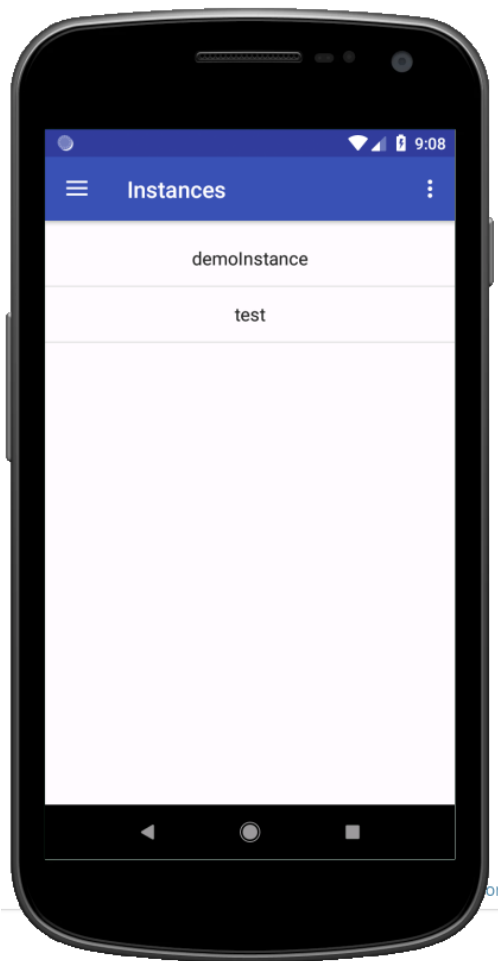
SLIKA 8: POSTAVKE

Nakon uspješnog logina, aplikacija prikazuje popis projekata koji pripadaju korisniku



SLIKA 9: PROJEKTI

Odabirom projekta prikazuje se popis instanci.



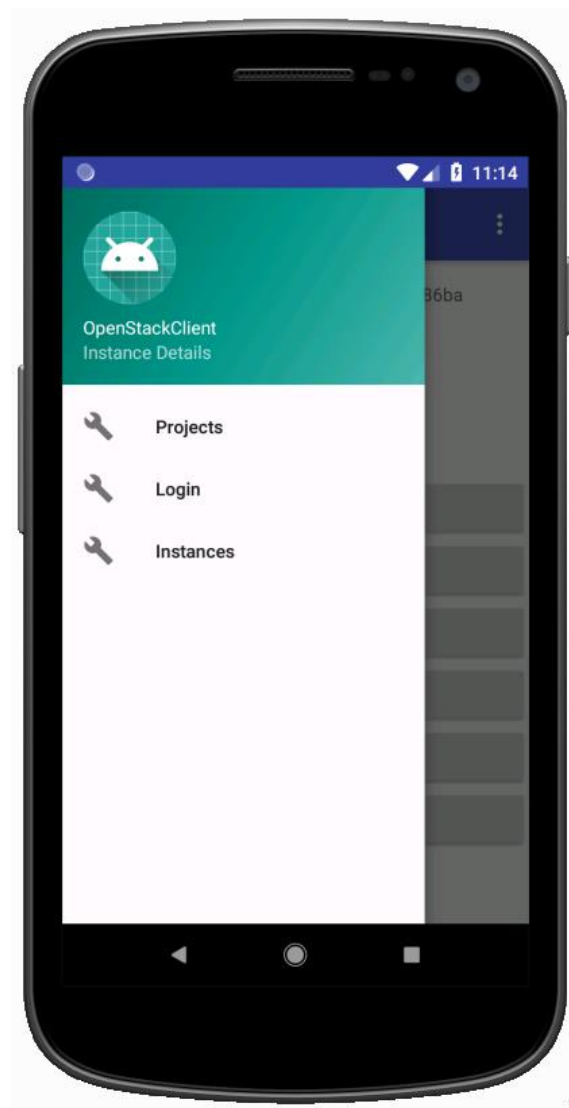
SLIKA 10: INSTANCE

Odabirom instance prikazuju se podatci o instanci, te gumbi za pojedine akcije na instanci.



SLIKA 11: INSTANCA

Kod aplikacije je priložen ovom radu. Obnavljanje tokena je implementirano tako da aplikacija u slučaju greške pokuša zatražiti novi token i do 5 puta dohvatiti traženi resurs. Ako i dalje odgovor nije uspješan, aplikacija se vraća na početni ekran. Na početni ekran je moguće vratiti se i pomoću izbornika u gornjem desnom kutu, a na početnom ekranu su polja popunjena već unesenim podacima, tako da je dovoljno kliknuti Login za nastavak rada. Uspješno se mogu podesiti URL adrese servera i vjerodajnice korisnika, dohvatiti lista projekata i instanci za pojedini projekt, vidjeti osnovne informacije o instanci, te izvršiti akcije na instanci. Ako postoji više od pet instanci, iznad liste se prikaže polje za filtriranje instanci po nazivu. Svi upiti na bazi podataka i slanje zahtjeva preko mreže se izvršavaju na pozadinskoj dretvi. Gumb za postavke nije implementiran na svakom ekranu, kako je originalno planirano, već samo na Login ekranu jer su sve spremljene postavke direktno vezane za prijavu korisnika. Na ekrane koji prikazuju listu projekata, instanci i odabranu instancu, dodan je *drawer* izbornik s lijeve strane, prikazan na slici 12, pomoću kojeg se može navigirati unutar aplikacije.



SLIKA 12: DRAWER IZBORNIK

Rasprava

Aplikacija pruža *lightweight* sučelje za pristup OpenStack resursima s mobilnog uređaja. Rad aplikacije ovisi o dostupnosti OpenStack servera kojem aplikacija pristupa, a brzina izvršavanja zahtjeva ovisi o brzini odgovora servera. Kako bi sučelje aplikacije bilo upotrebljivo neovisno o trajanju obrade zahtjeva na serveru, API pozivi se izvršavaju u pozadinskom procesu.

Greške u aplikaciji obrađuju se tako da aplikacija korisnika informira o grešci, te nudi mogućnost čitanja detalja greške, nastavka s radom, ili povratka na početni ekran. Jedino u slučaju greške u autentikaciji, aplikacija se automatski vraća na Login ekran, također uz ispis informacije o grešci.

Dizajn aplikacije prati standarde objektno orijentiranog programiranja, s naglaskom na modularnost i mogućnost ponovnog korištenja pojedinih komponenti. Takav pristup olakšava testiranje aplikacije i omogućuje laku zamjenu pojedine komponente. Komponenta

OpenStackClient je potpuno neovisna o Android sučelju aplikacije, te se može iskoristiti za izradu slične aplikacije na drugoj platformi.

Aplikacija je dizajnirana da podržava samo nekoliko osnovnih akcija na virtualnim serverima, no, kako koristi OpenStack API, moguće je proširiti skup funkcionalnosti na bilo koju od postojećih API metoda. Skup mogućih funkcionalnosti aplikacije je ograničen jedino funkcionalnostima koje pruža OpenStack API. Ako želimo funkcionalnost koju ne implementira nijedan od OpenStack servisa, bilo bi nužno prvo kreirati servis koji izvršava željenu funkcionalnost.

Postoji mogućnost da OpenStack server nije dostupan na Internetu, već samo na lokalnoj mreži. Kako se radi o mobilnoj aplikaciji, nužno je da može komunicirati s API servisima. Ako je API servisima moguće pristupiti s virtualnog stroja na OpenStack serveru, može se kreirati *proxy* servis koji će biti javno dostupan i usmjeravati zahtjeve OpenStack serveru. Tada je u aplikaciju potrebno unijeti IP virtualnog stroja preko kojeg se pristupa API servisima.

Zaključak

„Android aplikacija za OpenStack“ je namijenjena korisnicima OpenStacka koji žele na brz i jednostavan način dobiti uvid u stanje svojih virtualnih servera putem mobilnog uređaja. Dovoljno je unijeti korisničko ime i lozinku, te postaviti URL kako bi aplikacija mogla dohvatiti sve potrebne informacije s udaljenog servera.

Aplikacija je napisana u Java jeziku i namijenjena Android uređajima. Podijeljena je na dva dijela - komunikacija s OpenStack API-jem i Android sučelje. Pregled i upravljanje instancama servera na OpenStacku omogućeno je kroz temeljne funkcionalnosti kao što su: pregled i filtriranje liste servera, pretraživanje liste servera, pregled podataka o serveru, te operacije na serveru: pauza, start, stop i reboot.

Kratki pregled literature

Korisna iskustva sa Sveučilištu Amity u implementaciji hibridne infrastrukture kao usluge u oblaku, kako bi se poboljšalo iskustvo učenja i iskustva učenika, vide se u radu Deepak, A., Kumar, V., Jagdeep, B., Verma, P., *Proposed Model for Virtual Labs Interaction With Openstack Integration Using KVM Hypervisor*, International Journal of Scientific and Technology Research, vol. 3, issue 7, pp 64-67, July 2014. Ovaj članak pokriva metode konfiguracije i implementacije OpenStack okruženja uz sigurnosne mjere koje prate isporuku željenog hardvera računala.

OpenStack i KVM hypervisor spadaju u softver otvorenog koda i jedan su od lidera na tržištu za računarstvo u oblaku. Upotrebom aplikacije MyCloud korisnik može pristupiti svojim dokumentima s bilo koje lokacije s pristupom Internetu i u bilo koje vrijeme. Posebnu primjenu te aplikacije na Androidu možemo vidjeti u radu Gulekar, A., Rawat, K., Selvin, M., Shaikh, M., Devmane, M., *Secure File Storage in Cloud and Easy Access using Android – MyCloud*, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol 2, Issue 2, pp 983-988, 2017.

U knjizi Meier, R., *Professional Android 4 Application Development*, Wrox, 2012, opisan je Android SDK i API-ji dostupni u Android verzijama 3 i 4. Knjiga pruža praktične primjere koji se mogu koristiti kao osnova za razvoj Android aplikacija.

Važnost OpenStacka kao pružatelja usluga u oblaku i postupak instalacije istaknuli su u radu Pol, Urmila R., *Cloud Computing with Open Source Tool: OpenStack*, American Journal of Engineering Research (AJER), vol 3, issue 9, pp 233-240, 2014. U radu je opisano nekoliko

platformi za računarstvo u oblaku, kao što su Openstack, Eucalyptus, CloudStack i OpenNebula.

U knjizi Barton, T., Herrmann, F., Meister, V. G., Müller, C., Seel, C., *Prozesse, Technologie, Anwendungen, Systeme und Management*, Technischen Hochschule Brandenburg, University of Applied Sciences, 2016, u poglavlju *Development of an Android app for user administration in private clouds*, naglašen je porast upotrebe privatnih oblaka u Njemačkoj s obzirom da veliki dio tvrtki koristi upravo privatne oblake umjesto javnih ili hibridnih oblaka, zbog prednosti koje se odnose na sigurnosne aspekte. U poglavlju je opisano planiranje i implementacija aplikacija za mobilne korisnike u okruženju privatnog oblaka temeljem OpenStacka.

Knjiga Kirk Knoernschild, *Java Application Architecture*, Robert C. Martin Series, Prentice Hall, 2012, daje pregled prednosti korištenja modularnog pristupa u arhitekturi Java aplikacija, analizu do koje mjere se primjenjuje modularnost u različitim slučajevima, te daje primjere uz opise različitih uzoraka dizajna (eng. design patterns) koji podržavaju modularnost.

Knjiga T.Fifield, L. Hochstein, et al, *OpenStack Operations Guide*, OpenStack Foundation, 2015, namijenjena je prvenstveno administratormima, te pruža detaljne tehničke informacije o upravljanju oblakom na dnevnoj bazi.

Literatura

- [1] Barton, T., Herrmann, F., Meister, V. G., Müller, C., Seel, C., *Prozesse, Technologie, Anwendungen, Systeme und Management*, Technischen Hochschule Brandenburg, University of Applied Sciences, 2016, Poglavlje „Development of an Android app for user administration in private clouds“, Roberto Damm, Karsten Würth, Thomas Barton., pp 193-201.
- [2] Gulekar, A., Rawat, K., Selvin, M., Shaikh, M., Devmane, M., *Secure File Storage in Cloud and Easy Access using Android – MyCloud*, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol 2, Issue 2, pp 983-988, 2017
- [3] Meier, R., *Professional Android 4 Application Development*, Wrox, 2012
- [4] Deepak, A., Kumar, V., Jagdeep, B., Verma, P., *Proposed Model for Virtual Labs Interaction With Openstack Integration Using KVM Hypervisor*, International Journal of Scientific and Technology Research, vol. 3, issue 7, pp 64-67, July 2014
- [5] Pol, Urmila R., *Cloud Computing with Open Source Tool: OpenStack*, American Journal of Engineering Research (AJER), vol 3, issue 9, pp 233-240, 2014
- [6] T.Fifield, L. Hochstein, et al, *OpenStack Operations Guide*, OpenStack Foundation, 2015
- [7] Kirk Knoernschild, *Java Application Architecture*, Robert C. Martin Series, Prentice Hall, 2012

Web stranice korištene u izradi aplikacije

- [8] OpenStack Docs: DevStack, <https://docs.openstack.org/devstack/latest/> pristupljeno 09.04.2018.
- [9] Apache HttpComponents – Apache HttpComponents, <https://hc.apache.org/> pristupljeno 09.04.2018.
- [10] OpenStack Docs: Identity API v3 (CURRENT), <https://developer.openstack.org/api-ref/identity/v3/> pristupljeno 09.04.2018.
- [11] OpenStack Docs: Compute API, <https://developer.openstack.org/api-ref/compute/> pristupljeno 09.04.2018.
- [12] Meet Android Studio | Android Studio, <https://developer.android.com/studio/intro/index.html> pristupljeno 09.04.2018.