Specifikacija softverskih zahteva

za

Firefly – postavite svoj backend brzo

Verzija 1.0

Student Petar Otovic

Broj indeksa 5460

OAS Softversko inzenjerstvo, 2. godina

Predmet: SE222 Inzenjerstvo zahteva

Skolska 2023/24. godina

4.1.2024

Copyright © 2013 by Karl Wiegers and Seilevel. Permission is granted to use and modify this document

**Istorijat verzija dokumenta**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ime** | **Datum** | **Razlog za promenu** | **Verzija** |
| Petar Otovic | 4.1.2024 | Kreiranje dokumenta | 1.0.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**SADRZAJ**

[1. Uvod 3](#_Toc155464423)

[1.1 Svrha 3](#_Toc155464424)

[1.2 Konvencije o dokumentima 4](#_Toc155464425)

[1.3 Obim projekta 4](#_Toc155464426)

[1.4 Reference 5](#_Toc155464427)

[2. Opsti opis 6](#_Toc155464428)

[2.1 Perspektive proizvoda 6](#_Toc155464429)

[2.2 Klase i karakteristike korisnika 6](#_Toc155464430)

[2.3 Operativno okruzenje 8](#_Toc155464431)

[2.4 Ogranicenja u projektovanju i primeni 8](#_Toc155464432)

[2.5 Pretpostavke i zavisnosti 9](#_Toc155464433)

[3. Funkcije sistema 10](#_Toc155464434)

[3.1 Kreiranje aplikacije 10](#_Toc155464435)

[3.1.1 Opis 10](#_Toc155464436)

[3.1.2 Sekvence stimulusa/odgovora 10](#_Toc155464437)

[3.1.3 Funkcionalni zahtevi 10](#_Toc155464438)

[3.2 Pravljenje baze podataka i komunikacija sa istom 10](#_Toc155464439)

[3.2.1 Opis 10](#_Toc155464440)

[3.2.2 Sekvence stimulusa/odgovora 10](#_Toc155464441)

[3.2.3 Funkcionalni zahtevi 11](#_Toc155464442)

[3.3 Autentifikacija i Autorizacija 11](#_Toc155464443)

[3.3.1 Opis 11](#_Toc155464444)

[3.3.2 Sekvence stimulusa/odgovora 11](#_Toc155464445)

[3.3.3 Funkcionalni zahtevi 11](#_Toc155464446)

[3.4 Upravljanje korisnicima 11](#_Toc155464447)

[3.4.1 Opis 11](#_Toc155464448)

[3.4.2 Sekvence stimulusa/odgovora 11](#_Toc155464449)

[3.4.3 Funkcionalni zahtevi 12](#_Toc155464450)

[4. Zahtevi za podatke 12](#_Toc155464451)

[4.1 Logicki model podataka 13](#_Toc155464452)

[4.2 Recnik podataka 13](#_Toc155464453)

[4.3 Izvestaji 14](#_Toc155464454)

[5. Zahtevi za spoljni interfejs 15](#_Toc155464455)

[5.1 Korisnicki interfejsi 15](#_Toc155464456)

[5.2 Softverski interfejsi 17](#_Toc155464457)

[5.3 Hardverski interfejsi 17](#_Toc155464458)

[5.4 Komunikacioni interfejsi 18](#_Toc155464459)

[6. Atributi kvaliteta 18](#_Toc155464460)

[6.1 Upotrebljivost 18](#_Toc155464461)

[6.2 Performanse 19](#_Toc155464462)

[6.3 Bezbednost 19](#_Toc155464463)

[6.4 Sigurnost 19](#_Toc155464464)

[6.5 Efikasnost 19](#_Toc155464465)

[7. Uslovi internacionalizacije i lokalizacije 19](#_Toc155464466)

# Uvod

## Svrha

Postavljajuci svoj backend od nule moze da predstavlja pravi izazov za mnoge softverske kompanije, zato sto oduzima previse vremena I zahteva dosta posla.

Sa sve vecim zahtevima za backend funkcionalnostima, sigurnosnim standardima i personalizovanim korisnickim iskustvima, prepoznali smo potrebu za inovativnim pristupom backend tehnologiji. Ova potreba proizilazi iz rasta korisnickih ocekivanja i zelje za efikasnijim razvojem aplikacija.

Analizom trenutnih resenja na trzistu, uocili smo prostor za unapredjenje koje nasa nova backend-as-a-service aplikacija moze da pruzi. Inspirisani modelom usluge kao sto je Firefly, prepoznali smo mogucnost da stvorimo platformu koja olaksava upravljanje I skladistenje podataka i informacija o korisnicima.

Ovaj proizvod proizlazi iz jasne potrebe za efikasnim, skalabilnim i sigurnim backend resenjem koje ce omoguciti softverskim kompanijama da se fokusiraju na inovacijama u svojim softverskim resenjima, umesto da trose resurse na kompleksno odrzavanje i upravljanje backend infrastrukturom.

**Programeri:**

Interesovanje: Detaljni tehnicki zahtevi, opis arhitekture, alatke i tehnologije koje ce biti koriscene.

Cilj: Razumevanje kako implementirati funkcionalnosti i pridrzavati se postavljenih standarda.

**Rukovodioci Projekta:**

Interesovanje: Vremenski okviri, alokacija resursa, rizici i planovi.

Cilj: Efikasno upravljanje projektom, identifikacija kljucnih tacaka i resursa.

**Marketinsko Osoblje:**

Interesovanje: Kljucne karakteristike proizvoda, prednosti za krajnje korisnike, mogucnosti proizvoda

Cilj: Razvijanje marketinskih strategija, razumevanje kako proizvod pruza vrednost korisnicima.

**Korisnici:**

Interesovanje: Funkcionalnosti proizvoda, korisnicko iskustvo, podrzane platforme.

Cilj: Razumevanje sta proizvod nudi i kako ce olaksati njihov posao I ciljeve.

**Testeri:**

Interesovanje: Test scenariji, zahtevi za performanse, uskladjenost sa specifikacijama.

Cilj: Identifikacija i sprovodjenje testova kako bi se garantovala ispravnost proizvoda.

**Pisci Dokumentacije:**

Interesovanje: Jasan opis funkcionalnosti, uputstva za korisnike, tehnicka dokumentacija.

Cilj: Kreiranje relevantne dokumentacije koja pruza potpunu sliku o proizvodu.

## Konvencije o dokumentima

**Oznake zahteva:**

Svaki zahtev ce biti oznacen jedinstvenim identifikatorom radi lakse reference.

Format identifikatora: [TipZahteva]-[BrojZahteva], na primer: F-01 za funkcionalni zahtev, N-01 za nefunkcionalni zahtev.

**Stilovi Teksta:**

Bold: Koriscen za naglasavanje kljucnih termina, naslova odeljaka, i posebno zahteva koji su od posebnog znacaja.

Italic: Koriscen za citate, definicije pojmova, i naglasavanje dodatnih informacija.

**Reference i Hiperlinkovi:**

Reference na druge delove dokumenta ili vanjske izvore bice jasno oznacene i, gde je moguce, pruzene kao hiperlinkovi za brzi pristup dodatnim informacijama.

**Format Datuma i Vremena:**

Datumi ce biti napisani u formatu D-M-G (dan-mesec-godina), a vreme u 24-satnom formatu.

## Obim projekta

Softver koji se specificira ovim dokumentom ima za cilj razvoj Backend-as-a-Service (BaaS) aplikacije, koja ce funkcionisati slicno kao Firefly platforma. Ova BaaS aplikacija imace kljucnu ulogu u olaksavanju razvoja i upravljanja backend infrastrukturom za razlicite aplikacije.

**Svrha Softvera:**

Pruzanje korisnicima (programerima i razvojnim timovima) platforme koja omogucava brz i efikasan razvoj aplikacija fokusiranjem na backend funkcionalnosti.

Omogucavanje jednostavnog upravljanja bazama podataka, korisnicima, autentifikacijom i drugim backend aspektima bez potrebe zarucnim I kompleksnim resenjima.

**Povezanost sa Korisnickim Ciljevima:**

Programeri: Omogucavanje brze i efikasne implementacije backend infrastrukture bez potrebe za poznavanjem backend tehnologija.

Razvojni Timovi: Povecanje produktivnosti i smanjenje vremena potrebnog za razvoj i implementaciju backend funkcionalnosti.

**Povezanost sa Poslovnim Ciljevima i Strategijama:**

Agilnost: Pruzanje fleksibilnosti i agilnosti u razvoju aplikacija, sto doprinosi brzem plasmanu aplikacija na trziste.

Efikasnost: Smanjenje opterecenja razvojnih timova kroz pruzanje spremnih i pouzdanih backend resenja.

**Glavne Funkcionalnosti:**

Upravljanje Bazama Podataka: Pruzanje funkcionalnosti za lako upravljanje i manipulaciju podacima.

Autentikacija i Autorizacija: Omogucavanje autentifikacije korisnika i upravljanje pravima pristupa korisnika raznim sadrzajima.

Upravljanje Korisnicima: Kreiranje i upravljanje korisnickim nalozima.

Real time funkcionalnosti: Mogucnost uvida u real-time azuriranje podataka I sinhronizaciju izmedju korisnika.

## Reference

**Slicni proizvodi:**

* Firebase:

Veb sajt: <https://firebase.google.com/>

Dokumentacija: <https://firebase.google.com/docs/guides>

* Amazon DynamoDB

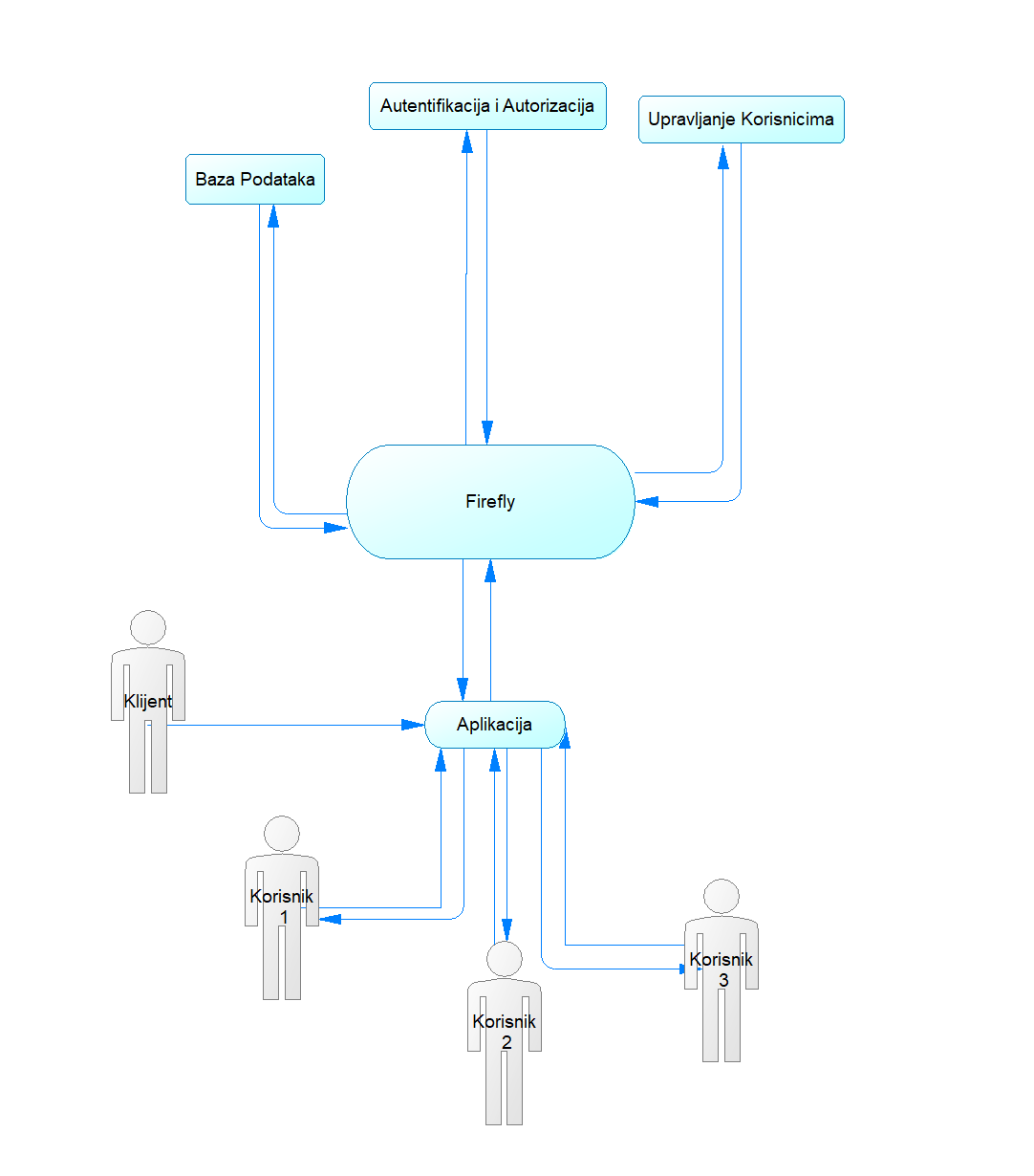
Veb sajt: <https://aws.amazon.com/dynamodb/>

Dokumentacija: <https://aws.amazon.com/dynamodb/features/>

# Opsti opis

## Perspektive proizvoda

Ovaj softver predstavlja potpuno novi proizvod u skladu sa zahtevima kompanije. Nije deo rastuce linije proizvoda niti sledeca verzija postojeceg sistema. Ova platforma se razvija kako bi zadovoljila potrebe trzista za Backend-as-a-Service (BaaS) funkcionalnostima. Ova platforma omogucava brz i jednostavan razvoj backend infrastrukture za razlicite aplikacije.



## Klase i karakteristike korisnika

Ove klase korisnika predstavljaju razlicite grupe zainteresovanih strana sa specificnim potrebama i ocekivanjima od proizvoda.

**Razvojni Timovi:**

Karakteristike:

* Napredno znanje u razvoju aplikacija.
* Potreba za brzim i efikasnim backend resenjima.

Ocekivanja:

* Pruzanje alata za ubrzanje backend razvoja.
* Mogucnost jednostavnog upravljanja backend infrastrukturom.

**Programeri:**

Karakteristike:

* Razliciti nivoi iskustva u razvoju.
* Traze efikasne i lake za upotrebu alate.

Ocekivanja:

* Jasna dokumentacija za lako integrisanje sa backend funkcionalnostima.
* Dokumentacija koja podrzava brzu implementaciju.

**Sistem Administratori**:

Karakteristike:

* Vestine u upravljanju infrastrukturom.
* Potreba za bezbednim i skalabilnim backend resenjima.

Ocekivanja:

* Mogucnost efikasnog upravljanja i skaliranja sistema.
* Alati za pracenje performansi i otklanjanje problema.

**Menadzeri Proizvoda:**

Karakteristike:

* Razumevanje trzista i potreba korisnika.
* Fokus na poslovnim ciljevima i strategijama.

Ocekivanja:

* Izvestaji o performansama i upotrebi kako bi informisali o strategijskim odlukama.
* Alati koji doprinose agilnosti i smanjenju vremena do plasmana na trziste.

**Korisnici Aplikacija:**

Karakteristike:

* Krajnji korisnici aplikacija koje koriste backend podrsku.
* Raznoliki nivoi tehnicke pismenosti.

Ocekivanja:

* Sigurnost I brzi odziv prilikom kontaktiranja sa backendom.
* Pouzdanost u radu i smanjenje vremena cekanja.

## Operativno okruzenje

**Platforma:**

Softver ce biti hostovan na serverima koje pokrece Linux operativni sistem. Serveri ce se nalaziti u Srbiji na jednoj lokaciji. Sto se tice samih korisnika, API ce biti dostupan za sva tri veca operativna sistema, a to su: Windows, MacOS, Linux sistemi. Jezici u kojima ce API biti dostupan je Java I C# za desktop I web aplikacije, Kotlin – za Android aplikacije, Swift – za IOs aplikacije I Flutter.

**Operativni Sistemi (Verzije):**

Softver ce biti kompatibilan sa sledecim operativnim sistemima:

Windows (verzije 10 i kasnije),

Linux (razlicite distribucije),

macOS (Big Sur i kasnije).

**Geografske Lokacije:**

Korisnici: U najranijoj fazi lansiranja se ocekuju korisnici prvenstveno sa prostora balkana, a sa kasnijim razvitkom I sirom sveta.

Serveri i Baze Podataka: Serveri ce u pocetku biti samo u Srbiji, dok ce sa porastom broja korisnika biti prosireni I na nove lokacije.

## Ogranicenja u projektovanju i primeni

**Korporativne i regulatorne politike:**

Programeri ce biti ograniceni regulatornim politikama u vezi sa bezbednoscu podataka i zastitom privatnosti. Ovo ukljucuje pridrzavanju standarda kao sto su GDPR, u zavisnosti od vrste podataka sa kojima se manipulise I na kojim geografskim podrucjima se radi sa podacima (Evropa I Amerika imaju razlicite zakone o zastiti podataka korisnika).

**Ogranicenja hardvera:**

Vreme odziva: Svaki zahtev ka bazi mora da se izvrsi u odredjenom vremenskom okviru kako bi se zadovoljile potrebe klijenata.

Memorijski zahtevi: Softver mora efikasno da upravlja memorijom kako bi se podaci sto efikasnije skladistili.

Skaliranje: Sistem mora da detektuje kada se obim saobracaja ka bazi povecava kako bi blagovremeno povecao resurse I omogucio nesmetan I brz pristup bazi.

**Integracija sa drugim aplikacijama I radnim okvirima**:

Integracija sa spoljnim aplikacijama moze biti ogranicena tehnickim specifikacijama tih aplikacija, ukljucujuci pdorsku za odredjene API-je ili protokole komunikacije.

Programeri ce biti ograniceni u izboru tehnologija i alata koje koriste.

**Zahtevi ili ogranicenja programskog jezika**:

Odredjeni delovi sistema mogu biti napisani u odredjenim programskim jezicima koji su prethodnoodredjeni zahtevima, ogranicavajuci slobodu izbora programskih jezika.

**Odrzavanje kompatibilnosti:**

Odrzavanje kompatibilnosti sa razlicitim operativnim sistemima i verzijama moze predstavljati ogranicenje u projektovanju i primeni softvera.

**Raspolozivost resursa:**

Ogranicenja u resursima, kao sto su budzet, ljudstvo i infrastruktura, mogu ograniciti obim i kompleksnost implementacije.

## Pretpostavke i zavisnosti

**Pretpostavke:**

* Stabilnost tehnologija I radnih okvira koji se koriste:

Pretpostavka: Tehnologije koje se koriste u razvoju ce ostati stabilne i podrzane tokom trajanja projekta.

Implikacije: Ako dodje do znacajnih promena u tehnologijama, verovatno ce biti neophodne I promene u implementaciji.

* Saradnja izmedju timova:

Pretpostavka: Timovi ce efikasno saradjivati i komunicirati tokom razvoja.

Implikacije: Problemi u komunikaciji mogu dovesti do kasnjenja ili nesporazuma u implementaciji.

* Dostupnost resursa:

Pretpostavka: Potrebni resursi, kao sto su serveri ili licencirani softver, bice dostupni prema planu.

Implikacije: Ogranicenja u dostupnosti resursa mogu uticati na razvoj.

**Zavisnosti:**

* Eksterni API-ji i servisi:

Zavisnost: Ispravno funkcionisanje eksternih API-ja i servisa na koje se aplikacija oslanja.

Implikacije: Ako eksterni servisi postanu nedostupni ili promene svoje specifikacije, potrebno je brzo reagovati I prilagoditi se.

* Odobrenje od regulatornih organizacija:

Zavisnost: Odobrenje od regulatornih organa u vezi sa uskladjivanjem sa standardima ili zakonima.

Implikacije: Odlaganje odobrenja moze uticati na planiranje pustanja proizvoda u upotrebu.

* Pravilno funkcionisanje trenutne infrastrukture:

Zavisnost: Da trenutna infrastruktura, na kojoj se oslanja projekat, ispravno funkcionise.

Implikacije: Problemi sa infrastrukturom mogu dovesti do prekida u radu ili gubitka podataka.

# Funkcije sistema

## Kreiranje aplikacije

### Opis

Svaki klijent koji odluci da koristi nasu platformu za svoju aplikaciju je u obavezi da napravi profil na nasoj platformi. Standardno, unose se neophodni podaci kao sto su email, korisnicko ime i lozinka. Po kreiranju svog profila neophodno je potvrditi identitet klikom na link koji ce stici na mejl klijenta nakon toga klijent je u mogucnosti da registruje svoju aplikaciju i koristi nase usluge za brzo postavljanje backend-a.

### Sekvence stimulusa/odgovora

1. Korisnik odlucuje da koristi nasu platfomu.
2. Korisnik odlazi na sajt nase platforme.
3. Korisnik se registruje na nas sajt.
4. Korisnik potvrdjuje svoj identitet.
5. Korisnik regtistruje svoju aplikaciju na nasoj platformi.
6. Korisnik moze da koristi nase usluge.

### Funkcionalni zahtevi

1. Korisnik moze da se registruje na nasoj platformi.
2. Korisnik moze da napravi aplikaciju na nasoj platformi.

## Pravljenje baze podataka i komunikacija sa istom

### Opis

U slucaju da je klijentu potrebna funkcionalnost za citanje i pisanje podatak iz i u bazu podataka, on je u mogucnosti da za svoju kreiranu aplikaciju na nasoj platformi napravi bazu podataka. Nakon toga koristeci nas SDK moze da implementira funkcionalnost komunikacije sa bazom podataka u svojoj aplikaciji.

### Sekvence stimulusa/odgovora

1. Korisnik se prijavljuje na svoj nalog na nasoj platformi.
2. Korisnik otvara panel sa kreiranim aplikacijama
3. Korisnik bira aplikaciju u koju zeli da integrise funkcionalnost baze podataka
4. Korisnik pravi bazu podataka I podesava njenu strukturu
5. Korisnik preuzima bezbednosne kredencijale koje generise server
6. Korisnik preuzima nas SDK
7. Korisnik implmenetira komunikaciju sa bazom podataka u svojoj aplikaciji.

### Funkcionalni zahtevi

1. Korisnik moze da napravi bazu podataka za odredjenu aplikaciju
2. Korisnik moze da podesi strukturu novokreirane baze podataka
3. Korisnik moze da pomocu SDK-a implementira komunikaciju sa bazom podataka

## Autentifikacija i Autorizacija

### Opis

U slucaju da je klijentu potrebna autentifikacija i autorizacija, sistem treba da pruzi i resenje za ovaj problem. U nekim slucajevima je potrebno autentifikovati korisnika koji pokusava da pristupi nekim podacima u bazi podataka kako bi se utvrdio identitet istog i odredilo da li moze da pristupi odredjenom resursu. U slucaju da je potrebna autorizacija neophodno je cuvati ulogu svakog registrovanog korisnika kako bi mu se omogucio pristup onim delovima koji su u skladu sa njegovom rolom.

### Sekvence stimulusa/odgovora

1. Korisniku je potrebno da ima autentifikaciju ili autorizaciju (ili oba) kao funkcionalnost u svojoj aplikaciji
2. Korisnik se loguje na svoj profil na nasem sajtu
3. Korisnik bira za koju aplikaciju zeli da implementira ovu funkcionalnost
4. Korisnik odlazi u odeljak za autentifikaciju I autorizaciju
5. Korisnik cita nasu dokumentaciju za implementaciju ove funkcionalnosti.
6. Korisnik preuzima nas sdk
7. Korisnik implementira autorizaciju I autentifikaciju u svojoj aplikaciji.

### Funkcionalni zahtevi

1. Korisnik skida SDK za autentifikaciju i autorizaciju
2. Korisnik integrise autentifikaciju i autorizaciju pomocu SDK-a

## Upravljanje korisnicima

### Opis

U slucaju da je klijentu potrebno upravljanje korisnickim nalozima, sistem treba pruzati i tu funkcionalnost. Ova funkcionalnost treba da pruza kreiranje profila, izmenu podataka vec registrovanih profila i brisanje postojeceg profila korisnika.

### Sekvence stimulusa/odgovora

1. Korisniku je potrebna funkcionalnost da upravlja korisnickim nalozima
2. Korisnik se loguje na svoj profil na nasoj platformi
3. Korisnik bira aplikaciju za koju zeli da podesi upravljanje korisnickim nalozima
4. Korisnik odlazi u odeljak za korisnike naloge
5. Korisnik cita dokumentaciju o integraciji upravljanja korisnickim nalozima u svojoj aplikaciji
6. Korisnik skida nas SDK za upravljanje korisnickim nalozima
7. Korisnik implementira funkcionalnost u svojoj aplikaciji

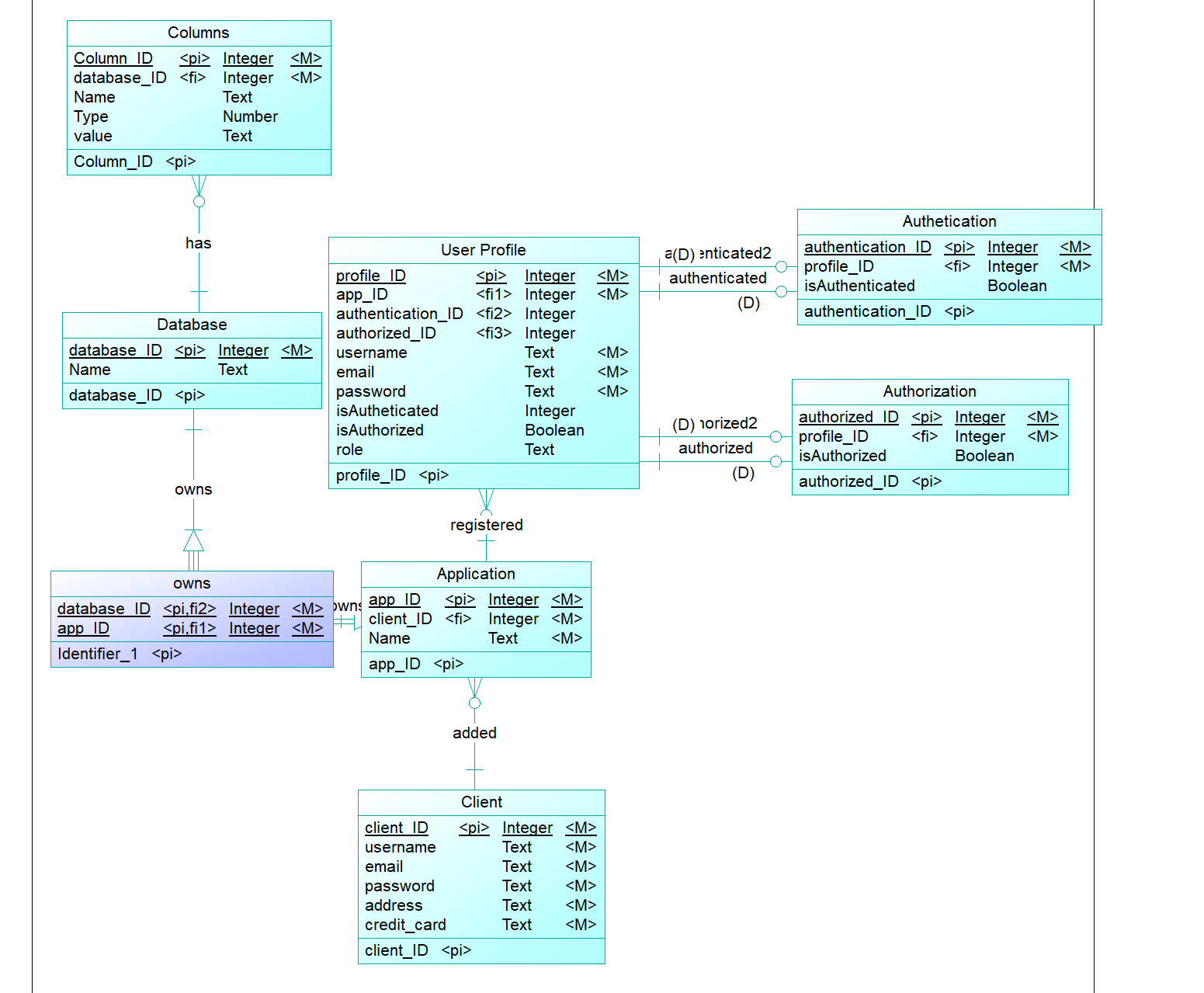
### Funkcionalni zahtevi

1. Korisnik skida SDK za upravljanje korisnickim nalozima
2. Korisnik pomocu SDK-a integrise upravljanje korisnickim nalozima u svoju aplikaciju

# Zahtevi za podatke

Ovaj odeljak opisuje razlicite aspekte podataka koje ce sistem koristiti kao ulazne parametre, obraditi na neki nacin ili kreirati izlaze.

## Logicki model podataka



## Recnik podataka

|  |  |  |  |
| --- | --- | --- | --- |
| Ime podatka | Opis | Tip | Duzina |
| Client | Klijent koji registruje aplikaciju na nasoj platformi. | -client\_ID <pk>  -username  -email  -password  -address  -credit\_card |  |
| Application | Aplikacija za koju ce klijent integrisati backend. | -app\_ID <pk>  -client\_ID  -Name |  |
| User Profiles | Korisnicki profili koje ce napraviti sami korisnici aplikacije prilikom registracije. | -Profile\_ID <pk>  -app\_ID  -authetication\_ID  -authorized\_ID  -username  -email  -password  -isAuthenticated  -isAuthorized  -role |  |
| Authentication | Ovde ce se cuvati status autentifikacije korisnika. | -Authentication\_ID  -profile\_ID  -isAuthorized |  |
| Authorization | Ovde ce se cuvati status autorizacije korisnika. | -authorized\_ID  -profile\_ID  -isAuthorized |  |
| Owns | Ovo je prelazni atribut koji povezuje tabelu Application i tabelu Database. | -database\_ID  -app\_ID |  |
| Database | Ovde ce se cuvati informacije o bazi podataka. | -database\_ID  -Name |  |
| Columns | Ovde ce se cuvati kolone koje sadrzi odredjena baza podataka kao i njihove tipove i vrednosti. | -Column\_ID  -database\_ID  -Name  -Type  -value |  |

## Izvestaji

Svi generisani izvestaji su u elektronskom formatu.

|  |  |
| --- | --- |
| Ime Izvestaja | Opis |
| Finansijski izvestaj | Ovaj izvestaj ce se generisati na kraju svakog meseca automatski od strane sistema. U ovom izvestaju ce biti moguce pregledati kompletno stanje sto se tice finansija. |
| Izvestaj saobracaja | Ovaj izvestaj ce se generisati svakog meseca i prikazivati kompletan saobracaj koji je bio upucen prema nasoj platformi, skaliranje i sve ostale metrike koje ce pomoci unapredjenju nase platforme. |
| Korisnicki izvestaj | Ovaj izvestaj se salje svakom korisniku ponaosob svakog meseca kako bi imao uvid u potrsonju resursa na nasoj platformi. |

# Zahtevi za spoljni interfejs

Ovaj odeljak pruza informacije koje ce osigurati da sistem pravilno obavlja komunikaciju sa korisnicima i eksternim hardverskim i softverskim elementima.

## Korisnicki interfejsi

**1. Registracija na platformi:**

Opis: Korisnicima se omogucava jednostavna i intuitivna registracija putem web interfejsa.

Karakteristike:

* Polja za unos osnovnih podataka (ime, prezime, e-posta).
* Validacija unetih informacija u realnom vremenu.
* Obavezan klik na potvrdni link koji se salje na e-postu za potvrdu identiteta.
* Jednostavan I intuitivan dizajn u skladu sa standardima.

**2. Upravljanje aplikacijama:**

Opis: Panel za upravljanje aplikacijama omogucava korisnicima da dodaju, uredjuju i brisu svoje aplikacije.

Karakteristike:

* Pregled svih registrovanih aplikacija.
* Opcije za dodavanje nove aplikacije.
* Mogucnost podesavanja postavki aplikacija.

**3. Kreiranje baze podataka za odredjenu aplikaciju:**

Opis: Interfejs za unos podataka potrebnih za povezivanje aplikacije sa bazom podataka

Karakteristike:

* Polja za unos osnovnih podataka o bazi I njenoj strukturi

**4. Autentifikacija i Autorizacija:**

Opis: Korisnicki interfejs za upravljanje autentifikacijom i autorizacijom.

Karakteristike:

* Postavljanje pravila autorizacije na nivou aplikacije.
* Pregled korisnika i njihovih privilegija.
* Opcije za definisanje korisnickih rola.

**5. Prikaz Podataka iz izabrane baze podataka:**

Opis: Interfejs za prikaz podataka koji se cuvaju u bazi.

Karakteristike:

* Jasan prikaz podataka u tabelarnom ili grafikonu formatu.
* Opcije za pretragu i filtriranje podataka.

**6. Sigurnost i prijavljivanje:**

Opis: Interfejs za upravljanje bezbednosnim postavkama i evidencijom prijava.

Karakteristike:

* Pregled bezbednosnih logova.
* Postavljanje pravila zastite i enkripcije.

**7. Help i Support:**

Opis: Pristup resursima za pomoc i podrsku korisnicima.

Karakteristike:

* Pristup cesto postavljanim pitanjima, live chat-u.
* Interaktivni vodici ili tutorijali za nove korisnike.

**8. Dashboard:**

Opis: Glavna kontrolna tabla koja prikazuje kljucne informacije o aplikacijama I njihovim performansama.

Karakteristike:

* Grafikoni i statistike o upotrebi.
* Pregled trenutnog statusa aplikacija.

**9. Obavestenja i Komunikacija:**

Opis: Sistem za slanje obavestenja i komunikaciju sa korisnicima.

Karakteristike:

* Obavestenja o promenama statusa aplikacije ili vaznim informacijama.
* Mogucnost postavljanja preferencija za obavestenja.

## Softverski interfejsi

Aplikacija ne zavisi ni od kakvih servisa ili usluga trece strane. Sve sto je razivjeno na platformi je napravljeno od nule za potrebe platforme.

## Hardverski interfejsi

**Fizicko povezivanje (kablovi, bezicne mreze):**

- Podrzani tipovi uredjaja: Razlicite vrste uredjaja povezanih mrezom.

- Podaci i kontrolne interakcije: Ethernet, Wi-Fi.

- Ulazi i Izlazi: Prenos podataka, komunikacija izmedju uredjaja.

- Problemi sa vremenom: Brzina prenosa, stabilnost veze.

**Operativni Sistemi (Windows, Linux, MacOS):**

Podrzani tipovi uredjaja: Racunari, serveri.

Podaci i kontrolne interakcije: Sistemski pozivi, upravljanje resursima.

Ulazi i izlazi: Naredbe, podaci iz fajlova.

Problemi sa vremenom: Odziv sistema, brzina operacija.

**Korisnicki uredjaji (racunari, pametni telefoni):**

Podrzani tipovi uredjaja: Racunari, pametni telefoni, tableti.

Podaci i kontrolne Interakcije: Web pregledaci, Firefly SDK.

Ulazi i izlazi: Korisnicki interfejs, HTTP zahtevi.

Problemi sa vremenom: Performanse uredjaja, brzina mrezne veze.

**Web Server i mrezna infrastruktura:**

Podrzani tipovi uredjaja: Racunari, serveri.

Podaci i kontrolne interakcije: Ethernet, Wi-Fi, HTTP/HTTPS protokoli.

Ulazi i izlazi: Zahtevi i odgovori preko mreze.

Problemi sa vremenom: Latencija mreze, brzina prenosa podataka.

## Komunikacioni interfejsi

**E-posta:**

Funkcije: Potvrda identiteta, obavestenja o registraciji.

Format poruke: Tekstualne poruke sa linkovima za potvrdu.

Sigurnost/Enkripcija: SSL/TLS zastita.

Brzina prenosa: Brza isporuka e-poste.

**Veb Pretrazivac:**

Funkcije: Pregledanje i interakcija sa web aplikacijom.

Format poruke: HTTP/HTTPS protokoli.

Sigurnost/Enkripcija: HTTPS zastita.

Brzina prenosa: Brza i pouzdana veza.

Sinhronizacija: Realno vreme, brza odzivnost.

**Mrezni Protokoli (Firefly SDK):**

Funkcije: Komunikacija izmedju korisnickih uredjaja i FireflyCloud servisa.

Format Poruke: JSON podaci prenose se putem HTTPS.

Sigurnost/Enkripcija: Firefly Authentication, HTTPS.

Brzina Prenosa: Efikasna komunikacija sa cloud servisima.

# Atributi kvaliteta

## Upotrebljivost

**UP1 –** Svaki korisnik mora pri pristupu platformi da odmah uvidi lakocu za registrovanje na istoj. Dugmici za logovanje i registraciju moraju biti upadljivi.

**UP2** – Na pocetnoj strani treba uproscenim modelom biti prikazanao sta nudi nasa platforma.

**UP3 –** Na pocetnoj strani za svaku uslugu mora odmah biti vidlji okvirni cenovnik za proizvoljne vrednosti, kako bi korisnik bez dodatne navigacije imao ideju o tome koliko bi ga kostala neka usluga.

**UP4 –** Na dashboard strani mora postojati jednostavan meni gde ce korisniku koji se registrovao na nasu platformu odmah biti jasno kom odlejku pristupa i gde moze da nadje podesavanja za odredjenu uslugu.

**UP5 –** U odeljku za izabranu uslugu ce na pocetku, pre bilo kakave postavke, biti prikazan kratak i najosnovnijij primer implementacije izabrane usluge.

## Performanse

**PER1 –** U zavisnosti od kompleksnosti upita i velicine informacija kojoj zeli da se pristupi, svaki upit mora da se izvrsi u datom vremenskom opsegu.

## Bezbednost

**SAF1 –** Autentifikacija i autorizacija moraju besprekorno da rade kako ne bi doslo do neovlascenog pristupa podacima.

**SAF2 –** Svi osetljivi podaci koji se cuvaju u bazi moraju da budu enkriptovani.

## Sigurnost

**SEC1 –** Korisnik mora da ima opciju da postavi granicu svog budzeta

**SEC2 –** Ukoliko potrosnja dodje do granice uslugu treba obustaviti.

## Efikasnost

**EFF1 -** Najmanje 10 posto kapaciteta procesora i memorije dostupne aplikaciji ne sme se koristiti pri planiranim uslovima najveceg opterecenja.

# Uslovi internacionalizacije i lokalizacije

Po lansiranju platforma mora da sadrzi srpski jezik sa opcijom za latinicu i cirilicu, posto je ciljno podrucje rada na lansiranju platforme balkan. Pored toga mora biti dostupna opcija da se izabere engleski jezik. Kada dodje do sirenja podrucja poslovanja, prosiricemo listu dostupnih jezika aplikacije na francuski, nemacki i spanski.

Dodatak A: Recnik pojmova

**HTTP –** Hyper Text Transfer Protocol

**HTTPS –** Hyper Text Transfer Protocol Secure

**JSON –** Javascript Object Notation