

**CARNET sustav za poruke**

Funkcionalna specifikacija

**Sadržaj**

[**Sadržaj** 2](#_Toc119418124)

[**1.** **Uvod** 3](#_Toc119418125)

[**2.** **Funkcionalni zahtjevi** 3](#_Toc119418126)

[2.1 Autentikacija 3](#_Toc119418127)

[2.2 Autorizacija 3](#_Toc119418128)

[**3.** **Komunikacija između korisnika** 4](#_Toc119418129)

[3.1 Općeniti zahtjevi 4](#_Toc119418130)

[3.2 Korisnici 5](#_Toc119418131)

[3.3 Tipovi korisnika 5](#_Toc119418132)

[3.4 Poruke 6](#_Toc119418133)

[3.5 Dretve 6](#_Toc119418134)

[4.1 Administracijsko sučelje 7](#_Toc119418135)

[4.2 Tehnološki zahtjevi 8](#_Toc119418136)

[4.3 API 8](#_Toc119418137)

[4.4 Komunikacija u realnom vremenu 8](#_Toc119418138)

[4.5 Pohrana podataka 8](#_Toc119418139)

[4.6 Sistemski zahtjevi 8](#_Toc119418140)

[4.7 Performanse 9](#_Toc119418141)

[4.8 Prijedlog strukture sustava 9](#_Toc119418142)

[Prijedlog REST API metoda 9](#_Toc119418143)

# **Uvod**

Hrvatska akademska i istraživačka mreža – CARNET planira razvoj platforme koja će omogućiti razmjenu poruka između korisnika pojedinih sustava iz CARNET-ovog ekosustava aplikacija. Ovaj dokument opisuje skup funkcionalnosti koja ta platforma treba sadržavati.

Radno ime sustava: CARNET sustav za poruke (u nastavku teksta „platforma“)

Sustav je zamišljen kao centralna platforma za izmjenjivanje i distribuciju tekstualnih poruka (eng. "messaging") unutar CARNET-ovog ekosustava aplikacija u inicijalnoj fazi implementacije, te glasovnu i video komunikacije u kasnijim fazama. Preko njega bi se obavljala razmjena poruka između korisnika svih CARNET-ovih aplikacija i sustava koji će podržavati navedenu funkcionalnost, uključujući e-Dnevnik, CARNET data i ostale CARNET-ove aplikacije koje se koriste u obrazovnom sustavu Republike Hrvatske. Korisnicima CARNET-ovih aplikacija platforma će omogućiti slanje i zaprimanje poruka na sučelju pojedinih aplikacija iz CARNET-ovog ekosustava ovisno o ulogama koje korisnici imaju u sustavu.

Platforma treba omogućavati:

* sve serverske (eng. "back-end") funkcionalnosti potrebne za ostvarivanje dvosmjerne tekstualne komunikacije u realnom vremenu između dvoje ili više korisnika (grupe korisnika), korištenjem WebSocket ili slične tehnologije
* slanje često korištenih tipova datoteka (primjerice .jpeg ili .pdf) kao priloga tekstualnoj komunikaciji
* administratorsko web-sučelje za upravljanje i nadzor sustava
* potrebnu modularnost/proširivost kako bi se u kasnijim nadogradnjama platforme mogla implementirati glasovna i video komunikacija

# **Funkcionalni zahtjevi**

## 2.1 Autentikacija

Platforma treba autenticirati zahtjeve na sljedeći način:

* od povezanih aplikacija korištenjem API ključeva jedinstvenih za svaku povezanu aplikaciju
* od strane korisnika korištenjem JWT tokena ([https://jwt.io](https://jwt.io/)) ili slične metode
* povezana aplikacija od platforme zahtijeva JWT/potpisni ključ/neki drugi podatak potreban za autentikaciju za korisnika prilikom pokretanja komunikacijske funkcionalnosti te ga prosljeđuje korisniku ukoliko je potrebno - primjerice, prilikom korištenja samostalnih front-end komponenti koje implementiraju komunikacijske funkcionalnosti

## 2.2 Autorizacija

Platforma mora autorizirati svaki poziv na API, WebSocket ili drugu podržanu metodu komunikacije s platformom. Korisnik smije pristupiti samo određenim API metodama koje su potrebne za funkcionalnosti dostupne korisnicima, dok povezane aplikacije smiju pristupiti svim metodama. Korisnik smije:

* vidjeti sve poruke iz dretvi (pojedinačnih ili grupnih) u kojima trenutno sudjeluje
* kreirati novu pojedinačnu dretvu s korisnicima s kojima mu je trenutno dozvoljeno koristiti komunikacijske funkcionalnosti (razrednici, ostali roditelji iz grupa u kojima trenutno sudjeluje, predmetni nastavnici i sl.)
* pristupiti osnovnim informacijama o sebi i korisnicima s kojima mu je trenutno dozvoljeno koristiti komunikacijske funkcionalnosti (razrednici, roditelji u grupama u kojima je korisnik voditelj i sl.). Pod osnovnim informacijama se podrazumijeva:
	+ ime i prezime
	+ škola i tip korisnika na školi (roditelj, nastavnik...)
	+ dostupnost za komunikaciju (trenutna dostupnost za komunikaciju i opcionalno vremenski interval definiran od strane korisnika u kojem je dostupan za komunikaciju)
* tipovi korisnika s odgovarajućim ovlastima smiju pristupiti web administracijskom sučelju

Povezane aplikacije smiju:

* dohvatiti podatke svih korisnika unutar platforme, koristeći OIB ili neki drugi atribut jedinstven za svakog korisnika
* upravljati podacima svih korisnika
* upravljati svim grupama korisnika (kreiranje, uređivanje i brisanje)

# **Komunikacija između korisnika**

3.1 Općeniti zahtjevi

Komunikacija između korisnika se odvija putem tekstualnih poruka („chat“) podijeljenih po dretvama (eng. „thread“). U kasnijim fazama razvoja planirana je glasovna i video komunikacija. Tekstualna poruka se sastoji od Unicode znakova, bez ograničenja dužine poruke. Pojedina poruka može imati priloge – binarne datoteke često korištenih tipova (slike, dokumenti i sl.). Prilozi mogu biti ograničene veličine i određenih tipova, podržane tipove i veličinu datoteka je potrebno definirati u dogovoru s CARNET-om.

Platforma treba podržavati slanje i primanje poruka u realnom vremenu (eng. „chat“) korištenjem WebSocket ili slične tehnologije, te slanje i primanje poruka korištenjem API-ja za korisnike/sustave koji nisu u mogućnosti koristiti odabranu tehnologiju za slanje/primanje u realnom vremenu. Također, za podržane klijente, platforma mora omogućavati slanje i primanje ostalih eventualno potrebnih podataka u realnom vremenu (primjerice, indikatori pročitanosti poruke).

Platforma treba podržavati obavještavanje povezanih sustava o pristiglim porukama, korištenjem definiranih API endpoint-ova na povezanim sustavima (primjerice, u svrhu slanja „push“ notifikacija iz povezanih sustava i sl.).

## 3.2 Korisnici

Platforma treba razlikovati korisnike po jedinstvenom internom ID-u na razini sustava, te osim navedenog ID-a, čuvati/pružati povezanim sustavima sljedeće podatke o korisnicima:

* OIB
* e-mail (opcionalno)
* Dodatni atributi iz drugih CARNET-ovih sustava
* tipovi korisnika za pojedinu ustanovu/razred
* dozvola za vidljivost je li korisnik trenutno dostupan za komunikaciju
* dostupnost korisnika za komunikaciju na razini platforme:
	+ nedostupan
	+ dostupan u određenom vremenskom intervalu/intervalima
	+ uvijek dostupan
* odobrenje za slanje potvrde pročitanosti poruka ostalim sudionicima dretvi
* koristi li neki od povezanih sustava i treba li u taj sustav/sustave slati obavijesti o novim porukama ako korisnik nije trenutno aktivan

Grupa korisnika se sastoji od više pojedinačnih korisnika koji mogu, sukladno dozvolama u platformi, međusobno komunicirati putem pojedinačnih dretvi ili jedne grupne dretve. Platforma razlikuje grupe po jedinstvenom ID-u te osim toga čuva sljedeće podatke o grupama:

* ID-ovi korisnika koji pripadaju grupi
* škola
* razred
* školska godina
* tip grupe (opcionalno)

## 3.3 Tipovi korisnika

Platforma treba razlikovati sljedeće tipove (razine ovlasti) korisnika:

* Roditelj
* Predmetni nastavnik
* Razrednik
* Stručni suradnik, pedagog
* Ravnatelj
* Centralni administrator sustava
* Helpdesk
* Učenik

Uloge se dodjeljuju od strane povezanih sustava/na administracijskom sučelju na razini škole (stručni suradnik, pedagog, ravnatelj) i razreda (razrednik, predmetni nastavnik i roditelj). Korisnik mora moći biti trenutno aktivan u više škola/razreda i imati odgovarajući tip korisnika za svaku pojedinu školu razred – primjerice, roditelji više djece koja pohađaju različite škole, stručni suradnici i nastavnici koji rade na više škola i sl.

## 3.4 Poruke

Poruka se sastoji od neograničenog broja Unicode znakova, te platforma osim samog teksta poruke mora čuvati sljedeće atribute:

* jedinstveni ID na razini platforme
* ID pošiljatelja (interni ID korisnika)
* (opcionalno) naslov
* datum i vrijeme slanja
* (opcionalno) prilozi poruke
* važnost poruke (poruke mogu imati više razina važnosti – normalna, povišena, visoka)

Privici porukama su binarne datoteke ograničene veličine, te platforma o njima čuva sljedeće atribute:

* ID priloga
* ID poruke uz koju je prilog povezan
* naziv i interna putanja datoteke na poslužitelju
* (opcionalno) naziv i putanja datoteke s pretpregledom priloga (eng. „thumbnail“)

Platforma mora omogućavati sigurnosno skeniranje priloga unutar same platforme ili prosljeđivanje priloga odvojenom sustavu na sigurnosno skeniranje prije prosljeđivanja priloga ostalim sudionicima komunikacije.

Svaka poruka uz gore navedene podatke mora spremati i indikatore pročitanosti od strane drugih sudionika (u skladu s dozvolama korisnika iz točke 3.2):

* za pojedinačnu komunikaciju indikator pročitanosti od strane druge osobe
* za grupnu komunikaciju indikator broja sudionika grupe koji su pročitali poruku

## 3.5 Dretve

Dretva („thread“) je skup poruka između dvoje ili više korisnika. Može ju inicirati korisnik (s drugim korisnicima sukladno ovlastima u sustavu) ili povezani sustavi sukladno definiranim pravilima. Početna postavka u sustavu treba biti da je iniciranje dretve omogućeno samo od strane razrednika prema roditeljima. Potrebno je predvidjeti posebnu opciju kojom razrednik dozvoljava ostalim sudionicima slanje poruka.

 Primjerice:

* na početku školske godine automatski se kreira grupa korisnika i pripadajuća grupna dretva s razrednikom i roditeljima djece pojedinog razreda pri čemu roditelj treba potvrditi da ga se doda u grupu
* prilikom dodavanja kombinirane grupe učenika u e-Dnevniku isti kreira grupu i dretvu s voditeljem kombinirane grupe i roditeljima djece (kombinirana grupa je skup djece iz različitih razreda koji zajedno pohađaju nastavu nekog predmeta, primjerice izvannastavnih aktivnosti), pri tome roditelj treba potvrditi da ga se može dodati u grupu

Platforma treba omogućavati sljedeće funkcionalnosti vezane uz dretve :

* kreiranje i uređivanje dretvi:
	+ kreiranje pojedinačnih dretvi od strane razrednika ili grupnih od strane povezanih sustava/na administracijskom sučelju platforme
	+ uređivanje dretve: promjena naziva
	+ dodavanje i uklanjanje korisnika u grupu/grupnu dretvu od strane voditelja grupe ili povezanih aplikacija (primjerice, dodavanje novog člana grupe kod naknadnog uključivanja djeteta u kombiniranu grupu)
* dozvola za pisanje u grupne dretve – voditelj grupe ima opciju definiranja mogu li svi korisnici slati poruke u grupnu dretvu ili samo voditelj
* dozvola za slanje priloga u grupne dretve – voditelj grupe ima opciju definiranje mogu li svi korisnici slati priloge ili samo voditelj
* dohvat svih ili određenih poruka putem REST API-ja
* skrivanje pojedinih dretvi iz pregleda i isključivanje notifikacija o novim porukama za pojedine dretve
* primanje novih poruka u realnom vremenu (eng. “push”) ako je korisnik trenutno aktivan – odnosi se na slanje novih poruka u korisnikov preglednik preko WebSocket-a ili druge korištene tehnologije, ako korisnik trenutno sudjeluje u komunikaciji (ima otvoreno sučelje koje komunicira s platformom)
* obavještavanje povezanih sustava o novim porukama unutar pojedine dretve, ako korisnik trenutno nije aktivan – u svrhu slanja notifikacija iz povezanih sustava prema potrebi. Povezani sustavi će implementirati standardiziranu HTTP metodu koju će platforma pozivati prilikom pristizanja novih poruka, ako za korisnika ima podatak da koristi neki od povezanih sustava i taj povezani sustav treba obavještavati o novim porukama.
* slanje i preuzimanje datoteka kao prilog porukama
* indikatori pročitanosti poruke od strane druge osobe/drugih sudionika grupne dretve (ako su drugi sudionici dali suglasnost za vidljivost potvrda pročitanosti poruka)

## 4.1 Administracijsko sučelje

Administracijsko sučelje platforme je web-aplikacija koju koriste ovlašteni djelatnici CARNET-a u svrhe nadzora i osnovne administracije funkcionalnosti platforme. Omogućava pretraživanje, pregled i upravljanje podacima o pojedinim korisnicima te grupama korisnika.

Ovlašteni djelatnici se prijavljuju na sustav korištenjem CARNET login sustava (OAuth provider korišten za prijavu na većinu CARNET-ovih sustava i usluga).

Administracijske funkcionalnosti platforme uključuju:

* pretraživanje korisnika po OIB-u, jedinstvenom identifikatoru na razini platforme ili mail adresi
* pregled i uređivanje podataka o pojedinim korisnicima i grupama
* definiranje novih korisnika i grupa
* dodavanje i uklanjanje ovlaštenih CARNET djelatnika

## 4.2 Tehnološki zahtjevi

Platforma se sastoji od sljedećih dijelova:

* API koji izlaže sve potrebne metode
* Pristupna točka za real-time komunikaciju s klijentima
* Sustav za trajnu pohranu (baze podataka)
* Opcionalno, sustav za privremenu pohranu često korištenih podataka (eng. „cache“)

## 4.3 API

API treba biti implementiran korištenjem REST arhitekturalnog obrasca i komunicirati korištenjem HTTP protokola. Prilikom komunikacije putem HTTP protokola trebaju se koristiti standardni HTTP glagoli i statusni kodovi za odgovore na zahtjeve (<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>). Podaci između API-ja i povezanih sustava/klijenata se trebaju razmjenjivati korištenjem JSON formata (<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>). API mora biti dokumentiran korištenjem OpenAPI specifikacije (<https://swagger.io/specification> ) u JSON ili YAML formatu.

## 4.4 Komunikacija u realnom vremenu

Komunikacija u realnom vremenu mora biti implementirana korištenjem WebSocket (<https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API> ) ili druge slične tehnologije koju podržavaju moderni internetski preglednici na PC/Mac/Linux računalima, mobilnim telefonima i tablet uređajima. Internetski preglednici koji moraju podržavati odabranu tehnologiju uključuju trenutno podržane Google Chrome, Microsoft Edge, Mozilla Firefox i Apple Safari.

## 4.5 Pohrana podataka

Pohrana podataka se vrši unutar relacijske baze podataka (npr. PostgreSQL ili slične). Opcionalno, poruke se mogu spremati u dinamičku („NoSQL“) bazu podataka (primjerice, MongoDB). Bazu podataka koja sadrži poruke mora biti moguće horizontalno skalirati na više poslužitelja. Poruke unutar sustava se čuvaju minimalno 2 godine. Maksimalna veličina relacijske baze može biti 500GB.

## 4.6 Sistemski zahtjevi

Sve komponente sustava, osim onih za trajnu pohranu podataka, moraju moći funkcionirati unutar Docker ili drugog kompatibilnog standarda za aplikacijske kontejnere, te biti dostupne za „deployment“ na Kubernetes. Sve komponente sustava će se izvršavati na CARNET-ovoj infrastrukturi, te će svi pripadajući podaci koji zahtijevaju trajnu ili privremenu pohranu biti pohranjeni na CARNET-ovoj infrastrukturi.

## 4.7 Performanse

Prije puštanja platforme u produkcijsko testiranje, provodit će se testovi opterećenja (eng. „load test“). Testovi će se provoditi primjenom alata za automatizirano testiranje – k6, JMeter i sličnih. Platforma mora biti optimizirana za istovremeni rad većeg broja korisnika, te ispunjavati sljedeće uvjete (na prikladno skaliranom hardveru):

* prosječno vrijeme odziva svih API metoda mora biti manje od 100 milisekundi prilikom 500 tisuća istovremeno aktivnih korisnika koji šalju jednu poruku svakih 10 sekundi

## 4.8 Prijedlog strukture sustava

U nastavku se nalazi prijedlog strukture sustava - platforma mora implementirati navedene funkcionalnosti, ali ne mora potpuno slijediti strukturu u nastavku.

### Prijedlog REST API metoda

|  |  |  |
| --- | --- | --- |
| **HTTP metoda** | **URI** | **Funkcionalnost** |
| POST | /authorize | dohvat JWT tokena za korisnika od strane povezane aplikacije |
|  |  |  |
| GET | /users/{OIB} | dohvat informacija o korisniku |
| POST | /users | dodavanje novog korisnika |
| PATCH | /users/{OIB} | uređivanje informacija o korisniku |
| DELETE | /users/{OIB} | brisanje korisnika |
|  |  |  |
| POST | /groups/create | dodavanje nove grupe |
| GET | /groups/{groupID}/members | dohvat informacija o članovima grupe |
| POST | /groups/{groupID}/members | dodavanje članova u grupu |
| GET | /groups/{groupID}/members/{OIB} | dohvat informacija o članu grupe (osobni podaci, trenutno dostupan) |
| DELETE | /groups/{groupID}/members/{OIB} | uklanjanje člana iz grupe |
| GET | /groups/search | pretraživanje grupa u sustavu |
|  |  |  |
| POST | /threads | dodavanje nove dretve |
| GET | /threads/{OIB} | dohvat svih dretvi u kojima korisnik sudjeluje. Moguće filtriranje po GET parametrima |
| POST | /threads/{threadID}/reply | slanje nove poruke na dretvu |
| PATCH | /threads/{threadID}/message/{messageID}/{OIB}/markRead | označava poruku pročitanom od strane pojedinog korisnika |
| PATCH | /threads/{threadID}/{OIB}/markRead | označava sve poruke unutar dretve pročitanima od strane korisnika |
| DELETE | /threads/{threadID}/message/{messageID} | brisanje odabrane poruke |
| GET | /threads/{threadID}/{OIB}/from/{messageID} | dohvaća sve poruke za korisnika unutar dretve od odabrane poruke (cursor) |
| GET | /threads/{threadID}/{OIB}/unread | dohvaća sve nepročitane poruke za korisnika unutar dretve |
| GET | /threads/{threadID}/{OIB}/unread/count | dohvaća broj nepročitanih poruka za korisnika unutar dretve |