Razvoj web aplikacije za upravljanje procesnim modelima

Starčić, Toni

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:247287

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-16**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli

Fakultet informatike

TONI STARČIĆ

RAZVOJ WEB APLIKACIJE ZA UPRAVLJANJE PROCESNIM MODELIMA

Diplomski rad

Pula, rujan, 2022. godine

Sveučilište Jurja Dobrile u Puli

Fakultet informatike

TONI STARČIĆ

RAZVOJ WEB APLIKACIJE ZA UPRAVLJANJE PROCESNIM MODELIMA

Diplomski rad

JMBAG: 0303055459, redovni student

Studijski smjer: Sveučilišni diplomski studij Informatika

Predmet: Web aplikacije

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: doc. dr. sc Nikola Tanković

Pula, rujan, 2022. godine

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani **Toni Starčić**, kandidat za magistra **informatike** ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

U Puli, rujan 2022. godine

IZJAVA O KORIŠTENJU AUTORSKOG DIJELA

Ja, **Toni Starčić** dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom **„Razvoj web aplikacije za upravljanje procesnim modelima“** koristi tako da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama. Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, rujan 2022. godine

Potpis

Zahvala

Hvala profesorima doc. dr. sc Nikoli Tankoviću i izv. prof. dr. sc Darku Etingeru na svemu u ovih pet godina studija. Hvala djevojci što me trpi. Hvala roditeljima i mojoj dragoj sestri na potpori i razumijevanju.

SADRŽAJ

UVOD 1

1. POSLOVNI PROCES, BPM, BPMN 2

1.1. POSLOVNI PROCES 2

1.2. UPRAVLJANJE PROCESIMA 2

1.3. DEFINIRANJE PROCESA 2

2. ZNAČAJKE IMPLEMENTIRANOG SUSTAVA 4

2.1. PREGLED SUSTAVA 4

2.2. DINAMIČKI PRIKAZ SUSTAVA 4

2.3. STATIČKI PRIKAZ SUSTAVA 6

2.4. PRIKAZ FUNKCIONALNOSTI SUSTAVA 7

2.4.1. STRANICA PROCESA 7

2.4.2. STRANICA SERVISA 17

2.4.3. STRANICA MODELERA 20

3. OPIS KORIŠTENIH TEHNOLOGIJA 35

3.1. GENERALNI PRIKAZ 35

3.2. VUE.JS 35

3.3. VUETIFY 36

3.4. AIOHTTP 37

3.5. PONY ORM 37

3.6. POSTGRESQL 37

3.7. BPMN-JS 38

3.7.1. BPMN-MODDLE 38

3.7.2. DIAGRAM-JS 38

3.7.3. JEZGRENI SERVISI 39

3.7.4. POMOĆNI SERVISI 39

4. MOGUĆA POBOLJŠANJA 40

4.1. PREGLED SUSTAVA 40

4.2. DINAMIČKI PRIKAZ SUSTAVA 41

4.3. STATIČKI PRIKAZ SUSTAVA 42

4.4. OPIS DODATNIH TEHNOLOGIJA 43

ZAKLJUČAK 45
SAŽETAK 46
ABSTRACT 47
LITERATURA 48
PRILOG 49

UVOD

Tema ovog rada je implementacija web aplikacije čija je svrha upravljanje, nadzor i modeliranje poslovnih procesa definiranih modelima norme BPMN (eng. Business Process Model Notation). Aplikacija omogućuje definiranje podskupa BPMN notacije i to u dovoljnoj mjeri kako bi se temeljni slučaj uporabe proveo od početka do kraja pri razvoju aplikacije.

Temeljni korisnički zahtjev jest taj da se svaki procesni model definiran u aplikaciji može cjelovito definirati i da je upravljanje samim sustavom intuitivno te jednostavno za koristiti. Korisnik mora moći definirati web servise koji se koriste u samoj definiciji dijagrama kod zadataka koji se izvode postojećim sustavima. Servisi se promatraju kao crna kutija (eng. Black-box) zbog jednostavnosti pri implementaciji aplikacije. Osim mogućnosti kreiranja servisa, mora ih se moći ažurirati i brisati. Što se tiče poslovnih procesa, mora postojati funkcionalnost uvoza, izvoza, aktivacije, deaktivacije, modeliranja, brisanja i ažuriranja.

Na temelju opisanih zahtjeva web aplikacija je implementirana s klijentske strane u modernom i progresivnom aplikacijskom okviru Vue.js, poslužiteljska strana u asinkronom okviru jezika Python, baziranom na modulima Asyncio i *aiohttp*. Kao bazu podataka odabrana je robusna relacijska baza PostgreSQL.

1. POSLOVNI PROCES, BPM, BPMN

1.1. POSLOVNI PROCES

Pragmatično gledajući, poslovni proces je nešto što se izvodi u jednoj organizaciji [1]. No, isto tako poslovni proces se može promatrati kao skup koraka za koje je potrebno odrediti trajanje i potrebnu alokaciju resursa ili prema fundamentalnoj definiciji poslovni proces skup povezanih aktivnosti i odluka, koji se izvodi na vanjski podražaj radi ostvarenja organizacijskih ciljeva, određenog je trajanja i troši određene resurse pretvarajući ih u specifične proizvode ili usluge važne za kupce, tj. korisnike [2].

1.2. UPRAVLJANJE PROCESIMA

Među nekim mnogim poznatim procesno-orijentiranim metodologijama, odnosno disciplinama kao što su TQM (eng. Total Quality Management), OM (eng. Operations Management), Lean i Six Sigma. Razvijen je BPM (eng. Business Process Management) od strane konzorcija OMG (eng. Object Management Group) [3], a definiran je kao disciplina koja se koristi različitim metodama za otkrivanje, modeliranje, analizu, mjerenje, poboljšanje i optimizaciju poslovnih procesa [4]. Budući da je jedan od organizacijskih ciljeva jest optimalno izvršavanje poslovnih procesa, dakle efikasno i efektivno. Problem kod unaprijeđenja poslovnih procesa je taj što se ne može eksperimentirati u realnom vremenu, već je nužno koristiti vjernu ali umanjenu verziju stvarnog sustava – tzv. model poslovnog procesa [2]. Kako bi se izbjegla neistoznačnost, utvrđena je norma, tj. standard prema kojem se ti poslovni procesi modeliraju. Ime tog standarda jest BPMN (eng. *The Business Process Model and Notation*).

1.3. DEFINIRANJE PROCESA

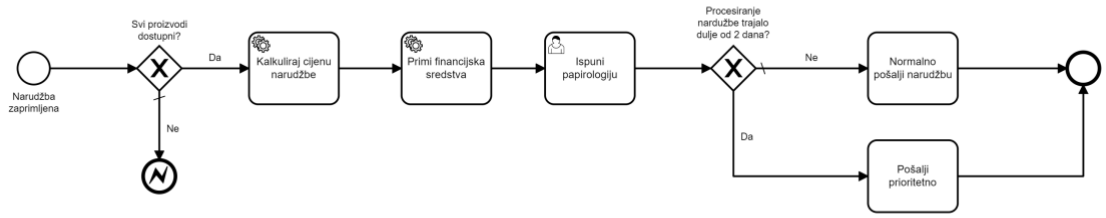
Specifikacija BPMN notacije je poduža, odnosno opširna. Generalno gledano BPMN objekti se mogu ugrubo podijeliti u četiri velike kategorije, a to su: objekti toka, spojni i hijerarhijski elementi te artefakti [5].

1. Objekti tijeka:
    a. Događaji (eng. Event) su elementi koji dodatno označuju nešto što se dogodilo,
    b. Aktivnosti (eng. Activities) su oni elementi koji indiciraju potrebni posao koji treba odraditi i
    c. Skretnice (eng. Gateway) upravljaju putem izvršavanja procesa.

2. Spojni elementi:
    a. Sekevencijni (eng. Sequential) grafički prikazuju smjer kretanja procesa punom linijom i strijelom,
    b. Asocijativni (eng. Association) grafički prikazani kao točkaste strijele, denotiraju srodnost elemenata i
    c. Poruke (eng. Messages) prikazuju dijeljenje poruka između aktera.

3. Hijerarhijski elementi:
    a. Polja (eng. Pool) apstrahira hijerarhijski značaj između aktera procesa i
    b. Trake (eng. Lanes) dodatno dijele aktere u manje jedinice.

4. Artefakti:
    a. Objekti (eng. Data objects) dodatno naznačuju podatke proizvedene u procesu,
    b. Anotacije (eng. Annotation) pomnije pojašnjavaju elemente i
    c. Grupe (eng. Groups) grupiraju elemente.



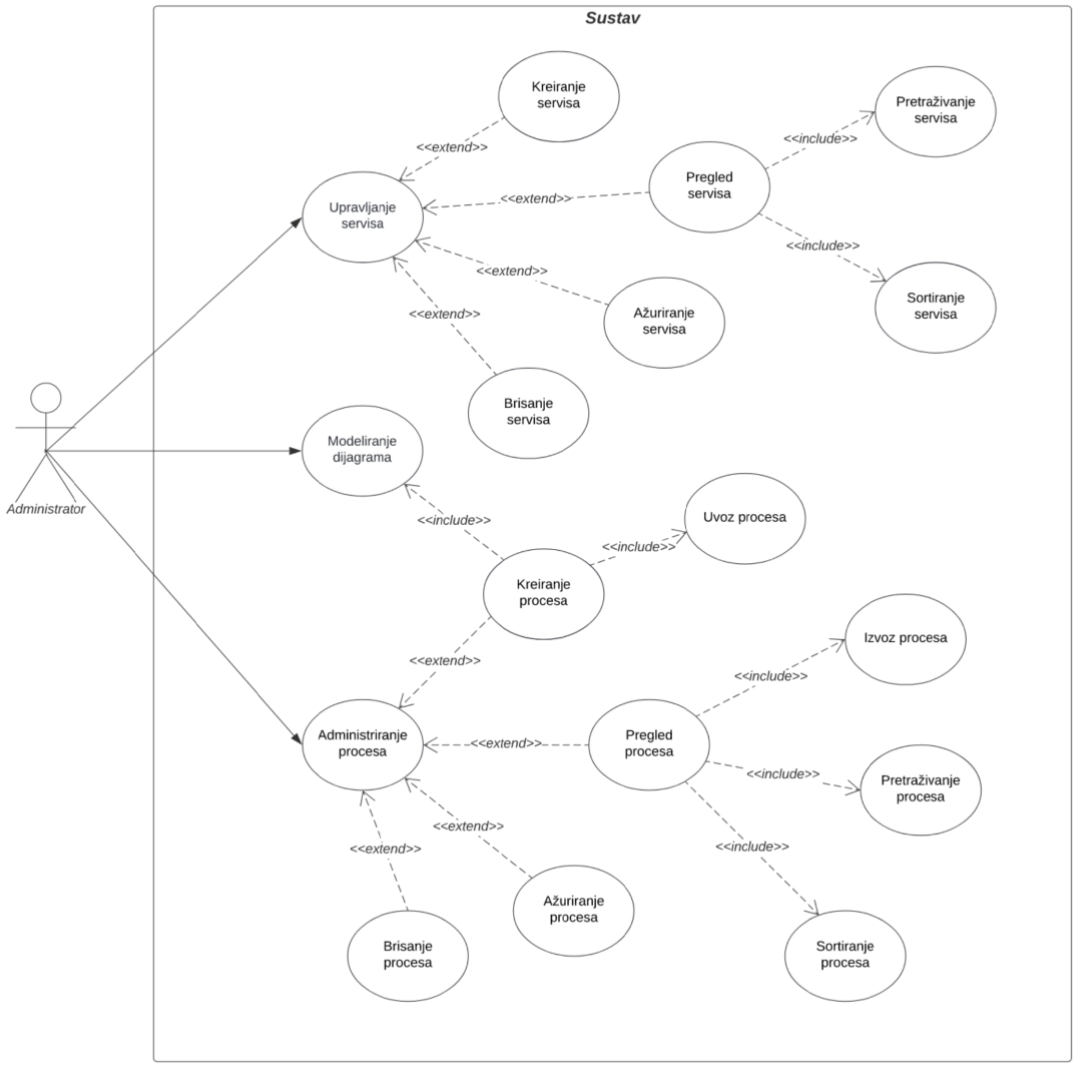
Slika 1: Primjer korištenja BPMN notacije u modeliranju poslovnih procesa (Izvor: Autor)

2. ZNAČAJKE IMPLEMENTIRANOG SUSTAVA

2.1. PREGLED SUSTAVA

U ovom odjeljku bit će prikazan dijagram ponašanja koji prikazuje dinamičan pogled na sustav, tzv. UML dijagram obrasca uporabe (eng. Use Case) i UML klasni dijagram koji prikazuje statičku stukturu sustava [6]. U podsekciji prikaz funkcionalnosti sustava bit će prikazane značajke sustava i način upravljanja sustavom.

2.2. DINAMIČKI PRIKAZ SUSTAVA

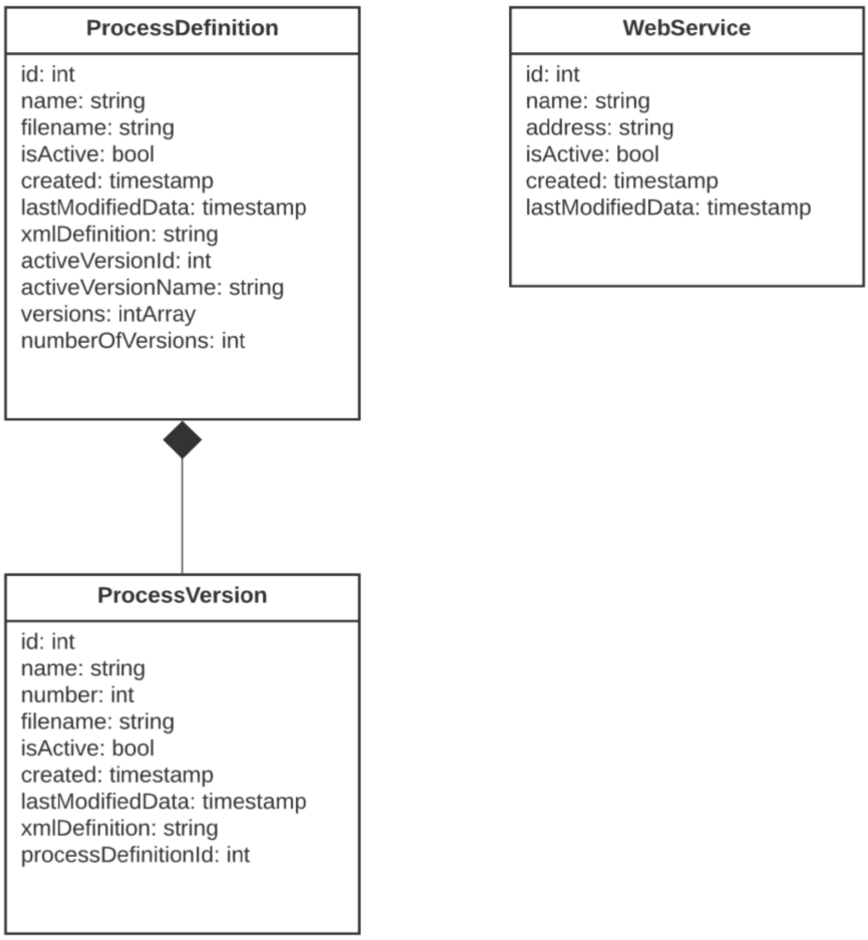


Slika 2: UML Dijagram obrasca uporabe (Izvor: Autor)

Akter (eng. Actor) prikazanog dijagrama UML obrasca uporabe (eng. Use Case) je Administrator. Naime, Administrator ima tri glavne funkcionalnosti:

1. Upravljanje servisima – na raspolaganju su mu kreiranje, pregled, ažuriranje i na kraju brisanje servisa. Dakle, u osnovi CRUD (eng. Create/Read/Update/Delete) operacije;
2. Modeliranje dijagrama – pomoću podsustava modeler korisnik ima mogućnost modeliranja poslovnih procesa;
3. Administriranje procesa – Administrator ima mogućnost CRUD operacija nad procesima, a kreiranje procesa se svodi na uvoz (eng. Import) procesa pomoću dediciranog podsustava zaduženog za uvoz ili modeliranje pomoću podsustava zaduženog za modeliranje;

2.3. STATIČKI PRIKAZ SUSTAVA



Slika 3: UML Klasni dijagram (Izvor: Autor)

Klasni dijagram prikazuje tri entiteta koja se pojavljuju u aplikaciji:

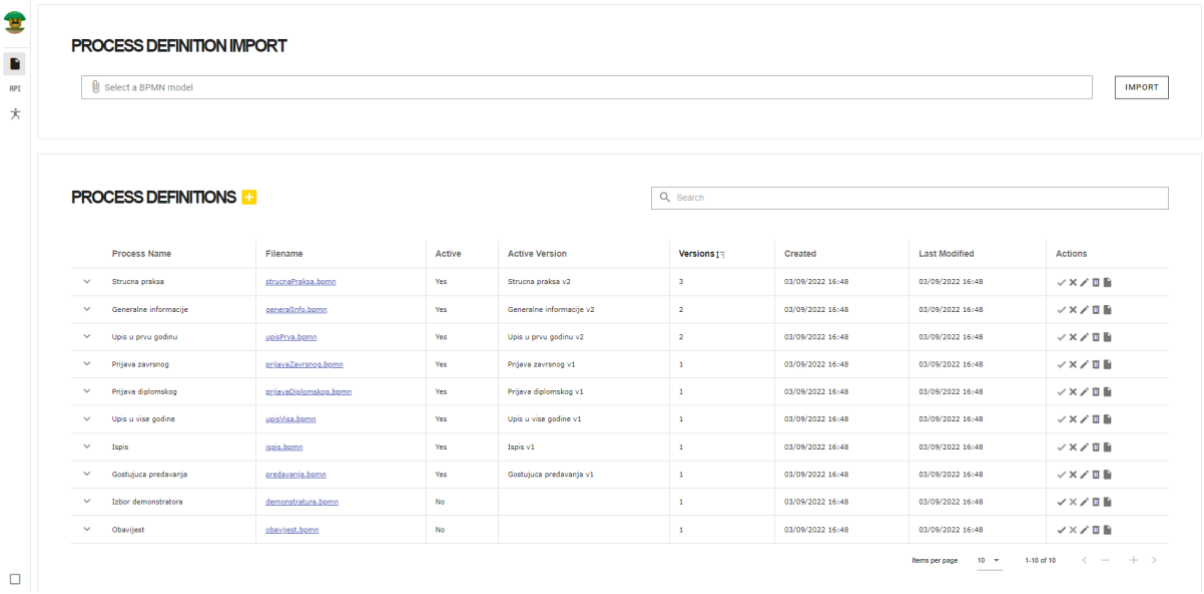
1. Procesna definicija (eng. Process definition) – početna verzija procesa;
2. Procesna verzija (eng. Process version) – različite verzije jedne procesne definicije;
3. Web servis (eng. Web service) – web servisi su entiteti koji se mapiraju na korisničke poslove (eng. User Task) unutar jednog procesa;

Postoji hijerarhijski odnos između procesne definicije i verzije, a entitet web servis postoji neovisno o procesima. Kreiranjem procesa nastaje jedna procesna definicija, bez obzira na način na koji je proces kreiran budući da postoje dva netom opisana. Definicija se može

objasniti koristeći objektno-orijentiranom analogijom, gdje se definicija može smatrati klasom, a verzija objektom, odnosno instancom klase. Verzije se koriste radi verzioniranja procesa. Značajke u korištenju procesnih definicija, odnosno verzija su gotovo pa jednake. Iz tog razloga većina operacija u radu sa sustavom bit će prikazana u radu s procesnim definicija.

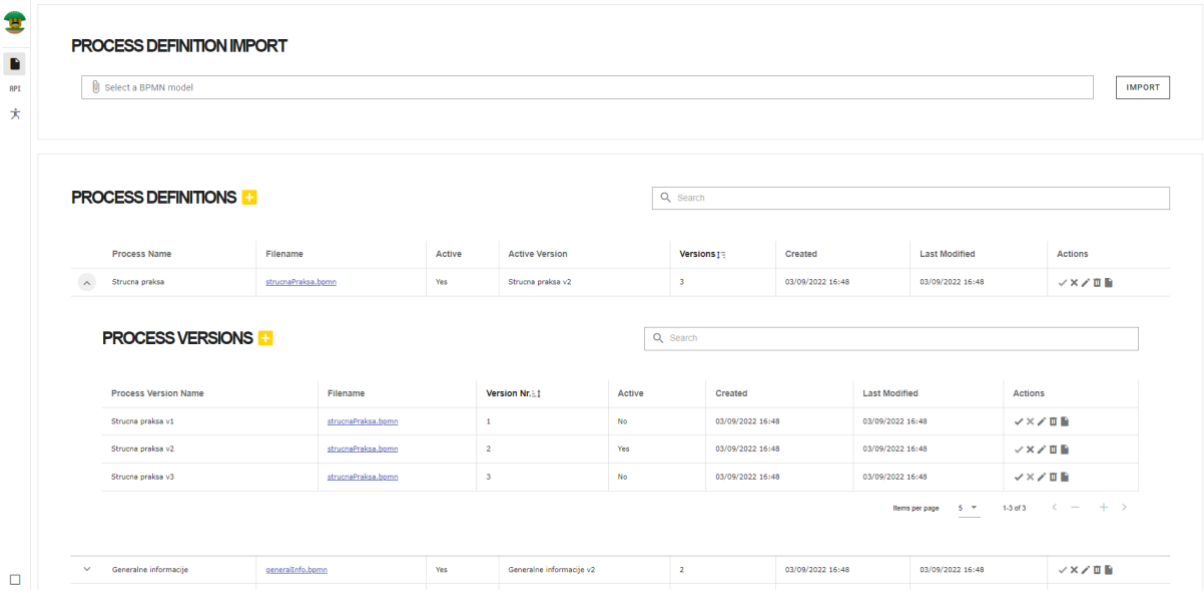
2.4. PRIKAZ FUNKCIONALNOSTI SUSTAVA

2.4.1. STRANICA PROCESA



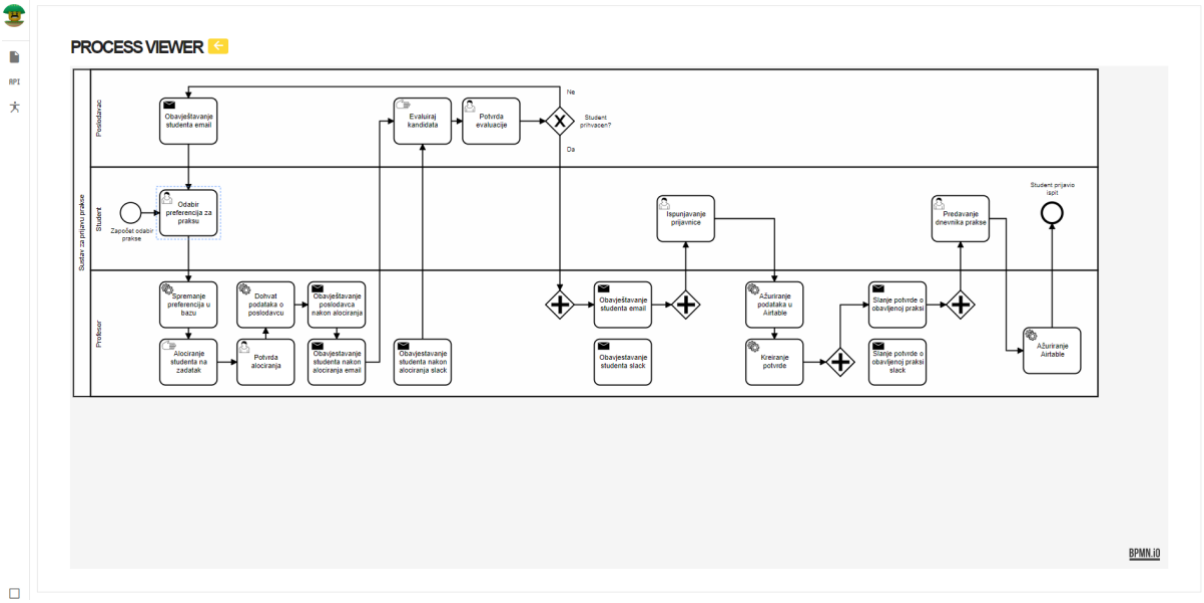
Slika 4: Prikaz sučelja procesnih definicija (Izvor: Autor)

Na jednoj od glavnih stranica zvanoj procesi, korisnik ima pregled svih procesa koji se nalaze u sustavu. Procesi su prikazani u tabličnom obliku što rezultira preglednošću i jednostavnosti uporabe.



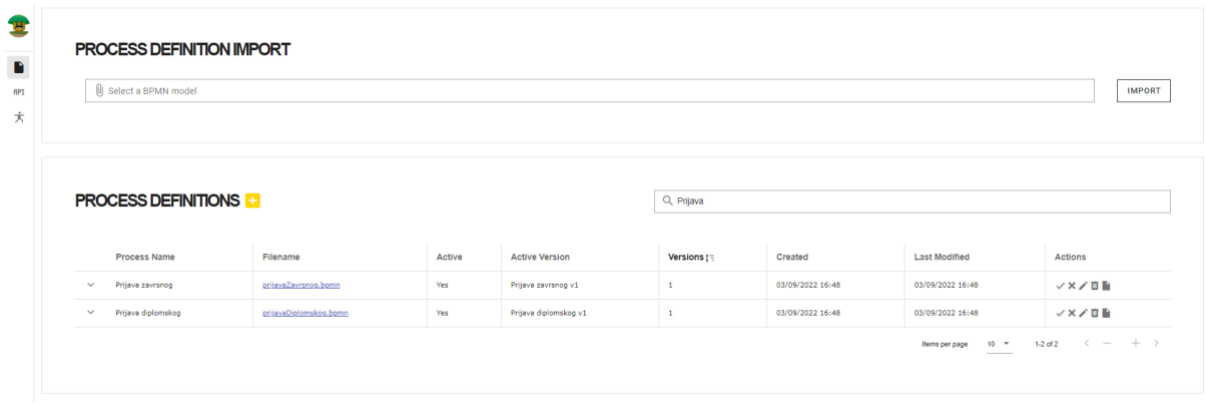
Slika 5: Prikaz sučelja procesnih verzija (Izvor: Autor)

Osim procesa, moguće je proširiti redak tablice i time prikazati sve procesne verzije jedne procesne definicije. Rad s verzijom se također temelji na korištenju tablice.



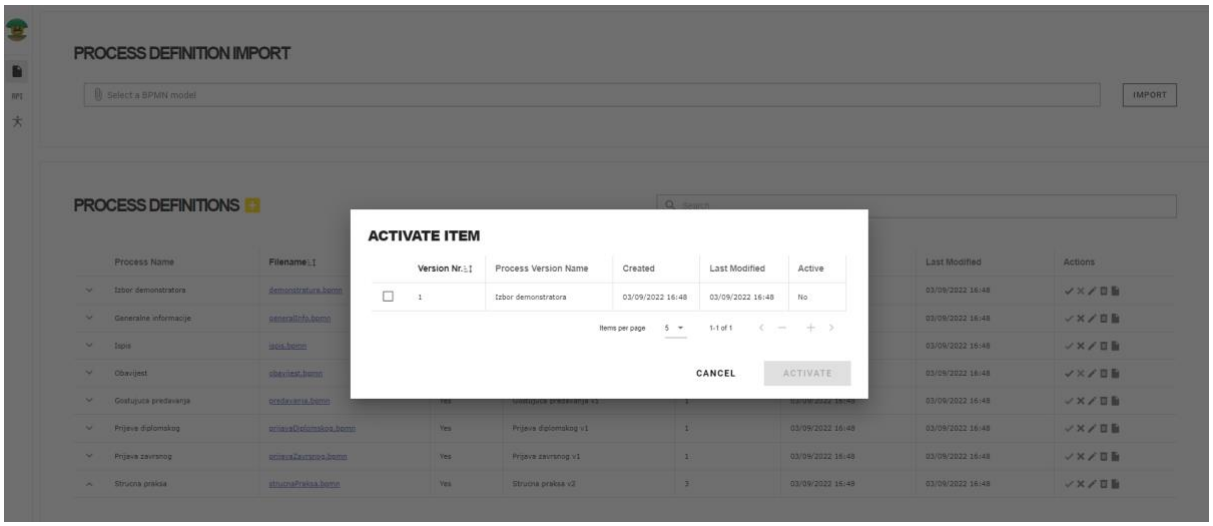
Slika 6: Prikaz funkcionalnosti preglednika procesa (Izvor: Autor)

Pritiskom na ime datoteke (eng. filename), korisnik ima mogućnost pregleda procesa ne samo u tekstualnom obliku nego ima i grafičku reprezentaciju BPMN dijagrama. Pritiskom na gumb pokraj naslova kartice, korisnik se vraća na prethodi ekran.



Slika 7: Tablične funkcionalnosti (Izvor: Autor)

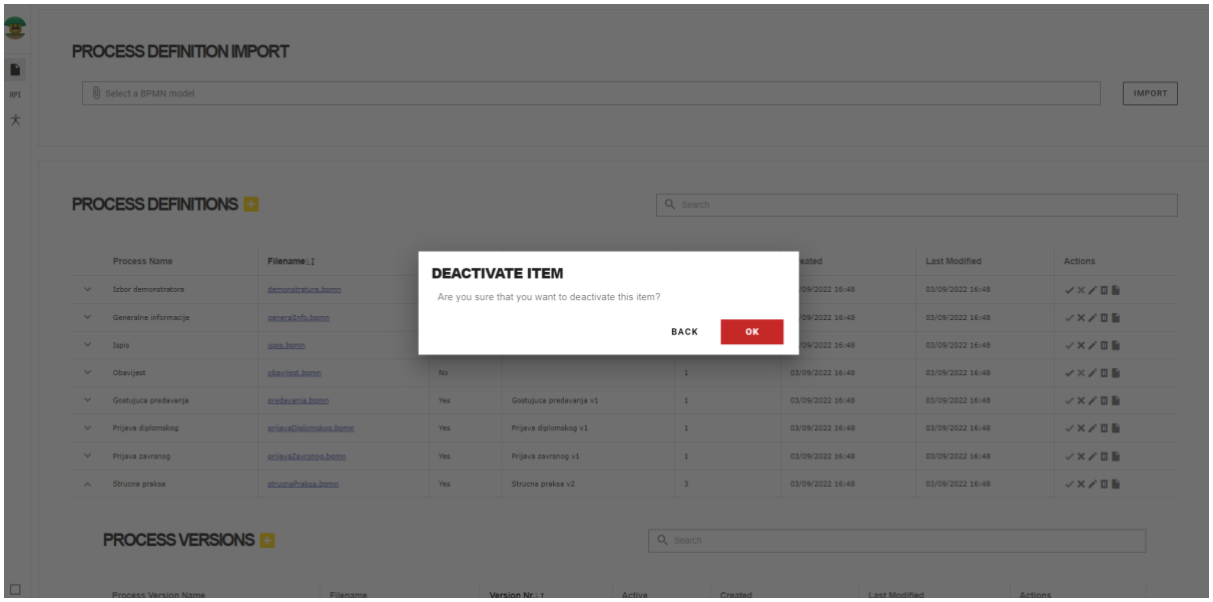
S ciljem poboljšanja rada s tabličnim podacima, implementirane su i mogućnosti sortiranja stupaca i pretraživanja tabličnih podataka. Sortiranje se provodi pritiskom na jedno od zaglavlja koje sadrži tu mogućnost, a pretraživanje na unos vrijednosti u tekstualni input koji se nalazi iznad same tablice.



Slika 8: Aktivacijski dijaloški okvir procesa (Izvor: Autor)

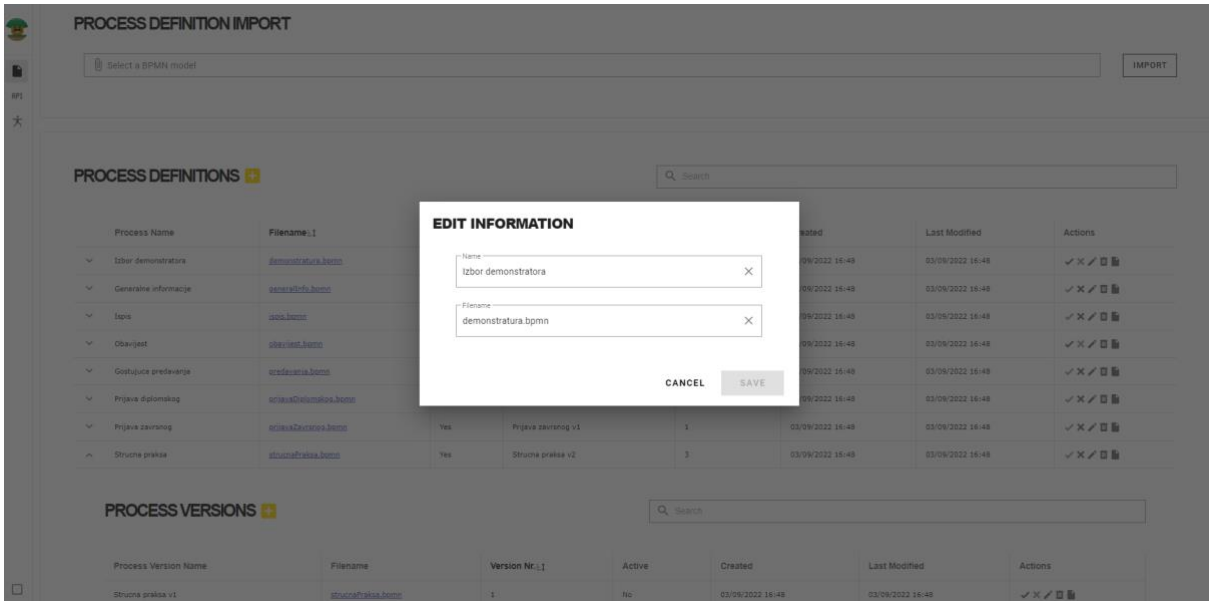
U zadnjem stupcu obiju tablica, imena akcije (eng. Actions) nalaze se operacije koje su moguće nad entitetom. Prva od operacija koja će biti spomenuta jest aktivacija procesa. Naime, pokrenut proces nije isto što i aktivan proces. Aktivnost procesa znači da je moguće pokrenuti i to točno definiranu verziju procesa koja je odabrana. Ukoliko je proces aktivan, utoliko je barem jedna procesna verzija aktivna. Ako je proces neaktivan, onda nijedna verzija nije aktivna. Dakle, bez aktivnih verzija nema aktivnih procesa i obrnuto. Aktivacija se radi pritiskom na dedicirani gumb, kada je gumb pritisnut otvara se dijaloški okvir koji služi za

odabir procesa. Postoji gumb odustani (eng. Cancel) i gumb aktiviraj (eng. Activate) za prigodne namjene.



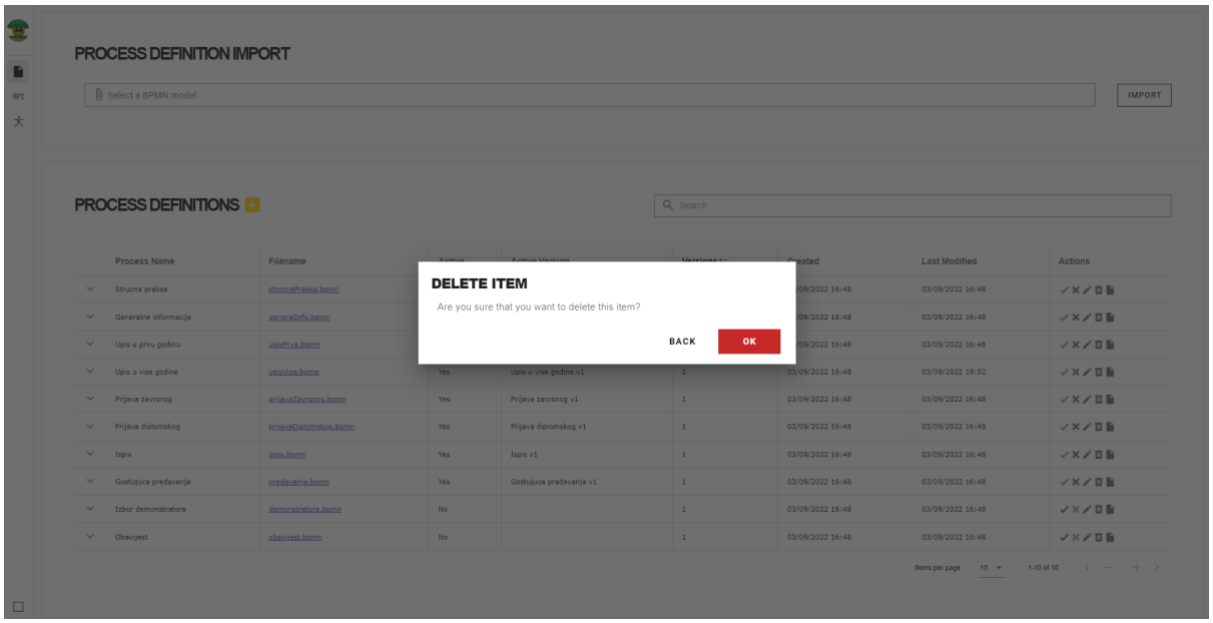
Slika 9: Deaktivacijski dijaloški okvir procesa (Izvor: Autor)

Pored opcije u tablici, odnosno stupcu, za aktivaciju procesa, nalazi se i opcija za deaktivaciju procesa. Semantika korištenja je analogna aktivaciji. Kod pritiska na gumb, otvara se dijalog koji traži dodatnu konfirmaciju pri akciji. Pritiskom na gumb odustani (eng. Cancel) korisnik odustaje od akcije i vraća se na prethodni ekran. Pritiskom na pristanak, deaktivacija biva dovršena kada se prigodna obavijest prikaže korisniku.



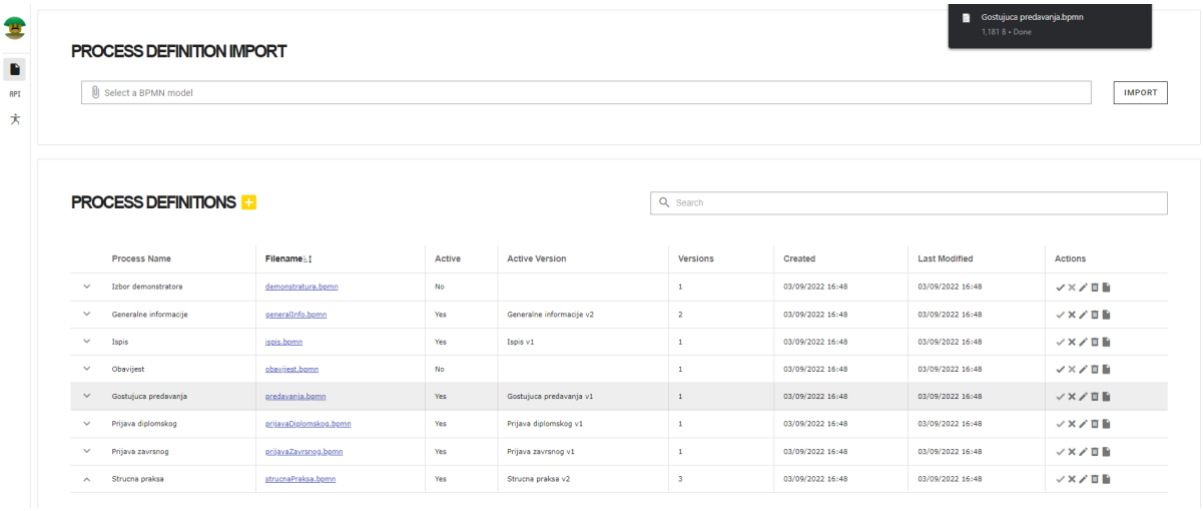
Slika 10: Dijaloški okvir mogućnosti uređivanja procesa (Izvor: Autor)

Jedina polja koja su zaista promjenjiva u procesnoj definiciji su promjena imena same definicije procesa i imena datoteke. Jednako je i u postupanju s procesnom verzijom. Korištenje opcije promjene svojstava procesa radi se pritiskom na gumb koji otvara dijaloški okvir. U dijaloškom okviru nalaze se dva tekstualna polja za upis, odnosno promjenu vrijednosti. Pritiskom na gumb odustani (eng. Cancel) korisnik odustaje od akcije i vraća se na prethodni ekran. Pritiskom na potvrdu, propagacija promjene se nastavlja do završetka.



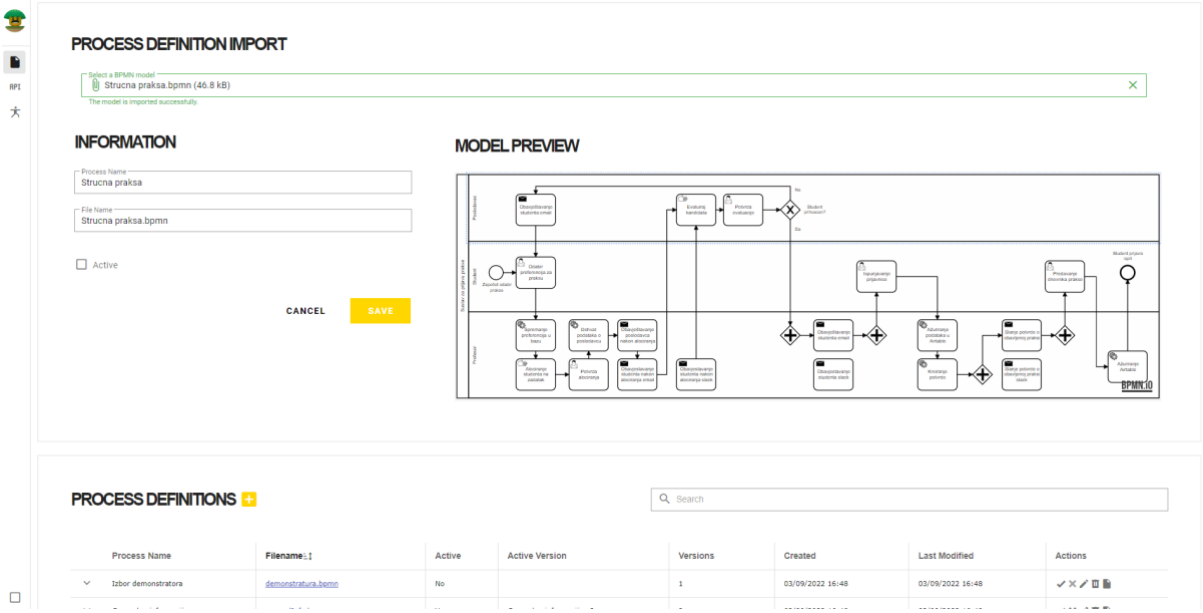
Slika 11: Dijaloški okvir mogućnosti brisanja procesa (Izvor: Autor)

Svaku procesnu definiciju moguće je i obrisati. Ako se obriše procesna definicija tada se brišu i sve verzije jer se smatraju djecom, a to je vidljivo i u klasnom dijagramu u prethodnom poglavlju. Kada se obriše jedina verzija u definicija tada se i definicija briše. Ako se briše verzija koja nije jedina, drugim riječima postoje i druge verzije unutar definicije tada je bitno da je verzije neaktivna, pod uvjetom da je aktivna tada se i procesna definicija deaktivira.



Slika 12: Prikaz funkcionalnosti izvoza procesa (Izvor: Autor)

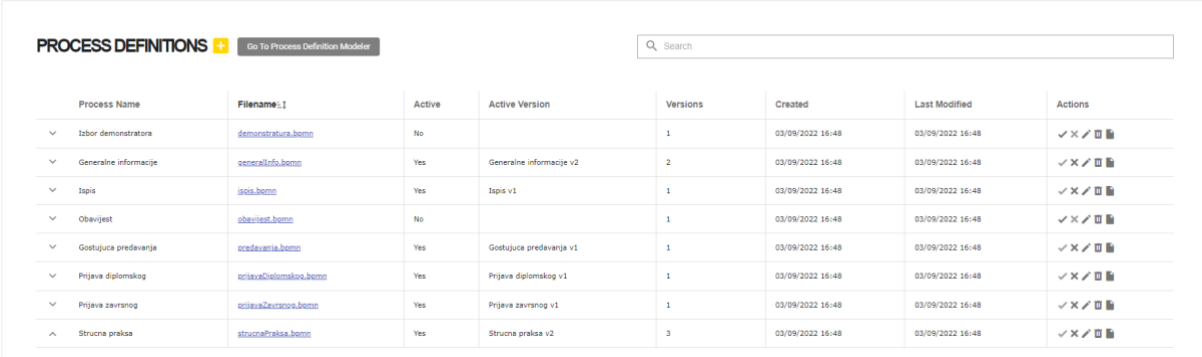
Osim aktivacije ili deaktivacije, uređivanja i brisanja. Korisnik ima mogućnost izvoza datoteke u „.bpmn“ formatu. Naime, „.bpmn“ datotečni format je zapravo podvrsta XML datoteke. Sam izvoz se radi pritiskom na dedicirani gumb u stupcu tablice pokreće operacija preuzimanja datoteke. To je zaista korisno kada se proces želi pobliže pregledati u izvanmrežnom radu ili modelirati u nekoj drugoj aplikaciji.



Slika 13: Prikaz funkcionalnosti unosa procesa (Izvor: Autor)

Uvoz procesne definicije moguć je korištenjem podsustavom za uvoz dijagrama u sustav. Potrebno je unijeti ime procesa, ime datoteke i postaviti vrijednost na to je li proces aktivan, odnosno neaktivan. Kod uvoza, s desne strane unosa generalnih informacija, postoji i prikaz

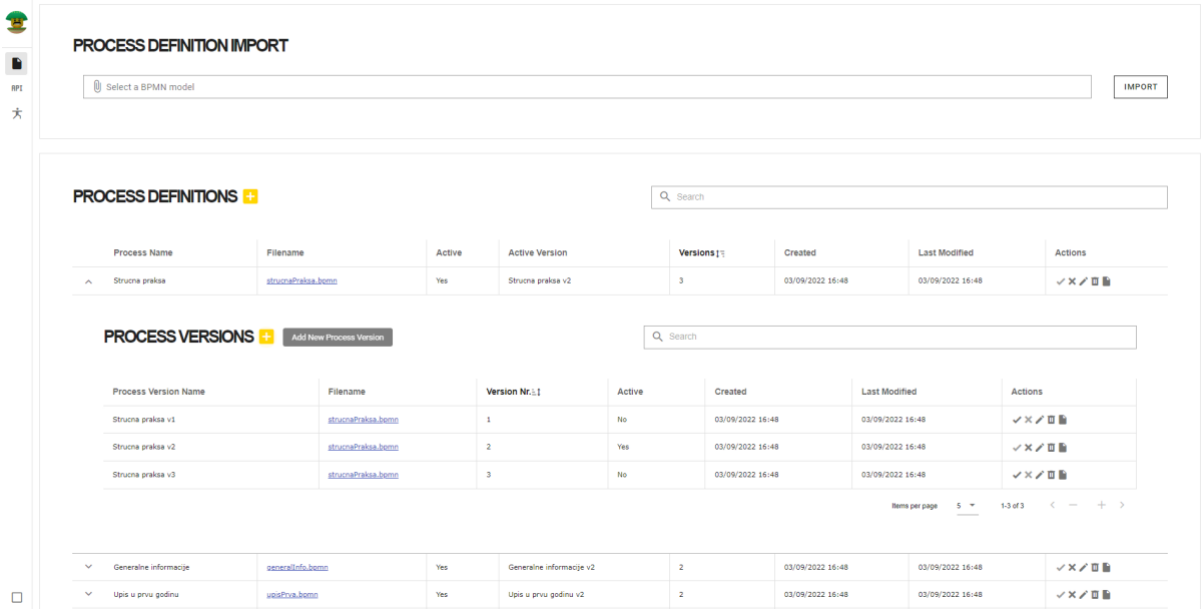
uvezenog procesa. Pritiskom na gumb odustani (eng. Cancel) korisnik odustaje od akcije i vraća se na prethodni ekran. Pritiskom na potvrdu, spremanje procesa u sustav biva dovršeno. Uvoz se može smatrati kao i prvim načinom kreacije procesa u sustav, a ograničen je na definiciju. Dakle, nije moguće uvesti procesnu verziju, već je moguće samo modelirati.



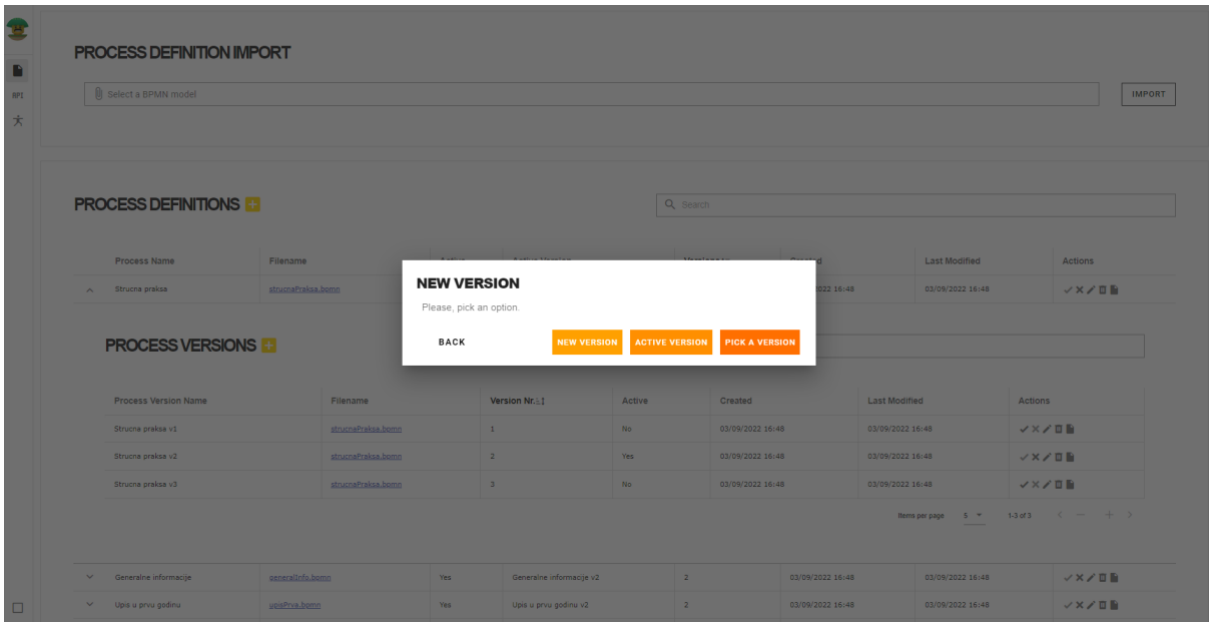
Slika 14: Prikaz načina uporabe kreiranja procesne definicije (1/2) (Izvor: Autor)

Slika 15: Prikaz načina uporabe kreiranja procesne definicije (2/2) (Izvor: Autor)

Druga mogućnost kreacije modela je onaj modeliranjem dijagrama u podsustavu zaduženom za modeliranje. Pritiskom na gumb pokraj naziva sekcije, korisnika se vodi u taj podsustav, gdje ima sve potrebne alate za definiranje jednog BPMN procesa. Modeliranje samog procesa bit će definirano kasnije.



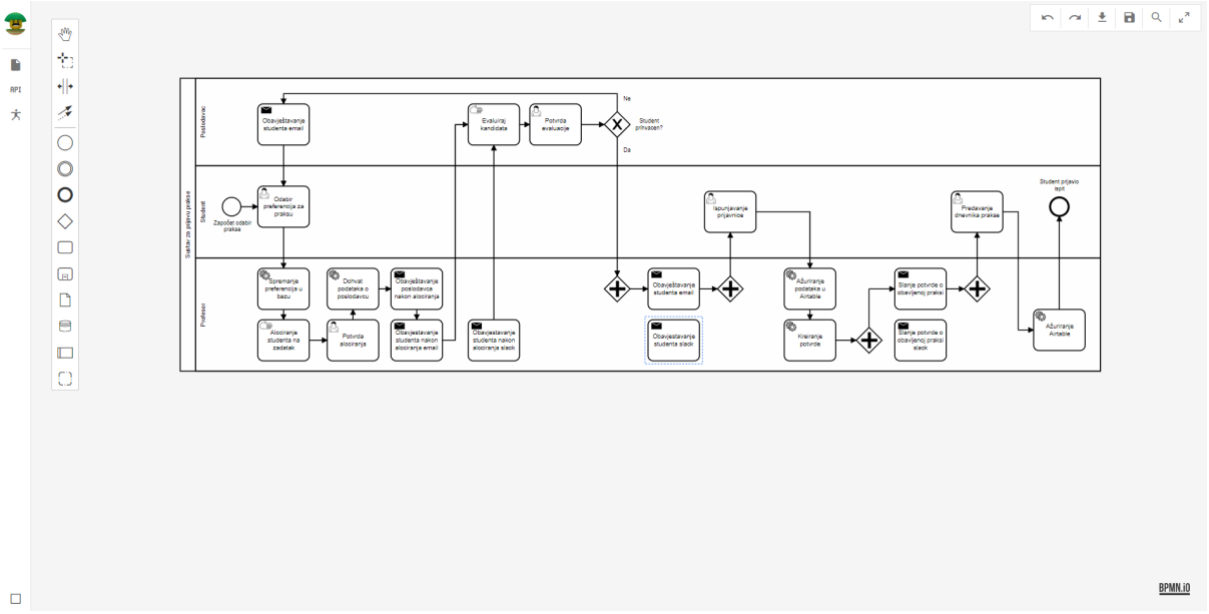
Slika 16: Prikaz mogućnosti kreiranja procesne verzije (1/6) (Izvor: Autor)



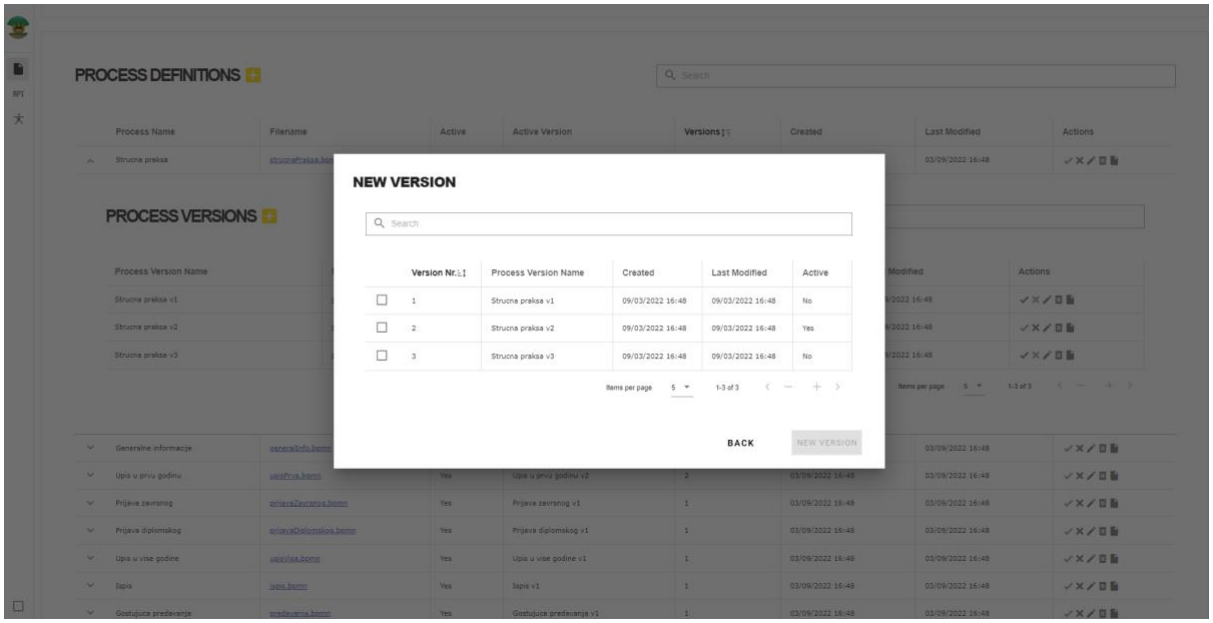
Slika 17: Prikaz mogućnosti kreiranja procesne verzije (2/6) (Izvor: Autor)

Kod procesnih verzija kreacija se pokreće pritiskom na gumb pokraj naslova sekcije. Pritisak na gumb dovodi do otvaranja dijaloškog okvira za odabir predloška pri modeliranju. Na slici 17. (2/6) prikazane su tri opcije. Naime, prva opcija uzima generalne podatke procesa, ali xml podatke postavlja na zadane (eng. Default) postavke kao što je vidljivo na slici 18. (3/6). Dakle, „.bpmn“ definicija koja će se koristiti kao predložak je ona bazična. Druga opcija uzima aktivnu verziju procesa te nju uzima kao predložak. Treća i zadnja opcija nudi mogućnost odabira verzije koja će se uzeti kao predložak, a vidljivo je na slici 20. (5/6).

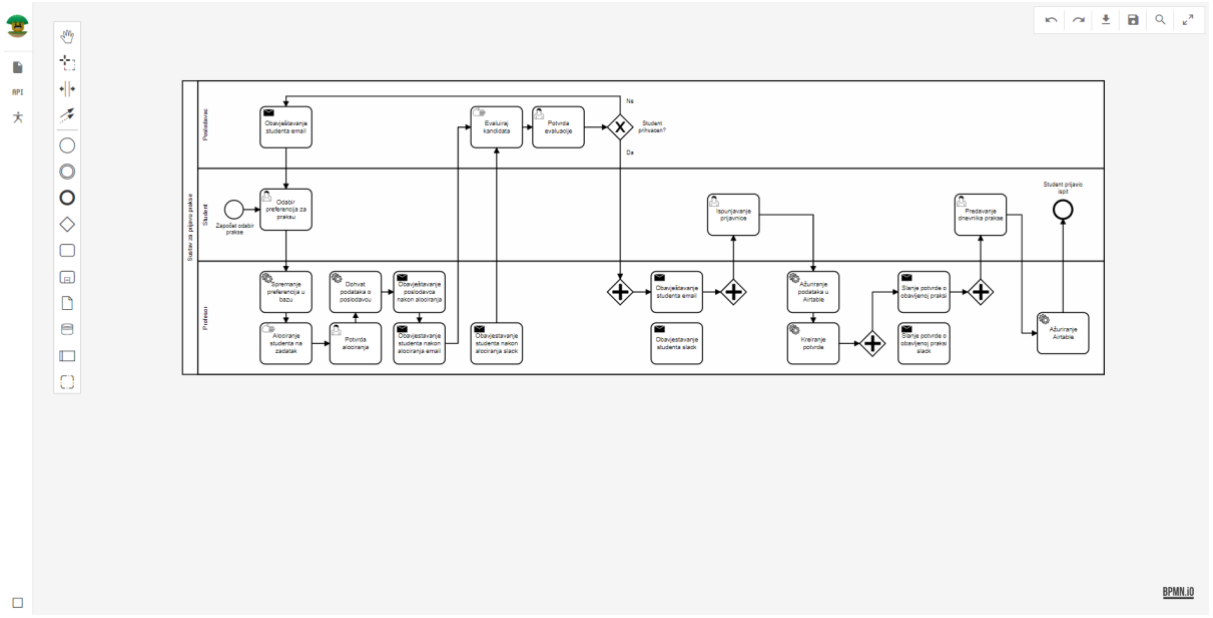
Slika 18: Prikaz mogućnosti kreiranja procesne verzije (3/6) (Izvor: Autor)



Slika 19: Prikaz mogućnosti kreiranja procesne verzije (4/6) (Izvor: Autor)

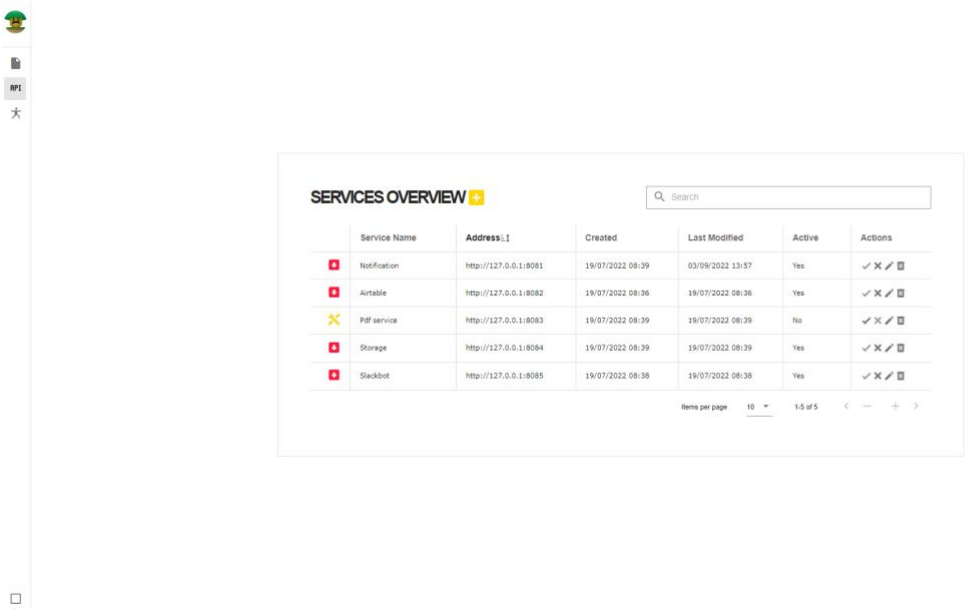


Slika 20: Prikaz mogućnosti kreiranja procesne verzije (5/6) (Izvor: Autor)



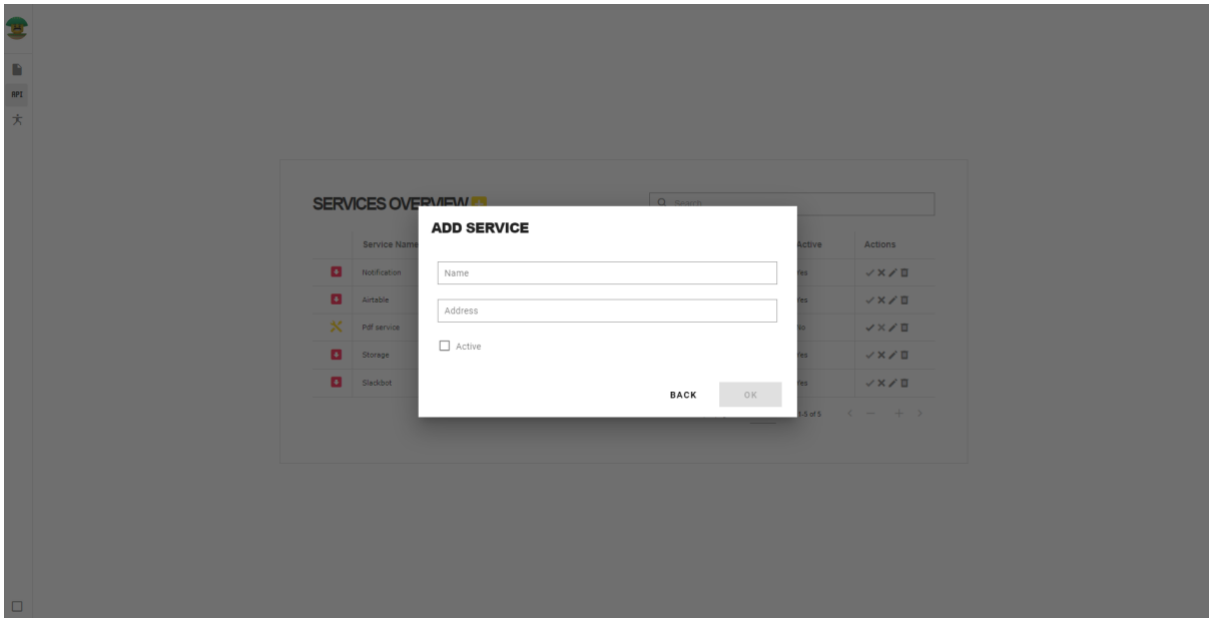
Slika 21: Prikaz mogućnosti kreiranja procesne verzije (6/6) (Izvor: Autor)

2.4.2. STRANICA SERVISA



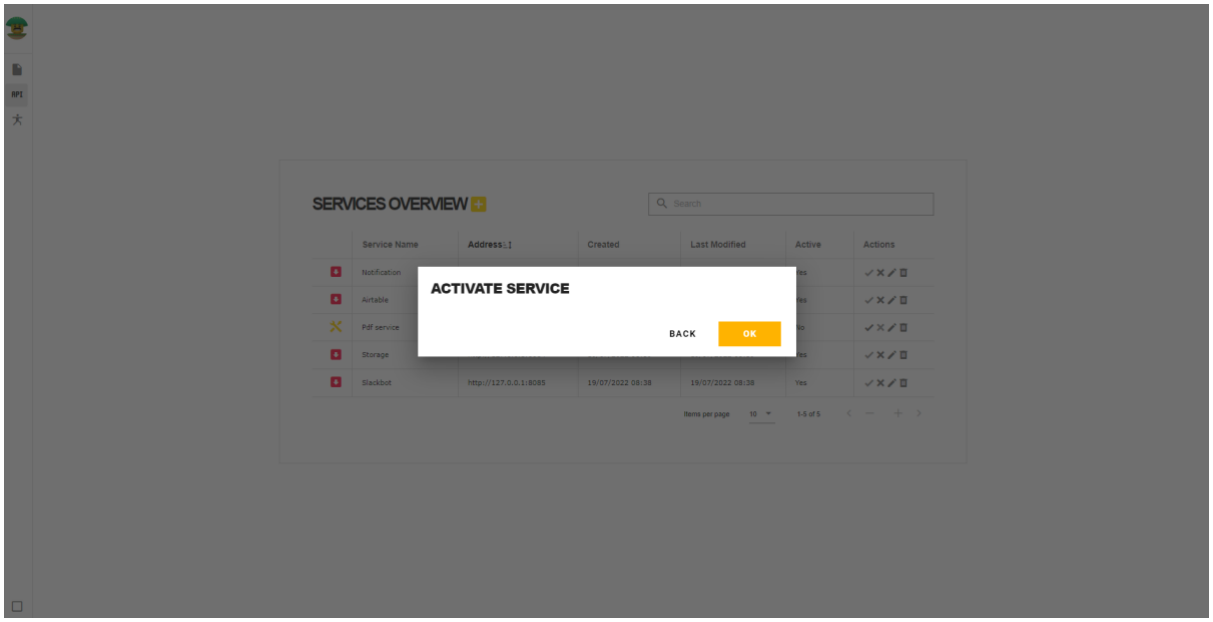
Slika 22: Prikaz sučelja servisa (Izvor: Autor)

Na stranici, tj. sekciji servisi (eng. Services) korisnik ima mogućnost upravljanja servisa. Servisi su prikazani u obliku tablice, a interakcija s tablicom je jednaka kao i u prethodnim prikazima. Moguće je i pretraživati te sortirati polja u tablici. Jedina iznimka, osim podataka, su ikonice koje iskazuju status servisa. Servis može biti na mreži (eng. Online), izvan mreže (eng. Offline) i na održavanju (eng. Under maintenance). Prva dva indikatora sugeriraju status dostupnosti servisa, dok status održavanja označava namjernu neaktivnost servisa. To je učinjeno kako bi se napravila distinkcija između namjerno i nenamjerno neaktivnog servisa.



Slika 23: Dijaloški okvir mogućnosti dodavanja servisa (Izvor: Autor)

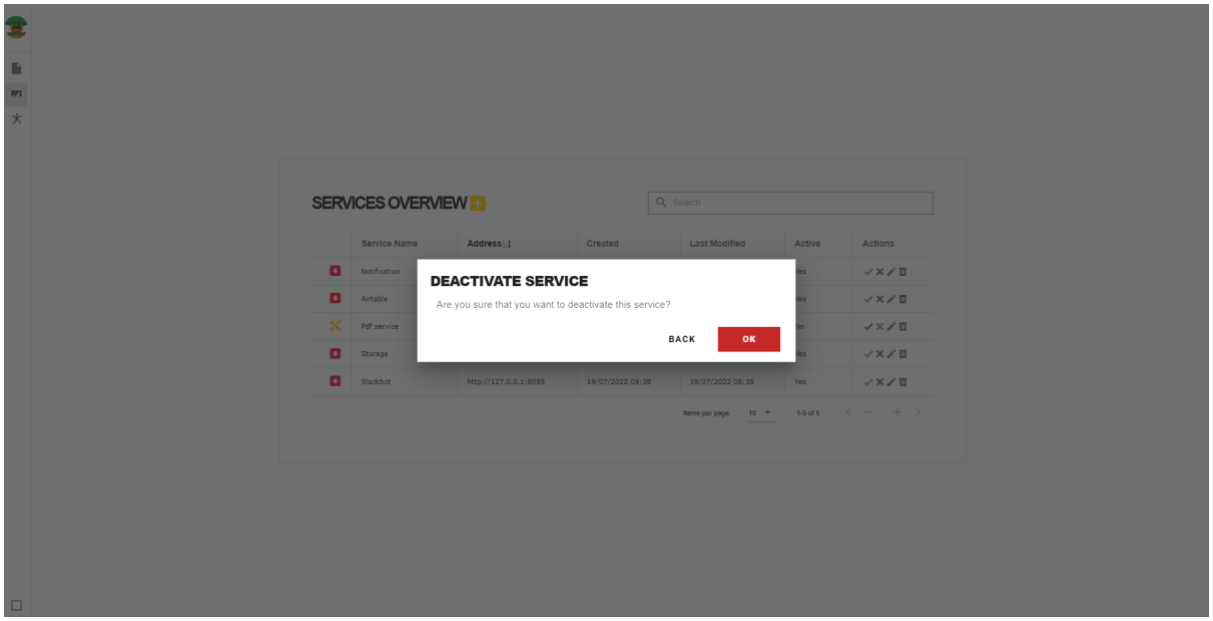
Dodavanjem servisa vrši se pritiskom na gumb. Gumb otvara dijaloški okvir u kojem je potrebno unijeti ime, url adresu i aktivnost samog servisa. Pritiskom na gumb odustani (eng. Cancel) korisnik odustaje od akcije i vraća se na prethodni ekran. Pritiskom na potvrdu, dodaje se proces u sustav. Kod unosa treba pripaziti zbog unikatnosti URL adrese.



Slika 24: Dijaloški okvir mogućnosti aktivacije servisa (Izvor: Autor)

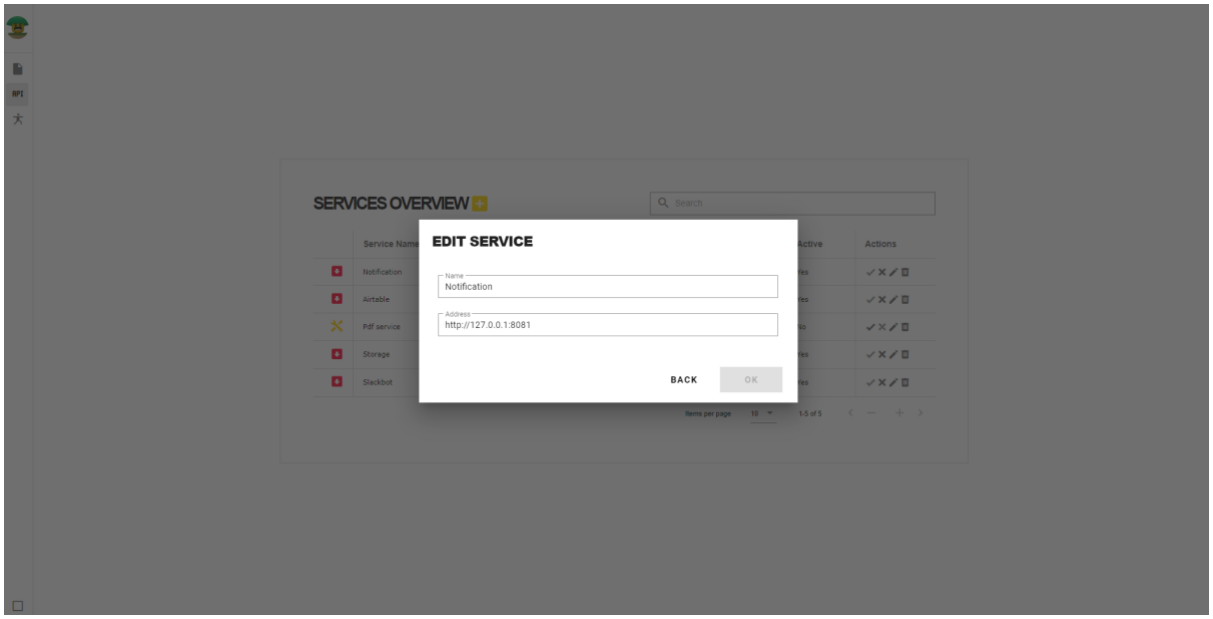
Aktivacija servisa se vrši na sličan način kao i prošle radnje. Pritiskom na gumb se otvara dijaloški okvir koji traži daljnju konfirmaciju. Pritiskom na gumb odustani (eng. Cancel) korisnik

odustaje od akcije i vraća se na prethodni ekran. Pritiskom na potvrdu, aktivacija servisa je dovršena. Samom aktivacijom servisa, servis postaje dostupan za korištenje.



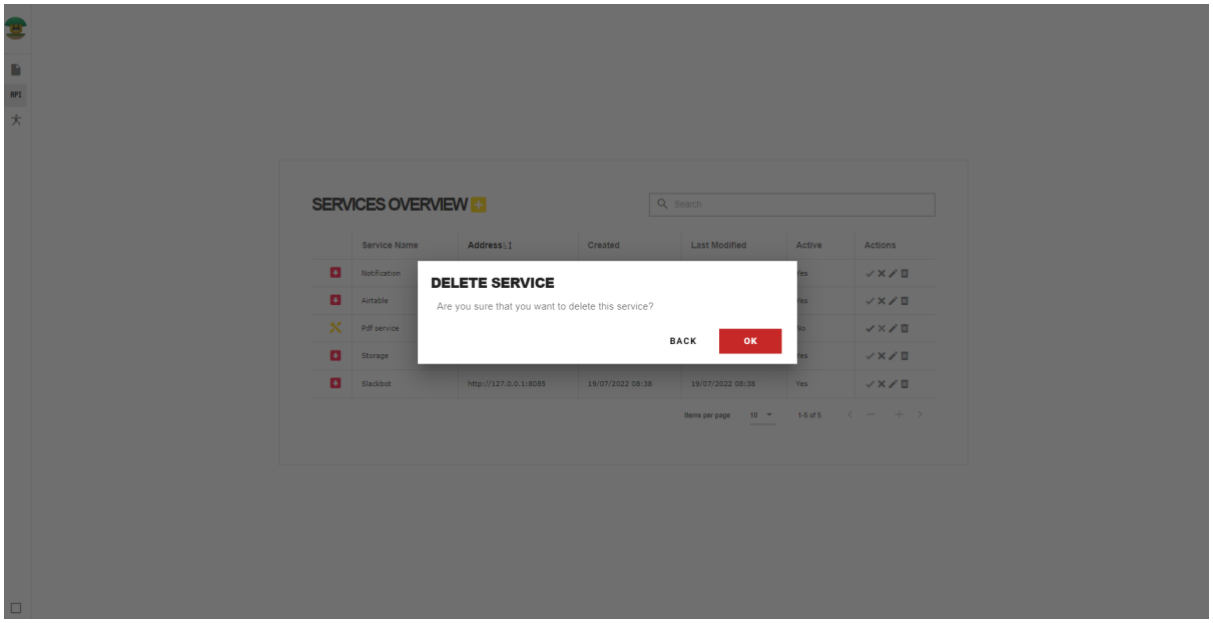
Slika 25: Dijaloški okvir mogućnosti deaktivacije servisa (Izvor: Autor)

Deaktivacija servisa je analogna aktivaciji servisa u semantičkom smislu, dakle servis postaje nedostupan i stavlja se indikator „Na održavanju“. Proces postupanja je jednak kao i u prethodnom slučaju.



Slika 26: Dijaloški okvir mogućnosti uređivanja servisa (Izvor: Autor)

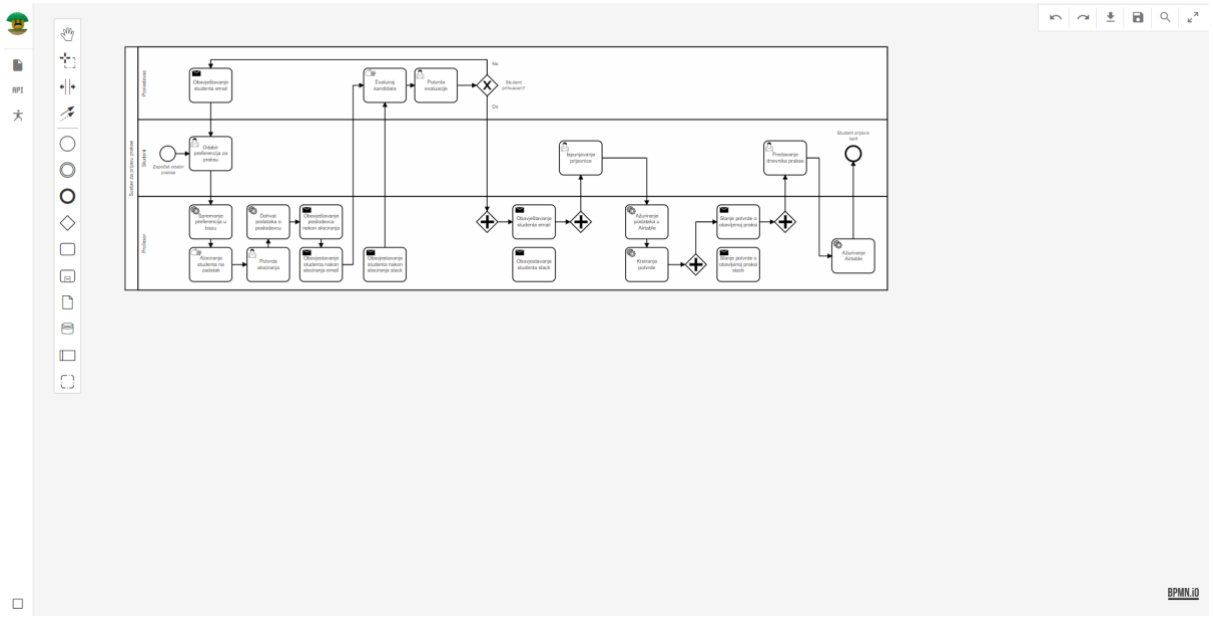
Servis je moguće i mijenjati. To se radi pritiskom na gumb u sekciji akcije. Otvara se dijalog koji omogućava samu promjenu informacija. Pritiskom na gumb „Ok“, propagacija se nastavlja.



Slika 27: Dijaloški okvir mogućnosti brisanja servisa (Izvor: Autor)

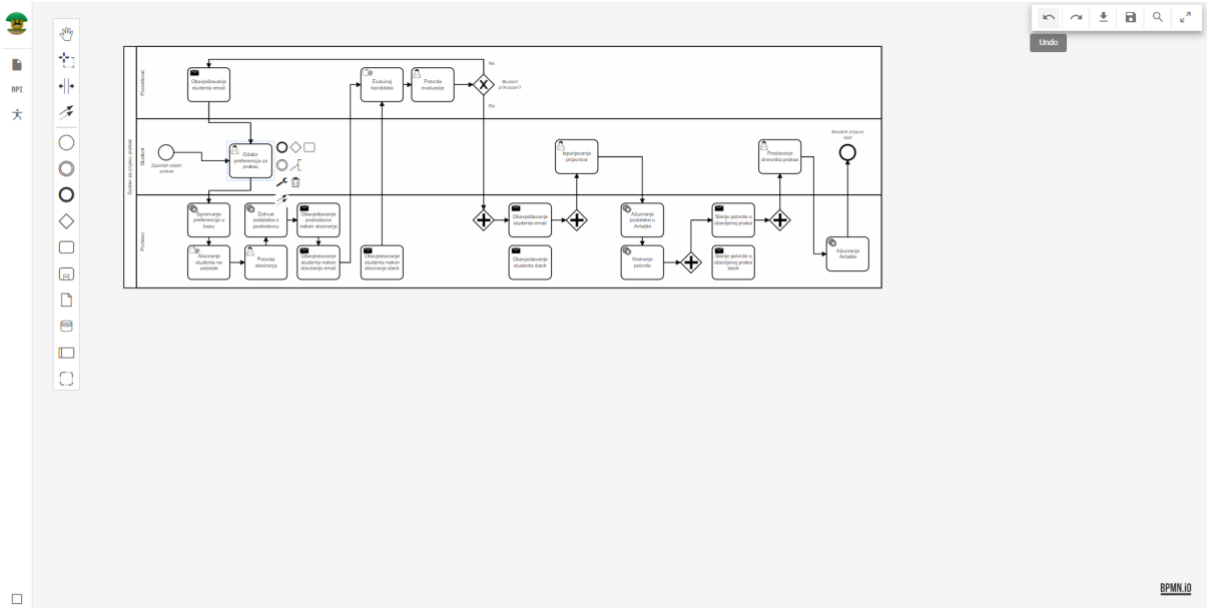
Postupak brisanja je jednak aktivaciji, odnosno deaktivaciji servisa.

2.4.3. STRANICA MODELERA

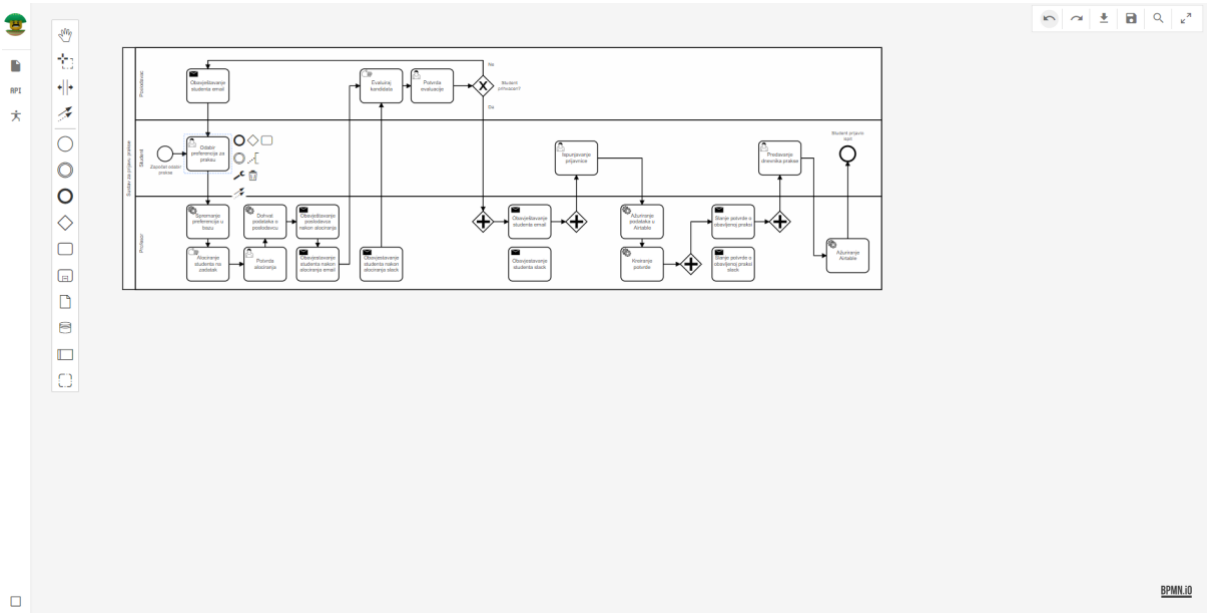


Slika 28: Prikaz sučelja modelera (Izvor: Autor)

Podsustavu zaduženom za modeliranje može se pristupiti na više načina, primjerice kod kreairanja nove procesne verzije ili definicije. Slika prikazuje sučelje sustava u kojemu je omogućeno modeliranje. S lijeve strane se nalaze razni objekti, artefakti, itd. koji omogućavaju interakciju u modeliranju i stvaranje BPMN modela. U centru se nalazi sam definirani proces, a s gornje desne strane se nalazi reducirana ploča akcija i svojstava (eng. actions and properties panel).

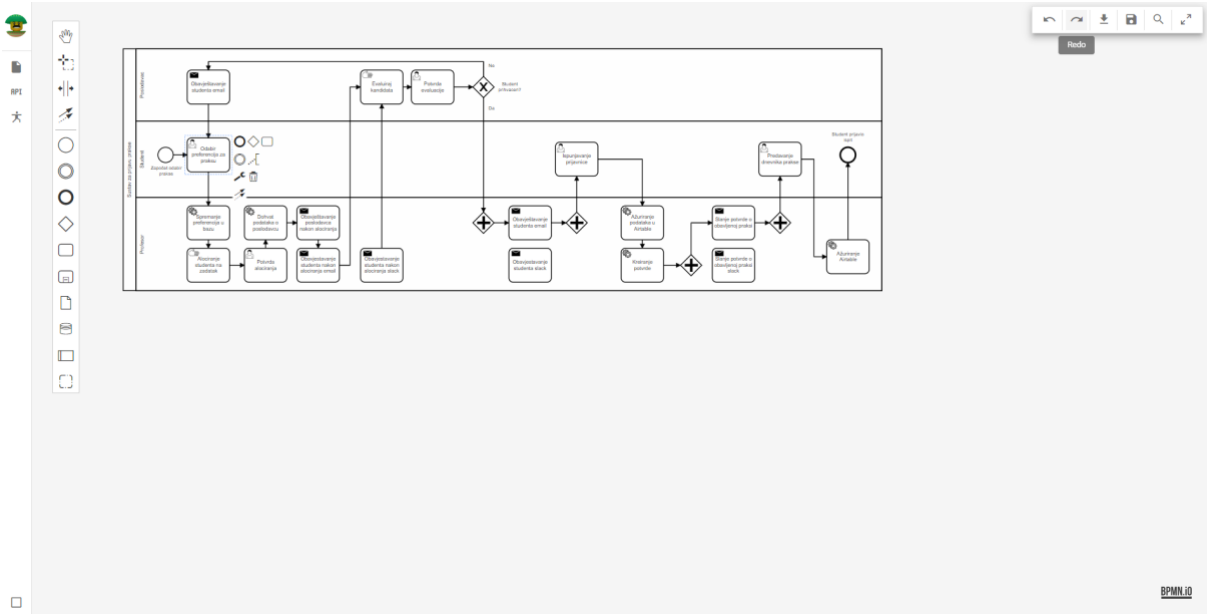


Slika 29: Prikaz funkcionalnosti "vrati" (1/2) (Izvor: Autor)

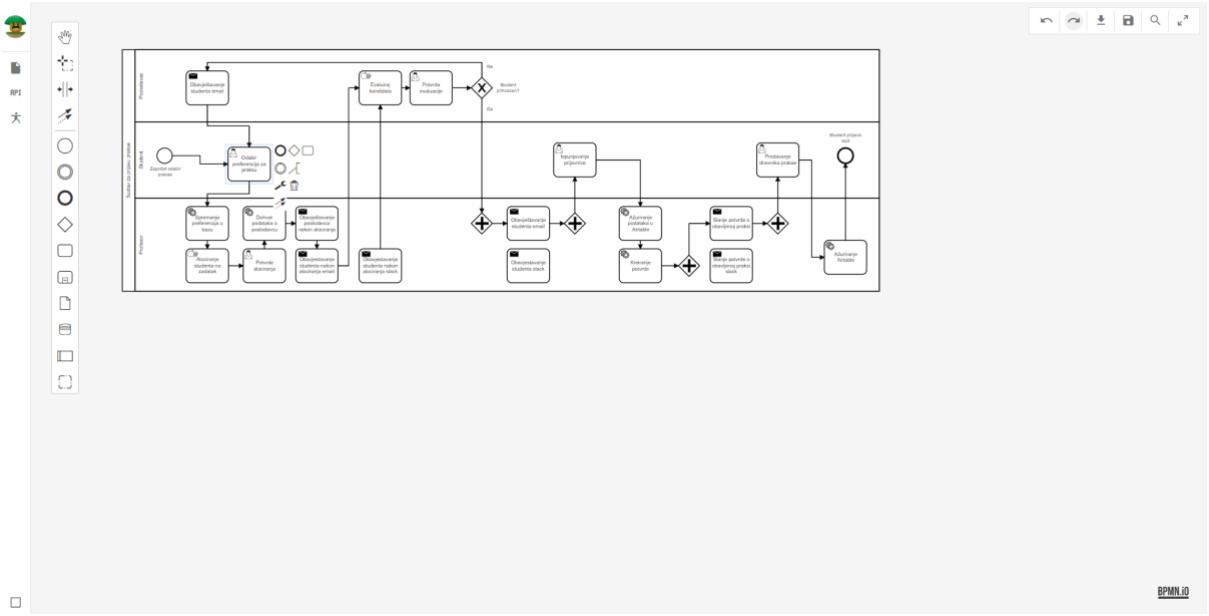


Slika 30: Prikaz funkcionalnosti "vrati" (2/2) (Izvor: Autor)

S krajne lijeve stane ploče svojstava, pritiskom na gumb povlači se napravljena akcija, što je vidljivo na slici 31. (1/2) i slici 32. (2/2).

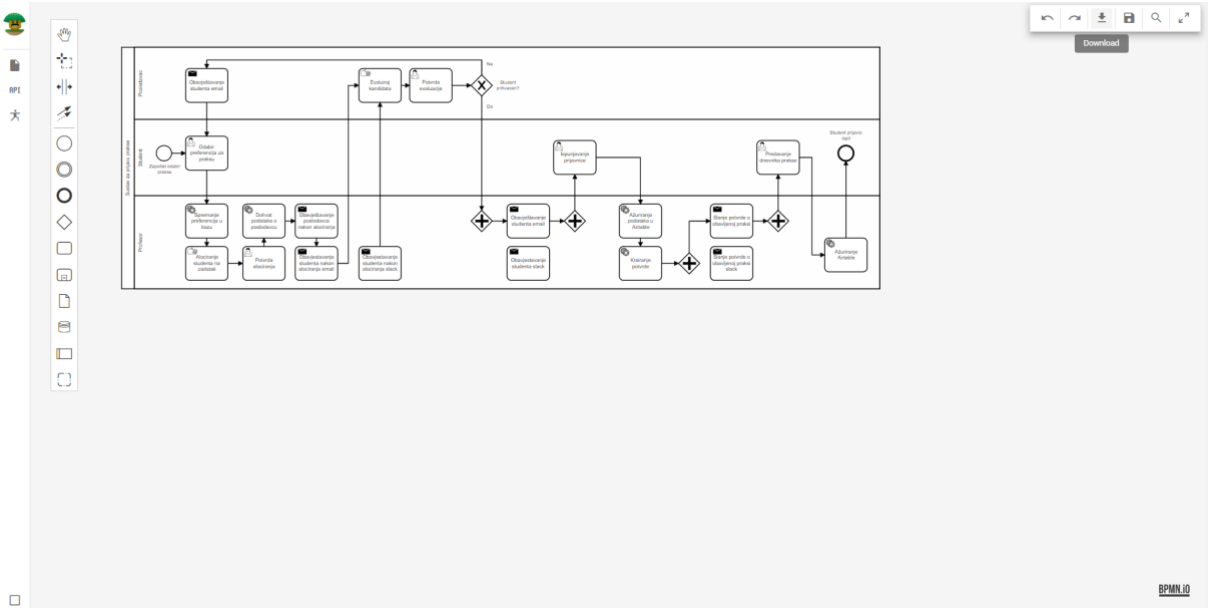


Slika 31: Prikaz funkcionalnosti "vrati prethodno" (1/2) (Izvor: Autor)

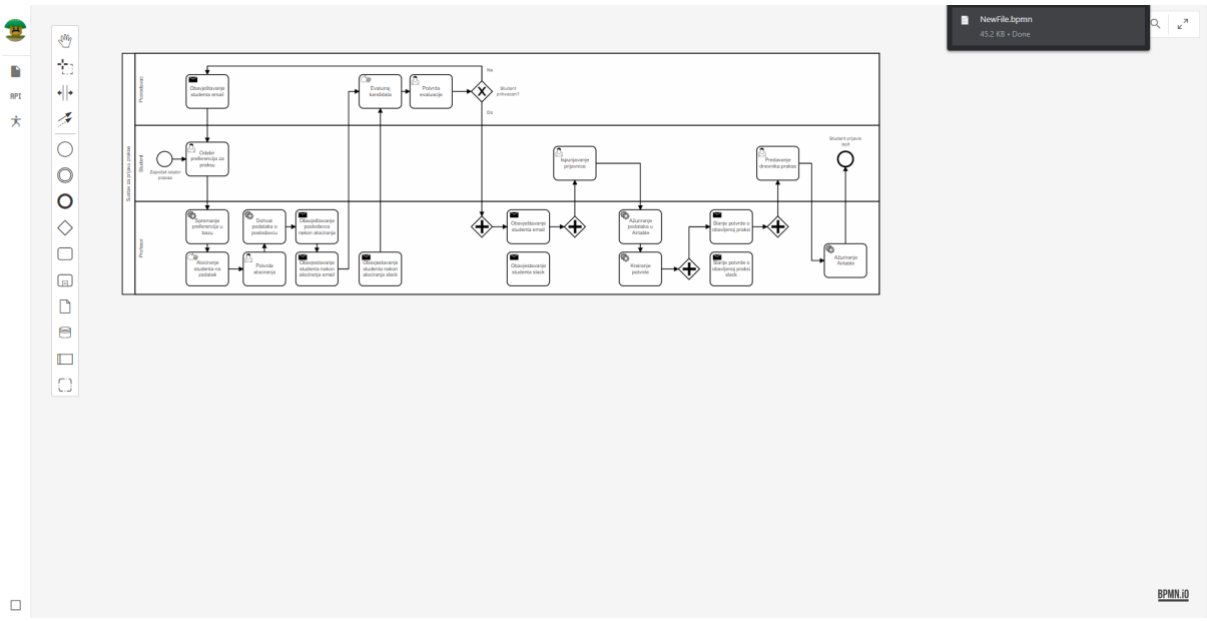


Slika 32: Prikaz funkcionalnosti "vrati prethodno" (2/2) (Izvor: Autor)

Kao i kod prethodnog primjera, operacija ponovno uradi (eng. Redo), moguće je vratiti prethodno izmijenjeni korak, odnosno ponovno odraditi prethodno napravljenu akciju.

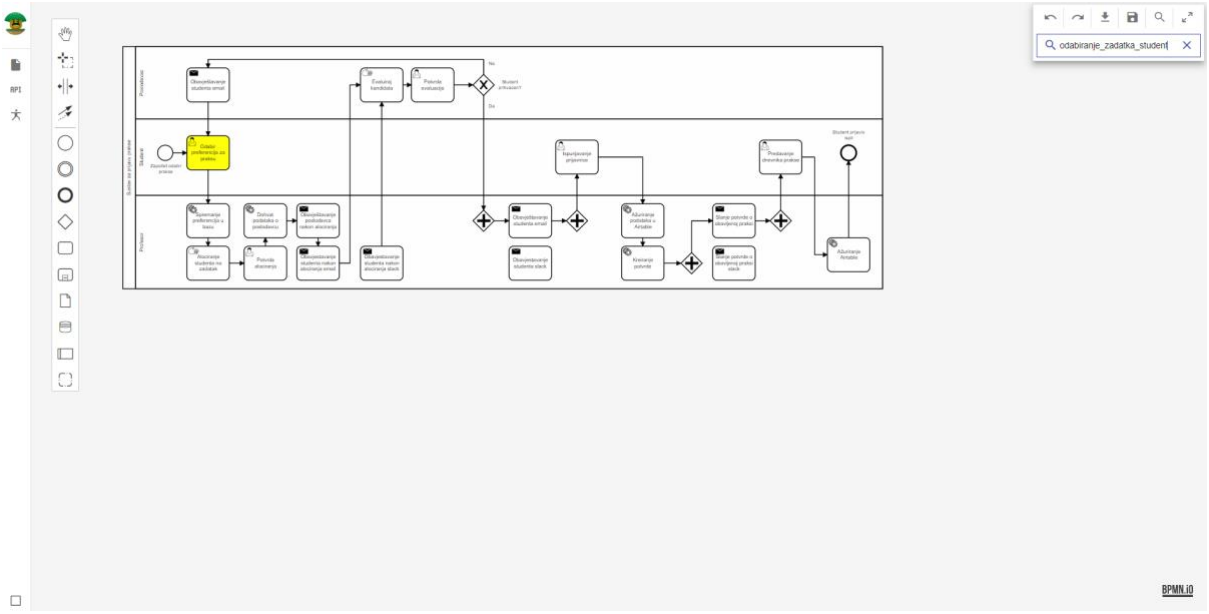


Slika 33: Prikaz funkcionalnosti preuzimanja (1/2) (Izvor: Autor)



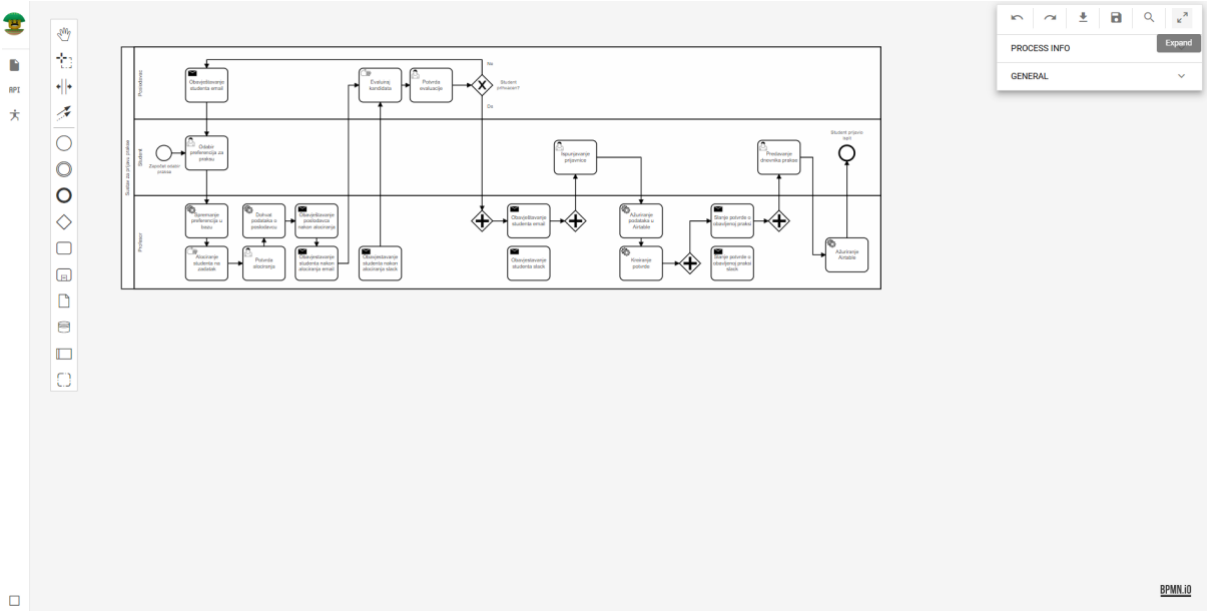
Slika 34: Prikaz funkcionalnosti preuzimanja (2/2) (Izvor: Autor)

Proces je u svakom trenutku moguće preuzeti lokalno. Što je korisno, ako se želi prenijeti rad u izvanmrežni rad ili spremiti lokalno. Samo preuzimanje se vrši pritiskom na gumb.



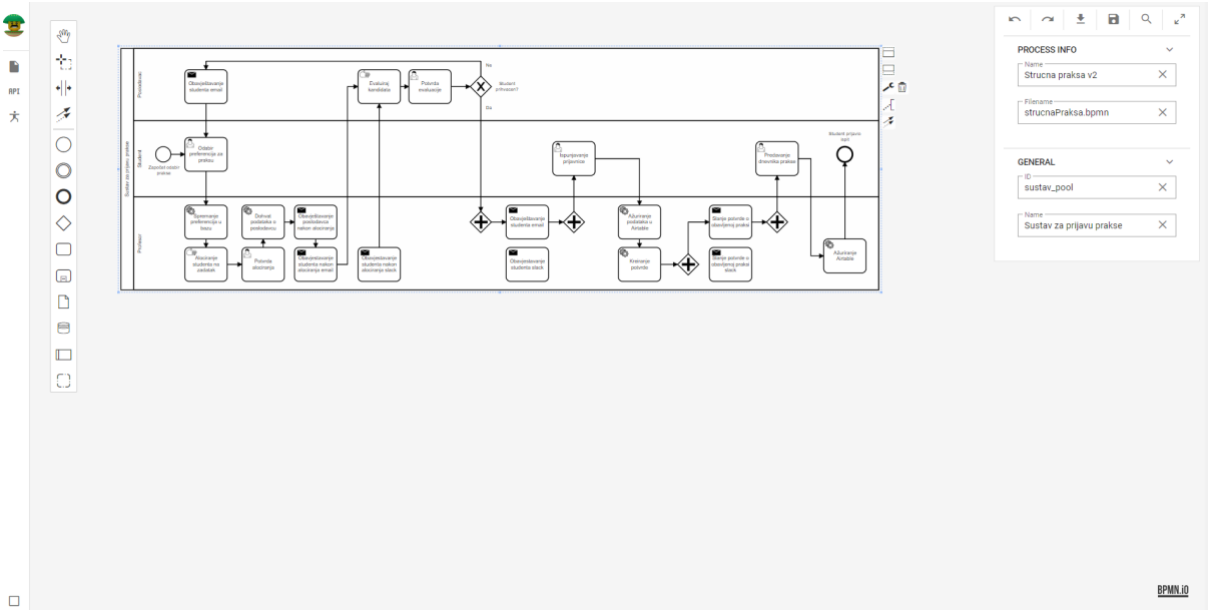
Slika 35: Prikaz funkcionalnosti pretraživanja (Izvor: Autor)

Opcija pretraživanja po identifikatoru radi tako da se pritisne gumb koji otvara tekstualni input i time omogućuje upis, odnosno pretraživanje svih elemenata dijagrama po id-ju dediciranog elementa.



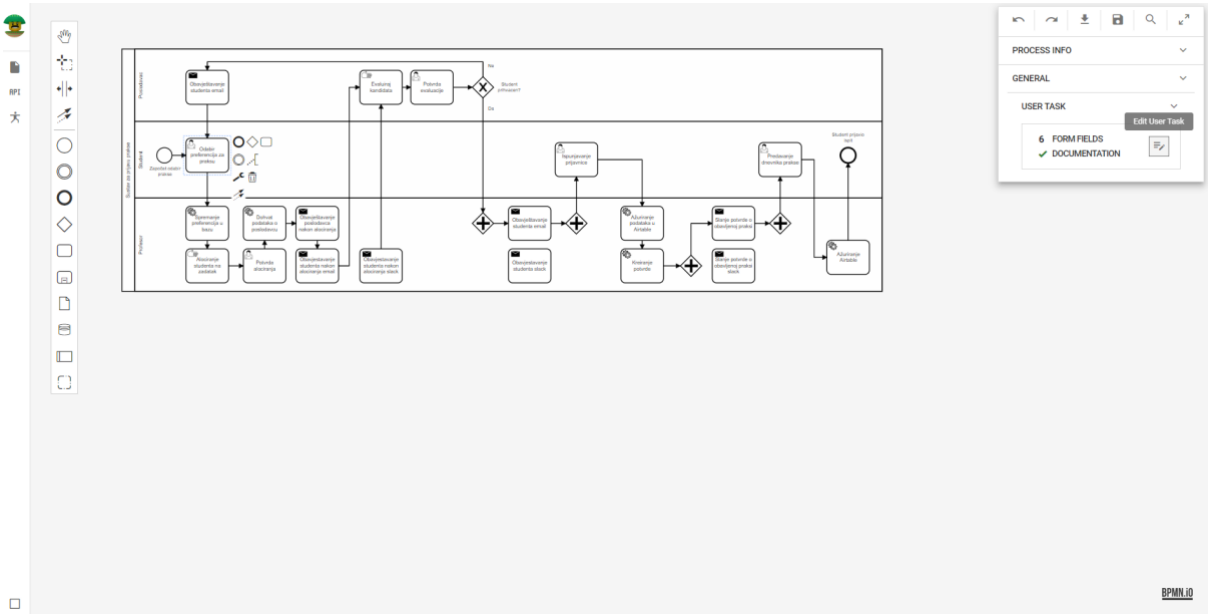
Slika 36: Prikaz funkcionalnosti povećanja panela (Izvor: Autor)

Pritiskom na gumb proširi (eng. Expand), panel svojstava procesa se proširuje. Interakcija s dijagramom, odnosno klik na elementu, prikazuje relevantne podatke.



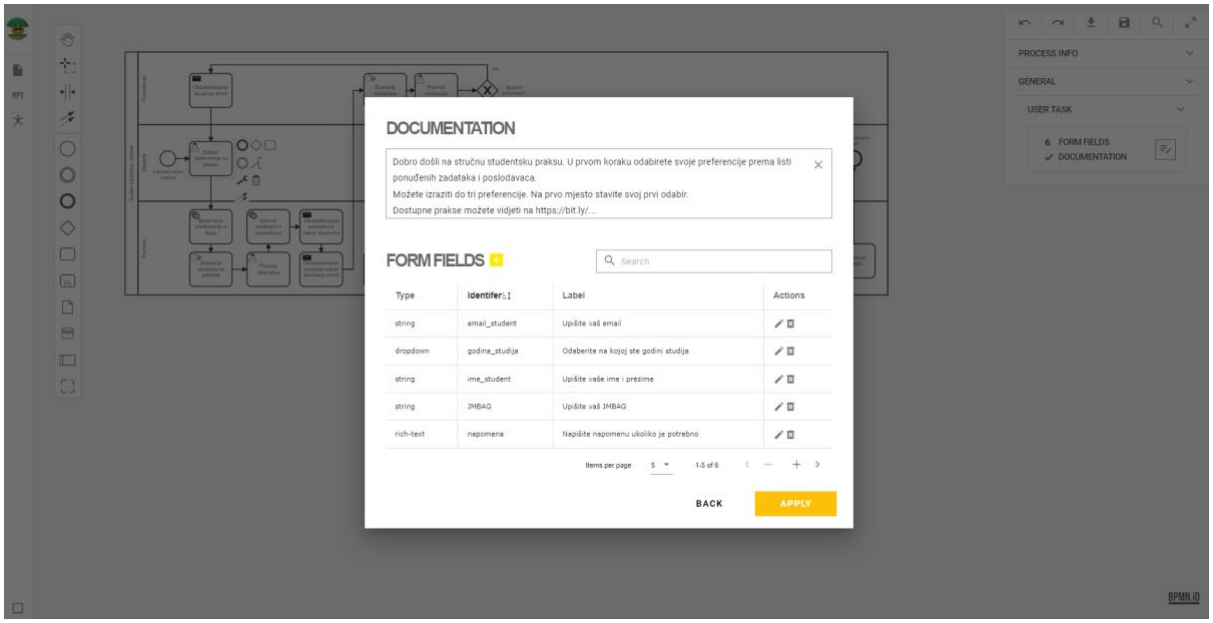
Slika 37: Prikaz informacijskog dijela panela (Izvor: Autor)

Pritiskom na gumb proširi, proširuju se i generalne informacije procesa, odnosno generalne informacije elementa.



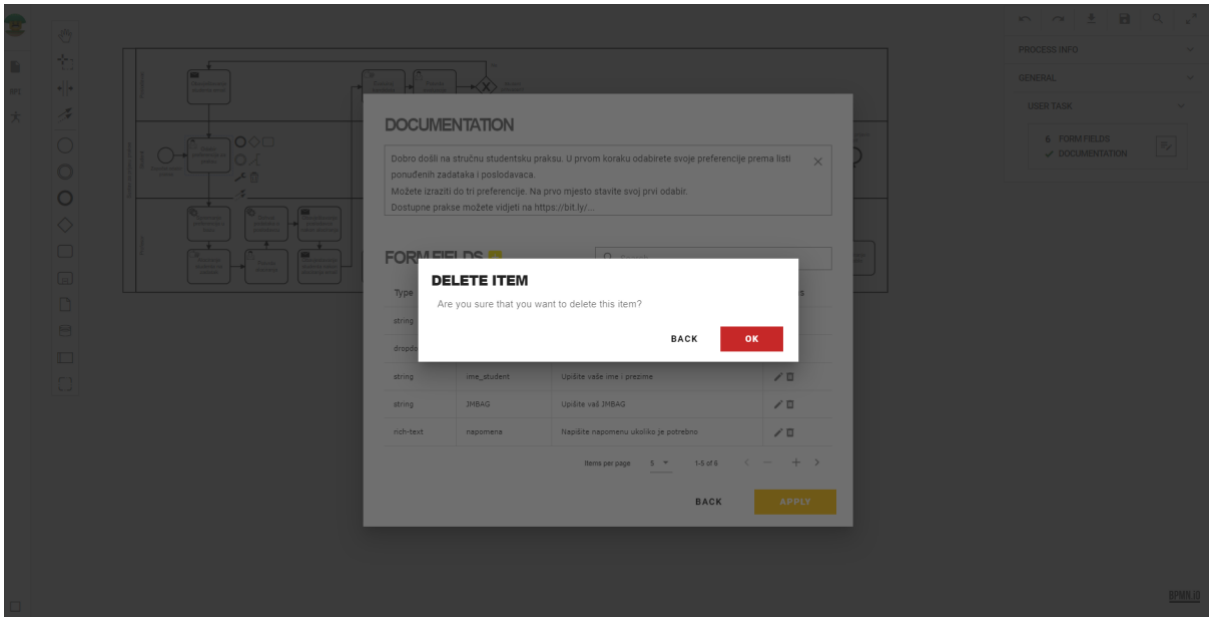
Slika 38: Prikaz otvaranja detalja korisničkog zadatka (Izvor: Autor)

Klikom na dedicirani element, kao što je u ovom slučaju korisnički zadatak (eng. User task). Panel prikazuje osnovne informacije elementa, a one detaljne zahtijevaju dodatni klik.



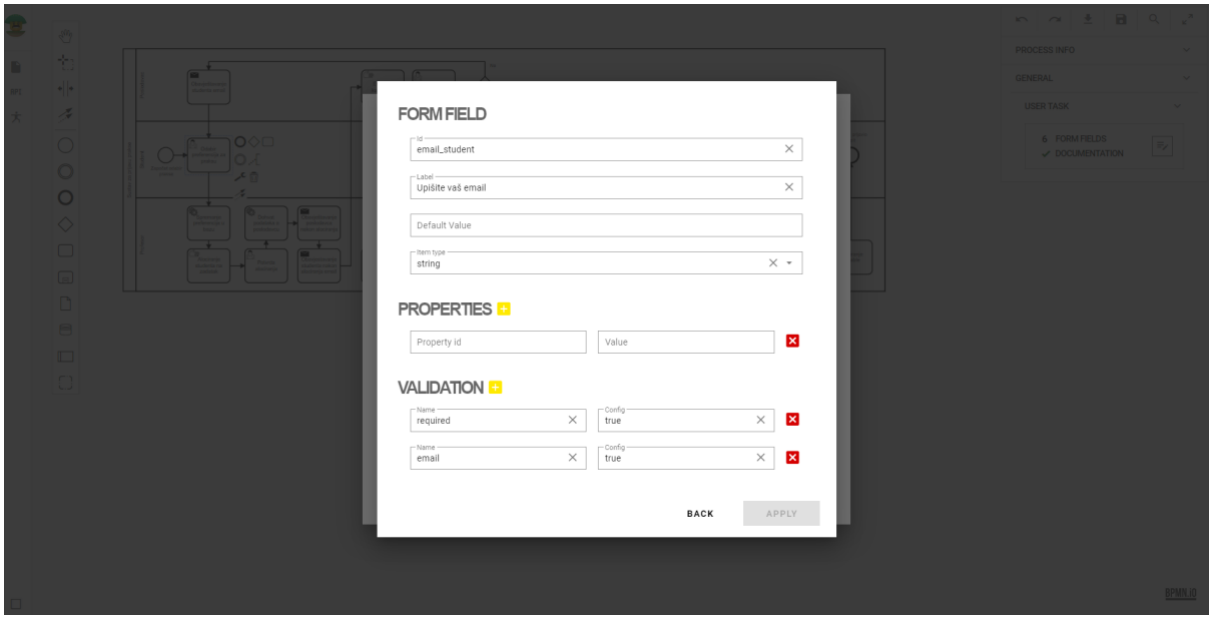
Slika 39: Prikaz detalja korisničkog zadatka (Izvor: Autor)

Klikom na gumb pored osnovnih informacija elementa, otvara se dijaloški okvir specifikacije korisničkog zadatka. U gornjoj strani dijaloga stoji dedicirana dokumentacija, a s donje strane je tablica koja prikazuje polja unutar forme koje je potrebno definirati.



Slika 40: Prikaz funkcionalnosti brisanja polja forme korisničkog zadatka (Izvor: Autor)

Polja forme je moguće brisati, a to se radi pritiskom na gumb briši. Otvori se dijalog koji traži dodatni pristanak na akciju. Samo brisanje je moguće poništiti, dovoljno je odustati ili ne spremiti promjenu.

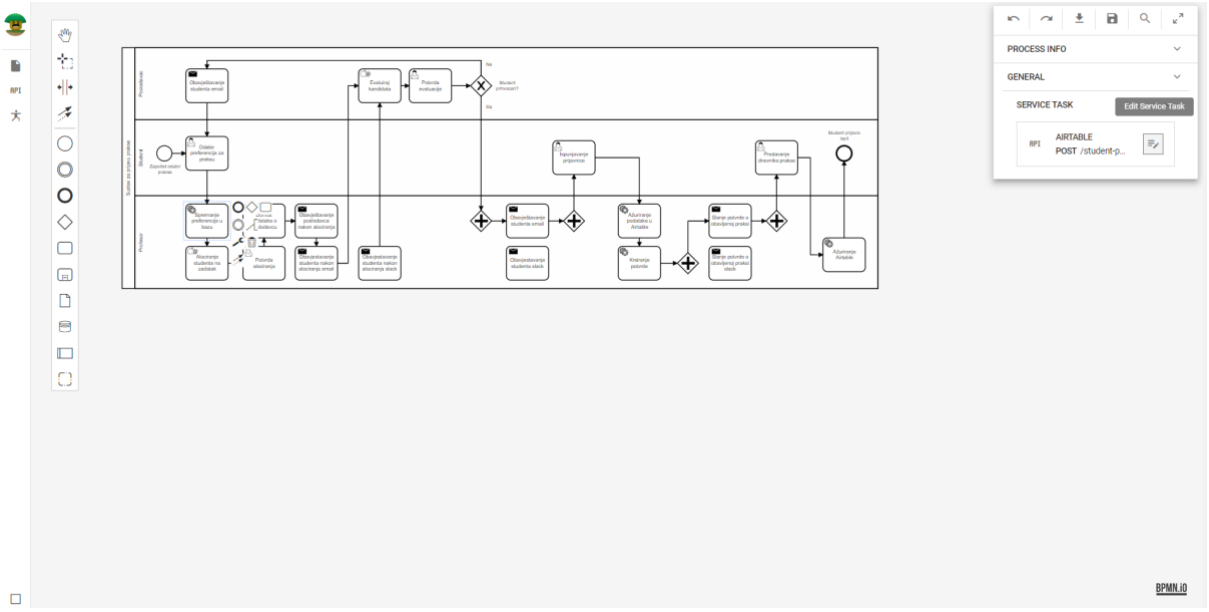


Slika 41: Prikaz funkcionalnosti uređivanja polja forme korisničkog zadatka (Izvor: Autor)

Pritiskom na gumb uredi (eng. Edit) otvara se dijalog koji omogućuje specifikaciju polja. U gornjem dijelu definirane su generalne opcije elementa, a ispod one specifičnijeg karaktera.

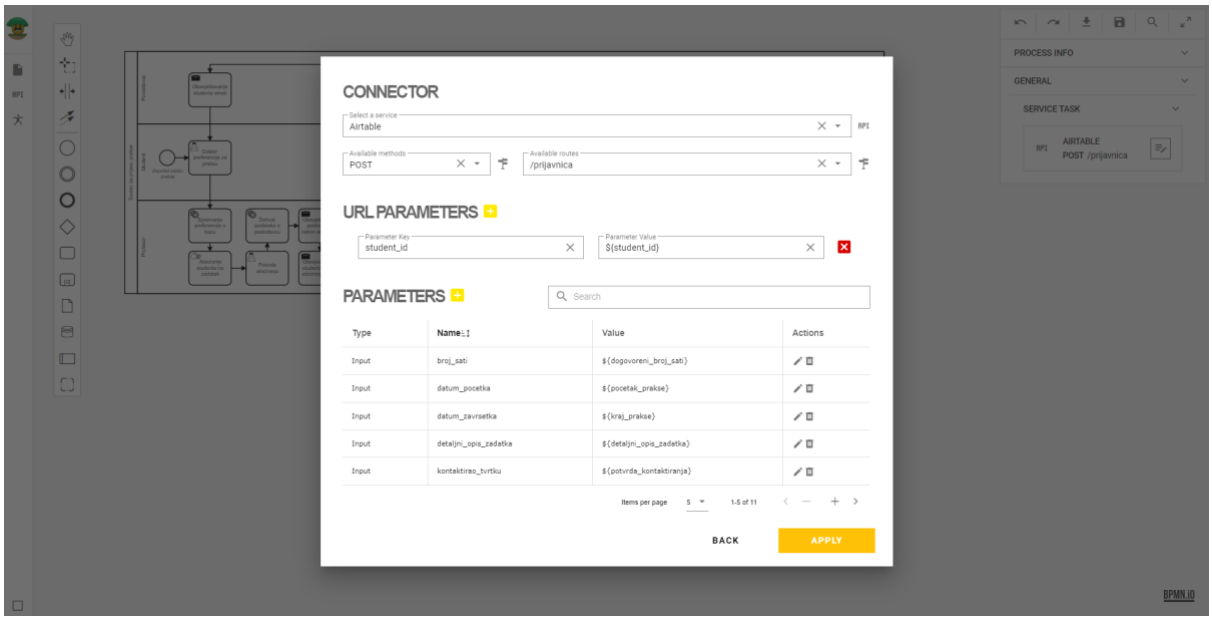
Slika 42: Prikaz funkcionalnosti kreiranja polja forme korisničkog zadatka (Izvor: Autor)

Kao i kod uređivanja, otvara se isti dijaloški okvir za dodavanje novog polja. Potrebno je definirati generalne informacije, a dodatna svojstva i validaciju su opcionalne postavke.



Slika 43: Prikaz otvaranja detalja servisnog zadatka (Izvor: Autor)

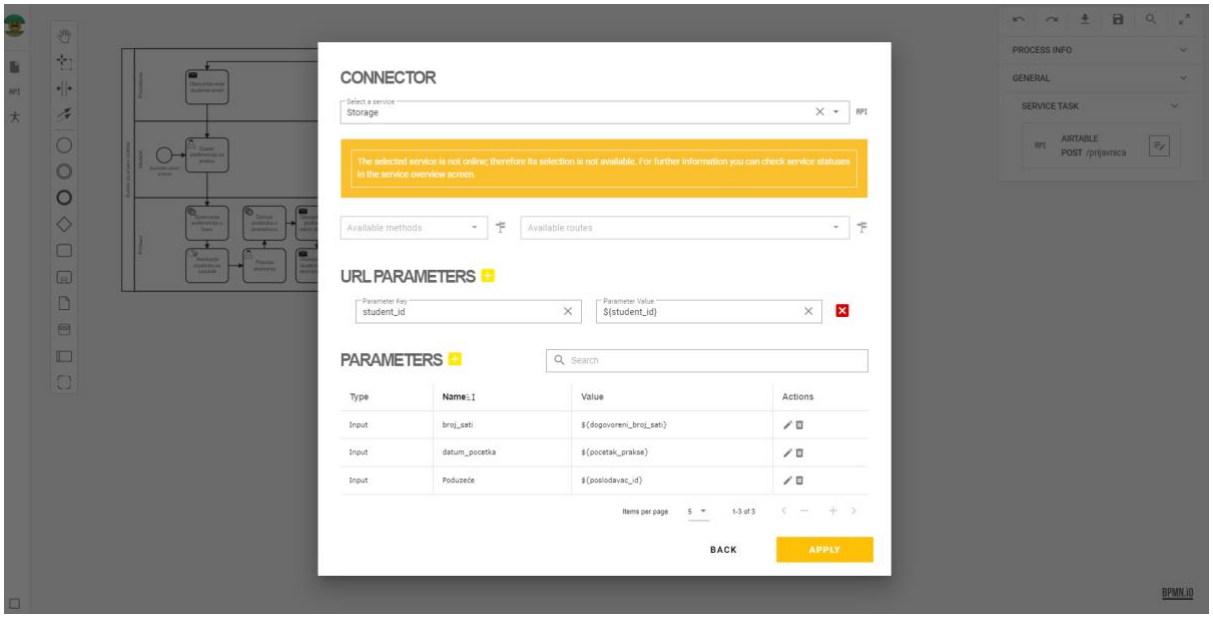
Kod otvaranja definiranog servisnog zadatka (eng. Service task) postupak upravljanja je sličan. Pritiskom na dedicirani gumb otvara se dijalog koji prikazuje i omogućuje detaljniji pregled, odnosno opciju uređivanja ili definiranja.



Slika 44: Prikaz detalja servisnog zadatka (Izvor: Autor)

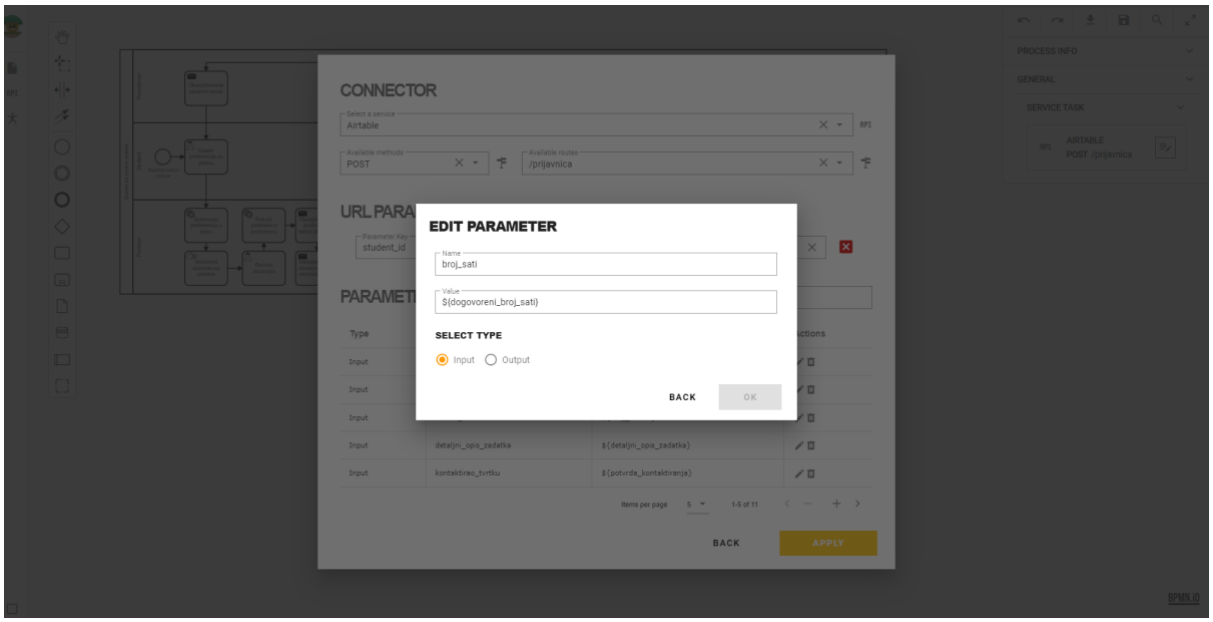
S gornje strane dijalog nalaze se informacije potrebne za definiciju konektora, a ispod konektora se nalaze url parametri koje korisnik može definirati. Konektor je u principu prethodno definirani servis u sekciji servisi. Konektor omogućuje izbor jednog od tih servisa.

Osim imena servisa, potrebno je definirati vrstu HTTP metode, kao što je primjerice GET ili POST. U donjem dijelu dijaloga nalazi se tablica prikazuje parametre koje taj zadatak stavlja u proces, odnosno sustav.



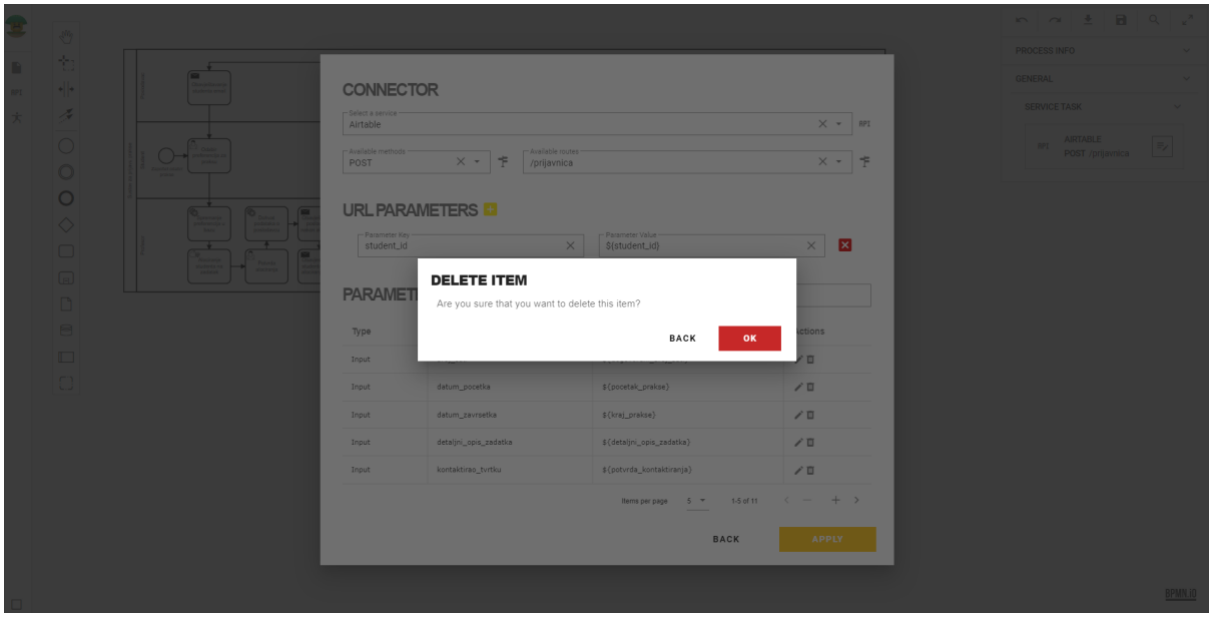
Slika 45: Prikaz notifikacije o nedostupnosti servisa u servisnom zadatku (Izvor: Autor)

Ukoliko je odabran servis koji nije aktivan, utoliko se isti smatra nedovoljnim za samu definiciju, pa iz toga slijedi obavijest o nepovoljnom statusu. Pregled svih statusa servisa je dostupan u sekciji servisi.



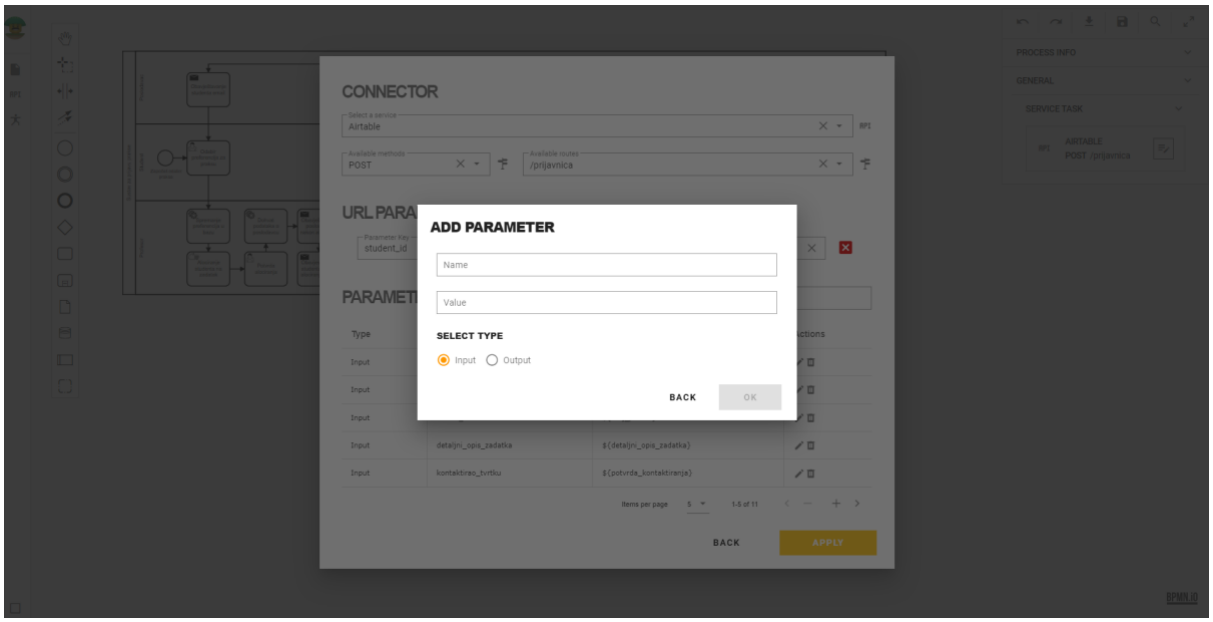
Slika 46: Prikaz uređivanja parametra servisnog zadatka (Izvor: Autor)

Parametri se mogu uređivati, a može se izmijeniti ime, vrijednost ili tip samog parametra.



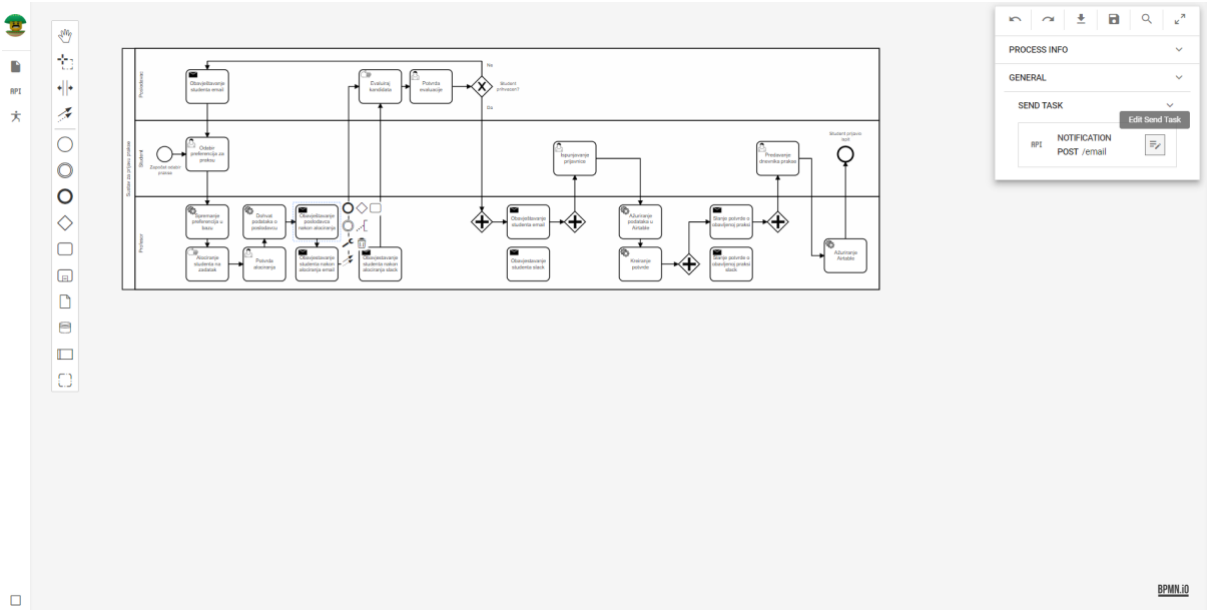
Slika 47: Prikaz brisanja parametra servisnog zadatka (Izvor: Autor)

Brisanje, kao i u prethodnim primjerima, radi na identičan način.



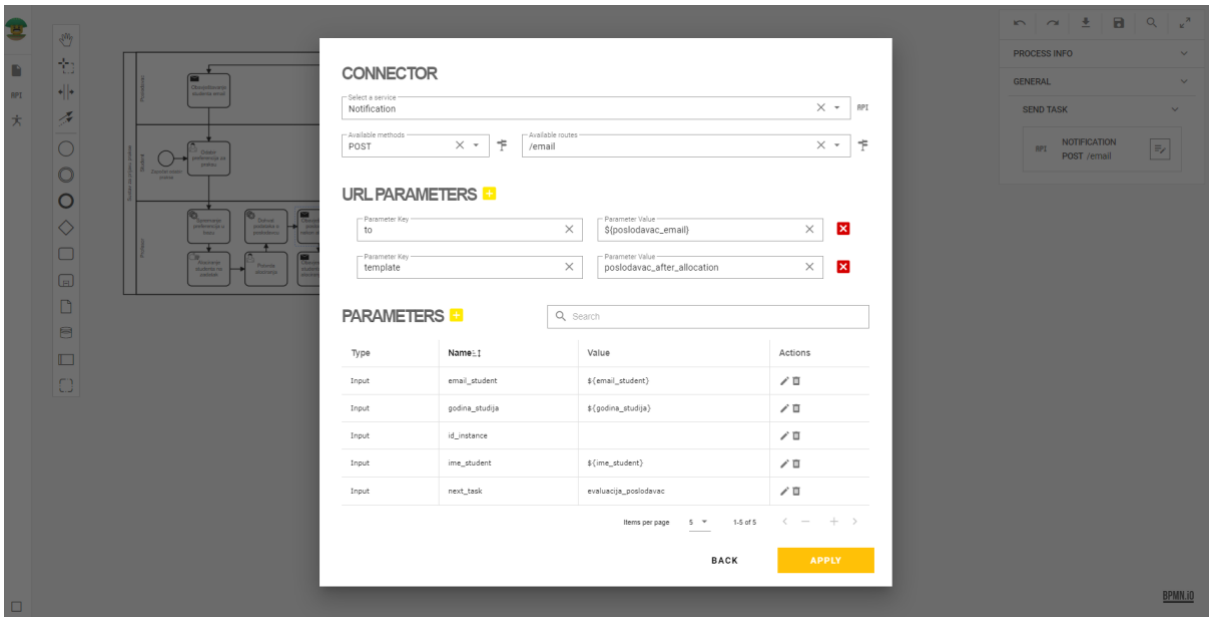
Slika 48: Prikaz dodavanja parametra servisnog zadatka (Izvor: Autor)

Dodavanje parametra je moguće pritiskom na gumb pokraj naslova tablice. Korisnik definira ime i vrijednost te zadani tip parametra zadatka.



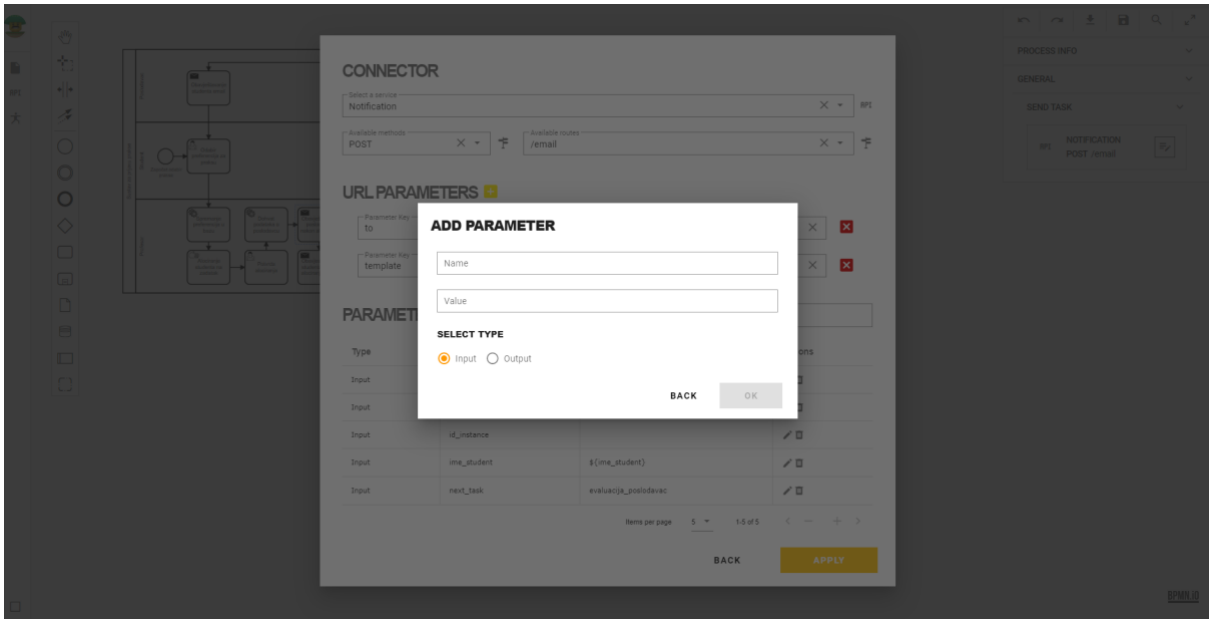
Slika 49: Prikaz otvaranja detalja pošiljateljskog zadatka (Izvor: Autor)

Otvaranje tzv. pošiljateljskog zadatka (eng. Send task) radi pritiskom na gumb pored osnovnih informacija.

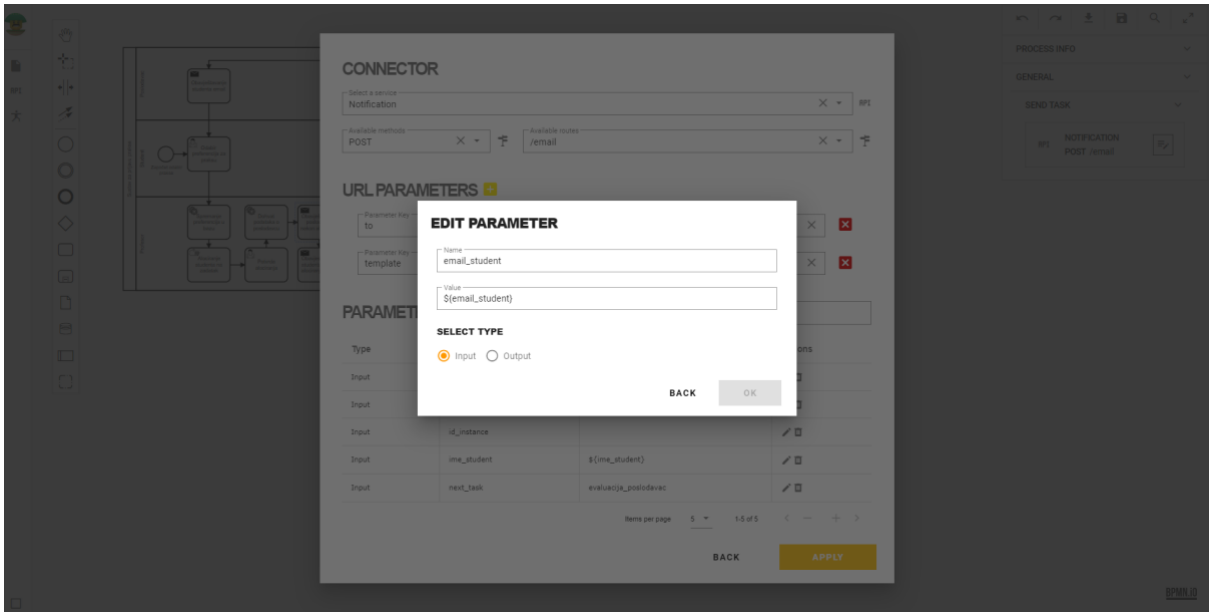


Slika 50: Prikaz detalja pošiljateljskog zadatka (Izvor: Autor)

Dijaloški okvir je jednak onom servisnom, pa samim time je i interakcija s tim dijalogom zapravo ista.

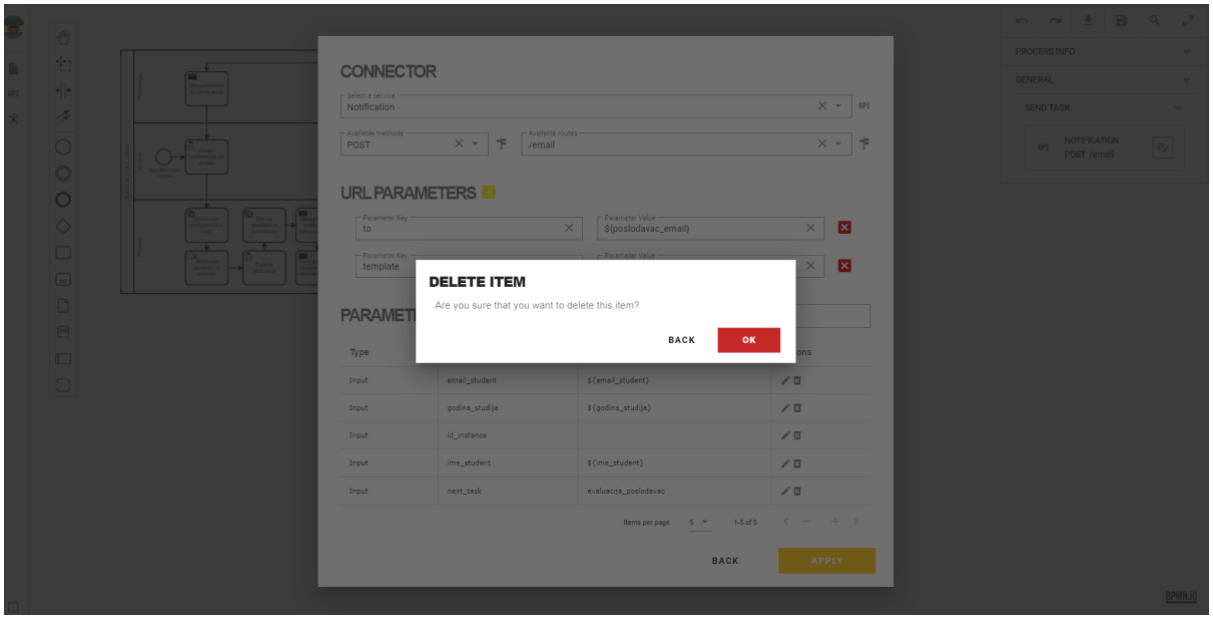


Slika 51: Prikaz dodavanja parametra pošiljateljskog zadatka (Izvor: Autor)



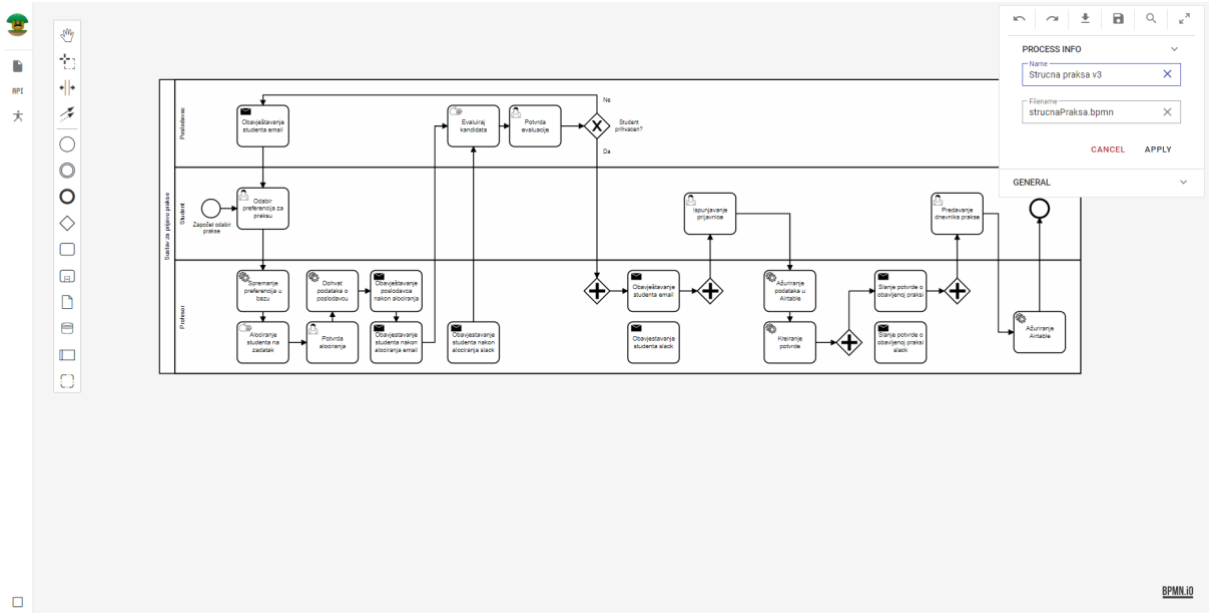
Slika 52: Prikaz uređivanja parametra pošiljateljskog zadatka (Izvor: Autor)

Dodavanje parametara i izmjena se vrši na sličan način. Potrebno je definirati, odnosno izmijeniti ime, vrijednosti i tip parametra unutar procesa.



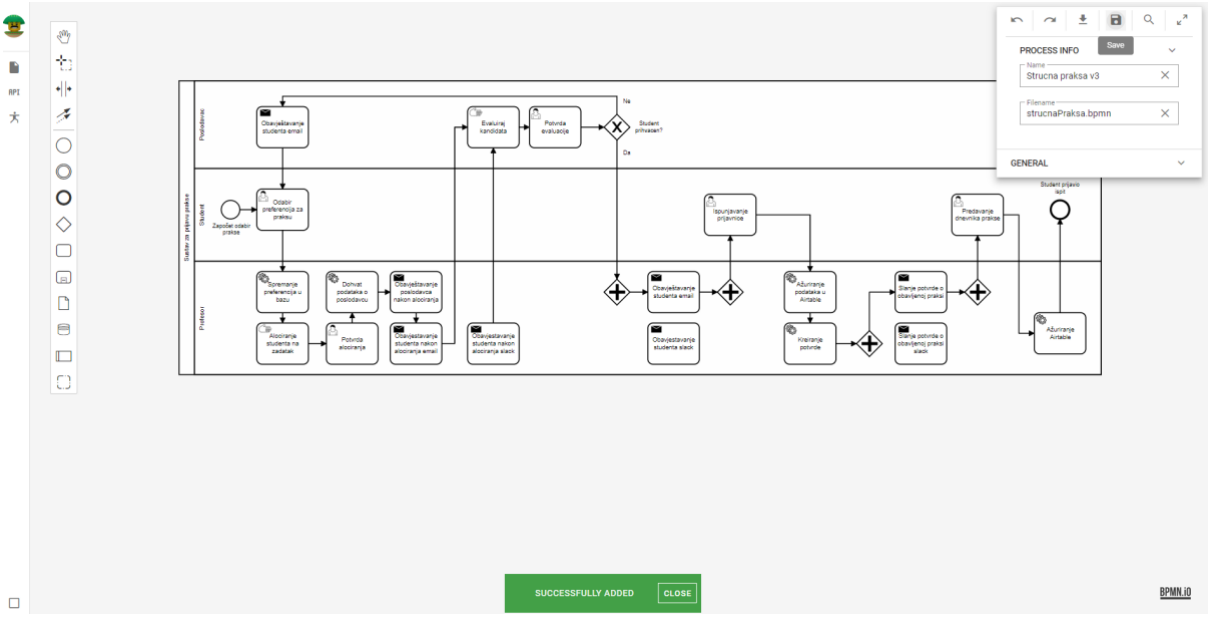
Slika 53: Prikaz brisanja parametra pošiljateljskog zadatka (Izvor: Autor)

Brisanje je također moguće, a ono se vrši pritiskom na gumb u tablici, odnosno retku procesa. Potom, otvoreni dijalog traži dodatni pristanak.



Slika 54: Prikaz unosa osnovnih informacija procesa (Izvor: Autor)

Kod spremanja procesa, potrebno je ispuniti procesne informacije, a one se definiraju u procesnim svojstvima. Ako se iste ne definiraju, spremanje nije moguće.

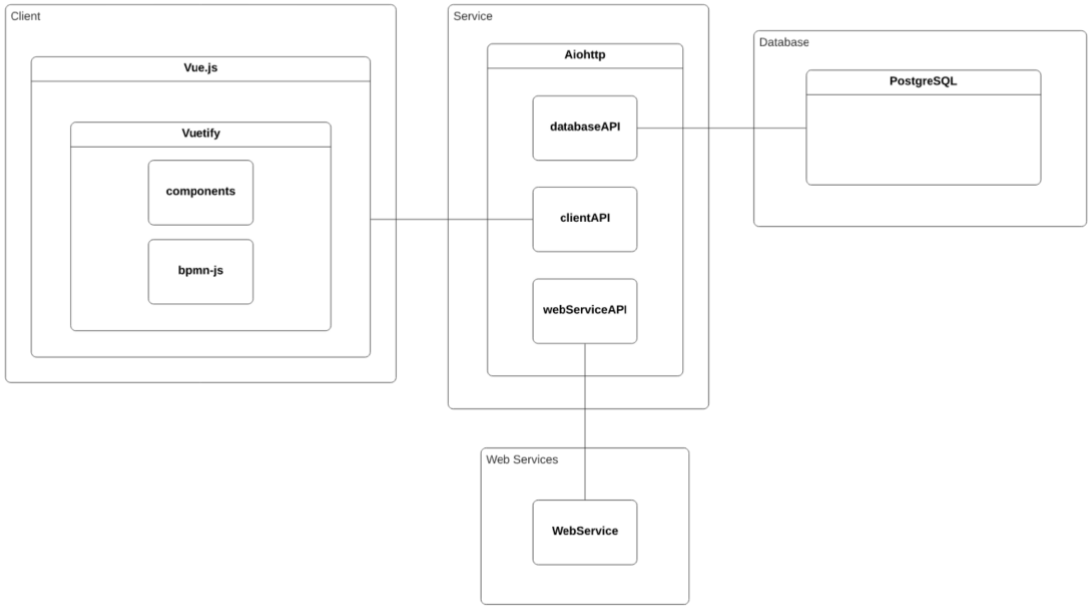


Slika 55: Prikaz spremanja procesa (Izvor: Autor)

Spremanje je moguće kada su sve procesne informacije popunjene. Spremanje se vrši pritiskom na gumb u samom u zaglavlju kartice, odnosno panela. Tu dolazi do kraja modeliranja jednog procesa i korisnika se seli na prethodni ekran. Time završavaju opisane funkcionalnosti implementirane aplikacije.

3. OPIS KORIŠTENIH TEHNOLOGIJA

3.1. GENERALNI PRIKAZ



Slika 56: Shematski prikaz korištenih tehnologija (Izvor: Autor)

Shema prikazuje korištene tehnologije pri izradi web aplikacije. Kao baza podataka korišten je PostgreSQL. Web servisi nisu dio ove aplikacije i gleda ih se kao pogled na crnu kutiju (eng. Black-box), dakle ne ulazimo u njihovu implementaciju. Kao aplikacijski okvir korišten je Aiohttp koji je logički podijeljen u tri podsustava: sučelje prema bazi podataka, sučelje prema web servisima i sučelje prema klijentu. Za implementaciju klijenta korišten je aplikacijski okvir Vue.js, a za komponente korisničkog sučelja (eng. User interface, UI) korišten je Vuetify. Za usluge modeliranja i održavanja logike upravljanja kontrolom samih dijagrama koristi se tzv. bpmn-js modul.

3.2. VUE.JS

Vue.js je progresivni aplikacijski okvir koji se koristi za izgradnju klijentskog dijela web-aplikacije s pogleda troslojne arhitekture [7]. Troslojna arhitektura je vrsta softverske arhitekture koja organizira aplikaciju u tri sloja:

1. Prezentacijski sloj ili bolje rečeno – korisničko sučelje (eng. User interface)
2. Aplikacijski sloj – sloj u kojem se procesiraju podaci i

3. Podatkovni sloj – zadnji sloj u kojemu se podaci skladište. [8]

Napravljen je kao pero-verzija AngularJSa. AngularJS je softver otvorenog koda koji se koristi poglavito za razvoj web aplikacija. Vodi i razvija ga tzv. Angular tim u Googleu, a favorizirani jezik je Microsoftov TypeScript. Naime, Vue je vrlo dobro dokumentiran i mala je krivulja svladavanja tehnologije (eng. Learning curve) [9] [10]. Vue.js se u projektu koristi za implementaciju klijentske logike. Danas Vue.js koristi napredne mogućnosti zbog kojih je jedan od lidera na tržištu u segmentu korištenja frontend aplikacijskih okvira. Samo neke od karakteristika su:

1. Virtualni DOM – prednost virtualnog DOM-a (eng. Document Object Model) nad onim običnim je činjenica da virtualni DOM je mnogo brži u samome radu jer prilikom izmjena elemenata stablaste strukture mijenja samo one elemente koji su pogođeni promjenom, dok se ostali ne diraju. Što nije slučaj običnog DOM-a;

2. Dvostrano povezivanje (eng. two-way binding) je svojstvo sustava u kojemu se korisničko sučelje ažurira prilikom izmjene podataka i kada korisničko sučelje ima mogućnost ažuriranja podataka. Dvostrano povezivanje je korisno radi jednostavnosti pri implementaciji funkcionalnosti, no problem nastaje kada aplikacija postane preglomazna jer nastaju problemi s performansama;

3. Jedno-datotečne komponente (eng. Single-file components) nudi mogućnost korištenja HTML-a (eng. HyperText Markup language), logika u jeziku JavaScript ili TypeScript i CSS-a (eng. Cascading Style Sheets) u jednoj datoteci što rezultira time da je kod čitljiviji i uredniji. Navedeno svojstvo postaje problem ako projekt postane prevelik u smislu broja samih datoteka unutar projekta.

4. MVVM (eng. Model-View-ViewModel) arhitekturalni obrazac je vrsta softverskog obrasca koji radi separaciju razvoja korisničkog sučelja od razvoja poslovne logike, odnosno podataka;

3.3. VUETIFY

Vuetify je Vue UI (eng. User interface) biblioteka koja nudi već vizualno izgrađene komponente koje olakšavaju dizajnerski dio razvijanja aplikacije i uvelike ubrzavaju rad. Baziran je na Material dizajnu (eng. Material Design) kreiranog od strane Googlea. Primjerice,

nudi elemente kao što su dugme (eng. button), razni form inputi i sl. napravljeni od JavsScripta i CSS-a [11].

3.4. AIOHTTP

Aiohttp je asinkroni HTTP klijent/poslužitelj za modul Asyncio i programski jezik Python [12]. Aiohttp osim asinkronosti nudi mogućnost korištenja Web pristupnih točaka (eng. Websocket) za poslužitelja i klijenta te uključuje i međuprogramsku logiku (eng. Middleware). U radu se koristi isključivo kao poslužitelj, a odabran je baš zbog asinkronih mogućnosti.

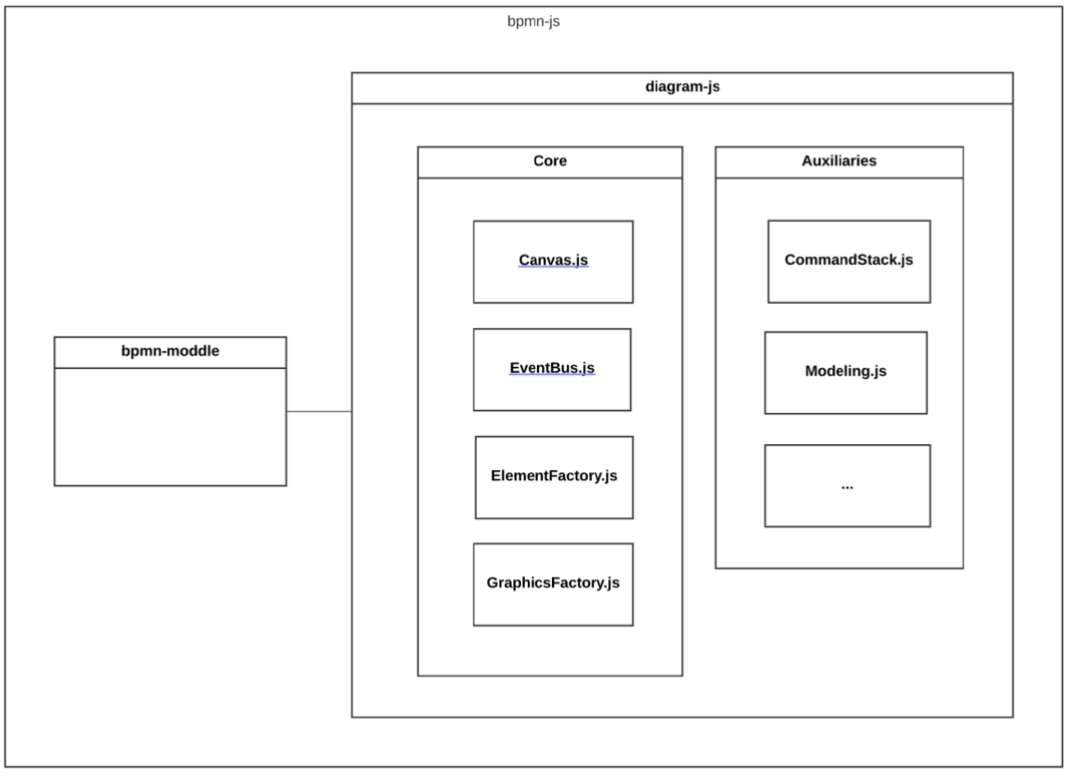
3.5. PONY ORM

Pony ORM je softverski proizvod koji služi za objektno-relacijsko mapiranje [13]. Mapiranje je tehnika pretvorbe tipa podataka između sustava koristeći objektno-orijentirani jezik ili objektno-orijentiranu funkcionalnost programskog jezika. Korištenje takvog alata za mapiranje podataka smanjuje potrebu za korištenjem programske logike koncentriranu u bazu podataka te ju translatira na servisni dio aplikacije i time ubrzava razvoj samog sustava.

3.6. POSTGRESQL

PostgreSQL je objektno-relacijski sustav baze podataka otvorenog koda koji koristi i proširuje SQL jezik u kombinaciji s mnogim značajkama koje sigurno pohranjuju podatke. Naime, nastao je na kalifornijskom sveučilištu Berkley 90-ih godina prošlog stoljeća. Prvotno razvijen da radi na UNIX operacijskim sustavima, a nakon se tek proširio na ostale OS-ove. Danas se koristi zbog svoje dokazane arhitekture, pouzdanosti, velikim skupom značajki, proširivošću i predanošću zajednice otvorenog koda koja stoji iza softvera [14]. U projektu se koristi kao upravitelj baze podataka.

3.7. BPMN-JS



Slika 57: Shematski prikaz bpmn-js komponente (Izvor: Autor)

Već spomenuta komponenta koja služi za izradu BPMN dijagrama poslovnog procesa, sastoji se od dvije cjeline, kao što je i vidljivo na slici 57. Sama komponenta nalazi se na Github repozitoriju i javno je dostupna [15].

3.7.1. BPMN-MODDLE

Budući da se BPMN interno pretvara u XML tip datoteke, potreban je modul koji je zadužen za serijalizaciju/deserijalizaciju XML dokumenata u tzv. BPMN stablasti model (eng. BPMN model tree) koji je u svojoj osnovi objektna reprezentacija stanja modela u programskom jeziku JavaScript.

3.7.2. DIAGRAM-JS

Diagram-js nudi programsko rješenje u pogledu modeliranja, a sam po sebi je modul koji čine više raznih modula koji su zaduženi za određenu skupinu posla kao što je i vidljivo u shematskom prikazu arhitekture. Podijeljen je na tzv. servise (eng. Services), odnosno jezgru

(eng. Core) i pomoćne (eng. Auxiliary) servise. Budući da ga čine više zasebnih modula, a sami moduli podijeljeni u dvije skupine, tako će biti prikazani i objašnjeni.

3.7.3. JEZGRENI SERVISI

Modul *Canvas* pruža programsko sučelje u vidu manipulacije grafikom, grafičkim elementima, prikazom dijagrama, zoom opciju i mnoge druge opcije. Modul *EventBus* nudi uslugu globalnog simplex komunikacijskog kanala, tj. stanja i radi tako da se događaji pri radu s dijagramom odašilju kroz taj kanal. Kanal nudi opciju slušanja, pa se time dobiva modularni pristup komunikaciji i gibanju podataka, odnosno paradigmi baziranoj na događajima (eng. event-based). Modul *ElementFactory* kreira vezu između elemenata te pruža ispravnost u povezivanju i automatsko ispravljanje pogrešaka u modeliranju. Radi s internim podacima. Služi kao registar elemenata jednog dijagrama. Nudi programsko sučelje za pretraživanje i filtriranje. Modul *GraphicsFactory* je modul zadužen za samu grafičku kreaciju elemenata unutar jednog dijagrama.

3.7.4. POMOĆNI SERVISI

Među mnogim ostalim, tj. pomoćnim modulima treba istaknuti: Modul *CommandStack* koji je zadužen za čuvanje, odnosno manipulaciju povijesti odrađenih akcija i Modul *Modeling* koji nudi programsko sučelje na samom dijagramu.

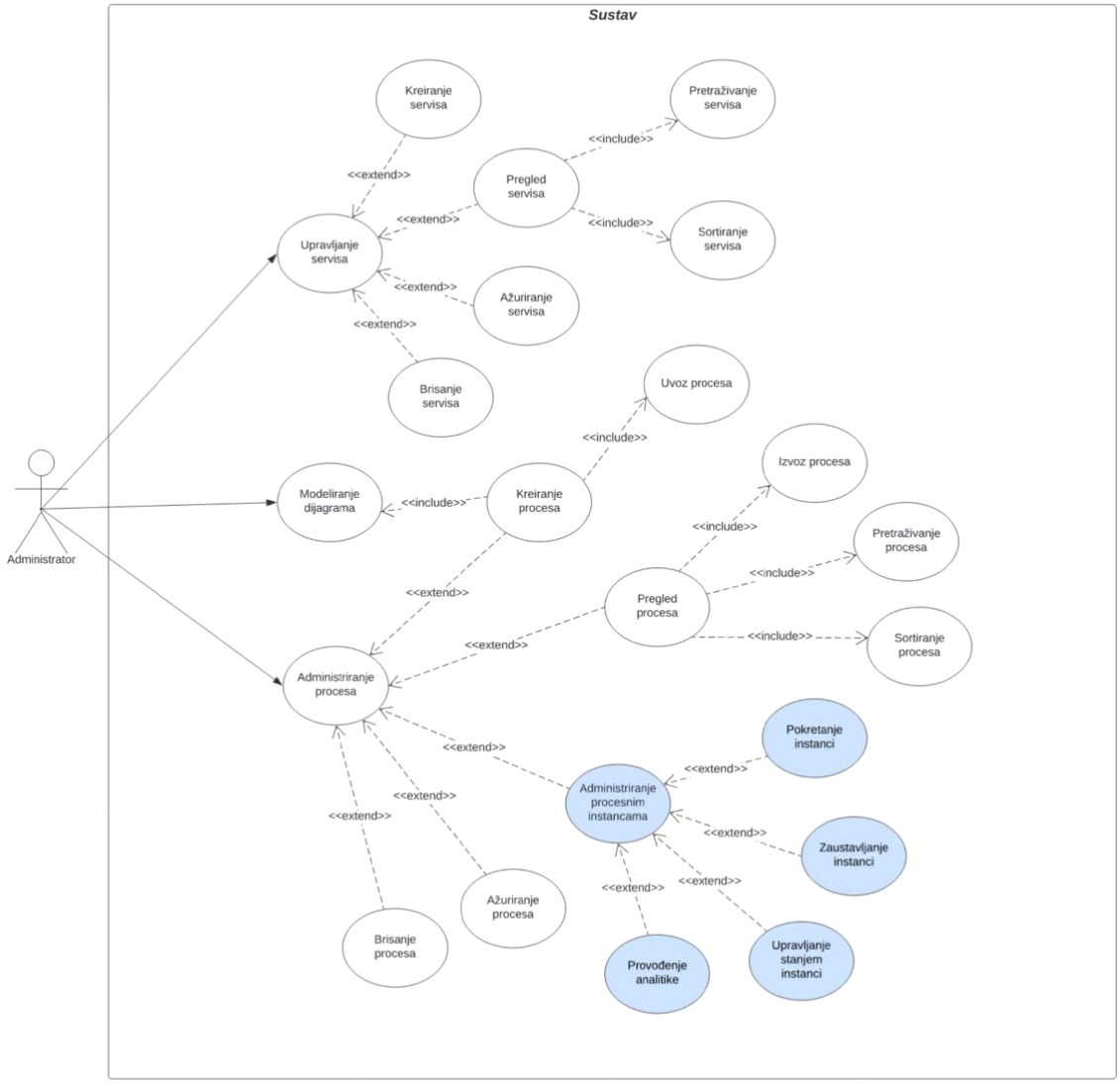
4. MOGUĆA POBOLJŠANJA

4.1. PREGLED SUSTAVA

Jedno od mnogih mogućih poboljšanja je pokretanje samih procesa, odnosno pokretanje procesnih verzija. Naime, implementirani sustav je zasad „statičan“, trenutno je moguće modelirati i uvoziti procese, ali ne i pokretati. Kako bi se ti procesi pokretali potrebno je napraviti sljedeće:

1. Implementirati tzv. BPMN stroj (eng. BPMN engine) na poslužitelju koji preuzima generiranje podataka na temelju XML datoteke procesne verzije,
2. Omogućiti klijentu komunikaciju s BPMN strojem radi uvida u stanje procesnih instanci,
3. Implementirati na klijentu dezignirano sučelje za operacije koje bi također moglo biti u tabličnom obliku,
4. Omogućiti BPMN stroju komunikaciju s bazom podataka radi perzistencije i očuvanja integriteta podataka i
5. Implementirati tablice u bazi podataka za povijesne podatke bi se mogla koristiti drugačija tehnologija, primjerice NoSQL baza podataka, budući da integritet podataka nije toliko izražen kao u relacijskim bazama podataka.

4.2. DINAMIČKI PRIKAZ SUSTAVA

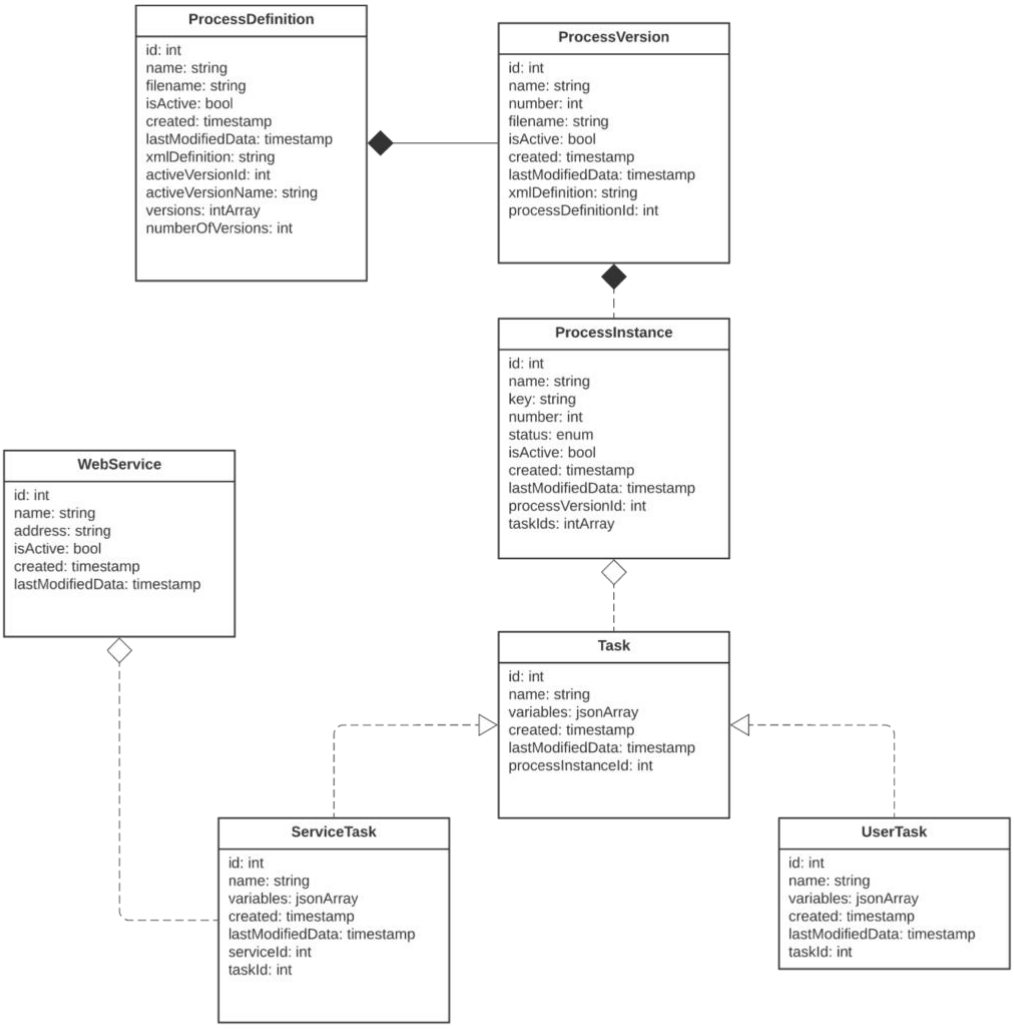


Slika 58: UML Dijagram obrasca uporabe moguće nadogradnje (Izvor: Autor)

Administriranje procesnim instancama može se promatrati kao novi slučaj uporabe administriranja procesa. Pokretanje instanci zapravo nudi mnogo mogućnosti. Hoće li se procesna verzija pokretati iz same aplikacije ili će se pokretati iz nekog drugog sučelja? Primjerice, virtualnog asistenta (eng. Chatbot) ili neke druge opcije? Tada bi aplikacija zaista postala administratorska. Zaustavljanje instanci se može promatrati na jednak način, s time da bi se mogao staviti naglasak na adminstratorska prava pri upravljanju samog procesa, gdje akteri procesa ne bi bili svjesni postojanja procesa. Pod upravljanjem stanja instance misli se na varijable u procesu koje su unesene pomoću korisničkih zadataka (eng. User task) ili servisnih zadataka (eng. Service task) proizvedenih u suradnji s nekim od mnogobrojnih

unesenih servisa u sustav. Ona najvažnija stavka jest ta da bi se mogla provoditi analitika nad dinamičkim procesima sa stvarnim procesnim instancama jer kao što je već spomenuto, kako bi se procesi mogli unaprijediti, odnosno optimizirati, potrebno ih je i analizirati.

4.3. STATIČKI PRIKAZ SUSTAVA

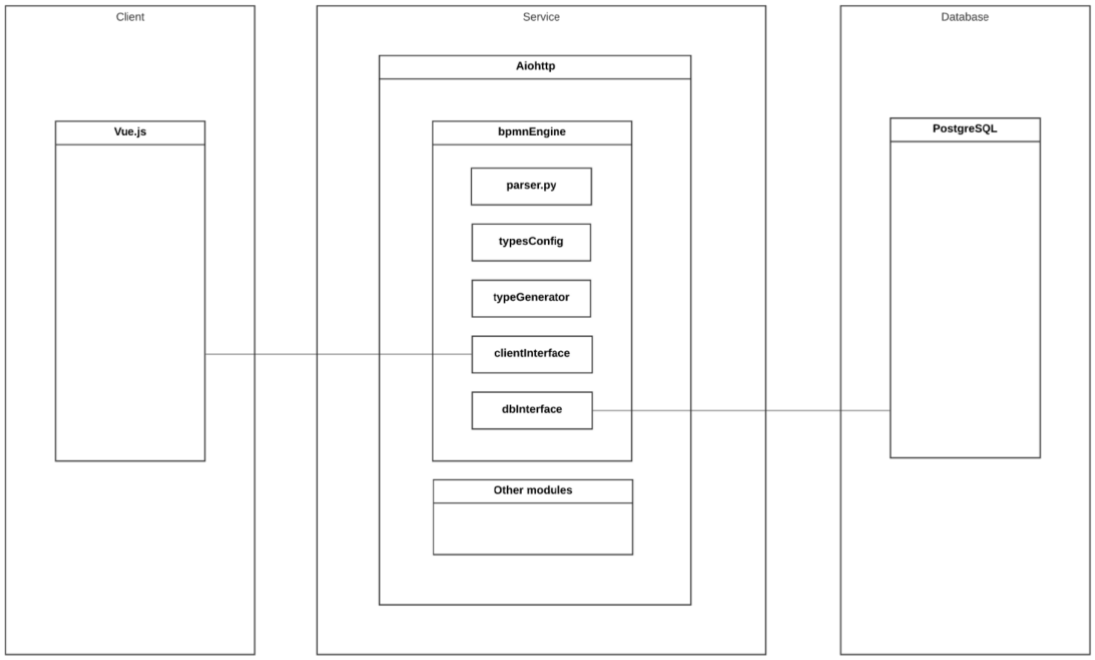


Slika 59: UML Klasni dijagram moguće nadogranje (Izvor: Autor)

Prikazana je kompozicijska spona između procesne definicije, verzije i instance. Nastavno na trenutnu implementaciju, pokretanje procesnih instanci radilo bi se tako da se procesne definicije i verzije definiraju kao i do sada, modeliranjem i uvozom u sustav te kreiraju pomoću

BPMN stroja. U scenariju gdje korisnik pokreće instancu kroz aplikaciju, išao bi zahtjev prema BPMN stroju s određenom verzijom, BPMN stroj bi parsirao verziju, odnosno XML datoteku, spremio instancu u bazu i dostavio obavijest administratoru. Ono što je tu zanimljivo jest činjenica da se taj proces može pokrenuti kroz koje god sučelje je implementirano, a aplikacija i dalje ima svoju funkciju u vidu nadzora, modeliranja i upravljanja procesima. Budući da prilikom parsiranja procesa, treba proizvesti i podatkovnu strukturu jednaku procesnim zadacima, iste treba i spremiti. S time da treba promisliti bi li se trebalo implementirati BPMN stroj tako da se procesni zadaci generiraju pri prvom parsiranju, tj. njihovoj kreaciji ili bi se trebali kreirati ad-hoc. Naime, ukoliko se radi o prvoj situaciji, utoliko bi i skladište podataka trebalo biti veće ili ako se radi o drugoj soluciji tada bi računalni poslužitelj trebao imati dovoljno računalnih resursa da može podnijeti znatnije opterećenje. Zadaci bi bili temeljeni kao i dosad, ali bi servisni i pošiljateljski zadaci dobili mogućnost povezivanja s web servisima, tj. njihovom logikom i produkcijom informacija. Korisnički zadaci bi se mogli iskoristiti kao oblik sučelja, odnosno formulara koje korisnik treba ispuniti.

4.4. OPIS DODATNIH TEHNOLOGIJA



Slika 60: Shematski prikaz nadogradnje

BPMN stroj može se promatrati kao web servis odvojen od trenutnog servisa, ali može se i promatrati kao jedan od modula uklopljen u poslužitelj kojeg čine manji moduli. Dakako prvi

modul bi bila mogućnost pretvaranja priloženog XML-a verzije u hijerarhijsku podatkovnu strukturu nalik onoj koju bpmn-js koristi interno prilikom interakcije s dijagramom. Ta podatkovna struktura bi se izgradila tako da se mapiranje tipova iz Konfiguracije tipova tzv. (eng. typesConfig) modula ugrade u proces. Time bi se proces usklađivanja tipova sa standardiziranom verzijom BPMN-a maksimalno pojednostavio. Sama generacija standardiziranih tipova ne bi bila u nadležnosti parsera, već bi taj teret pao na leđa tzv. Generatora tipova (eng. typeGeneratora). Budući da je potrebno programsko sučelje prema klijentu, nužno je implementirati isto. Optimalna solucija bi bila koristiti tehnologiju Web pristupne točke (eng. Websocket) jer je nužno da web aplikacija ima pristup podacima u realnom vremenu zbog same administratorske prirode. Naime, ako bi se implementirala razna sučelja nad BPMN strojem tada bi bilo još važnije imati pristup u realnom vremenu. Potrebna je i perzistencija samih podataka, stoga je potrebno i povezati BPMN stroj s bazom podataka.

ZAKLJUČAK

Poslovne organizacije danas moraju kontinuirano razvijati kompetitivnu prednost, a poznavanje i aktivno upravljanje ključnim poslovnima procesima omogućuje jednu takvu prednost. Softverska rješenja koja omogućuju uvid u problematiku poznavanja procesa su veoma poželjna.

Ova web aplikacija čija je namjena upravo praćenje, upravljanje i modeliranje procesnih modela koji su temeljeni na grafičkom standardu za modeliranje poslovnih procesa BPMN – implementirana je korištenjem suvremenih tehnologija. Korisničko sučelje je kreirano u provjerenom i etabliranom aplikacijskom okviru Vue.js što jamči optimiziranost u radu, moderan izgled i intuitivno korištenje. Modeliranje i sve u vezi BPMN logike postignuto je nezaobilaznom bpmn-js komponentom. Poslužiteljska logika je razvijena u asinkronom aplikacijskom okviru koji garantira visoke performanse, a baza podataka PostgreSQL je gotovo standard među relacijskim bazama otvorenog koda.

SAŽETAK

Svaka organizacija ima ili ju čini makar jedan poslovni proces. Problem kod poslovnih procesa je taj što su veoma često složeni i kompleksni. One poslovne organizacije koje ne obraćaju pozornost na svoje procese, tj. imaju neoptimizirane procese, nestaju s tržišta.

Naime, poznavanje i kontinuirano praćenje poslovnih procesa organizacijama nudi mogućnost unaprjeđivanja svojih poslovnih procesa tako da povećavaju efikasnost i efektivnost. Dakle, poslovni procesi predmet su razmatranja kako bi se isti obavljali na optimalan način.

Kako bi se proces optimizirao, potrebno ga je prvo modelirati budući da se u današnje vrijeme analitika vrši nad modelima, a ne u stvarnom vremenu. Drugim riječima, nije preporučljivo istovremeno empirijski provoditi analizu i voditi poslovanje. Predmet ovog rada je web aplikacija koja nudi mogućnosti modeliranja, analize i nadzora procesa. Ova aplikacije je programsko, odnosno softversko rješenje za sve poslovne organizacije koje traže rješenje za uspostavu kontrole nad svojim poslovnim procesima.

Ključne riječi: poslovni proces, poslovna organizacija, optimalnost, analiza, web aplikacija, softversko rješenje

ABSTRACT

Every organization has or consists of at least one business process. The problem with business processes is that they are very often complex. Those business organizations that do not pay attention to their processes, i.e. have unoptimized processes, disappear from the market.

Namely, knowledge and continuous monitoring of business processes offer organizations the opportunity to improve their business processes in a way that increases efficiency and effectiveness.

Therefore, business processes are subject to consideration so that they are performed optimally. To optimize the process, it needs to be modeled first, since analytics nowadays is done on models, not in real-time. In other words, it is not advisable to conduct empirical analysis and business at the same time. The subject of this paper is a web application that offers modeling, analysis, and monitoring capabilities that solve the problem in question. This application is a software solution for all business organizations that are looking for a solution to establish control over their business processes.

Keywords: business process, business organisation, optimality, analysis, web application, software solution

LITERATURA

[1] Laguna Manuel; Marklun Johan. 2013. Business process modeling, simulation adn design. Boca Raton. CRC Press

[2] Brumec Josip; Brumec Slaven. 2018. Modeliranje poslovnih procesa. Zagreb. Školska knjiga.

[3] n.d. OMG. Pristupljeno: 5. Rujan 2022. https://www.omg.org/

[4] n.d. Gartner. Pristupljeno: 5. Rujan 2022. https://www.gartner.com/en/information-technology/glossary/business-process-management-bpm

[5] n.d. BPMN 2.0. Pristupljeno: 5. Rujan 2022. https://www.omg.org/spec/BPMN/2.0/

[6] n.d. UML. Pristupljeno: 5. Rujan 2022. https://www.uml.org/

[7] n.d. Vue . Pristupljeno: 5. Rujan 2022. https://vuejs.org

[8] **https://www.ibm.com/cloud/learn/three-tier-architecture**

[9] n.d. Vue docs . Pristupljeno: 5. Rujan 2022. https://vuejs.org/guide/extras/rendering-mechanism.html

[10] Wohlgethan Eric. „Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js. Bachelor thesis“. Faculty of Engineering and Computer Science, Hamburg, 2018.

[11] n.d. Vuetify - . Pristupljeno: 5. Rujan 2022.

[12] n.d. Aiohttp - . Pristupljeno: 5. Rujan 2022.

[13] n.d. PonyORM - . Pristupljeno: 5. Rujan 2022.

[14] n.d. PostgreSQL- . Pristupljeno: 5. Rujan 2022. https://www.postgresql.org/about/

[15] n.d. bpmn.io - . Pristupljeno: 5. Rujan 2022. https://github.com/bpmn-io

PRILOG

Github repozitorij klijenta: https://github.com/starcictoni/bpmn-admin

Github repozitorij servisa: https://github.com/starcictoni/python-bpmn-engine

Github repozitorij web servisa: https://github.com/starcictoni/internship-application-services