

Veb aplikacija Evrtić

Nemanja Murgaški

Fakultet tehničkih nauka

Univerzitet u Novom Sadu

Trg Dositeja Obradovića 6, 21000 Novi Sad

murgaski.sr60.2020@uns.ac.rs

Sažetak

U ovom radu opisana je aplikacija koja je namenjena kao podrška sistema eUprave, radi olakšanog načina upisa dece u vrtić u elektronskom obliku. Primenom ovog sistema se ostvaruje značajna ušteda vremena prilikom upisa dece Republike Srbije u vrtić.

Ključne reči

Informacioni sistem, eUprava, vrtić.

Uvod

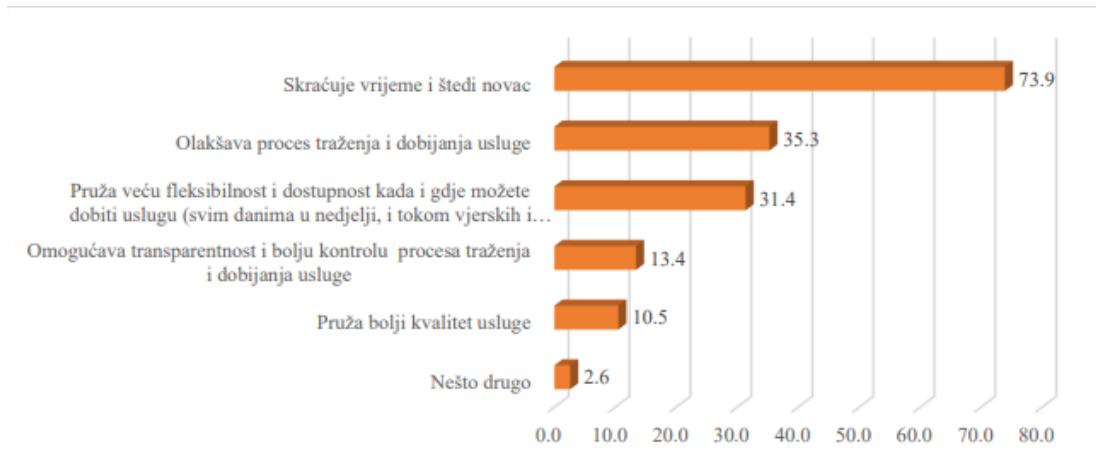
Republika Srbija je u odnosu na druge zemlje Evrope kasno uvela elektronski upis dece u vrtiće. Do pre nekoliko godina je osoba koja je htela da upiše dete morala da podnese zahtev lično u državnim službama gde je taj proces bio veoma mučan i za roditelje kao i za zaposlene u službi.

Nakon što je elektronski upis dece u vrtiće pušten u rad, ceo sistem počinje da radi nekoliko puta bolje i bez velikog broja grešaka.

Elektronska uprava uvodi elektronsku razmenu podataka i dokumenata, kojima se ukida potreba pribavljanja suvišne papirne dokumentacije i nepotrebnog čekanja za šalterom.

Sprovođenje postupaka u elektronskom obliku eliminiše potrebu stvaranja i čuvanja dokumentacije u papirnom obliku, čija izrada i arhiviranje koštaju i zahtevaju prostor za čuvanje materijala kod organa i stranaka.

Takođe se smanjuje uticaj ljudske greške prilikom elektronske razmene podataka i dokumenata.



Slika 1: Prednost podnošenja zahteva preko Euprave u odnosu na lično podnošenje zahteva. Statistika preuzeta sa [2] .

U konkretnom radu je napravljen sistem korišćenja Evrtića. Implementirana je veb aplikacija za elektronski upis deteta u vrtić gde osoba koja upisuje dete u vrtić može to da učini prilično jednostavno i bez prevelikog opterećenja.

Srodna istraživanja

Trenutno stanje je takvo da se upis sveo na samo nekoliko koraka, a ti koraci su:

- Pristup internetu i web sajtu <https://euprava.gov.rs/vrtic> [1]
- Zahtev za registraciju
- Logovanje na sistem
- Odlazak na stranicu za upis deteta
- Unos podataka i prikupljanje dokumentacije u elektronskom obliku

Име ●

Презиме ●

ЈМБГ ●

Пребивалиште

● Ознака за обавезно поље

Идентификациони документ (лична карта или пасош)
Приложите очитану, скенирану или фотографисану личну карту или пасош.
Уколико прилажете личну карту, потребно је снимити и приложити обе стране.

Тип документа ●

Број документа ●

Приложите документа

Максимално два документа, 3МБ сваки.
Допуштени формати: .pdf, .png, .jpg.

● Достављањем документа у електронском облику, потврђујете свој идентитет.

Како приложити идентификациони документ?
[Кликните овде](#)

Корисничко име и лозинка

Корисничко име ●

Унесите своју имејл адресу која ће представљати корисничко име.

Лозинка ●

Потврда лозинке ●

Унесите лозинку Поновите лозинку

Упишите лозинку која ће садржати најмање осам карактера, бар једно велико и мало слово, број и специјални карактер

Зачина лозинке

☐ Слажем се са Условима коришћења Портала еИД и прихватам опште услове и политику приватности пружања услуге електронске идентификације и шеме електронске идентификације Мој еИД

☐ Нисам робот

[Региструј ме](#) →

- Региструјте налог корисничким именом и лозинком**
 - Попуните форму за регистрацију. Потребно је приложити очитан, скениран или фотографисан лични документ (личну карту или пасош).
 - Послаћемо Вам мејл са линком за потврду адресе електронске поште. Потврдите адресу електронске поште у року од 24 сата.
 - Прегледаћемо достављене податке и у року од највише 48 сати одобрити Ваш захтев за регистрацију, о чему ћемо Вас обавестити путем мејла.

Пријава корисничким именом и лозинком Вам омогућава приступ основним услугама и функционалностима електронске управе.

2 Активирајте мобилну апликацију ConsentID

Пријава мобилном апликацијом ConsentID представља пријаву високог нивоа поузданости и омогућава Вам приступ свим услугама и функционалностима електронске управе.

3 Активирајте квалификовани електронски сертификат у клауду

Квалификовани електронски сертификат у клауду (KEC у клауду) Вам омогућава да користите свој квалификовани електронски потпис на даљину. Нисте везани за рачунар и потписивање можете да обавите у време и на месту које Вам одговара помоћу апликације ConsentID на мобилном уређају.

Постаните еГрађанин

Регистрацијом на Порталу еИД, активацијом мобилне апликације ConsentID и квалификованог електронског сертификата у клауду брзо и лако користите све услуге и функционалности еУправе.

[Више детаља](#) →

Слика 2: Регистрација на сајт Еуправе

Пријава корисничким именом и лозинком је пријава основног нивоа поузданости. [Сазнајте више.](#)

Корисничко име:
(Адреса електронске поште коришћена приликом регистрације)

Ваша лозинка:

[ПРИКАЖИТЕ](#)

Пријавите се

[Заборављена лозинка](#)

Немате налог на eid.gov.rs? Региструјте се [овде](#).

Слика 3: Пријава на сајт ЕУправе

еУПРАВА
Портал еУправа Републике Србије

Пријава деце у предшколску установу

Српски ▼

Назив установе

Предшколска установа "Хабиба Сточевић" Тутин

Општи подаци о упису детета

Врћић 1

Врћић 2

Врћић 3

Васпитна Група

Жељени месец уписа

Да ли сте конкурисали "Ако нисте конкурисали, унесите „не“ у неку другу установу и коју

Језик васпитно-образовног рада

Општи подаци о детету за које конкурисате:

Slika 4: Popunjavanje forme za upis deteta

Korišćene tehnologije

Za implementaciju klijentske aplikacije korišćen je programski jezik TypeScript[6] (framework Angular [5]).

Za implementaciju na serverskoj strani korišćen je programski jezik Golang [4] uz Docker [7] koji je zapravo aplikacija otvorenog koda koja automatizuje razvijanje aplikacija u kontejneru. Svi podaci se čuvaju u Mongo bazi podataka[3] .

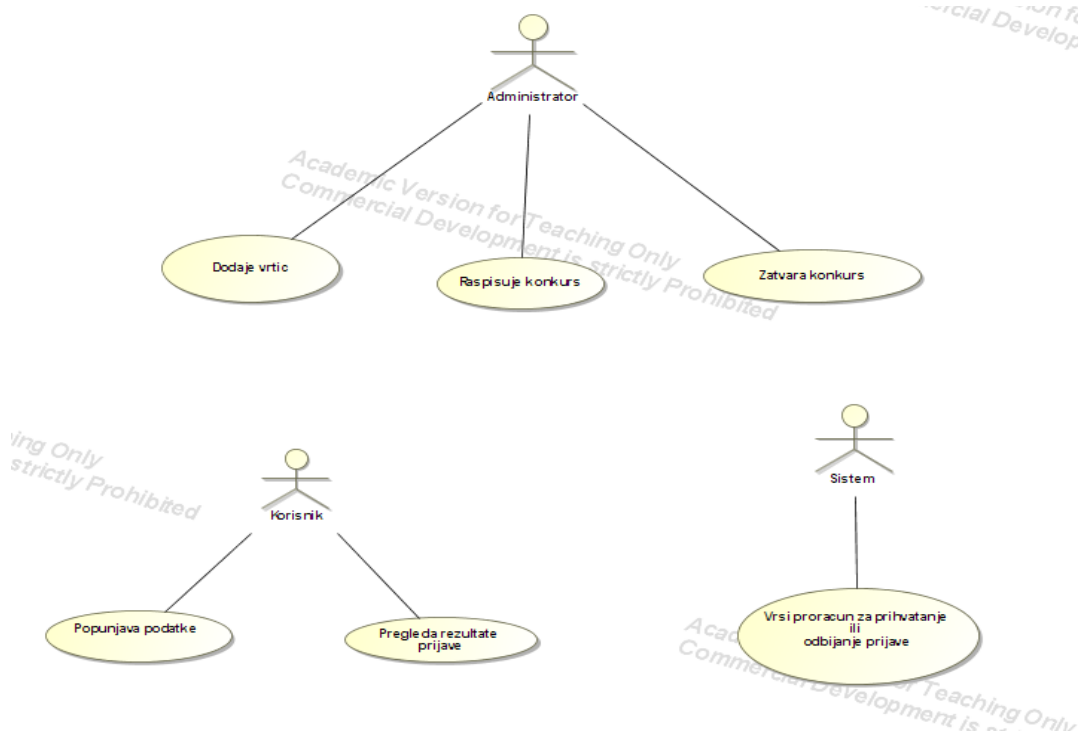
Za autorizaciju je korišćen Single Sign On (SSO) [9] koji omogućava korisniku da se prijavljuje sa jedinstvenim korisničkim imenom i pristupa više aplikacija i usluga.

Specifikacija zahteva

Korisnici sistema su administrator i običan korisnik. Administrator je zadužen za dodavanje vrtića i raspisivanje konkursa (unos informacije o konkursu kao što je naziv vrtića u kom se konkurs održava, datum početka, datum završetka itd.) i završetak konkursa.

Po završetku konkursa sistem ima uvid o svim prijavama na konkurs i rangira upisanu decu na osnovu osvojenog broja bodova (bodovi se osvajaju ukoliko su roditelji zaposleni, ukoliko dete ima zdravstvene probleme itd.). Rezultati prijave su dostupni po završetku konkursa.

Korisnik ima mogućnost da odabere konkurs koji je vezan za vrtić i da popuni potrebne podatke, a zatim pošalje prijavu.



Slika 5: Dijagram slučajeva korišćenja

Na tabeli 1 prikazan je opis slučajeve korišćenja "Dodavanje vrtića".

<i>Naziv</i>	Dodavanje vrtića
<i>Učesnici</i>	Administrator
<i>Preduslovi</i>	1. Administrator mora biti prijavljen na sistem
<i>Koraci</i>	1. Admin je na stranici za dodavanje vrtića 2. Admin popunjava formu o informacijama o vrtiću 3. Admin proverava podatke i potvrđuje formu
<i>Rezultati</i>	Vrtić je dodat u listu svih vrtića
<i>Izuzeci</i>	Vrtić već postoji ili su uneti nevalidni podaci

Tabela 1: Opis slučajeve korišćenja "Dodavanje vrtića"

Na tabeli 2 prikazan je opis slučajeve korišćenja "Raspisivanje konkursa".

<i>Naziv</i>	Raspisivanje konkursa
<i>Učesnici</i>	Administrator
<i>Preduslovi</i>	1. Administrator mora biti prijavljen na sistem
<i>Koraci</i>	1. Admin je na stranici za raspisivanje konkursa 2. Admin popunjava potrebne podatke o konkursu(vreme objave, početak i kraj konkursa, opština, grad, broj dece, uzrast) 3. Admin proverava podatke i potvrđuje formu
<i>Rezultati</i>	Konkurs je uspešno raspisan
<i>Izuzeci</i>	Uneti podaci su nevalidni

Tabela 2: Opis slučajeve korišćenja "Raspisivanje konkursa"

Na tabeli 3 prikazan je opis slučaja korišćenja "Zatvaranje konkursa".

<i>Naziv</i>	Zatvaranje konkursa
<i>Učesnici</i>	Administrator
<i>Preduslovi</i>	1. Administrator mora biti prijavljen na sistem
<i>Koraci</i>	1. Admin je na stranici gde ima prikaz liste konkursa 2. Admin bira konkurs koji želi da proglasi završenim 3. Administrator klikom na dugme uspešno završava konkurs
<i>Rezultati</i>	Konkurs je uspešno završen i sistem izbacuje rang listu prijava
<i>Izuzeci</i>	-

Tabela 3: Opis slučaja korišćenja "Zatvaranje konkursa"

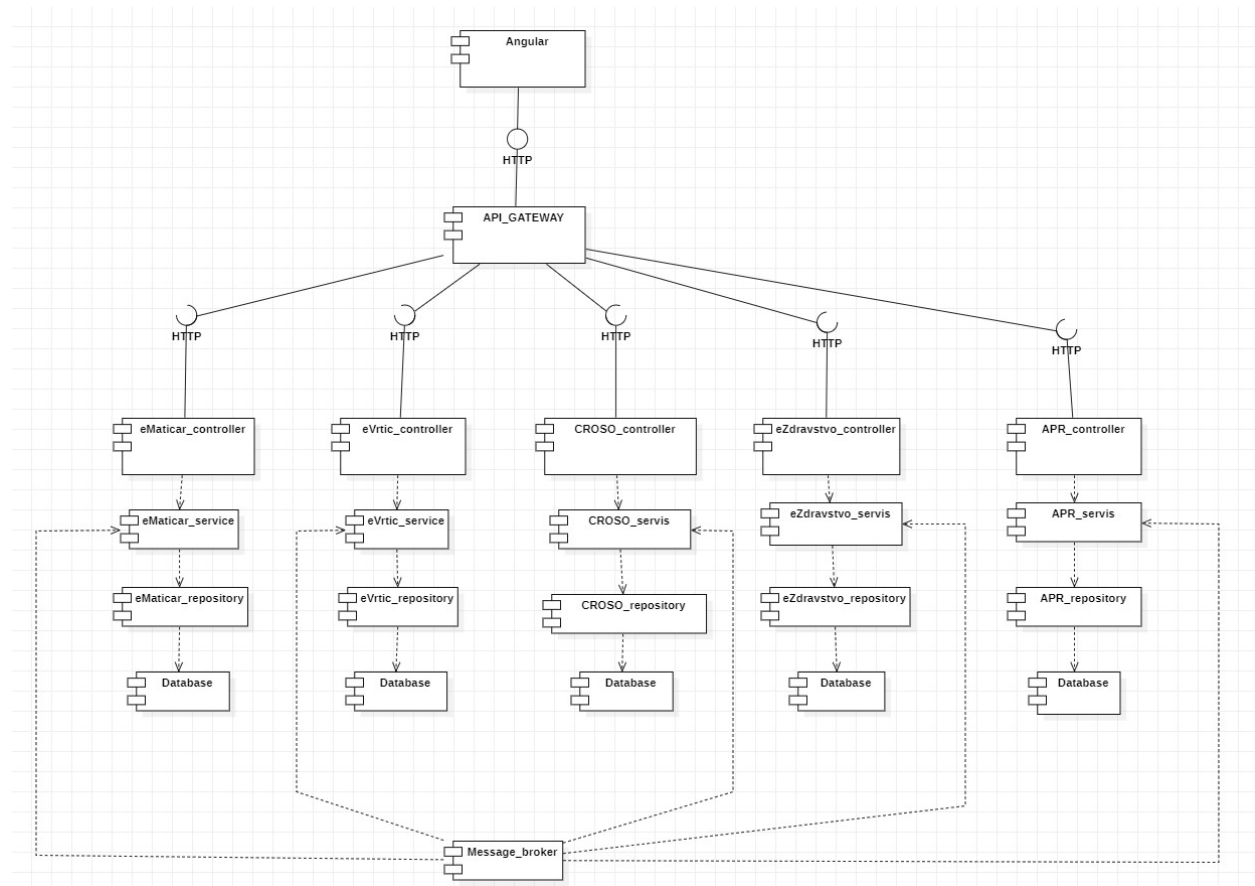
Na tabeli 4 prikazan je opis slučaja korišćenja "Popunjavanje podataka za upis".

<i>Naziv</i>	Popunjavanje podataka za upis deteta
<i>Učesnici</i>	Korisnik
<i>Preduslovi</i>	1. Korisnik mora biti prijavljen na sistem
<i>Koraci</i>	1. Korisnik je na stranici gde ima prikaz liste svih konkursa i vrtića u kojima se konkurs održava 2. Korisnik pretragom može da nađe sve otvorene konkurse u njegovom gradu/opštini ili po nazivu vrtića 3. Korisnik ulazi na stranicu odabranog konkursa 4. Korisnik popunjava podatke i potvrđuje ih
<i>Rezultati</i>	Dete je uspešno prijavljeno na konkurs
<i>Izuzeci</i>	1. Dete nije pregledano i sistem ne dozvoljava prijavu na konkurs 2. Korisnik nije dostavio svu potrebnu dokumentaciju 3. Korisnik je uneo nevalidne podatke

Tabela 4: Opis slučaja korišćenja "Popunjavanje podataka"

Specifikacija dizajna

Na slici 6 je prikazan komponentni dijagram koji predstavlja način funkcionisanja sistema.

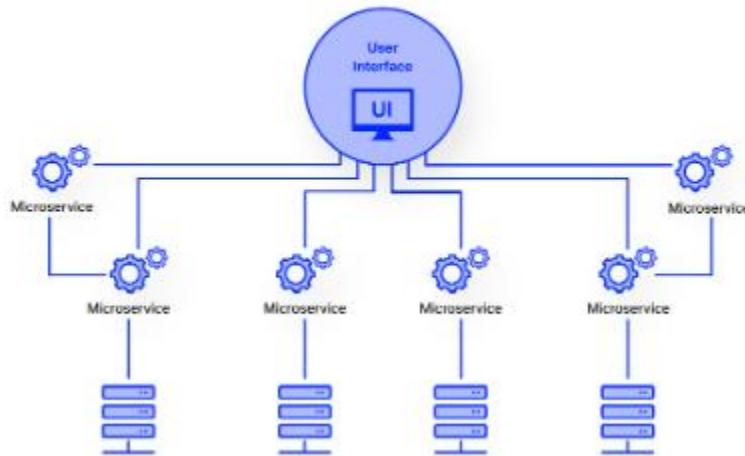


Slika 6: Komponentni dijagram

Arhitektura sistema

Sistem je realizovan kao mikroservisna arhitektura [11] gde postoje drugi servisi za prikupljanje podataka o detetu, roditelju, zdravstvenom stanju deteta i stanju zaposlenosti roditelja. Cilj ovih servisa je da su nezavisni jedni od drugih i da u slučaju otkaza jednog sistema ne otkáže cela aplikacija.

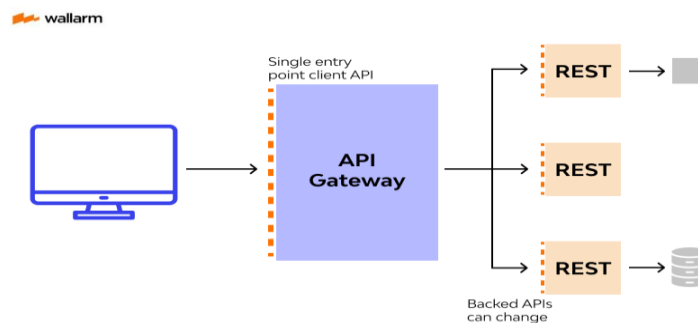
Microservice Architecture



Slika 7: Izgled mikroservisne arhitekture

Da bi se ostvarila komunikacija između dva sistema potrebno je da imaju “tačku” za kontakt, tzv. Interfejs koji je zapravo posrednik između dva odvojena sistema koji zajedno rade. Interfejs u ovom slučaju je REST API [10] .

Serverski deo se sastoji od više servisa koji zajedno funkcionišu preko API Gateway-a (korišćen nginx Api Gateway koji služi kao ulazna tačka i za raspoređivanje zahteva po servisima).



Slika 8: Način na koji funkcioniše API Gateway

Za komunikaciju sa serverom je potreban validan token izdat od strane SSO(Single Sign On) servera. Ukoliko je token validan, zahtev se prosleđuje dalje u sistem, a ukoliko nije vratiće se greška 403 – Forbidden.

Komunikacija izmedju servisa vrši se preko NATS-a [8] (open source sistem za razmenu poruka, može se posmatrati kao “message-oriented-middleware”)

```
func Conn() *nats.Conn { 5 usages  nananeko1305

    // NATS_URI is: nats://nats:4222 because instead of "localhost" we use name of container "nats"

    conn, err := nats.Connect(os.Getenv( key: "NATS_URI"))
    if err != nil {
        log.Fatal(err)
    }
    return conn
}
```

Slika 9: NATS konekcija

```
msg, err := p.nats.Request(os.Getenv( key: "GET_EMPLOYEE_STATUS_BY_ID"), requestBytes, 5*time.Second)
if err != nil {
    log.Println(err)
    println( args...: "Error on getting request")
}
var response map[string]bool
err = json.Unmarshal(msg.Data, &response)
```

Slika 10: Preuzimanje potrebnih podataka preko NATS-a

Pri samoj komunikaciji sa healthcare servisom preko NATS-a poziva se metoda `GetZdravstvenoStanjeByJMBG()` koja prima JMBG deteta i vraća zdravstveno stanje za dete. Ukoliko dete nije pregledano, prijava deteta na konkurs neće biti moguća

```

dataToSend, err := json.Marshal(insertComp.Dete.JMBG)
if err != nil {
    log.Println(v... "Error Marshaling JMBG")
}

response1, err := p.nats.Request(os.Getenv( key: "GET_STANJE_BY_JMBG"), dataToSend, 5*time.Second)

var deteZS data.ZdravstvenoStanje

if response1.Data == nil {
    http.Error(rw, error: "Dete nema ZS", http.StatusBadRequest)
    return
}

err = json.Unmarshal(response1.Data, &detेZS)
if err != nil {
    log.Println(v... "Error in Unmarshalling json")
    return
}

```

Slika 11: Komunikacija sa healthcare servisom pri dobavljanju zdravstvenog stanja

```

func (controller *HealthcareController) GetZdravstvenoStanjeByJMBG(writer http.ResponseWriter, req *http.Request) {
    vars := mux.Vars(req)
    jmbg := vars["jmbg"]

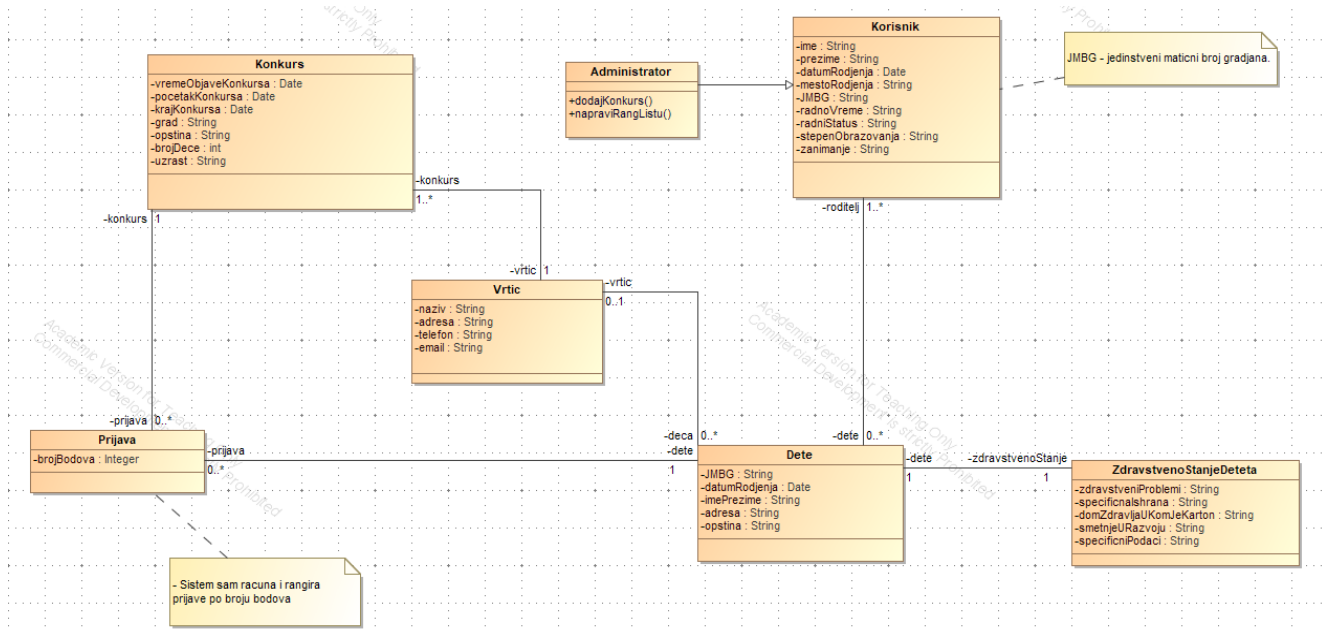
    zdravstvenoStanje, err := controller.service.GetZdravstvenoStanjeByJMBG(jmbg)
    if err != nil {
        log.Println(v... "Error finding Zdravstveno Stanje By JMBG")
        log.Println(v... "Found no Zdravstveno Stanje with that JMBG")
        writer.WriteHeader(http.StatusBadRequest)
        return
    }

    jsonResponse(zdravstvenoStanje, writer)
    writer.WriteHeader(http.StatusOK)
}

```

Slika 12: Prikaz metode koja se poziva preko NATS-a iz preschool servisa

Model podataka



Slika 13: Model podataka

Klasom Korisnik i Administrator predstavljeni su korisnici sistema koji imaju svoja prava pristupa.

Klasa Konkurs predstavlja entitet koji dodaje administrator i povezan je sa klasom Prijava gde jedan konkurs može imati više prijava, a prijava može imati samo jedan konkurs. Po zatvaranju konkursa sistem rangira prijave po broju bodova koje je dete osvojilo i prijave koje su prosle dobijaju status "Upisan".

Klasa Vrtić predstavlja same podatke o vrtiću i povezan je sa konkursom gde jedan vrtić može imati više različitih konkursa, a jedan konkurs može imati samo jedan vrtić.

Implementacija

Klijent

Klijentski deo aplikacije podrazumeva administratorske i korisničke funkcionalnosti, čiji se tip korisnika i privilegije istog nalaze u tokenu za autorizaciju.

Server

U main.go fajlu se nalazi kontroler koji prosleđuje zahtev sa odgovarajućeg REST API-ja na odgovarajuću metodu servisnog sloja. Rezultat se prevodi u HTTP odgovor.

```
getAllCompetitionsApplies := router.Methods(http.MethodGet).Subrouter()
getAllCompetitionsApplies.HandleFunc(path: "/competitions/applies", applyCompetitionsHandler.GetAllCompetitionApplies)
```

Slika 14: Primer kontroler metode

```
func (p *ApplyCompetitionHandler) GetAllCompetitionApplies(rw http.ResponseWriter, h *http.Request) { 1 usage  MurganS
    allCompetitions, err := p.repo.GetAllApplies()
    if err != nil {
        http.Error(rw, error: "Database exception", http.StatusInternalServerError)
        p.logger.Fatal(v...: "Database exception: ", err)
    }

    err = allCompetitions.ToJSON(rw)
    if err != nil {
        http.Error(rw, error: "Unable to convert to json", http.StatusInternalServerError)
        p.logger.Fatal(v...: "Unable to convert to json :", err)
        return
    }
}
```

Slika 15: Primer aplikativne logike implementirane u servisu

U servisu se poziva metoda iz repozitorijuma GetAllApplies() koja uzima kolekciju svih prijava iz Mongo baze podataka i pronalazi listu koju na kraju vrati.

```

func (pr *ApplyCompetitionRepo) GetAllApplies() (Prijava, error) { 1 usage  MurgaNS *
    ctx, cancel := context.WithTimeout(context.Background(), 5*time.Second)
    defer cancel()

    competitionCollection := pr.getCollectionCompetitionApply()

    var prijava Prijava
    usersCursor, err := competitionCollection.Find(ctx, bson.M{})
    if err != nil {
        pr.logger.Println(err)
        return nil, err
    }
    if err = usersCursor.All(ctx, &prijava); err != nil {
        pr.logger.Println(err)
        return nil, err
    }
    return prijava, nil
}

```

Slika 16: Primer repozitorijum metode

Demonstracija

Po prijavljivanju na sistem, korisniku je prikazana stranica koja odgovara njegovoj ulozi u sistemu. Administrator ima mogućnost dodavanja vrtica, da napravi konkurs za određeni vrtić, kao i da taj konkurs proglasi završenim. Korisnik ima mogućnost da se prijavi na konkurs, kao i da po završetku konkursa pregleda rezultate.

[PRIJAVA NA KONKURS ZA VRTIC](#)
[SERVISI](#)
[LOGOUT](#)

Napravi vrtic

Vrtic: "radosni vrtic"

Naziv: radosni vrtic

Adresa:

Telefon:

Email:

Grad: Beograd

Opština: Karaburma

Vrtic: "vidovdanski zvoncic"

Naziv: vidovdanski zvoncic

Adresa: novi sad

Telefon: 021-311-122

Email: vidovdanskizvoncic@gmail.com

Grad: novi-sad

Opština: novi-sad

Slika 17: Pregled vrtića

Dodaj konkurs

Datum objave

mm/dd/yyyy --:--

Pocetak konkursa

mm/dd/yyyy --:--

Kraj konkursa

mm/dd/yyyy --:--

Uzrast

Jaslice
Predskolsko
Mesovita grupa

Dodaj konkurs

Povratak na konkurse

Slika 18: Dodavanje konkursa

Prijava za konkurs u vrticu vidovdanski zvoncic

Podaci o konkursu:

Datum objave: Jun 29, 2023, 1:41:00 PM

Pocetak konkursa: Jun 30, 2023, 1:41:00 PM

Kraj konkursa: Jul 1, 2023, 1:41:00 PM

Uzrast: jaslice

Broj dece: 2

Status konkursa: Otvoren

Podaci o vrticu:

Naziv: vidovdanski zvoncic

Grad: novi-sad

Opstina: novi-sad

Adresa: novi sad

Telefon: 021-311-122

Email: vidovdanskizvoncic@gmail.com

Završi konkurs

Slika 19: Administrator završava konkurs

Zaključak

U radu je opisan opisan informacioni sistem za elektronski upis dece u vrtić. Prikazano je kako je dobavljanjem podataka od ostalih servisa moguće ubrzati upis dece, kao i smanjenje mogućnosti ljudske greške. U trenutnoj implementaciji vrtić i konkurs su objedinjeni u jedan sistem. U slučaju proširenja sistema, vrtić bi imao svoj sistem, a konkurs svoj i za to proširenje bi bilo potrebno uložiti mnogo truda i finansijskih sredstava.

Literatura

- [1] Portal Euprava- Vrtić. Pristupljeno 25.06.2023. URL: <https://euprava.gov.rs/vrtic>
- [2] Republički zavod za statistiku Srbije. Pristupljeno 25.06.2023. URL: <https://www.stat.gov.rs/>
- [3] Mongo DB. Pristupljeno 25.06.2023. URL: <https://www.mongodb.com/>
- [4] Go Programming Language. Pristupljeno 25.06.2023. URL: <https://go.dev/>
- [5] Angular, 2023. Pristupljeno 26.06.2023. URL: <https://angular.io/>
- [6] TypeScript, 2023. Pristupljeno 26.06.2023. URL: <https://www.typescriptlang.org/>
- [7] Docker, 2023. Pristupljeno 27.06.2023. URL: <https://www.docker.com/>
- [8] National Air Traffic Services(NATS). Pristupljeno 27.06.2023. URL: <https://nats.io/>
- [9] Single sign-on (SSO). Pristupljeno 28.06.2023. URL: <https://www.onelogin.com/learn/how-single-sign-on-works>
- [10] Representational state transfer(REST). Pristupljeno 28.06.2023. URL: https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- [11] Microservice architecture. Pristupljeno 29.06.2023. URL: <https://microservices.io/>