

Univerzitet u Beogradu  
Fakultet organizacionih nauka  
Laboratorija za softversko inženjerstvo

Softverski zahtevi

# Rečnik podataka

Mentor:

Prof. Dr. Dušan Savić

Studenti:

Ivona Heldrih 2020/3711

Jelena Srećković 2020/3712

Katarina Bresko 2020/3713

Miloš Cupara 2020/3721

Beograd, 2021

## Sadržaj

|   |    |
|---|----|
| 1. Sistemska strukturna analiza .....                             | 1  |
| 1.1. Hijerarhijski organizovan skup dijagrama toka podataka ..... | 1  |
| 1.2. Rečnik podataka .....  | 2  |
| 1.2.1. Tipovi rečnika podataka.....                               | 2  |
| 1.3. Specifikacija logike primitivnih procesa .....               | 3  |
| 2. Originalna notacija rečnika podataka SSA .....                 | 3  |
| 2.1. Složeni podaci .....   | 4  |
| 2.2. Elementarni podaci: .....                                    | 5  |
| 3. FON notacija rečnika podataka SSA.....                         | 6  |
| 3.1. Polja i domeni .....   | 6  |
| 3.2. Strukture .....  | 6  |
| 3.3. Meta model rečnika podataka .....                            | 7  |
| 4. Literatura.....  | 10 |

## Popis slika

|   |   |
|---|---|
| Slika 1.1 Osnovni Koncepti DTP .....                    | 1 |
| Slika 3.1 Meta model rečnika podataka.....              | 8 |
| Slika 3.2 Meta model rečnika podataka- Constraint ..... | 9 |

## Popis tabela

|                              |   |
|------------------------------|---|
| Tabela 1 Složeni podaci..... | 4 |
|------------------------------|---|

# 1. Sistemska strukturna analiza

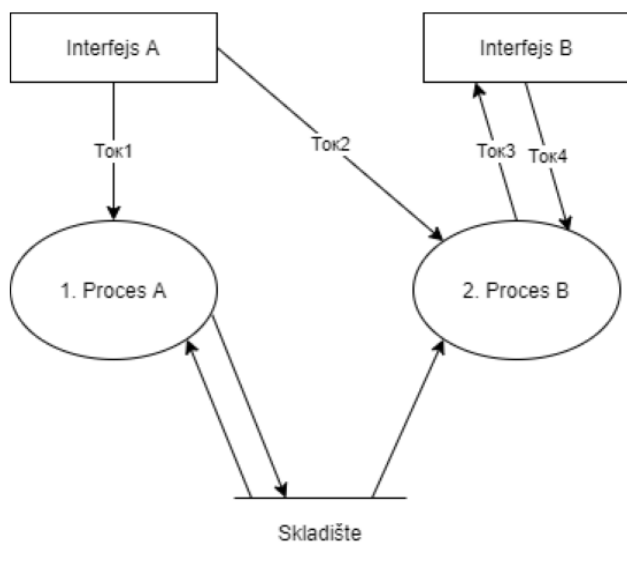
Sistemska strukturna analiza (SSA) je konvencionalna metoda za analizu i funkcionalnu specifikaciju informacionog sistema. Specifikacija treba da da odgovor na pitanje šta je proizvod informacionog sistema bez ulaženja u detalje implementacije. SSA je zasnovana na funkcionalnoj dekompoziciji sistema [1].

Potpunu specifikaciju informacionog sistema čine:

1. Hijerarhijski organizovan skup dijagrama toka podataka
2. Rečnik podataka
3. Specifikacija logike primitivnih procesa [3].

## 1.1. Hijerarhijski organizovan skup dijagrama toka podataka

Osnovni koncepti za specifikaciju informacionog sistema u SSA su: funkcije, interfejsi, tokovi i skladišta podataka. Dijagram toka podataka prikazuje međusobni odnos ovih koncepata. Na Slici 1 je predstavljen primer dijagrama toka podataka [3].



Slika 1.1 Osnovni Koncepti DTP

Skladište podataka poredstavlja stanje sistema. Na dijagramu toka podataka grafički se predstavlja sa dve paralelne linije. Usmerena strelica predstavlja tok podataka, on nosi podatke u različitim oblicima, dovodi ulazne podatke u proces obrade, a izlazne odvodi.

Interfejsi se grafički predstavljaju pravugaonikom i označavaju entitete sa kojima informacijski sistem komunicira. Pomoću kruga ili elipse mogu se predstaviti funkcije ili procesi obrade podataka. Proces obrade koristi podatke i menja podatke o stanju sistema razmenom tokova podataka sa skladištem. Za beleženje podataka koristi se baza podataka, a specifikacija se definiše pomoću rečnika podataka, što će u nastavku biti detaljnije objašnjeno.

Hijerarhijski opis sistema korišćenjem dijagrama toka se vrši tako što se prvo definišu globalniji procesi, a zatim se na sledećem nivou svaki od tih procesa predstavi novim dijagramom toka podataka. Na višem nivou imamo opštiji opis sistema, a na nižim nivoima se postepeno uvode detalji.

Dijagram toka podataka na vrhu ove hijerarhije naziva se dijagram konteksta, a procesi koji se dalje ne dekomponuju nazivaju se primitivni procesi [3].

## 1.2. Rečnik podataka

Rečnik podataka služi za opis sadržaja i strukture tokova i skladišta podataka, i definiše njihovu strukturu. Predstavlja alat za strukturirani opis sadržaja i strukture podataka u sistemu. Definiše ga poseban skup koncepata i pravila za opis podataka, tačnije, imaju svoju sintaksu za opis strukture podataka [4].

Rečnik podataka prikazuje dekompoziciju tokova i skladišta podataka korišćenjem sintakse za opis strukture podataka. Može se definisati kao prikupljanje informacija o svim elementima podataka ili sadržaju baza podataka kao što su tipovi podataka, tekstualni opisi sistema... Korisnicima olakšava upotrebu podataka, kao i razumevanje ulaza, izlaza, komponentama baze podataka...

Dakle, rečnik podataka je veoma bitan i potreban je za pravilno znanje i upotrebu sadržaja. Pruža sve informacije o imenima koja se koriste u modelima sistema, o entitetima, odnosima i atributima koji su prisutni u sistemskom modelu.

Kao deo alata za strukturnu analizu i dizajn, vrši se primena rečnika podataka [7].

Rečnik podataka može se odnositi na podatke u jednoj aplikaciji, podsistemu ili u celom informacijskom sistemu. Njegova izrada predstavlja početnu fazu u kreiranju aplikacije [4].

### 1.2.1. Tipovi rečnika podataka

Definisane su dve vrste rečnika podataka:

- Aktivni rečnici podataka- rečnici podataka stvoreni u bazi koju opisuju. Automatski ažuriraju promene u bazama pri pristupanju. Ovako se izbegava bilo kakve razlike

između rečnika podataka i njihovih struktura baza podataka I uvek su informacije ažurne.

- Pasivni rečnik podataka- rečnici podataka stvoreni kao nove baze podataka - odvojeni od baza podataka koje opisuju. Zahteva promenu od strane korisnika da bi bila ažurna [9].

### 1.3. Specifikacija logike primitivnih procesa

Za detaljan opis primitivnih funkcija (procesi koji se dalje ne dekomponuju) koristi se pseudokod. Ovaj opis se naziva specifikacija logike primitivnih procesa.

Strukturna sistemska analiza, originalno je nastala kao metodologija za analizu sistema krajem 70ih godina . Na Fakultetu organizacionih nauka koncepti SSA su dopunjeni strožijom formalizacijom za opis tokova i skladišta podataka [2]. Opis originalne notacije, kao i dopunjene biće prikazan u nastavku rada.

## 2. Originalna notacija rečnika podataka SSA

Rečnik podataka sadrži formalne definicije svih podatkovnih stavki prikazanih u dijagramima toka podataka. Ukoliko se u analizi ne koriste dijagrami toka podataka, svaka stavka u rešenju mora biti uključena u rečnik podataka. Ukoliko postoje slučajevi korišćenja, svaka stavka koja nije akter je podatkovna stavka ili svojstvo. Rečnik podataka se koristi za obezbeđivanje preciznih detalja koje se tiču podatkovnih entiteta. Važno je reći da se stavke u rečniku definišu onako kako se nalaze u domenu problema, a ne u implementaciji ili domenu rešenja. Stavke u rečniku podataka postaju kandidati za klase u fazi dizajna.

Rečnik podataka ima dve različite vrste elemenata: složene podatke i elementarne podatke [5].

Podatke višeg nivoa, tj. složeni, možemo definisati koristeći elementarne podatke, a elementarni podaci su stavke koje ne mogu biti dalje smanjene i definisane su svojim vrednostima koje mogu da imaju ili nekim drugim nedvosmislenim baznim tipom.

Generalni format rečnika podataka je sličan standardnom rečniku prirodnih jezika; sadrži leksikografski poređanu listu stavki. Svaka stavka se sastoji od formalne definicije i verbalnog opisa. Verbalni opis je suštinski rečenica ili dve na prirodnom jeziku koja opisuje podatak. Formalna deskripcija pruža precizniju definiciju koristeći vrstu matematičke notacije [5].

## 2.1. Složeni podaci

Složene podatke možemo konstruisati na tri načina, kao sekvencu, ponavljanje ili selekciju drugih tipova [5].

|             |                  |  |
|-------------|------------------|--|
| Sekvenca    | +                | Plus označava da jednom elementu sledi drugi element ili da mu je prisjedinjen                               |
| Selekcija   | [ ]              | Uglaste zagrade se koriste da obuhvate jedan ili više opcionih elemenata                                     |
|             |                  | Uspravna crta označava alternativne izbore u selekciji   |
| Ponavljanje | { }              | Vitičaste zagrade označavaju da definisani element čini serija ponavljanja elemen(a)ta sadržanih u zagradama |
|             | { } <sup>y</sup> | Gornja granica broja ponavljanja može se označiti celim brojem   |

Tabela 1 Složeni podaci

Primeri:

Sekvenca:

Adresnica = ime + adresa + grad + postanski-kod

\*Puna adresa na koju kupac može da primi poštu\*

Selekcija:

Transakcija = [uplata | isplata]

\*Klijentska transakcija na bankomatu\*

Ponavljanje:

Polaganje-ispita {ocena}<sup>3</sup>

\*Polaganje ispita na koji je moguće izaci triput pre obnavljanja\*

U ovim primerima koristimo zvezdice da označimo komentare ili verbalne opise stavki, ali može se koristiti i druga notacija\*

## 2.2. Elementarni podaci:

Elementarni podaci su opisani kao tip podataka ili lista vrednosti koje mogu da imaju

zeljena-temperatura = floating point

\*zeljena-temperatura je vrednost koju korisnik podesava za zeljenu temperaturu vode u bazenu.

To je decimalna vrednost sa pokretnim zarezom izmedju 0.0 i 100.0, ukljucujuci obe vrednosti.

Vrednosti su iskazane u stepenima Celzijusove skale\*

uzrast = nenegativan ceo broj

\*uzrast je broj korisnikovih godina. Uzrast je ceo broj veci od ili jednak nuli\*

broj-odgledanih-predstava = brojac

\*broj-odgledanih-predstava je broj predstava na kojima je prisustvovao kupac. Napomena:

posto se osoba ne dodaje na mejling listu pre odlaska na prvu predstavu, vrednost ne moze biti nula.\*

brojac = pozitivan ceo broj

\*brojac je ceo broj veci od nule koji samo moze biti povecan i nikad smanjen\*

Formalne stavke rečnika podataka se mogu lako napraviti i za neuobičajenije vrste elemenata.

U primeru ispod pokazano je kako bi sistem pretplate za časopis mogao da sadrži informaciju

tipa kategorije. U ovom slučaju, stavke podataka su enumerisani tipovi i definicija izlistava sve

dozvoljene vrednosti. Ne pravi se posebna stavka za svaku enumerisanu vrednost. Ukoliko je

potrebno, značenje enumerisanih vrednosti može se objasniti u napomenama.

Tip-transakcije = [obnova|nova|odjava]

Tip-transakcije oznacava vrstu transakcije

Nova je prva ikada pretplata klijenta

Obnova je druga i svaka sledeca pretplata

Odjava je zahtev za prestanak pretplate

Nacin-placanja [kes|cek|menica|kartica|nalog-za-prenos]

Nacin-placanja je nacin kojim klijent placa svoju transakciju (vrednosti ne moramo objasnjavati

Dalje, u rečniku podataka mogu se specificirati detalji koji se tiču formata ulaza ili izlaza.

Predpostavimo da sistem pretplate štampa mesečni izveštaj novih pretplatnika. U dijagramu

toka podataka, jedna aktivnost se prikazuje da proizvodi data element „novi-pretplatnici-izvestaj”

kao izlaz. U rečniku podataka, format se može definisati ovako:

Novi-pretplatnici-izvestaj = zaglavlje + novi-pretplatnici-lista + sumiran-izvestaj

Zaglavlje = naslov + datum

Naslov = ‘Mesečni izveštaj novih pretplatnika’

Novi-pretplatnici-lista = { pretplatnik-ime + pretplatnik-adresa }

Sumiran-izvestaj = ‘Ukupan broj novih pretplatnika’ + broj-novih-pretplatnika

Svi navedeni termini korišćeni u definiciji moraju biti definisani negde drugo u rečniku.

Jednostruki znaci navođenja koriste se da označe literale [5].

### 3. FON notacija rečnika podataka SSA

Izmenjena verzija originalne SSA metode se proučava nekoliko decenija na Fakultetu Organizacionih Nauka. Profesor Branislav Lazarević je napravio prvobitni koncept te ideje. U FON-ovoj izmenjenoj verziji, originalni SSA koncepti su zadržani, ali nova pravila su strože definisana [1].

Rečnik podataka je tehnika pomoću koje dajemo opis strukture i sadržaja svih tokova i skladišta podataka, definiše poseban skup koncepata i pravila za opis podataka. Tok ili skladište podataka mogu da predstavljaju i papirni dokument, datoteku..

Tokovi podataka i skladišta mogu imati složenu strukturu, pa je potrebno dekomponovati ih na proste elemente. Takva dekompozicija se opisuje u rečniku podataka.

Osnovni koncepti rečnika podataka su

- Polja i domen
- Strukture

#### 3.1. Polja i domen

Polje je primitivna i elementarna (atomska) struktura koja se dalje ne može dekomponovati i koja ima svoju vrednost. Polja svoje vrednosti uzimaju iz skupova koji se nazivaju domenima. Domen predstavlja kolekciju dozvoljenih vrednosti koje mogu da budu dodeljene polju. Ograničenje nad domenom sužava skup mogućih vrednosti koje polje može da uzme iz određenog domena.

Domeni mogu biti

- **Predefinisani** domen su standardni domen kao što su skup celih brojeva, skup realnih brojeva, skup karaktera ili skup logičkih vrednosti (tačno/netačno). (Integer, Character, Real, Logical, Date.. )
- **Semantički** domen daju novo ime predefinisanim domenima i obično kroz ograničenje definišu samo podskup predefinisano skupa elemenata.

#### 3.2. Strukture

Struktura je kompozicija komponenti gde komponenta može biti polje ili neka druga struktura [1].

Postoji nekoliko tipova strukture:



- **Agregacija komponenata:** Predstavlja složenu strukturu od N komponenti. Unutar same agregacije, može da se sadrže i druge strukture.  
Notacija je <a, b, c>
- **Ekskluzivna specijalizacija:** opisuje uniju komponente i označava da u strukturi može da se pojavljuje isključivo jedna komponenta.  
Notacija je [a, b, c]
- **Neekskluzivna specijalizacija:** Predstavlja uniju komponente i označava da je u ta struktura može imati samo jednu, dve ili sve komponente.  
Notacija je / a, b, c /
- **Skup komponenti:** Predstavlja kolekciju više vrednosti jedne komponenta i ukazuje da ova komponenta može pojavljivati se više puta u odgovarajućoj strukturi  
Notacija je {a, b, c}

Meta-model Rečnika podataka prikazan je na slici 1. (U ostatak ovog odeljka nazivi pojmova definisani u meta-modelu su oblikovani kurzivom.) Ovaj meta-model predstavlja apstraktnu sintaksu Rečnika podataka jezik koji se koristi za specifikaciju modela podataka na konceptualnom nivou.

### 3.3. Meta model rečnika podataka

StructureDefinition sadrži spisak FieldDefinition koncepata.

StructureElement može biti

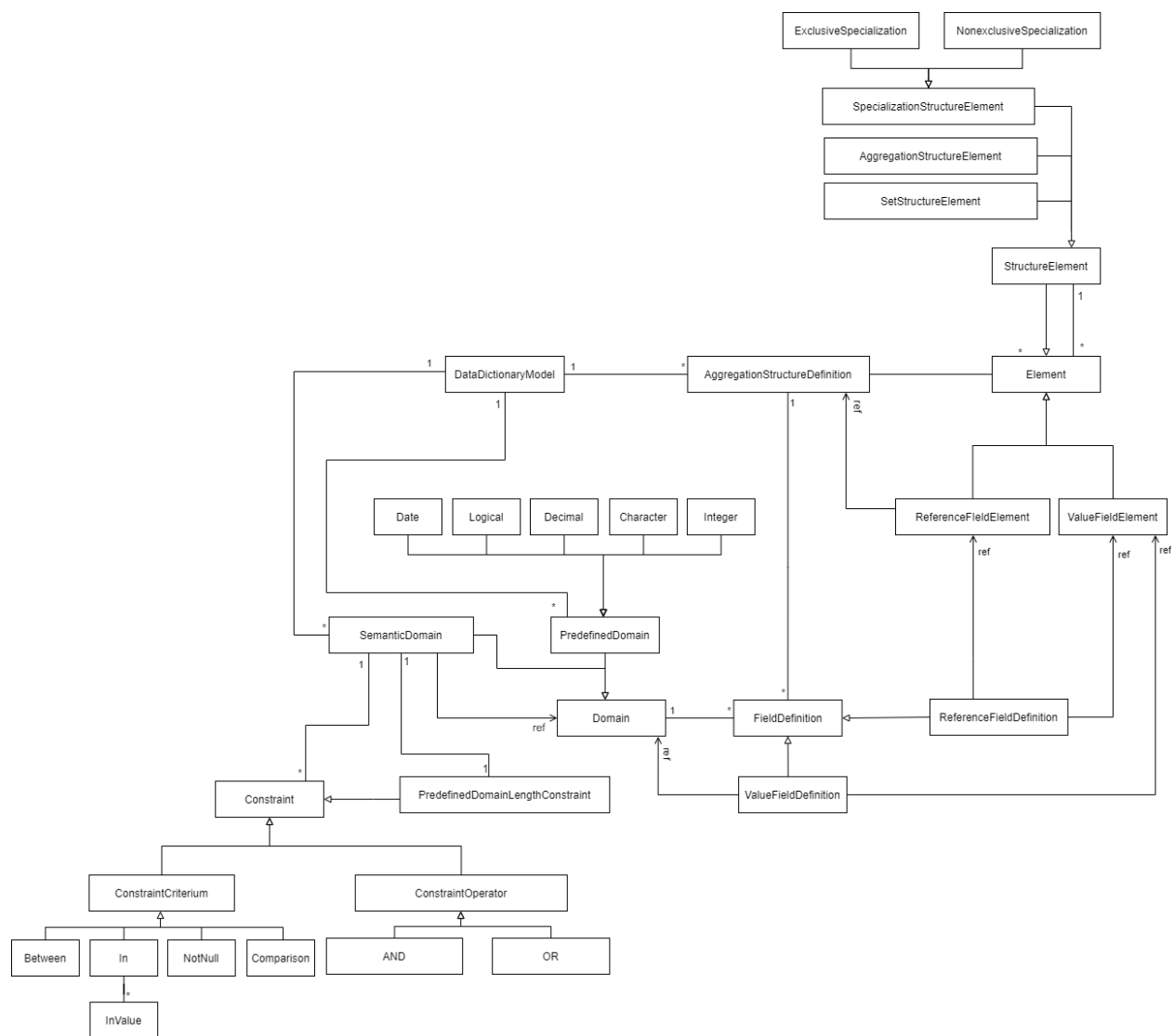
- agregacija (AggregationStructureElement),
- skup (SetStructureElement) ili
- specijalizacija (SpecializationStructureElement).
  - Postoje koncepti ExclusiveSpecialization i NonExclusiveSpecialization koja proširuju apstraktni koncept SpecializationStructureElement.

Svako polje (field) je opisano pomoću FieldDefinition.

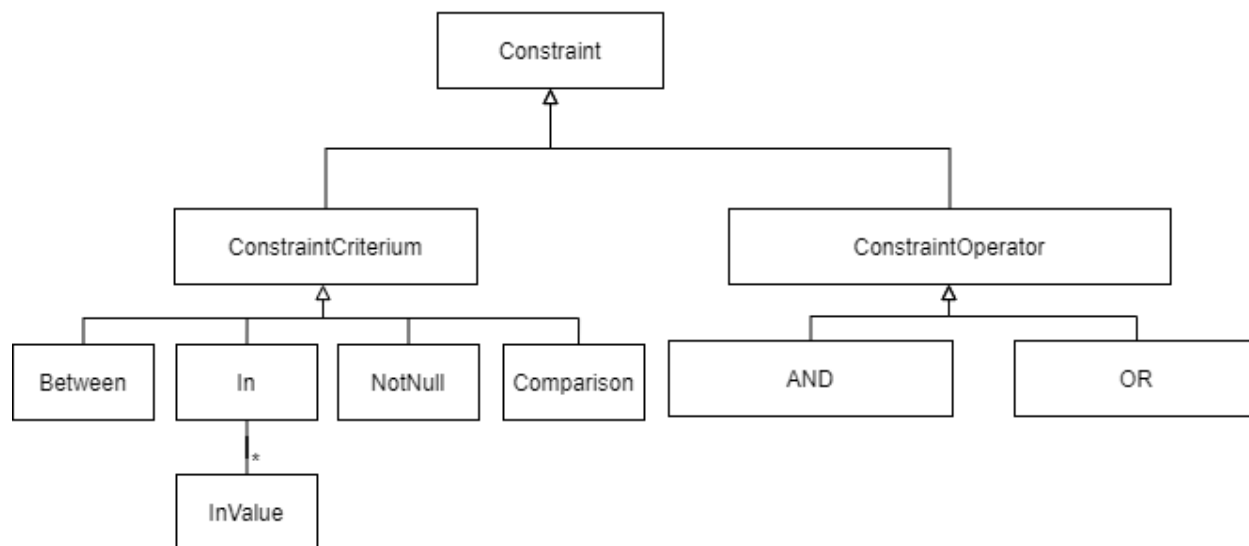
Domen može biti unapred definisan (PredefinedDomain) ili semantički (SemanticDomain); Predefinisani domen su ceo broj, string, real, boolean i date. Semantički domen se mogu zasnivati na predefinisanim domenima, kojima se dodaje jedno ili više ograničenja (Constraint), ili nekom drugom semantičkom domenu kom se dodaju dodatna ograničenja.

Unapred definisani domen modeluje se konceptom PredefinedDomain. Semantički domen (SemanticDomain) koji se zasniva na predefinisanom domenu i ograničenju (Constraint) nad tim domenom.

## ReferenceFieldElement I ValueFieldElement



Slika 3.1 Meta model rečnika podataka



**Slika 3.2 Meta model rečnika podataka- Constraint**

Ograničenja (Constraint) su definisana preko ConstraintCriterium I ConstraintOperatora. ConstraintCriterium su kriterijumi po kojima se prave ograničenja. ConstraintOperator se dodaju ukoliko postoji potreba za dva ili više operatora.

## 4. Literatura

- [1] D. Savic, "Razvoj softvera zasnovan na modelu slučajeva korišćenja i MDD pristupu", PhD thesis, Faculty of Organizational Sciences, Belgrade, 2016.
- [2] Zorana Ikonić, Valentina Andjelković, Dušan Savić and Alberto Rodrigues da Silva "Development DSL for data dictionary using MPS" 24th International Conference on Information Technology (IT) Zabljak, 2020
- [3] Fakultet Organizacionih Nauka, Laboratorija za informacione sisteme "Strukturna sistemska analiza" Beograd, 2000
- [4] Lazarević, B., Marjanović, Z., Aničić, N., & Babarogić, S. "Baze podataka". Univerzitet u Beogradu, Fakultet organizacionih nauka. 2016
- [5] Data Dictionary Notation, dostupno na:  
<http://users.csc.calpoly.edu/~jdalbey/SWE/DDnotation.html> [Pristupljeno: 12.maj 2021.]
- [6] Systems Analysis, dostupno na: <http://www.uky.edu/~dsianita/390A&D/lecture3.html>  
[Pristupljeno: 12.maj 2021.]
- [7] GeeksforGeeks. (2020, June). Short note on Data Dictionary, dostupno na:  
<https://www.geeksforgeeks.org/short-note-on-data-dictionary/> [Pristupljeno: 12.maj 2021.]
- [8] Strukturna sistemska analiza, dostupno na:  
[https://www.puskice.org/download/upravljanje\\_kvalitetom\\_dokumentacije/Strukturna-sistemska-analiza.pdf](https://www.puskice.org/download/upravljanje_kvalitetom_dokumentacije/Strukturna-sistemska-analiza.pdf) [Pristupljeno: 12.maj 2021.]
- [9] Coronel, C. et al. "Database Systems Design, Implementation, and Management", Cengage Learning, Boston, MA, 2010.