Digitalna knjižnica

*Specifikacija zahtjeva sustava*

Studenti:

Lovro Vidović, 583/RM

Kristijan Zadro, 584/RM

Toni Jelonjić, 585/RM

Mostar, 17.12.2021.

Autor

|  |  |  |
| --- | --- | --- |
| **Naziv** | **Uloga** | **Odjel/Department** |
| Lovro Vidović | Voditelj grupe |  |
| Kristijan Zadro | Član grupe |  |
| Toni Jelonjić | Član grupe |  |
|  |  |  |

Povijest izmjena

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datum** | **Verzija** | **Status** | **Detalji izmjene** | **Autor** |
| 17.12.2021. | 1.0 | WIP | - | Svi |
| 19.12.2021. | 1.1 |  | Dodan AD dijagram | Svi |
|  |  |  |  |  |
|  |  |  |  |  |

S A D R Ž A J

[1. Uvod 4](#_Toc532472376)

[1.1 Svrha dokumenta 4](#_Toc532472377)

[1.2 Definicije, akronimi i skraćenice 4](#_Toc532472378)

[1.3 Reference 4](#_Toc532472379)

[1.4 Sažetak dokumenta 4](#_Toc532472380)

[2. Opis SUSTAVA 6](#_Toc532472381)

[2.1 Svrha SUSTAVA 6](#_Toc532472382)

[2.2 OPSEG PROJEKTA 7](#_Toc532472383)

[2.3 Korisnici i njihove karakteristike 8](#_Toc532472384)

[2.4 Funkcionalnosti SUSTAVA 8](#_Toc532472385)

[2.5 Pretpostavke i ovisnosti 8](#_Toc532472386)

[2.6 Planiranje zahtjeva 9](#_Toc532472387)

[3. Razrada zahtjeva 10](#_Toc532472388)

[3.1 DIJAGRAM KORIŠTENJA 11](#_Toc532472389)

[3.2 sudionici 11](#_Toc532472390)

[3.2.1 Ljudski sudionici 11](#_Toc532472391)

[3.2.2 Hardware Actors 11](#_Toc532472392)

[3.2.3 Software System Actors 12](#_Toc532472393)

[3.3 Specifikacija USE Case dijagrama korištenja 12](#_Toc532472394)

[3.4 Funkcionalni zahtjevi za KORISNIKA <NAZIV KORISNIKA> 13](#_Toc532472395)

[3.4.1 Prijavi se u sustav – *Login* 13](#_Toc532472396)

[3.4.2 Registracija korisnika - *Registration* 14](#_Toc532472397)

[3.4.3 Use Case 1 16](#_Toc532472398)

[3.4.4 Use Case 2 16](#_Toc532472399)

[3.4.5 Use Case n 16](#_Toc532472400)

[3.5 Klasifikacija funkcionalnih zahtjeva ili USE CASE 16](#_Toc532472401)

[3.6 Performance Requirements 17](#_Toc532472402)

[3.7 Design Constraints 18](#_Toc532472403)

[3.8 Software system attributes 18](#_Toc532472404)

[3.9 Logical Database Requirements 18](#_Toc532472405)

[3.9.1 Relacijski model 19](#_Toc532472406)

[3.9.2 Opis tablica u relacijskom modelu 19](#_Toc532472407)

[3.10 Nefunkcionalni zahtjevi 29](#_Toc532472408)

[3.10.1 Korisnička sučelja 29](#_Toc532472409)

[3.10.2 Vanjska sučelja i integracije 29](#_Toc532472410)

[3.10.3 Upotrebljivost 30](#_Toc532472411)

[3.10.4 Performanse 30](#_Toc532472412)

[3.10.5 Korektnost 31](#_Toc532472413)

[3.10.6 Instalacija 32](#_Toc532472414)

[3.10.7 Upravljanje i održavanje sustava 32](#_Toc532472415)

[3.10.8 Prenosivost 33](#_Toc532472416)

[3.10.9 Skalabilnost 33](#_Toc532472417)

[3.10.10 Sigurnost 33](#_Toc532472418)

[3.10.11 Primjenjivi standardi 33](#_Toc532472419)

[3.10.12 Robusnost 34](#_Toc532472420)

[3.10.13 Internacionalizacija 34](#_Toc532472421)

[3.10.14 Personalizacija 34](#_Toc532472422)

[3.10.15 Ograničenja 34](#_Toc532472423)

[4. Prilozi 35](#_Toc532472424)

# Uvod

Specifikacija zahtjeva za projekt Digitalna knjižnica.

## Svrha dokumenta

Zahtjevi su za sustav za podršku rada knjižnice koja se nalazi u sklopu fakulteta.

## Definicije, akronimi i skraćenice

|  |  |  |
| --- | --- | --- |
| **Term/Acronym** | **Definicija** | **Opis** |
| DK | Digitalna knjižnica | Skraćenica za digitalnu knjižnicu |
| WIP | Work In Progress | Radovi u tijeku |
| TBD | To Be Decided | Nije još odlučeno |

## Reference

|  |  |  |
| --- | --- | --- |
| **KOmponenta** | **Name (with link to the document)** | **Opis povezanosti** |
|  |  |  |

## Sažetak dokumenta

U dokumentu se detaljno opisuje kako se koristi digitalna knjižnica koja se nalazi u sklopu fakulteta. Studenti će moći rezervirati određene knjige, posuditi i čitati. Po završetku čitanja, studenti su dužni vratiti knjige u knjižnicu. Ako kasne s vraćanje, sustav ih obavještava da kasne.

# Opis SUSTAVA

U sustavu se detaljno opisuje kako se koristi digitalna knjižnica koja se nalazi u sklopu fakulteta. Studenti će moći rezervirati određene knjige, posuditi i čitati. Po završetku čitanja, studenti su dužni vratiti knjige u knjižnicu. Ako kasne s vraćanje, sustav ih obavještava da kasne.

## Svrha SUSTAVA

* Ovaj sustav omogućava korisnicima brz pronalazak knjige i posuđivanje iste, te ostavljanje recenzcije o knjizi
* Sustav će pomoći studentima pri posuđivanju određene knjige
* Što je to sustav s korisničkog gledišta? Koje će ciljeve ispuniti?

S korisničkog gledišta, cilj sustava je ušteda vremena studentu, jer pomoću sustava može vidjeti postoji li knjiga uopće u knjižnici, da ne mora ići do knjižnice ukoliko knjige nema. Također olakšava posao radnicima knjižnice jer na vrijeme mogu pripremiti knjige koje studenti rezerviraju preko sustava.

## OPSEG PROJEKTA

Da li je sustav samostalan ili dio većeg sustava? Opišite da li je to nova verzija već

* Je li sustav samostalan?

Da, sustav je samostalan.

* Je li to dio većeg sustava? (kojeg?)

Ne, nije dio većeg sustava.

* Dajte na visokoj razini pregled temeljnih vanjskih sučelja (HW, SW, API i ostali komunikacijska sučelja, itd).

Nemamo još

## Korisnici i njihove karakteristike

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User/Role** | **Example** | **Frequency of Use** | **Security/Access, Features Used** | **Additional Notes** |
| *Admin* | *Radnici u knjižnici* | Često | Korisnik treba znati koristiti računalo, moći će dodavati nove informacije u sustav, kreirati korisničke račune i voditi evidenciju o posuđivanju knjiga | */* |
| *Student* | *Student* | *Ponekad* | *Student treba znati čitati* | */* |
|  |  |  |  |  |

## Funkcionalnosti SUSTAVA

Registriranje na sustav, pregled knjiga, rezervacija knjige, pretraga knjiga, ostavljanje recenzije, pregled detalja o knjizi itd

## Pretpostavke i ovisnosti

Potrebno je koristiti Windows sustav i preglednik Chrome.

## Planiranje zahtjeva

Moguće je promijeniti zahtjeve, samo je potrebno na vrijeme komunicirati s ljudima koji razvijaju sustav.

# Razrada zahtjeva

<Funkcionalni zahtjevi predstavljaju izjave o uslugama koje sustav mora pružiti, te kako bi se sustav morao ponašati u određenim situacijama. Funkcionalni zahtjevi aplikacije za upravljanje troškovima:>

<Funkcionalni i nefunkcionalni zahtjevi mogu bit grupirani na različite načine:

* Registracija na sustav
* Pregled knjiga
* Dodavanje novih informacija
* Pretraživanje knjiga
* Rezervacija knjiga
* Recenzija knjige
* Pregled detalja o knjizi

Za detalje pogledajte *IEEE Recommended Practice for Software Requirements Specifications IEEE Std 830-1998* → Poglavlje 5.3.7 (Organizing the specific requirements)

<U ovom poglavlju navesti popis zahtjeva sukladno traženoj dokumentaciji ili inicijalnom tekstu projektnog zadatka>

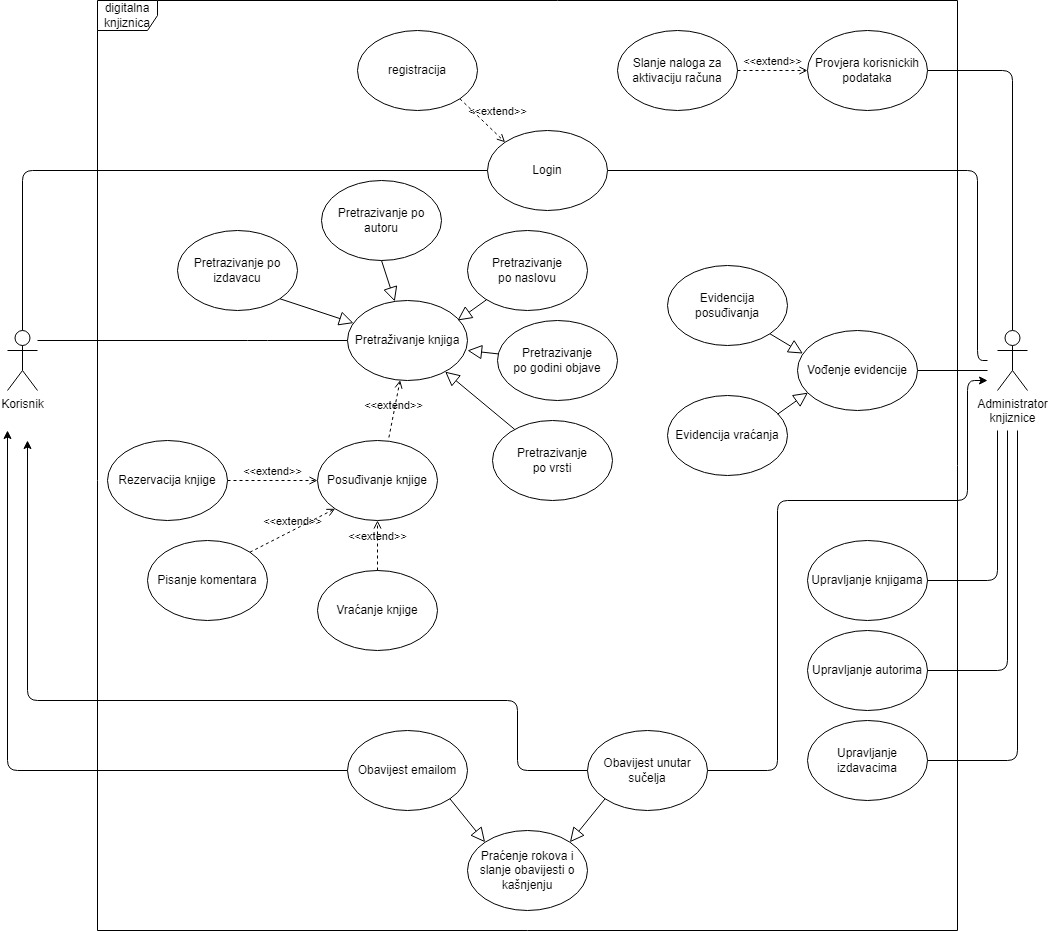
< npr.:

1. Aplikacija mora imati glavni *menu* sa dnevnim pregledom i posljednjim izmjenama.
2. Sustav mora imati mogućnost ulaska u polje „posljednje izmjene“ gdje možemo pobliže vidjeti sve izmjene.
3. Software mora dati mogućnost izmjene trenutnog računa kojeg želimo vidjeti u glavnom meniju.
4. Itd.

>

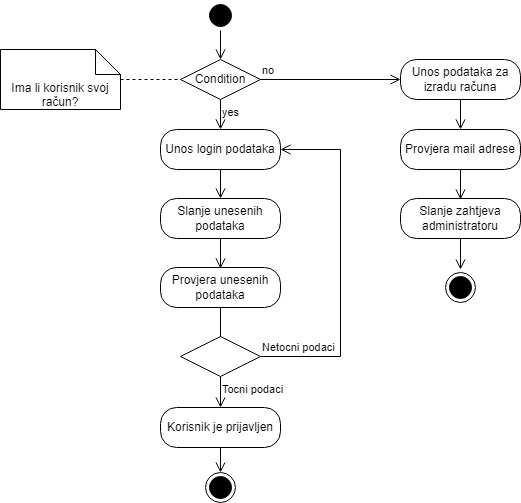
Dodati ćemo primjer kada budemo imali više informacija, zasad ne znamo.

## DIJAGRAM KORIŠTENJA



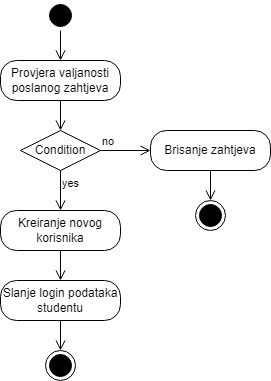
|  |  |
| --- | --- |
| **ID:** | **1** |
| **Title:** | Login |
| **Description:** | Student se prijavljuje u sustav |
| **Primary Actor:** | Student, admin |
| **Preconditions:** | Student se nije prijavio u sustav |
| **Postconditions:** | Student je prijavljen u sustav i može nastaviti dalje |
| **Main  Success Scenario:** | Student unosi svoje korisničko ime i lozinku, sustav provjerava unesene podatke i ako su ispravni studentu se daje pristup sustavu. |
| **Extensions:** | Ako uneseni podaci nisu ispravni, student se daje do znanja kako je pogriješio i da unese ispravne podatke.  Registracija - Ako student nema račun mora se prvo registrirati tako što će poslati zahtjev za izradu računa |
| **Frequency of Use:** | Svaki put kada student prvi put pristupa sustavu, ulazna točka sustava |
| **Status:** | WIP |
| **Owner:** | TBD – To Be Decided |
| **Priority:** | Visok |

Login AC diagram:



|  |  |
| --- | --- |
| **ID:** | **2** |
| **Title:** | Provjera korisničkih podataka |
| **Description:** | Admin mora provjeriti podatke koji su poslani u zahtjevu za izradu računa |
| **Primary Actor:** | Admin |
| **Preconditions:** | Student mora prethodno poslati zahtjev za računom |
| **Postconditions:** | - |
| **Main  Success Scenario:** | Admin sustava prima zahtjev za novim računom, zatim mora provjeriti postoji li taj mail u ISS sustavu  Slanje naloga za aktivaciju računa – ako su podaci točni admin kreira novog korisnika i salje login podatke studentu |
| **Extensions:** | - |
| **Frequency of Use:** | Jedino kada postoje zahtjevi za računima |
| **Status:** | WIP |
| **Owner:** | TBD |
| **Priority:** | Srednji |

Provjera korisničkih podataka AC diagram:

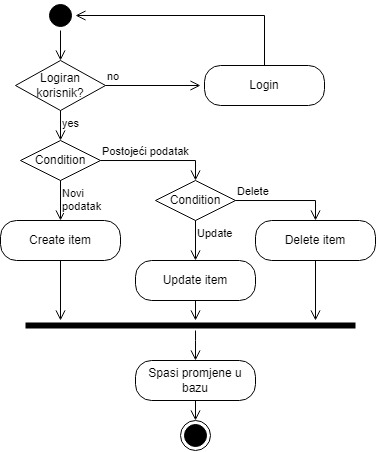


|  |  |
| --- | --- |
| **ID:** | **3** |
| **Title:** | Upravljanje knjigama |
| **Description:** | Admin može kreirati nove knjige, mijenjati ili brisati postojeće |
| **Primary Actor:** | Admin |
| **Preconditions:** | Za brisanje i mijenjanje knjiga, knjiga mora prvo postojati |
| **Postconditions:** | Knjiga stvorena, promijenjena ili izbrisana |
| **Main  Success Scenario:** | Za kreiranje nove knjige student će imati formu koju mora popuniti kako bi napravio knjigu. Ako su uneseni podaci validni knjiga će se stvoriti. Svaki od tih podataka se može naknadno promjeniti. Ako te knjige više nema u sustavu, ona se može izbrisati. |
| **Extensions:** | Ako admin unese pogrešne podatke bit će obaviješten i tražit će se da unese validne |
| **Frequency of Use:** | Često |
| **Status:** | WIP |
| **Owner:** | TBD |
| **Priority:** | Visok |

|  |  |
| --- | --- |
| **ID:** | **4** |
| **Title:** | Upravljanje autorima |
| **Description:** | Admin može kreirati nove autore, mijenjati ili brisati postojeće |
| **Primary Actor:** | Admin |
| **Preconditions:** | Za brisanje i mijenjanje autora, autor mora prvo postojati |
| **Postconditions:** | Autor stvoren, promijenjen ili izbrisan |
| **Main  Success Scenario:** | Za kreiranje novog autora student će imati formu koju mora popuniti kako bi napravio autora. Ako su uneseni podaci validni autor će se stvoriti. Svaki od tih podataka se može naknadno promjeniti. Ako tog autora više nema u sustavu, on se može izbrisati. |
| **Extensions:** | Ako admin unese pogrešne podatke bit će obaviješten i tražit će se da unese validne |
| **Frequency of Use:** | Često |
| **Status:** | WIP |
| **Owner:** | TBD |
| **Priority:** | Visok |

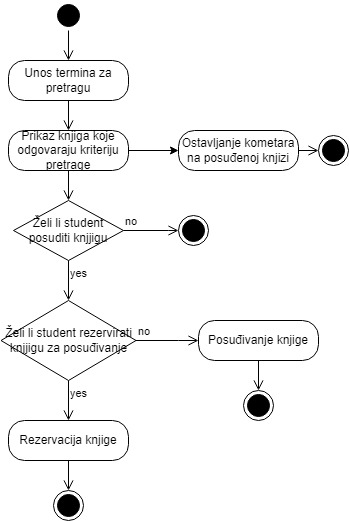
|  |  |
| --- | --- |
| **ID:** | **5** |
| **Title:** | Upravljanje izdavačima |
| **Description:** | Admin može kreirati nove izdavače, mijenjati ili brisati postojeće |
| **Primary Actor:** | Admin |
| **Preconditions:** | Za brisanje i mijenjanje izdavača, izdavač mora prvo postojati |
| **Postconditions:** | Izdavač stvoren, promijenjen ili izbrisan |
| **Main  Success Scenario:** | Za kreiranje novog izdavača student će imati formu koju mora popuniti kako bi napravio izdavača. Ako su uneseni podaci validni izdavača će se stvoriti. Svaki od tih podataka se može naknadno promjeniti. Ako tog izdavača više nema u sustavu, on se može izbrisati. |
| **Extensions:** | Ako admin unese pogrešne podatke bit će obaviješten i tražit će se da unese validne |
| **Frequency of Use:** | Često |
| **Status:** | WIP |
| **Owner:** | TBD |
| **Priority:** | Visok |

AC diagram za use caseove 3,4,5:



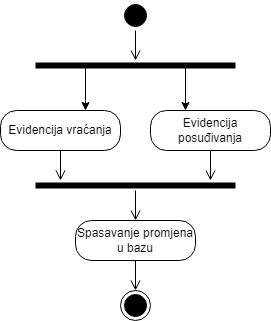
|  |  |
| --- | --- |
| **ID:** | **6** |
| **Title:** | Pretraživanje |
| **Description:** | Studenti mogu pretraživati naslove u knjižnici, po većem broju kriterija (ime i/ili prezime autora i/ili naslov knjige i/ili područje i/ili vrsta naslova, izdavač, godina objave i sl. itd.). |
| **Primary Actor:** | Student |
| **Preconditions:** | Prijavljen student |
| **Postconditions:** | Pronalazak tražene knjige |
| **Main  Success Scenario:** | Student će u polje za unos unijeti pojam za pretragu, a ispod toga će se prikazati tablica knjiga koji odgovaraju kriteriju pretrage |
| **Extensions:** | Student ne može unijeti samo whitespace ili manje od dva znaka jer onda pretraga neće raditi  Posuđivanje knjige – nakon sto student pronađe knjigu može posuditi knjigu  Rezervacija – student može rezervirati knjigu koju hoce posuditi  Pisanje komentara – korisnik na posuđenim knjigama može ostavljati komentare  Vraćanje knjige – korisnik može vratiti posuđenu knjigu |
| **Frequency of Use:** | Često |
| **Status:** | WIP |
| **Owner:** | TBD |
| **Priority:** | Srednji |

AC diagram za use case 6:



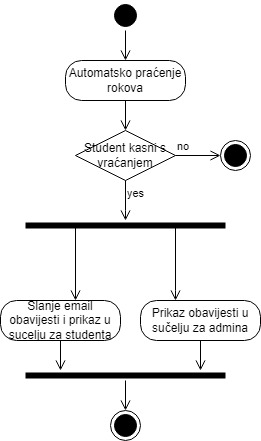
|  |  |
| --- | --- |
| **ID:** | **7** |
| **Title:** | Vođenje evidencije |
| **Description:** | Admin vodi evidenciju |
| **Primary Actor:** | Admin |
| **Preconditions:** | Prijavljen admin |
| **Postconditions:** | Evidentirana posudba ili vraćanje |
| **Main  Success Scenario:** | Admin evidentira posuđivanje knjige ili vraćanje |
| **Extensions:** | Kada student posudi knjigu, admin evidentira da je posuđenja  Kada student vrati knjigu admin evidentira vraćanje |
| **Frequency of Use:** | Često |
| **Status:** | WIP |
| **Owner:** | TBD |
| **Priority:** | Srednji |

AC diagram za aktivnost 7:



|  |  |
| --- | --- |
| **ID:** | **8** |
| **Title:** | Praćenje rokova i slanje obavijesti |
| **Description:** | Sustav prati rokove za vraćanje knjiga i obavješta korisnike o kašnjenju |
| **Primary Actor:** | Sustav |
| **Preconditions:** | Postojanje evidencije o posuđenoj knjizi |
| **Postconditions:** | Slanje obavijesti korisnicima o kašnjenju |
| **Main  Success Scenario:** | Sustav automatski prati rokove za vraćanje knjiga i ako se dogodi da student kasni s vraćanjem, salje se obavijest adminu i studentu |
| **Extensions:** | Admin dobiva obavijest unutar sučelja o kašnjenju  Student dobiva obavijest unutar sučelja i putem maila |
| **Frequency of Use:** | Ponekad |
| **Status:** | WIP |
| **Owner:** | TBD |
| **Priority:** | Srednji |

AC diagram use casea 8:



## sudionici

### Ljudski sudionici

Zaposlenici knjižnice i studenti.

### Hardware Actors

<kratko opisati sudionike koji su HW predmetnog ili nekog drugog sustava >

Ne razumijemo

### Software System Actors

<kratko opisati sudionike koji su SW predmetnog ili nekog drugog sustava >

Ne razumijemo ni ovo

## Specifikacija USE Case dijagrama korištenja

<Ovo poglavlje sadrži sve specifikacije/zahtjeve do nivoa detalja.

Svaki zahtjev treba da bude:

* Jednoznačno označen (obično sa brojevima *requirement* RQ-1.5 ili *funkcionalni zahtjev* FZ-1.5 ili UC1 itd)
* Ispravno definiran
* Jednoznačan
* Kompletan
* Konzistentan
* Prioritiziran (nivo prioriteta)
* Provjerljiv (da može bit testiran)
* Izmjenljiv
* Sljedljiv (*traceable* - da se može pratiti kroz prethodne i buduće verzije dokumenata)

Za detalje pogledajte *IEEE Recommended Practice for Software Requirements Specifications IEEE Std 830-1998* → Poglavlje 4.3 (Characteristics of a good SRS)

<Za pojedini use case ispod možete izdvojiti dio UC dijagrama koji se odnosi samo na jednog *user-a* ili interakciju s use case-om>

<Također možete ili ne radi lakšeg praćenja iz USE CASE dijagrama korištenja izvoditi funkcionalne zahtjeva ali i ne morate>

<U nastavku slijedi nekoliko primjera specifikacije zahtjeva. Inače, ovaj predložak ne definira format zahtjeva, način označavanja prioriteta, način organiziranja zahtjeva niti nivo detalja zahtjeva. Obratite pažnju zahtjevi su označeni sa Wordovim stilom „FZ 1“, „FZ 2“, „FZ 3“ ili „FZ 4“ (na taj način Word je automatski numerirati zahtjeve).>

Zahtjevi još nisu utvrđeni.

<U ovom dijelu možete organizirati use case dijagrame slijedno ili po useru, npr ako idemo po sudionicima iz Poglavlja iznad, npr useru onda ćemo otvoriti novo poglavlje i ispod raspisati njegove uključenje use case-ove>

## Funkcionalni zahtjevi za KORISNIKA <NAZIV KORISNIKA>



*<dio UC dijagrama koji se odnosi na <korisnika>>*

*Ili*

### Prijavi se u sustav – *Login*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Use Case ID:* | 2 | | | |
| *Use Case Name:* | Prijava na sustav | | | |
| *Created By:* | Tim 9 | | *Last Updated By:* | Toni |
| *Date Created:* | 17.12.2021. | | *Date Last Updated:* | 17.12.2021. |
| *Actors:* | | Korisnik;Administrator | | |
| *Description:* | | Prijavom na sustav Korisnik;Operater;Administrator pristupaju svojim korisničkim računima i svim mogućnostima vezanim za njega | | |
| *Trigger:* | | Odabir opcije Prijava | | |
| *Preconditions:* | | 1. Korisnik;Operater mora biti registriran u sustav | | |
| *Postconditions:* | | 1. Korisnik:Operater;Administrator je prijavljen u sustav | | |
| *Normal Flow:* | | 1. Korisnik;Operater;Administrator je odabrao opciju Prijava 2. Sustav zatraži unos email-a i lozinke 3. Nakon unosa potrebnih podataka, korisnik treba potvrditi unos | | |
| *Alternative Flows:* | | 1. Korisnik;Operater;Administrator nije unio točnu kombinaciju email-a i lozinke, tijek se nastavlja u 2. točki. 2. Korisnik;Operater nije registriran u sustav, nije moguće izvršiti prijavu – tijek se završava. | | |
| *Exceptions:* | |  | | |
| *Includes:* | |  | | |
| *Priority:* | |  | | |
| *Frequency of Use:* | |  | | |
| *Business Rules:* | |  | | |
| *Special Requirements:* | |  | | |
| *Assumptions:* | |  | | |
| *Additional documents* | |  | | |
| *Source* | |  | | |
| *Notes and Issues:* | |  | | |

1. Potrebno je omogućiti prijavu korisnika na sustav
   1. Aplikacija smije dozvoliti više logiranih korisnika na istom računaru, izuzev administratora.
   2. Forma za prijavu potrebno je unijeti „username“ i „password“.
   3. Ukoliko se radi o registriranom običnom korisniku nakon logiranja otvara se forma za unos podataka o aktivnostima
   4. Ukoliko se radi o novom korisniku koji do sada nije koristio aplikaciju potrebno je omogućiti njegovu registraciju.

### Registracija korisnika - *Registration*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Use Case ID:* | UC2 | | | |
| *Use Case Name:* | Registracija korisnika na sustav | | | |
| *Created By:* | Tim 7 | | *Last Updated By:* |  |
| *Date Created:* | 27.1.2014. | | *Date Last Updated:* |  |
| *Actors:* | | Korisnik | | |
| *Description:* | | Pri prvom pristupanju sustavu, korisnik se mora registrirati kako bi mu se omogućio pristup sustavu | | |
| *Trigger:* | | Odabir opcije za registraciju | | |
| *Preconditions:* | |  | | |
| *Postconditions:* | | 1. Registriran je novi korisnik u sustav | | |
| *Normal Flow:* | | 1. Korisnik odabire opciju registracije 2. Otvara mu se forma u koju mora unijeti potrebne podatke (sva polja moraju biti obavezna): 3. Ime i prezime 4. Datum rođenja 5. Email 6. Lozinka 7. Nakon unosa podataka, korisnik mora potvrditi registraciju | | |
| *Alternative Flows:* | | 1. Nisu popunjeni obavezni podaci za registraciju, tijek se nastavlja u točki 2. 2. Korisnik je već registriran u sustav, ponovna registracija s istim podatcima nije moguća, tijek se zaustavlja | | |
| *Exceptions:* | |  | | |
| *Includes:* | |  | | |
| *Priority:* | |  | | |
| *Frequency of Use:* | |  | | |
| *Business Rules:* | |  | | |
| *Special Requirements:* | |  | | |
| *Assumptions:* | |  | | |
| *Additional documents* | |  | | |
| *Source* | |  | | |
| *Notes and Issues:* | |  | | |

1. Registracija korisnika -Registration
   1. Za registracija je potrebno unijeti slijedeće informacije :

|  |  |  |  |
| --- | --- | --- | --- |
| **Ime polja** | **Opis** | **Validacija** | **Dodatni zahtjevi** |
| Ime | Ime korisnika | Obavezno: DA |  |
| Prezime | prezime korisnika | Obavezno: DA |  |
| Email | Email adresa korisnika | Obavezno: DA |  |

* 1. Nakon što se korisnik uspješno registrira, sustav će automatski poslati email poruku na njegovu email adresu. Kako mi aktivirao svoj korisnički nalog, korisnik mora kliknuti na link koju mu je prosljeđen.
     1. Subjekt email-a:

|  |
| --- |
| Aktiviraj book2eat nalog |

* + 1. Sadržaj email-a:

|  |
| --- |
| Poštovanje [Ime] [Prezime],  Želimo vam srdačnu dobrodošlicu na book2eat, sustav za online rezervaciju restorana. Da bi aktivirali svoj korisnički nalog, molimo vas [klikni te ovdje].  Ukoliko imate bilo kakvih pitanje i sugestija, molimo vas da nas kontaktirate putem email-a info@book2eat.ba  Srdačno vas pozdravlja,  Vaš [www.book2eat.ba](http://www.book2eat.ba) |

* 1. Nakon aktivacije korisničkog naloga, sustav treba omogućiti printanje Ugovora o pristupanju. Izgled i sadržaj ugovora je dat u prilogu 3.2.

### Use Case 1

<detaljna specifikacija Use case-a 2>

### Use Case 2

<detaljna specifikacija Use case-a 2>

### Use Case n

<detaljna specifikacija Use case-a N>

## Klasifikacija funkcionalnih zahtjeva ili USE CASE

Ukoliko samo želimo implementirati kroz UC:

|  |  |
| --- | --- |
| **Oznaka slučaja korištenja** | **Nivo prioriteta** |
| **UC1** | visok |
| **UC2** | nizak |
| **itd.** |  |

<Ili ako želimo kroz izvedene FZ>

U ovom prvom primjeru (FZ-1) prioriteti su definirani u posebnoj tabeli, npr.

|  |  |
| --- | --- |
| **Oznaka funkcionalnog zahtjeva** | **Nivo prioriteta** |
| **FZ-1.2** | visok |
| **FZ-1.3** | nizak |
| **itd.** |  |

## Performance Requirements

*This subsection specifies both the static and the dynamic numerical requirements placed on the software or on human interaction with the software, as a whole. Static numerical requirements may include:*

*(a) The number of terminals to be supported*

*(b) The number of simultaneous users to be supported*

*(c) Amount and type of information to be handled*

*Static numerical requirements are sometimes identified under a separate section entitled capacity.*

*Dynamic numerical requirements may include, for example, the numbers of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions.*

*All of these requirements should be stated in measurable terms.*

*For example,*

*95% of the transactions shall be processed in less than 1 second*

*rather than,*

*An operator shall not have to wait for the transaction to complete.*

*(Note: Numerical limits applied to one specific function are normally specified as part of the processing subparagraph description of that function.)*

## Design Constraints

*Specify design constraints that can be imposed by other standards, hardware limitations, etc.*

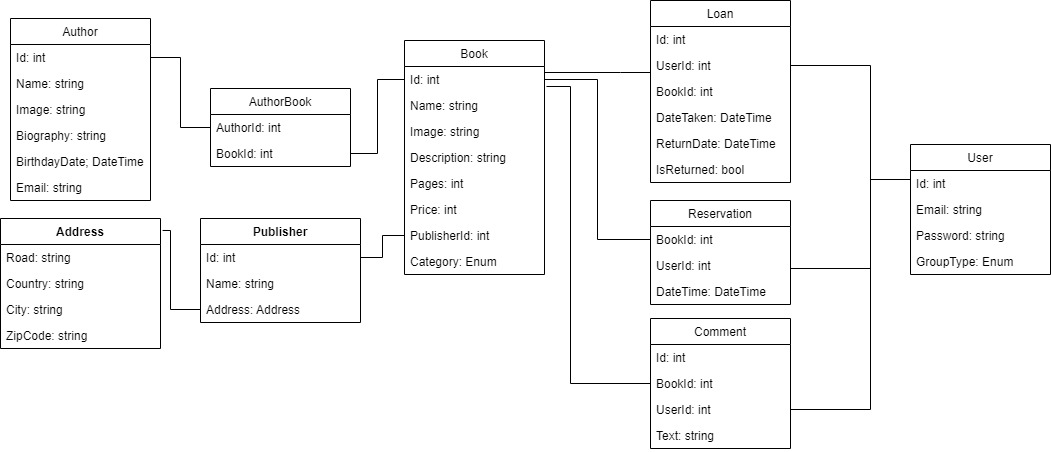
## Software system attributes

*There are a number of attributes of software that can serve as requirements. It is important that required attributes by specified so that their achievement can be objectively verified. The following items provide a partial list of examples. These are also known as non-functional requirements or quality attributes.*

*These are characteristics the system must possess, but that pervade (or cross-cut) the design. These requirements have to be testable just like the functional requirements. Its easy to start philosophizing here, but keep it specific.*

## Logical Database Requirements

### Dijagram klasa



### Opis klasa

Popis klasa:

* User
* Author
* Book
* Publisher
* AuthorBook
* Loan
* Reservation
* Comment

## Nefunkcionalni zahtjevi

### Korisnička sučelja

<Opišite logičke karakteristike svakog korisničkog sučelja koji su potrebni sustavu. Neke moguće stavke su reference za GUI ili dizajn standarde koji se moraju slijediti, standardi za fontove, ikone, *button labels*, slike, *color scheme*, uobičajeno korištene kontrole, izgled ekrana i ograničenja rezolucije, standard *buttons*, funkcije ili linkovi navigacije koji će se pojavljivati na svakom ekranu, (npr. *help button*), *shortcut keys*, standardi prikazivanja upozorenja i poruka, mogućnost prilagodbe za ljude sa oštećenjima vida.

Ovdje se ne treba nalaziti izgled finalne aplikacije.>

Korištenje material standarda za izgled aplikacije. Još nemamo definiran broj korisničkih sučelja, ali svaki će se temeljiti na material standardu.

### Vanjska sučelja i integracije

<Sučelja kao što su hardverski, softverski (API, formati datoteka), komunikacijski. Pogledajte opis u nastavku.>

#### Mrežna i hardverska sučelja

<Opišite logičke i fizičke karakteristike svake poveznice između hardver komponenti sustava. Ovaj opis može uključivati tipove uređaja za koje postoji podrška, kontrola softvera/hardvera interakcije i protoka podataka te komunikacijski protokoli koji će se koristiti.

Sustav će se moći koristiti na bilo kojem računalu koje ima monitor i miš, te na prijenosnim računalima.

Zahtjeve za sve komunikacijske funkcije koje će sustav koristiti, mrežne protokole i elektronske forme. Identificirajte sve komunikacijske standarde koji će se koristiti, kao što su HTTP ili FTP. Specificirajte sve činjenice koje se tiču sigurnosti komunikacije i moguće enkripcije, brzine prenosa podataka i mehanizme njihovih usklađivanja.>

#### Sustavski i softverski interfejsi

<Opišite konekcije između ovog sustava i drugih aplikacija (identificirajte sa imenom i verzijom), uključivši baze podataka, operativne sustave, alate, biblioteke i integrirane komercijalne komponente. Iznesite svrhu poruka, podataka i kontrolnih procesa koje SW komponente međusobno izmjenjuju. Opišite usluge koje su potrebne vanjskim poveznicama i prirodu njihove komunikacije. Identificirajte podatke koji će se dijeliti među komponentama - ako se radi toga treba na poseban način implementirati mehanizam za dijeljenje podataka, tada to specificirajte kao ograničenje.

Primjer jednog zahtjeva sustavskog interfejsa:

Pogledati primjer:

<https://wiki.cac.washington.edu/download/attachments/4273783/Requirements%20Specification%20Template.doc> → poglavlje 3.6.2>

### Upotrebljivost

<Upotrebljivost (eng. *usability*). Ovo poglavlje treba da uključuje sve nabrojane zahtjeve koji utiču na upotrebljivost, kao što su:

* Korisnički dokumentacija i online pomoć trebaju biti kompletirani.
* Sustav treba biti lagan za učenje.
* Specificirati zahtijevano vrijeme obuke za tipične i privilegovane korisnike za postizanje produktivnosti u određenim operacijama.

Pogledati primjer [www.it.uu.se/edu/course/homepage/pvt/**SRS**.pdf](http://www.it.uu.se/edu/course/homepage/pvt/SRS.pdf) → poglavlje 5.21

Za dodatna objašnjenja pogledate <http://www.usabilitynet.org/> >

### Performanse

<Karakteristike performansi sustava se moraju istaći u ovom dijelu. Uključite i specifična vremena odgovora. Gdje god je primjenjivo, povežite odgovarajući use- case po imenu. Podsjetnik: Ove vrijednosti vrijede pod uslovima koje su navedene u poglavlju „Pretpostavke i ovisnosti“ .

#### Kapacitet

* Kapacitet (npr. broj kupaca ili transakcija koje sustav može da opsluži)

*Pogledati primjer* [*www.it.uu.se/edu/course/homepage/pvt/****SRS****.pdf*](http://www.it.uu.se/edu/course/homepage/pvt/SRS.pdf) *→ poglavlje 5.12.1>*

#### Propusnost

* Propusnost (npr. broj kupaca ili transakcija u sekundi koje sustav može da opsluži)*>*

#### Dostupnost i pouzdanost

* Dostupnost (Availability) – specificirati postotak vremena dostupnosti ( xx.xx%), sati korištenja, pristup za održavanje, degradirano stanje operacije, pokrivenost različitih lokacija i sl.

*Pogledati primjer* [*www.it.uu.se/edu/course/homepage/pvt/****SRS****.pdf*](http://www.it.uu.se/edu/course/homepage/pvt/SRS.pdf) *→ poglavlje 5.11>*

* Srednje vrijeme između dva otkaza (Mean Time Between Failures (MTBF)) – se obično specificira u satima ali može takođe biti zadato i u danima, mjesecima ili godinama

*Pogledati primjer* [*www.it.uu.se/edu/course/homepage/pvt/****SRS****.pdf*](http://www.it.uu.se/edu/course/homepage/pvt/SRS.pdf) *→ poglavlje 5.14>*

* Srednje vrijeme otklanjanja greške (Mean Time To Repair (MTTR) ili The Mean Time To Fix (MTTF)) – koliko dugo je dozvoljeno da sustav ne bude u funkciji, nakon pojave kvara?

*Pogledati primjer* [*www.it.uu.se/edu/course/homepage/pvt/****SRS****.pdf*](http://www.it.uu.se/edu/course/homepage/pvt/SRS.pdf) *→ poglavlje 5.10>*

#### Latencija

* Vrijeme koje je potrebno da se izvrši zahtijevani servis (srednje, maksimalno)

*Pogledati primjer* [*www.it.uu.se/edu/course/homepage/pvt/****SRS****.pdf*](http://www.it.uu.se/edu/course/homepage/pvt/SRS.pdf) *→ poglavlje 5.12.3>*

* Stanja degradacije (šta je prihvatljivo stanje operacije kada je sustav degradiran u određenom smislu)>

### Korektnost

<Ovo podpoglavlje navodi zahtjeve koji se odnose na stepen do kojeg sustav može sadržavati nedostatke, a dalje biti prihvatljiv za kupca.>

#### Skriveni nedostaci

<Maksimalna stopa greški (bugs) i stepen kvarova (defects) – Greške je potrebno kategorizirani prema pojmovima niži (engl. minor), visoki (engl. significant), i kritični (engl. critical bugs). Zahtjevi moraju definisati šta se misli pod “kritična greška". Npr. potpuni gubitak podataka ili potpuna nemogućnost korištenja određenih dijelova funkcionalnosti sustava.>

*Pogledati primjer* [*www.it.uu.se/edu/course/homepage/pvt/****SRS****.pdf*](http://www.it.uu.se/edu/course/homepage/pvt/SRS.pdf) *→ poglavlje 5.5.1>*

#### Točnost

<Tačnost (Accuracy) – specificira tačnost (po nekom prihvaćenom standardu) koja se zahtijeva na izlazu sustava.

*Pogledati primjer* [*www.it.uu.se/edu/course/homepage/pvt/****SRS****.pdf*](http://www.it.uu.se/edu/course/homepage/pvt/SRS.pdf) *→ poglavlje 5.5.2 >*

#### Preciznost

<Preciznost (Precision) – specificira preciznost (po nekom prihvaćenom standardu) koja se zahtijeva na izlazu sustava.

*Pogledati primjer* [*www.it.uu.se/edu/course/homepage/pvt/****SRS****.pdf*](http://www.it.uu.se/edu/course/homepage/pvt/SRS.pdf) *→ poglavlje 5.5.3>*

#### Ažurnost

<Stepen do kojeg sustav mora osigurati da njegov pohranjena informacija je aktuelna (tj. up-to-date)>

*Pogledati primjer* [*www.it.uu.se/edu/course/homepage/pvt/****SRS****.pdf*](http://www.it.uu.se/edu/course/homepage/pvt/SRS.pdf) *→ poglavlje 5.5.4>*

### Instalacija

<Specifični zahtjevi za instaliranje sustava na serveru, klijentski računarima itd. Npr. prosječno vrijeme koje je potrebno da bi se instalirao sav potrebni softver na klijentskom računaru ne smije biti duži od 5 minuta.

Pogledati primjer [www.it.uu.se/edu/course/homepage/pvt/**SRS**.pdf](http://www.it.uu.se/edu/course/homepage/pvt/SRS.pdf) → poglavlje 5.8>

### Upravljanje i održavanje sustava

<Ovo poglavlje opisuje sve zahtjeve koje će unaprijediti podršku ili održavanje sustava koji se gradi, uključujući standarde kodiranja, konvencije imenovanja, fizički pristup za održavanje, uslužne programe za održavanje (odnosi se na sve faktore koji olakšavaju održavanje sustava u njegovoj upotrebi)

#### Nadzor

< engl. Monitoring: Navedite sve zahtjeve za nadzor ispravnosti sustava kroz praćenja uslova koji dovode do kritičnog stanja, otkrivanje grešaka, pregled log-datoteka itd.>

#### Održavanje

<engl. Maintenance. Odredite atribute sustava koji se odnose na jednostavnost održavanja. Ti zahtjevi se mogu odnositi na modularnost, kompleksnost itd. Zahtjevi se ne bi trebao biti postavljeni ovdje jednostavno zato što su mislili da će dobar dizajn u implementaciji.>

#### Operativnost

<Navedite sve normalne i specijalne operacije koje se zahtijevaju od strane korisnika sustava, uključujući:

• periodi rada bez ljudskog nadzora

• backup i recovery operacije>

### Prenosivost

<Ako je potrebno, navedite osobine koje su potrebne da bi se sustav premjestio na druge lokacije, računarske konfiguracije ili operativne sustave.>

### Skalabilnost

<Ovdje se definira mogućnost proširenja sustava kroz povećanje kapaciteta i propusnosti.

### Sigurnost

<npr. Sigurnost: Identificira podatke koji trebaju biti zaštićeni kao i tipove prijetnji kojima je svaki tip podataka izložen (posebni ili neuobičajeni sigurnosni zahtjevi posebno u odnosu na: pristup sustavu, enkripciju podataka, praćenje tragova). >

#### Zaštita sustava

<Primjer:

* enkripciju podataka
* praćenje tragova
* provjera integriteta podataka>

#### Autorizacija i autentifikacija

#### Privatnost

### Primjenjivi standardi

<Ovo poglavlje opisuje po značaju bilo koje standarde koji se mogu primijeniti, kao i specifične dijelove bilo kojeg od tih standarda koji se primjenjuju na opisani sustav. Npr. to može uključivati pravne, standarde kvaliteta i regulatorne standarde, industrijske standarde za korisnost, interoperabilnost itd..

1. Dizajn softveratreba da bude specificiran prema standardu xyz

<http://www.env.gov.bc.ca/csd/imb/3star/sdlc/4design/Software_Design_Description.doc>

1. Dizajn arhitekture treba da bude specificiran prema standardu xyz

<http://www.env.gov.bc.ca/csd/imb/3star/arch/docs/Deployment_Patterns.pdf>

1. Implementacija softverau programskom jeziku Java treba da bude izvršena prema standardu xyz

<http://www.env.gov.bc.ca/csd/imb/3star/sdlc/5build/java/java_standards.html>

1. Test prihvaćenosti treba da bude specificiran prema standardu xyz

<http://www.env.gov.bc.ca/csd/imb/3star/sdlc/6transtn/uat/uat.htm>

Prethodno navedeni linkovi na standarde su preuzeti sa:

<http://www.env.gov.bc.ca/csd/imb/3star/alpa_standards.html>*>*

### Robusnost

<Definira se stupanj po kojim će sustav nastaviti ispravno raditi u slučaju abnormalnog korištenja sustava. Primjer jednog zahtjeva: Svaka forma mora sadržavati validaciju unosa (pogledati FZ-2.1)

### Internacionalizacija

### Personalizacija

### Ograničenja

#### Ograničenja sustavskih resursa

<Ograničenja na hardver. Npr. diskovni prostor ili procesor: Dijeljeni iznajmljeni web-server (shared web hosting) se ne smije procesorski opteretiti 100% duže od X sekundi. >

#### Ograničenja na dizajn softvera i implementaciju

<Ovo poglavlje opisuje ograničenja dizajna softvera koji utiču na sustav. Primjeri uključuju ograničenja programskih jezika, zahtjevi softverskih procesa, razvojnih alata, ograničenje softverske arhitekture, nabavljene komponente ili biblioteke itd..

*Pogledati primjer* [*www.it.uu.se/edu/course/homepage/pvt/****SRS****.pdf*](http://www.it.uu.se/edu/course/homepage/pvt/SRS.pdf) *→ poglavlje 6.4.1>*

#### Ograničenja testiranja softvera

#### Pravna i regulatorna ograničenja

<Ovo poglavlje opisuje sve potrebne pravne odgovornosti, jamstva, autorska prava, patente, wordmark, žigove ili logotip-ove u softveru>

# Prilozi

<Ukoliko se prilozi koriste, potrebno je u dokumentu specificirati koji prilozi predstavljaju zahtjeve a koji to ne čine.>

<primjer priloga je npr. Obrazac za prijavu i sl,. iz realnog života>