Calorie Counter App

*Specifikacija zahtjeva sustava*

Studenti:

Marko Rezić, 590/RM

Hrvoje Zovko, 571/RM

Marin Zovko, 578/RM

Mostar, 19.01.2022.

Autor

|  |  |  |
| --- | --- | --- |
| **Naziv** | **Uloga** | **Odjel/Department** |
| Marko Rezić | Voditelj grupe |  |
| Hrvoje Zovko | Član grupe |  |
| Marin Zovko | Član grupe |  |
|  |  |  |

Povijest izmjena

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datum** | **Verzija** | **Status** | **Detalji izmjene** | **Autor** |
| 19.01.2022. | 1.0 | - | Dodan Use Case Diagram | Svi |
| 20.01.2022. | 1.1 | - | Dodan Class Diagram | Marko Rezić |
|  |  |  |  |  |
|  |  |  |  |  |

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 Calorie Counter App.

## Svrha dokumenta

Zahtjevi za sustav koji obavlja sve potrebne funkcionalnosti spremanja, održavanja i dohvaćanja podataka o proizvodima i ciljevima ishrane.

## Definicije, akronimi i skraćenice

|  |  |  |
| --- | --- | --- |
| **Term/Acronym** | **Definicija** | **Opis** |
| CCA | Calorie Counter App | Skraćenica aplikaciju za praćenje kalorija |
| 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 Calorie Counter App. Korisnici mogu unositi proizvode u svoj dnevnik ishrane i pratiti napredak prema svom cilju. Za cilj mogu birati da gube, dobivaju, ili održavaju kilažu. Aplikacija će omogućiti korisnicima i da vide detaljne informacije o hrani koju unose, kao što su mikro i makronutrijenti.

# Opis SUSTAVA

Sustav omogućava korisnicima da unose i mijenjaju proizvode u svojem dnevniku ishrane te da prate napredak prema svom cilju. Za cilj mogu birati da gube, dobivaju, ili održavaju kilažu. Aplikacija će omogućiti korisnicima i da vide detaljne informacije o hrani koju unose, kao što su mikro i makronutrijenti. Također mogu gledati prethodne upisane dane u dnevniku.

## Svrha SUSTAVA

* Ovaj sustav omogućava korisnicima brz i učinkovit način zapisivanja i praćenja dnevnih kalorija
* Sustav će pomoći korisnicima u postizanju ciljane težine
* Što je to sustav s korisničkog gledišta? Koje će ciljeve ispuniti?

S korisničkog gledišta, cilj sustava je ušteda vremena i truda kod praćenja kalorija. Najčešća prepreka u postizanju cilja je manjak volje za zapisivanjem kalorija, ili pak provjeravanje koliki vam treba biti dnevni unos. Stoga ova aplikacija olakšava taj postupak kako bi korisnici morali uložiti minimalan trud da dobiju potrebne informacije.

## 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** |
| *Superadmin* | *Vlasnik sustava* | Rijetko | Superadmin nije potreban za konstantno nadziranje sustava, ali u ponekim situacijama su potrebne više ovlasti prilikom rada s osjetljivim podacima, naročito što se tiče sigurnosti sustava | */* |
| *Moderator* | *Admin* | *Često* | *Moderator je zadužen za redovno pregledavanje novih proizvoda u bazi, te ispravljanje i brisanje neispravnih kako nebi došlo do neispravnosti podataka* | */* |
| *Korisnik* | *Osoba* | *Često* | *Korisnici moraju znati što im je kalorijski/težinski cilj kako bi popunili adekvatno podatke na temelju kojih aplikacija određuje dnevni kalorijski unos. Korisnik u dnevnik upisuje proizvode koje konzumira* | */* |

## Funkcionalnosti SUSTAVA

Registriranje na sustav, pregled dnevnika, unos proizvoda, biranje cilja, grafički pregled napretka

## Pretpostavke i ovisnosti

Potrebno je koristiti Mobilni uređaj platforme Android ili iOS.

## 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
* Biranje cilja
* Pregled dnevnika ishrane
* Dodavanje novih proizvoda
* Pretraživanje proizvoda
* Pregled nutrijenata
* Grafički pregled napretka

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>

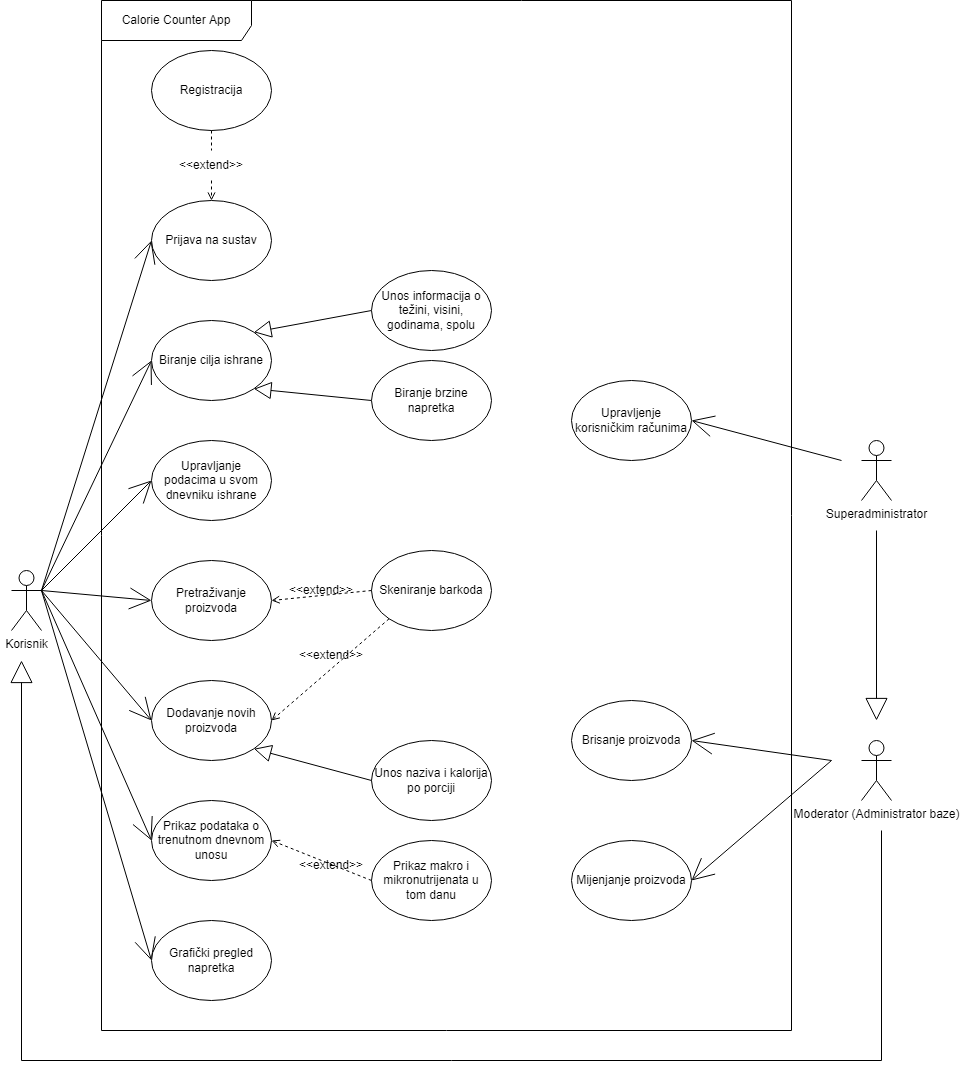
< npr.:

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:** | Prijava na sustav |
| **Description:** | Korisnik se prijavljuje u sustav |
| **Primary Actor:** | Korisnik, moderator, superadministrator |
| **Preconditions:** | Korisnik se nije prijavio u sustav |
| **Postconditions:** | Korisnik je prijavljen u sustav i može nastaviti dalje |
| **Main  Success Scenario:** | Korisnik unosi svoje korisničko ime i lozinku, sustav provjerava unesene podatke i ako su ispravni korisniku se daje pristup sustavu. |
| **Extensions:** | Ako uneseni podaci nisu ispravni, korisniku se daje do znanja kako je pogriješio i da unese ispravne podatke |
| **Frequency of Use:** | Svaki put kada korisnik prvi put pristupa sustavu, ulazna točka sustava |
| **Status:** | - |
| **Owner:** | - |
| **Priority:** | Visok |

|  |  |
| --- | --- |
| **ID:** | **2** |
| **Title:** | Registracija |
| **Description:** | Korisnik se registrira na sustav |
| **Primary Actor:** | Korisnik, moderator, superadministrator |
| **Preconditions:** | Korisnik se nije registrirao na sustav |
| **Postconditions:** | Korisnik je registriran na sustav i može početi proces prijave |
| **Main  Success Scenario:** | Korisnik unosi podatke za registraciju, sustav provjerava unesene podatke za registraciju i ako su valjani, korisnik je uspješno registriran |
| **Extensions:** | Ako podaci nisu ispravni, šalje se obavijest korisniku da mora ponoviti proces registracije, odnosno unijeti ispravne podatke za registraciju |
| **Frequency of Use:** | Korisnik se samo jednom registrira u sustav |
| **Status:** |  |
| **Owner:** |  |
| **Priority:** | Visok |

|  |  |
| --- | --- |
| **ID:** | **3** |
| **Title:** | Upravljanje korisničkim računima |
| **Description:** | Nadgledanje i upravljanje nad svim registriranim korisnicima i moderatorima u bazi podataka |
| **Primary Actor:** | Superadministrator |
| **Preconditions:** | Superadministrator ne nadgleda korisnike u bazi podataka sustava |
| **Postconditions:** | Superadministrator stalno nadgleda korisnike i može upravljati nad njima |
| **Main  Success Scenario:** | Superadministrator pristupa bazi podataka , provjerava sve korisnike trenutno registrirane na sustav, te ne nad njima radi akcije pod potrebi |
| **Extensions:** |  |
| **Frequency of Use:** | Svaki put kad superadministrator pristupi bazi podataka sustava, vrlo često |
| **Status:** |  |
| **Owner:** |  |
| **Priority:** | Visok |
|  |  |

|  |  |
| --- | --- |
| **ID:** | **4** |
| **Title:** | Brisanje proizvoda |
| **Description:** | Brisanje prije dodanih prozivoda u Calorie Count aplikaciji |
| **Primary Actor:** | Moderator |
| **Preconditions:** | Moderator je uvidio dodani proizvod koji nije valjan |
| **Postconditions:** | Moderator je izbrisao proizvod |
| **Main  Success Scenario:** | Moderator ima pregled svih proizvoda i ako vidi neki proizvod koji je dodan a nije ispravan, on mu pristupa i briše ga iz sustava trajno |
| **Extensions:** | Moderator izbriše ispravno dodan proizvod, te mora ga opet dodati na sustav |
| **Frequency of Use:** | Svaki put kada je pogrešno dodan proizvod, odnosno kada se nalazi neispravan proizvod na aplikaciji |
| **Status:** |  |
| **Owner:** |  |
| **Priority:** | Srednji |

|  |  |
| --- | --- |
| **ID:** | **5** |
| **Title:** | Mijenjanje proizvoda |
| **Description:** | Mijenjanje podataka proizvoda na sustavu |
| **Primary Actor:** | Moderator |
| **Preconditions:** | Neki podaci vezani za proizvod(informacije, npr. broj kalorija) su neispravni |
| **Postconditions:** | Neispravni podaci su promijenjeni od strane moderatora. |
| **Main  Success Scenario:** | Moderator uvidi neispravnost podataka kod proizvoda, pristupi proizvoda, te ažurira podatke tako da budu potpuno ispravni |
| **Extensions:** | Neki podaci opet mogu ostati neispravni koje moderator nije uvidio, pa se proces za njih mora ponoviti |
| **Frequency of Use:** | Svaki put kada su podaci neispravni za neki proizvod |
| **Status:** |  |
| **Owner:** |  |
| **Priority:** | Srednji |

|  |  |
| --- | --- |
| **ID:** | **6** |
| **Title:** | Biranje cilja ishrane |
| **Description:** | Korisnik u aplikaciji može izabrati odgovarajuću dijetu, odnosno ishranu kako bi postigao svoj krajnji cilj |
| **Primary Actor:** | Korisnik |
| **Preconditions:** | Korisnik je registriran i prijavljen na sustav no još nije izabrao kakvu će ishranu imati, te koji su njegovi ciljevi |
| **Postconditions:** | Korisnik je odabrao odgovarajući plan, te aplikacija mu je prikazala njegov cilj ishrane |
| **Main  Success Scenario:** | Nakon uspješne prijave, korisnik upiše svoje ciljeve i podatke, te uzvrat dobije cilj prehrane i odgovarajuću dijetu da bi ispunio taj cilj |
| **Extensions:** | Korisnik utipka pogrešne podatke te može dobiti plan koji nije idealan za njega |
| **Frequency of Use:** | Većinom samo jednom, no korisnik može promijeniti svoj plan ako to želi |
| **Status:** |  |
| **Owner:** |  |
| **Priority:** | Srednji |

|  |  |
| --- | --- |
| **ID:** | **7** |
| **Title:** | Upravljanje podacima u svom dnevniku ishrane |
| **Description:** | Korisnik unosi podatke u svom dnevniku ishrane dnevno, tako da regularno prati svoju ishranu te svoj dnevni unos |
| **Primary Actor:** | Korisnik |
| **Preconditions:** | Korisnik nije unio svoj dnevni unos u dnevnik ishrane |
| **Postconditions:** | Korisnik je unio svoj dnevni unos u dnevnik ishrane |
| **Main  Success Scenario:** | Korisnik pristupa aplikaciji te odmah unosi pojedeni obrok u dnevnik ishrane |
| **Extensions:** | Korisnik može unijeti pogrešne podatke, što može doći do netočne kalkulacije i nepravilnosti u dnevniku ishrane |
| **Frequency of Use:** | Jako čest, korisnik pristupa dnevniku ishrane nakon svakog obroka |
| **Status:** |  |
| **Owner:** |  |
| **Priority:** | Srednji |

|  |  |
| --- | --- |
| **ID:** | **8** |
| **Title:** | Pretraživanje proizvoda |
| **Description:** | Većinom se proces odvija kada korisnik želi unijeti određeni proizvod dnevnik ishrane, no i ako želi saznati kalorijsku vrijednost proizvoda |
| **Primary Actor:** | Korisnik |
| **Preconditions:** | Korisnik želi naći određeni proizvod ili podatke vezane za taj proizvod |
| **Postconditions:** | Korisnik nakon što je utipkao na search bar u aplikaciji je našao traženi proizvod |
| **Main  Success Scenario:** | Korisnik je našao traženi proizvod u aplikaciji te može pristupiti podacima tog proizvoda ili drugo |
| **Extensions:** | Proizvod koji korisnik traži se ne nalazi na aplikaciji, te je potrebno da manualno dodati |
| **Frequency of Use:** | Jako čest |
| **Status:** |  |
| **Owner:** |  |
| **Priority:** | Srednji |

|  |  |
| --- | --- |
| **ID:** | **9** |
| **Title:** | Dodavanje novih proizvoda |
| **Description:** | Korisnik ima opciju dodavanja novih proizvoda u aplikaciju, za olakšavanja sebi i drugim korisnicima u aplikaciji |
| **Primary Actor:** | Korisnik |
| **Preconditions:** | Korisnik pretražuje proizvod no on se ne nalazi u aplikaciji |
| **Postconditions:** | Korisnik je dodao željeni proizvod i on se je vidljiv i dostupan svima |
| **Main  Success Scenario:** | Korisnik upiše sve podatke vezane za proizvod te ga dodaje u aplikaciju. On postaje dostupan i svaki mu korisnik u aplikaciji može pristupiti i drugo. |
| **Extensions:** | Korisnik je unio pogrešne podatke vezane uz novi proizvod, te proces se mora ponoviti |
| **Frequency of Use:** | Jako često, svaki put kada korisnik uvidi da nema željenog proizvoda, on taj proizvod doda na sustav |
| **Status:** |  |
| **Owner:** |  |
| **Priority:** | Srednji |

|  |  |
| --- | --- |
| **ID:** | **10** |
| **Title:** | Skeniranje barkoda |
| **Description:** | Skeniranjem barkoda automatski se unosi obrok u dnevnik ishrane, ili možemo dodati proizvod u aplikaciju |
| **Primary Actor:** | Korisnik |
| **Preconditions:** | Korisnik u aplikaciji ima opciju skeniranja barkodom , no potrebno je da obrok koji želi unijeti ima barkod |
| **Postconditions:** | Skeniranjem barkoda proizvod je unesen, ili u sustav ili u korisnikov dnevnik ishrane |
| **Main  Success Scenario:** | Nakon što se skenirao barkod, svim makro i mikronutrijenti proizvoda se automatski dodaju, te njegov naziv i kalorijska vrijednost |
| **Extensions:** | Ako je došlo do greške tijekom skeniranja, korisnik dobije obavijesti da mora ponoviti proces skeniranja proizvoda |
| **Frequency of Use:** | Jako čest, koristi se često pri dodavanju proizvoda ili dodavanja obroka u dnevnu prehranu |
| **Status:** |  |
| **Owner:** |  |
| **Priority:** | Srednji |

|  |  |
| --- | --- |
| **ID:** | **11** |
| **Title:** | Prikaz podataka o trenutnom dnevnom unosu |
| **Description:** | Korisnik može stalno pristupiti prikazu trenutnog dnevnog unosa, da uvidi koliko je kalorija i obroka unio i dr. |
| **Primary Actor:** | Korisnik |
| **Preconditions:** | Korisnik nakon stvaranja dnevnika ishrane, dnevno unosi obroke u dnevnik, te želi u svakom trenutku pristupiti i uvidjeti dnevni unos |
| **Postconditions:** | Korisnik pristupa trenutnom dnevnom unosu te ima prikaz trenutno unesenih obroka, ukupan broj kalorija i ostale vrijednosti |
| **Main  Success Scenario:** | Korisnik želi znati trenutni unos te u aplikaciji on dobiva prikaz trenutno unesenih vrijednosti, obroka, makro i mikronutrijenti. |
| **Extensions:** | Prikaz unesenih podataka se ne prikazuje, pa korisnik možda mora ponoviti proces da uvidi prikaz dnevnog unosa |
| **Frequency of Use:** | Jako čest, korisnik dnevno će dosta puta pristupiti dnevnom unosu |
| **Status:** |  |
| **Owner:** |  |
| **Priority:** | Srednji |

|  |  |
| --- | --- |
| **ID:** | **12** |
| **Title:** | Prikaz makro i mikronutrijenata u tom danu |
| **Description:** | Mogućnost pregleda trenutno unesenih makro i mikronutrijenata u danu |
| **Primary Actor:** | Korisnik |
| **Preconditions:** | Korisnik je prijavljen na sustavu i želi pristupiti i uvidjeti njegov trenutni unos makro i mikronutrijenata na dnevnoj bazi |
| **Postconditions:** | Korisnik dobiva prikaz unesenih makro i mikronutrijenata u tom danu |
| **Main  Success Scenario:** | Korisnik pristupa njegovom dnevnom unosu i dobiva prikaz svih trenutno unesenih vrijednosti, a jedne od njih su makro i mikronutrijenti |
| **Extensions:** | Prikaz se možda ne učitava, pa korisnik mora ponoviti proces da uvidi trenutne unesene vrijednosti |
| **Frequency of Use:** | Čest |
| **Status:** |  |
| **Owner:** |  |
| **Priority:** | Nizak |

|  |  |
| --- | --- |
| **ID:** | **13** |
| **Title:** | Grafički pregled napretka |
| **Description:** | Korisnik ima mogućnost uvidjeti svoj napredak preko grafičkog prikaza koji slikovito prikazuje korisnikov napredak, stagnaciju ili mogući nazadak njegovog plana |
| **Primary Actor:** | Korisnik |
| **Preconditions:** | Korisnik koji je prijavljen na sustav ima opciju uvidjeti svoj napredak putem grafičkog prikaza |
| **Postconditions:** | Grafičkim prikazom korisnikovog trenutnog plana on dobije slikovit i jasan prikaz njegovog napretka |
| **Main  Success Scenario:** | Korisnik pristupa grafičkom prikazu napretka koja mu prikazuje njegovo stanje ishrane kroz određeni period, te korisnik uspješno dobiva saznanje o trenutnom napretku |
| **Extensions:** | Grafički prikaz ovisi o korisniku i njegovom postupnom dodavanju proizvoda u dnevnom unosu, ako je korisnik pogrešno unosio podatke u dnevnik ishrane, graf neće biti ispravan |
| **Frequency of Use:** | Ne koristi se najčešće nego nakon nekog perioda |
| **Status:** |  |
| **Owner:** |  |
| **Priority:** | Nizak |

## sudionici

### Ljudski sudionici

Vlasnik sustava, Moderatori i Korisnici.

### Hardware Actors

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

/

### Software System Actors

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

/

## 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.*

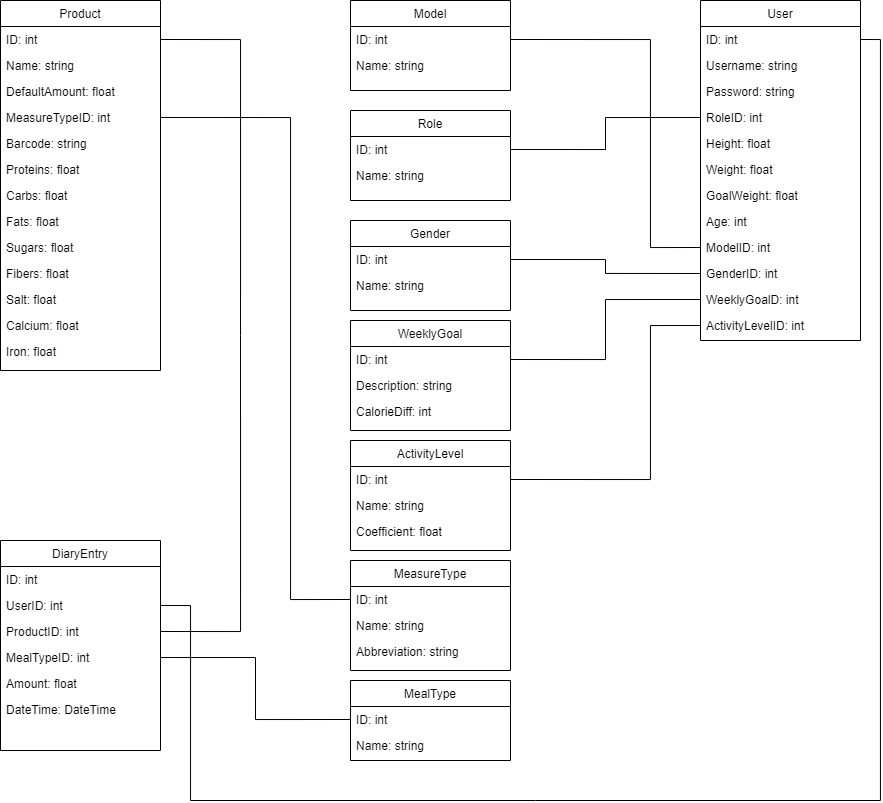
## Software system attributes

*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

### Relacijski model



### Opis tablica u relacijskom modelu

Popis tablica:

* User
* Product
* DiaryEntry
* Model
* Role
* Gender
* WeeklyGoal
* ActivityLevel
* MeasureType
* MealType

## 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>