

TEHNIČKA DOKUMENTACIJA

Programiranje	2
Veb programiranje	5

Programiranje

1. Opis sistema

Opis sistema objašnjava svrhu programa, kome je namenjen i koje probleme rešava. Bez detalja implementacije.

Odgovori koje opis sistema treba da pruži:

- Šta je sistem? – Primer: *Ovaj sistem je desktop aplikacija za evidenciju čitalaca u biblioteci.*
- Čemu služi? – Primer: *Sistem omogućava unos, čuvanje i pregled podataka o čitaocima i njihovim iznajmljivanjima.*
- Ko koristi sistem? – Primer: *Aplikaciju koriste zaposleni u biblioteci.*
- U kom okruženju radi? – Primer: *Sistem se koristi na računaru sa operativnim sistemom Windows.*

2. Zahtevi za projektovanje

Funkcionalni zahtevi govore šta sistem radi, a nefunkcionalni pod kojim uslovima to radi.

Primeri funkcionalnih zahteva:

- *Sistem mora omogućiti unos podataka o korisniku.*
- *Sistem mora omogućiti pregled sačuvanih podataka.*
- *Sistem mora prikazati rezultate u tabelarnom obliku.*

Primeri nefunkcionalnih zahteva:

- *Sistem mora biti jednostavan za korišćenje.*
- *Sistem mora obraditi greške pri radu sa bazom podataka.*
- *Sistem mora čuvati podatke u bazi podataka.*

3. Struktura sistema

Struktura sistema opisuje od kojih delova se sistem sastoji i kako su ti delovi međusobno povezani.

Primer strukture sistema:

Sistem se sastoji od sledećih delova:

- *Korisnički interfejs, omogućava unos i pregled podataka putem formi.*
- *Logički deo aplikacije, obrađuje podatke i izvršava pravila rada.*
- *Baza podataka, u kojoj se čuvaju podaci aplikacije.*

Korisnički interfejs omogućava komunikaciju korisnika sa sistemom, logički deo obrađuje podatke, a baza podataka čuva podatke aplikacije.

4. Opis korisničkog interfejsa

Opis korisničkog interfejsa prikazuje kako izgleda aplikacija iz ugla korisnika i koje elemente korisnik koristi za rad sa sistemom. Opis korisničkog interfejsa govori šta korisnik vidi i koristi, a ne kako je to programirano.

Primer opisa korisničkog interfejsa:

Forma Čitaoci omogućava unos i pregled podataka o čitaocima.

Sadrži sledeće elemente:

- *polja za unos osnovnih podataka o čitaocu*
- *dugme za unos podataka*
- *listu čitalaca*
- *dugme za izlaz iz aplikacije*

Opis bi se ponovio za svaku formu koja se nalazi u projektu.

5. Pravila rada i validacije

Pravila rada i validacije opisuju ograničenja, uslove i pravila koja važe tokom rada sistema i obezbeđuju ispravnost podataka i pravilno ponašanje aplikacije. Pravila rada i validacije opisuju šta je dozvoljeno i šta nije tokom rada aplikacije, bez objašnjenja kako je to tehnički urađeno.

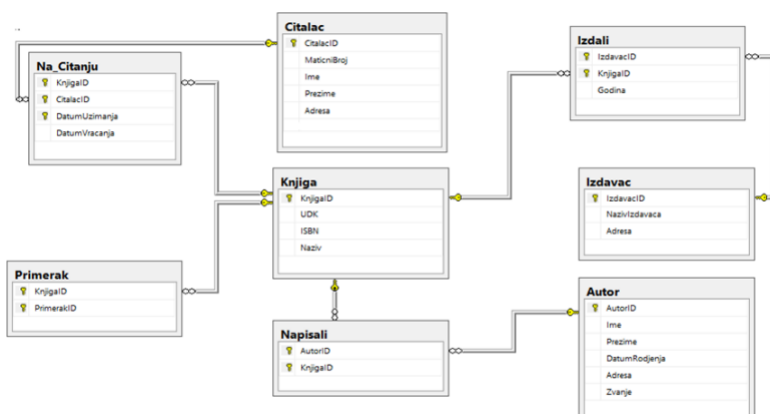
Primer pravila rada i validacije:

- *Broj članske karte mora biti jedinstven.*
- *U polje Broj članske karte dozvoljen je unos samo cifara.*
- *JMBG mora sadržati tačno 13 cifara.*
- *Sva obavezna polja moraju biti popunjena pre unosa podataka.*
- *Sistem ne dozvoljava unos neispravnih ili nepotpunih podataka.*
- *U slučaju greške pri radu sa bazom, sistem obaveštava korisnika porukom.*

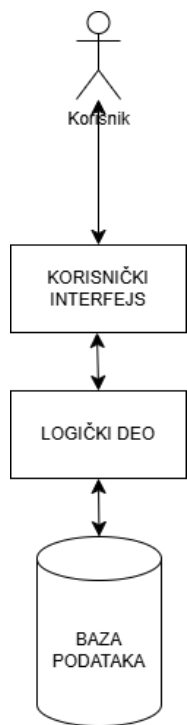
6. Dijagrami

Dijagrami služe za grafički prikaz sistema, njegovih delova i međusobnih veza. Omogućavaju lakše razumevanje strukture sistema i odnosa između njegovih komponenti. Dijagrami su crteži koji pokazuju kako je sistem organizovan, a ne kako je programiran.

Primer dijagrama baze podataka:



Primer dijagrama sistema:



U dijagramima su korišćeni osnovni i opšteprihvaćeni simboli za prikaz komponenti sistema i baze podataka.

Veb programiranje

1. Opis sistema

Opis sistema objašnjava svrhu veb sajta, kome je namenjen i koje probleme rešava.

Odgovori koje opis sistema treba da pruži:

- Šta je sistem? – Primer: *Ovaj sistem je veb sajt namenjen prikazu informacija o domaćim životinjama.*
- Čemu služi? – Primer: *Sistem omogućava pregled slika, zvučnih zapisa i kratkih opisa domaćih životinja.*
- Ko koristi sistem? – Primer: *Sajt koriste učenici i drugi korisnici koji žele da se informišu o domaćim životinjama.*
- U kom okruženju radi? – Primer: *Sistem se koristi putem internet pretraživača.*

2. Zahtevi za projektovanje

Funkcionalni zahtevi govore šta korisnik može da uradi, a nefunkcionalni kako sajt treba da funkcioniše.

Primeri funkcionalnih zahteva:

- *Sistem mora omogućiti navigaciju između svih stranica sajta.*
- *Sistem mora omogućiti reprodukciju zvuka pri prelasku miša preko slike.*
- *Sistem mora omogućiti prikaz dodatnih informacija o životinji klikom na sliku.*

Primeri nefunkcionalnih zahteva:

- *Sajt mora biti pregledan i jednostavan za korišćenje.*
- *Sajt mora pravilno funkcionisati u internet pretraživaču.*
- *Svi linkovi na sajtu moraju biti ispravni.*

3. Struktura veb sajta

Struktura veb sajta opisuje od kojih se delova veb sajt sastoji i kako su ti delovi međusobno povezani radi pravilnog funkcionisanja i lakog snalaženja korisnika.

Primer strukture veb sajta:

Veb sajt se sastoji od sledećih delova:

- *Stranice veb sajta koje sadrže informacije za korisnika.*
- *Navigacija koja omogućava kretanje između stranica.*
- *Korisnički interfejs za pregled i interakciju sa sadržajem.*

Navigacija povezuje stranice veb sajta, korisnički interfejs omogućava komunikaciju korisnika sa sajtom.

4. Struktura fajlova i resursa

Struktura fajlova i resursa opisuje način organizacije datoteka koje čine veb sajt, kao i raspored dodatnih resursa potrebnih za njegov pravilan rad i prikaz u internet pretraživaču.

Primer strukture fajlova i resursa:

Veb sajt se sastoji od sledećih fajlova i resursa:

- *HTML fajlovi, koji predstavljaju stranice veb sajta i sadrže njegov sadržaj*
 - *index.html, o-autoru.html, uputstvo.html*
- *CSS fajlovi, koji su zaduženi za izgled i raspored elemenata na stranicama.*
 - *CSS/ - stil.css*
- *JavaScript fajlovi, koji omogućavaju interakciju i dinamičko ponašanje elemenata na sajtu.*
 - *JS/ - skripta.js*
- *Folderi sa resursima, u kojima se nalaze slike, zvukovi i drugi multimedijalni sadržaji.*
 - *Slike/ - pas.jpg, kokoska.jpg, ovca.jpg, svinja.jpg, curka.jpg*
 - *Zvukovi/ - pas.mp3, kokoska.mp3, ovca.mp3, svinja.mp3, curka.mp3*

Fajlovi su organizovani u odgovarajuće foldere radi bolje preglednosti i lakšeg održavanja sajta. Razdvajanje strukture, izgleda i interakcije omogućava jasnu organizaciju projekta i ispravan prikaz sadržaja u internet pretraživaču.

5. Opis korisničkog interfejsa

Opis korisničkog interfejsa prikazuje kako veb sajt izgleda iz ugla korisnika i na koji način korisnik ostvaruje interakciju sa sadržajem na pojedinim stranicama sajta.

Primer opisa korisničkog interfejsa:

Na početnoj stranici (index.html) korisnik vidi pozadinu koja predstavlja prirodno stanište domaćih životinja. Na toj pozadini su prikazane slike pet domaćih životinja. Na vrhu stranice nalazi se navigacioni meni koji omogućava prelazak na ostale stranice sajta.

Opis bi se ponovio za svaku veb stranicu koja se nalazi u projektu.

6. Pravila rada i interakcije

Pravila rada i interakcije opisuju način na koji se veb sajt ponaša tokom korišćenja i kako korisnik ostvaruje interakciju sa njegovim elementima.

Primer pravila rada i interakcije:

- *Prelaskom miša preko slike domaće životinje aktivira se karakterističan zvuk te životinje.*
- *Klikom levim tasterom miša na sliku domaće životinje otvara se prozor sa kratkim opisom izabrane životinje.*
- *Prozor sa opisom može se zatvoriti na predviđen način, nakon čega se korisnik vraća na prethodni prikaz.*
- *Navigacioni linkovi omogućavaju prelazak između svih stranica sajta, pri čemu se stranice otvaraju bez grešaka.*
- *Sadržaj se prikazuje jasno i pregledno, bez preklapanja elemenata.*

7. Dijagram strukture sajta

Dijagram strukture sajta (sitemap) prikazuje sve stranice veb sajta i veze (navigaciju) između njih. Dijagram ne prikazuje fajlove i foldere.

Primer dijagrama strukture sajta:

