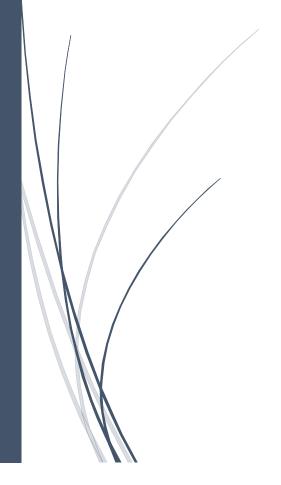
5/12/2019

Ticketeer

Dizajn sistema



Autori:

BRANKA STANKOVIĆ, FILIP ADAMOVIĆ, SUKARA MILOŠ, NIKOLA BLAGOJEVIĆ

SADRŽAJ

1.	Uvo	d	1
	1.1.	Namjena sistema	1
	1.2.	Projektni ciljevi	1
	1.3.	Referentni dokumenti	2
	1.4.	Kratak pregled dokumenta	2
2.	Arhi	tektura postojećeg sistema	3
3.	Prec	Iložena arhitektura	3
	3.1.	Kratak pregled arhitekture i funkcionalnosti podsistema	3
	3.2.	Dekompozicija sistema	4
	3.3.	HW/SW mapiranje	6
	3.4.	Perzistentni sloj	7
	3.5.	Kontrola prava pristupa i sigurnosti	9
	3.6.	Kontrola toka	9
	3.7.	Granična stanja	10

1. UVOD

1.1. NAMJENA SISTEMA

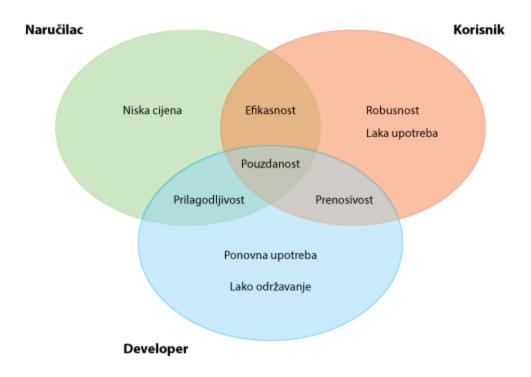
Ticketeer je web aplikacija za rezervaciju ulaznica koja je namijenjena za različite klijente i vrste događaja. Prevashodno ima za cilj ubrazavanje procesa rezervisanja karata za različite vrste događaja kao i unapređenje kvaliteta usluga koje naručioci softvera pružaju krajnjim korisnicima. Sistem je namijenjen širokom spektru organizatora različitih događaja (pozorišta, bioskopi, restorani, koncertne dvorane itd.) Naručioci sistema će imati lakši uvid u rezervacije ulaznica, dok će krajnjim korisnicima rezervacija ulaznica biti dostupnija. Dobra karakterisitka ovog sistema je prilagodljivost različitim potrebama i željama konkretnog naručioca softvera.

1.2. PROJEKTNI CILJEVI

Ticketeer je klijent-server aplikacija koja podržava veći broj funkcija za interakciju korisnika sa sistemom i odgovarajućim grafičkim korisničkim interfejsom (GUI). Ticketeer nudi ograničen broj funkcionalnosti i za korisnike koji nisu registrovani na sistem.

Ovaj softver je dizajniran tako da ispuni sledeće projektne ciljeve:

- Pouzdanost podrazumjeva da je softver u mogućnosti da izvrši traženu funkciju pod određenim uslovima u određenom vremenskom periodu.
- Ponovna upotreba-
- Efikasnost se ogleda u nastojanju da se putem softvera omogući brza i laka rezervacija ulaznica.
- Prenosivost se ogleda u mogućnosti korišćenja softvera i svih njegovih funkcionalnosti na različitim uređajima i u različitim okruženjima.
- Prilagodljivost se ogleda u mogućnosti ovog softvera da se primjeni za različite naručioce i različite tipove događaja.
- Robusnost predstavlja mogućnost sistema da radi pri velikim opterećenjima ili da reaguje na nevalidne situacije.
- Laka upotreba se zasniva na jednostavnom i intuitivnom dizajnu, što omogućava korišćenje sistema na jednsotavan način.
- Lako održavanje se odnosi na mogućnost zamjene i nadogradnje hardvera i softvera.
- Niska cijena u zavisnosti od zahtjeva naručioca



Slika 1Projektni ciljevi

1.3. REFERENTNI DOKUMENTI

[1] Specifikacija softverskih zahtjeva, Ticketeer

1.4. KRATAK PREGLED DOKUMENTA

Ovaj dokument se sastoji od 3 tri poglavlja:

- 1. Uvod
- 2. Arhitektura postojećeg sistema
- 3. Predložena arhitektura

U uvodnom dijelu dokumenta opisani su: Namjena sistema, Projektni ciljevi, te su navedeni referentni dokumenti.

U drugom dijelu navedene su arhitekture postojećih sistema, na osnovu kojih je bilo potrebno izabrati jednu za dizajniranje našeg sistema.

Treće poglavlje dokumenta opisuje izabranu arhitekturu kao i razloge zbog kojih je ona izabrana, te opise svih bitnih aktivnosti prilikom projektovanja softvera.

2. ARHITEKTURA POSTOJEĆEG SISTEMA

Pod pretpostavkom da nema postojećeg sistema sa ovakvim skupom funkcionalnosti, ovaj sistem će biti izgrađe od nule, tj. predstavljaće greenfield projekat. Zbog toga se mora pažljivo odabrati arhitekturni stil.

Arhitekturni stilovi koje se najčešće koriste su:

- Klijent-Server arhitekturni stil
- Peer-to-Peer arhitekturni stil
- MVC (Model-View-Controller) arhitekturni stil
- Troslojni arhitekturni stil
- Četvoroslojni arhitekturni stil

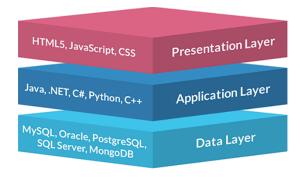
Budući da je **Ticketeer** web aplikacija, izabran je troslojni arhitekturni stil. Web browser će implementirati korisnički interfejs, web server servisira zahtjev web browser-a i DBMS upravlja podacima.

3. PREDLOŽENA ARHITEKTURA

3.1. KRATAK PREGLED ARHITEKTURE I FUNKCIONALNOSTI PODSISTEMA

Troslojni arhitekturni stil je arhitekturni stil u kojem je sistema strukturisan na tri hijerarhijska podsistema (sloja). Ovaj arhitekturni stil se tipično koristi u web programiranju. Često se koristi u aplikacijama kao specifičan tip Klijent-Server sistema.

Troslojni arhitekturni stil pruža mnoge pogodnosti za proizvodna i razvojna okruženja modularizacijom korisničkih interfejsa, poslovne logike i slojeva za skladištenje podataka.



Slika 2 Troslojni arhitekturni stil

Sistem je podijeljen na tri podsistema:

- Sloj prezentacije korisnički interfejsi
- Sloj aplikacije poslovna logika
- Sloj podataka reprezentacija i manipulacija objektima

3.2. DEKOMPOZICIJA SISTEMA

Dekompozicija sistema je postupak kojim se sistem reprezentuje kao kolekcija podsistema. Podsistem je kolekcija blisko povezanih klasa, asocijacija, operacija, događaja i ograničenje. Dekompozicijom sistema na (relativno nezavisne) podsisteme, više timova može istovremeno da razvija pojedine podsisteme uz minimalnu međusobnu komunikaciju.

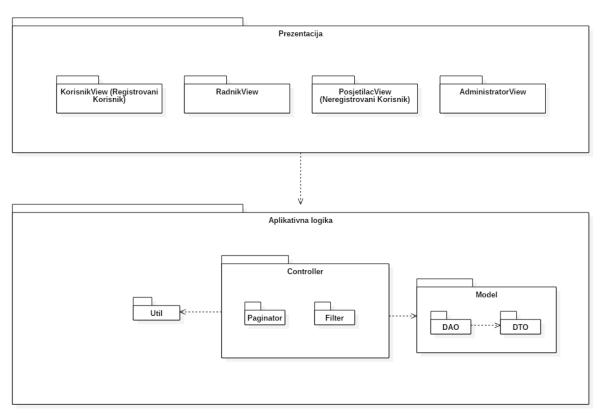
Dekompozicija sistema Ticketeer je zasnovana na Troslojnom arhitekturnom stilu.

Podsistem prezentacije se sastoji od:

- Podsistema za rad sa neregistrovanim korisnicima omogućava interakciju sa korisnicima koji nisu kreirali nalog. Oni mogu da pregledaju repertoare, događaje, te kreiraju nalog.
- Podsistem za rad sa registrovanim korisnicima omogućava interakciju sa korisnicima koji imaju kreiran nalog. Oni mogu da pregledaju repertoare, događaje i kreiraju rezervaciju za željeni događaj.
- Podsistem za radnike omogućava interakciju sa radnicima koji vrše obradu rezervacija.
- Podsistem za administratore omogućava nadgledanje i upravljanje sistemom.

Podsistem aplikativne logike služi za interakciju sa korisnicima.

Podsistem reprezentacije i pristupa podacima obezbjeđuje interakciju sa DMBS.



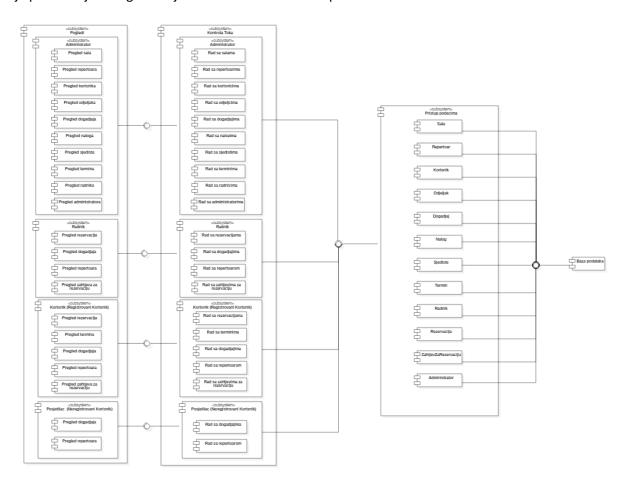
Slika 3 Dijagram paketa ciljnog sistema

3.3. HW/SW MAPIRANJE

HW/SW mapiranje daje odgovor na to kako se hardverski ili softverski realizuju podsistemi te prikazuje njihovu međusobnu povezanost. Za prikaz mapiranja se koriste dvije vrste dijagrama:

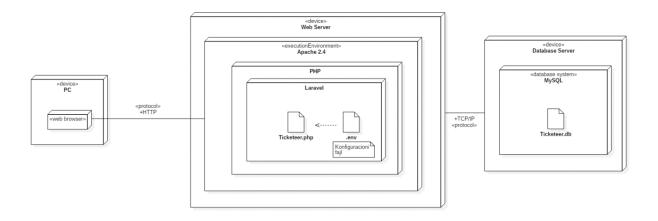
- Dijagram komponenata (eng. component diagram)
- Dijagram razmještaja (eng. deployment diagram)

Komponenta predstavlja zamjenjivi dio sistema koji realizuje skup interfejsa. Na dijagramu komponenata je predstavljena organizacija i zavisnosti između komponenata.



Slika 4 Dijagram komponenata ciljnog sistema

Dijagram razmještaja služi za modelovanje rasporeda komponenata u eksploataciji sistema. To je graf kog čine čvorovi (eng. *nodes* – apstrakcije fizičkih objekata) i veze čvorova (eng. *communication associations* – apstrakcije fizičkih veza između objekata).

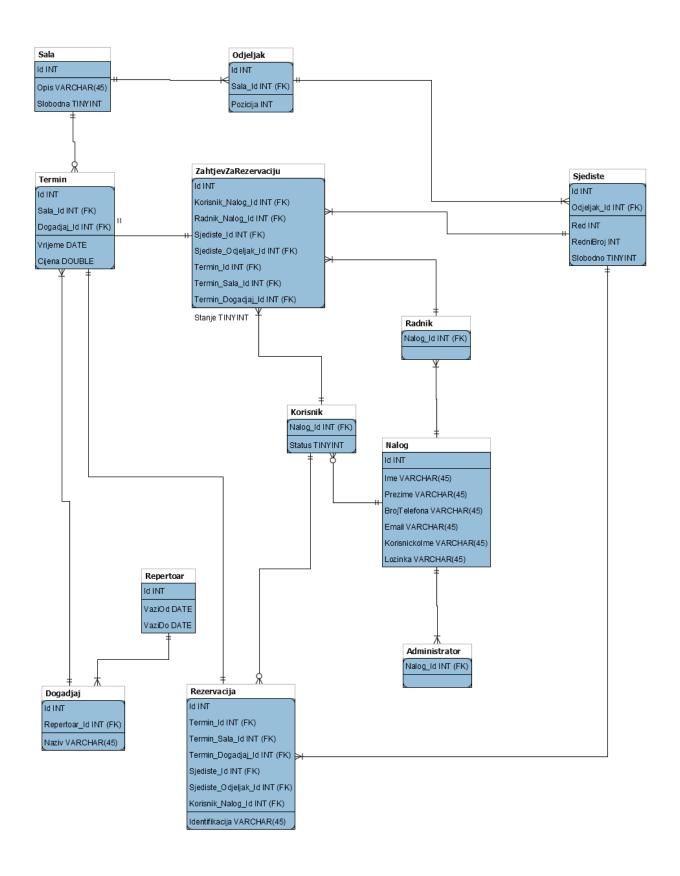


Slika 5 Dijagram razmještaja ciljnog sistema

3.4. PERZISTENTNI SLOJ

Perzistentni sloj upravlja perzistentnim objektima,odnosno objektima čije stanje mora da se trajno čuva. Ticketeer čuva podatke o korisnicima tj. njihovim nalozima,zahtjevima za rezervacije koje su poslali, kreiranim rezervacijama, salama, zauzetim sjedištima u salama, događajima, i repertoarima. Svi ovi podaci su neophodni za ispravno funkcionisanje sistema. Obzirom da je Ticketeer zamišljen kao klijent server aplikacija, neophodno je omogućiti konkurentan pristup podacima. Konkurentan pristup podacima omogućava da svi korisnici sistema imaju mogućnost istovremenom pristupu podacima.

Podaci se čuvaju u serverskom dijelu aplikacije. Zbog navedenih karakteristika, perzistentni sloj smo realizovali pomoću relacionih baza podataka.



3.5. KONTROLA PRAVA PRISTUPA I SIGURNOSTI

Kontrola pristupa i sigurnost su obezbijeđeni podjelom korisnika u različite grupe od kojih svaka grupa korisnika ima pristup samo onim funkcionalnostima i podacima za koje ima privilegije. Spisak učesnika, klasa, kao i lista operacija koje svaki učesnik može da izvrši nad objektima date klase su dati u matrici pristupa.

			Vlasa			
Učesnici	Klase					
	Rezervacija	Termin	Dogadjaj	Repertoar	ZahtjevZaRezervaciju	
Administrator		kreiraj() azuriraj() obrisi()	pregledaj() kreiraj() azuriraj() obrisi()	pregledaj() kreiraj() azuriraj() obrisi()		
Registrovani korisnik	pregledaj() otkazi()	izaberi()	pregledaj()	pregledaj()	kreiraj()	
Radnik	pregledaj() azuriraj()		pregledaj()	pregledaj()	pregledaj() obradi()	
Neregistrovani korisnik			pregledaj()	pregledaj()		
	Sala	Odjeljak	Sjediste	Nalog		
Administrator	pregledaj() kreiraj() azuriraj() obrisi()	pregledaj() kreiraj() azuriraj() obrisi()	pregledaj() kreiraj() azuriraj() obrisi()	pregledaj() azuriraj()		
Registrovani korisnik						
Radnik						
Neregistrovani korisnik						

Tabela 1 Kontrola prava pristupa

3.6. KONTROLA TOKA

Kontrola toka u okviru Ticketeer sistema je distribuirana (decentralizovana). Komunikacioni obrazac u distribuiranoj kontroli toka koji je korišćen je **asinhrona komunikcija.** Ovaj obrazac karakteriše da klijet šalje zahtjev i ne čeka odgovor nego nastavlja sa radom. Odgovor nije odmah raspoloživ nego će biti raspoloživ naknadno. Komunikacija se vrši razmjenom poruka.

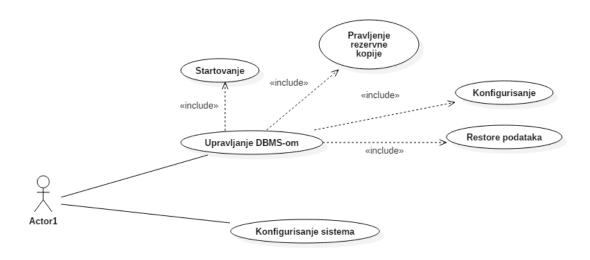
Da bi se omogućilo konkurentno izvršavanja, tj. da bi se održalo konzistentno stanje sistema u slučaju konkuretnog izvršavanja odredjenjih operacija, potrebno je da se transakcije realizuju u izolaciji, odnosno da se sljedeća transakcija može obaviti tek nakon što je prethodna u redu čekanja obrađena.

3.7. GRANIČNA STANJA

Granična stanja sistema se modeluju graničnim slučajevima upotrebe (boundarz use case). Granični slučajevi upotrebe tipično se ne identifikuju tokom analize sistema, jer su mnogi granični slučajevi posljedica dizajna.

Ticketeer sadrži sljedeće slučajeve graničnih stanja:

- Početak rada sistema sistem prikazuje formu za prijavu. Nakon prijave na sistem prikazuje se odgovarajuća forma u zavisnosti od statusa korisnika koji se prijavio. Korisnički interfjest omogućava korisnicima da uđu na sistem, promjene lozinku, ili prikazuje obavještenje u slučaju da korisničko ime i/ili lozinka nisu tačni.
- Terminacija rada sistema nakon prestanka rada na sistemu, oslobađaju se svi zauzeti resursi kao što su mrežna konekcija, veza sa bazom itd.
- Manipulacija izuzecima sistem će obavijestiti korisnika o svim nekorektnim unosima i neuobičajenim stanjima u kojima se može naći (nedostatak kredita prilikom online kupovine ulaznice, unos nekorektne vrijednosti prilikom dpune kredita itd.)
- Otkazi obuhvataju greške, bug-ove i nepredvidive pojave kao što su nestanak električnog napajanja, hardverski otkazi ili eventualno pad čitavog sistema. Za pravilno funkcionisanje sistema odgovoran je administrator. Administrator obavlja određene aktivnosti u cilju boljeg funkcionisanja sistema: Analiza sistema i identifikacija potencijalnih problema u sisitemu, izrada rezervnih kopija podataka, rješavanje zastoja u sistemu i izvještavanje itd.



Slika 6 Granični slučajevi upotrebe