

Specifikacija sistemskog dizajna

Pozorište na dlanu

Studenti:

1. Jelena Jandrić 1112/18
2. Bojana Jovičić 1118/18
3. Violeta Jaguzović 1134/18
4. Danijela Vukosav 1138/18

Sadržaj

1.	Uvod	3
	Namjena sistema.....	3
	Projektni ciljevi	4
	Definicije i skraćenice	6
	Referentni dokumenti.....	6
	Kratak pregled dokumenta i predložene arhitekture.....	7
2.	Arhitektura postojećeg sistema.....	8
3.	Predložena arhitektura	9
	Kratak pregled arhitekture funkcionalnosti podsistema	9
	Dekompozicija sistema	9
	HW/SW mapiranje	12
	Perzistentni sloj.....	15
	Kontrola prava pristupa i sigurnost	16
	Kontrola toka.....	17
	Granična stanja sistema	18

1. Uvod

Namjena ovog dokumenta je jasno predstavljanje osnovnih koncepata sistemskog dizajna za projekat pod nazivom "Pozorište na dlanu". Dokument će služiti kao osnova za implementaciju sistema, a nakon svake aktivnosti projektovanja biće ažuriran. U dokumentu su izloženi primarni projektni ciljevi i njihovo obrazloženje, pregled podsistema u skladu sa odabranim arhitekturnim stilom te dijagrami komponenti i razmještaja za inicijalnu verziju. Opisana je analiza realizacije pristupa perzistentnom sloju kao i najvažniji koncepti vezani za kontrolu prava pristupa i sigurnost kontrole toka i granična stanja sistema.

Namjena sistema

"Pozorište na dlanu" je sistem koji prvenstveno treba da omogućiti korisniku online rezervisanje ulaznica za pozorište, osim toga sistem nudi brži i jednostavniji uvid u sve buduće i prošle predstave.

Funkcionalnosti koje sistem treba da obezbijedi su:

- Registracija posjetilaca
- Registracija biletarnika
- Dodavanje i izmjena predstava
- Dodavanje i izmjena vijesti u pozorištu
- Rezervacija karata za predstave
- Komentarisane i ocjenjivanje predstava

U fazi izrade aplikacije definiše se jedan administrator koji ima najviše privilegije. Sistem obezbjeđuje funkcionalnost kreiranja naloga za različite vrste korisnika. Sistem takođe obezbjeđuje funkcionalnost registracije posjetioce pomoću aplikacije za posjetioce.

Administrator ima privilegije registrovanja naloga biletarnika, kao i mogućnost izmjene i brisanja tih naloga. Ova funkcionalnost sistema omogućava potpunu kontrolu unutar hijerarhijskog sistema učesnika u aplikaciji.

Administrator ima mogućnost da dodaje predstave u listu predstava i da vrši izmjenu već dodatih

predstava. Pored predstava, administrator upravlja i dodavanjem vijesti u pozorištu. Glavna funkcionalnost koju sistem nudi je mogućnost rezervacije karte. Ta funkcionalnost je omogućena samo registrovanim posjetiocima. Rezervacija se vrši popunjavanjem odgovarajuće forme za rezervaciju karte.

Registrovanim posjetiocima je omogućeno da pri pregledu odgovarajuće predstave pregledaju već postojeće komentare i ocjene, kao i da popunjavanjem odgovarajuće forme komentarišu i ocjenjuju predstave.

Projektni ciljevi

Ovaj sistem omogućava posjetiocu da efikasno online rezerviše ulaznicu za predstavu u pozorišti. Osim toga, sistem nudi brži i jednostavniji uvid u sve buduće i prošle predstave, te mogućnosti komentaranja i ocjenjivanja predstava lakom upotrebom sistema. Sistem će biti dizajniran tako da korisnik maksimalno uveća produktivnost pomaganjem u automatizovanju procesa koji bi se u suprotnom morao raditi ručno. Cilj ovog softverskog sistema je sprovođenje procesa rezervisanja i praćenja pozorišnih dešavanja na efikasniji način nego što se to do sad sprovodilo. Ovaj sistem donosi značajnu vremensku efiksanost.

Ostali važniji ciljevi su:

- Lakoća održavanja

Najvažniji projektni cilj jeste lakoća održavanja softverskog sistema. Ovaj cilj će biti ostvaren kroz grafički korisnički interfejs koji će administrator omogućiti da na jednostavan način prati rad same aplikacije, da mijenja, da dodaje novi i briše postojeći sadržaj aplikacije. Pored toga pri projektovanju sistema odabran je MVC arhitekturni stil koji bi trebao da olakša intervenciju u slučaju nekakvih problema u radu same aplikacije.

- Dobra dokumentacija

Važno je redovno ažuriranje dokumentacije sistema nakon svake verzije aplikacije na način da ona oslikava trenutno stanje. Redovno će se ažurirati dokumentacija realizovanih funkcionalnih

i nefunkcionalnih zahtjeva, kao i karakteristike dizajna sistema, radi lakšeg upoznavanja zainteresovanih strana sa funkcionalnostima sistema.

- Laka upotreba

Neophodno je obezbijediti i realizovati sistem koji će biti jednostavan za korištenje te kroz realizaciju interfejsa omogućiti budućim korisnicima sistema nesmetano korištenje istog u skladu sa njihovom informatičkom pismenošću i poznavanjem tehnologija.

Definicije i skraćenice

Biletarnik	<i>Osoba zadužena za rad na biletarnici</i>
Administrator	Osoba koja ima najviše privilegije u radu sa administrativnom aplikacijom
Nalog	Učesnicima se proširuje mogućnost rada sa sistemom
Registrovani korisnik	Osoba koja koristi usluge sistema „Pozorište na dlanu“ preko Web aplikacije i posjeduje korisnički nalog
Framework	Univerzalno okruženje za razvoj aplikacija koje softveru pruža generičke funkcionalnosti
MySQL	Višenitni korisnički SQL sistem za upravljanjem bazama podataka
MVC	Model View Controller arhitekturni stil
DBMS	Sistem za upravljanje bazom podataka
HTTP	Hypertext Transfer Protocol

Referentni dokumenti

Specifikacija korisničkih zahtjeva – “Pozorište na dlanu”.

Kratak pregled dokumenta i predložene arhitekture

Svrha ovog dokumenta je da predstavi tehničke aspekte sistema Pozorište na dlanu i tehnologije korištene za razvoj i implementaciju aplikacije. Cilj ovog dokumenta je pružiti čitaocu bolje razumijevanje načina na koji se aplikacija razvija i implementira kroz primjere zahtjeva, ograničenja i arhitekturu sistema.

Dokument je organizovan u nekoliko cjelina:

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

U uvodnom dijelu dokumenta data je namjena sistema koji se razvija. Navedeni su glavni ciljevi kojima teži softver koji se razvija, akronimi i skraćenice koje su korištene u dokumentu kao i referentni dokumenti.

Pošto sistem ne postoji, u pitanju je razvijanje novog sistema, u arhitekturi postojećeg sistema dat je pregled postojećih relevantnih arhitektura za dati sistem. U poglavlju predložena arhitektura nalazi se kratak pregled predložene arhitekture iz ptičje perspektive i funkcionalnosti podsistema.

2. Arhitektura postojećeg sistema

Pozorište na dlanu je nov softverski sistem i u procesu analize dizajna sistema naophodno je definisati arhitekturu ciljnog sistema. Za implementaciju ove aplikacije izabran je MVC arhitekturni stil. Ovaj arhitekturni stil pruža mnogo lakše održavanje sistema, a performanse nisu značajno narušene. Arhitektura će biti zatvorenog tipa pri čemu prezentacioni sloj interaguje sa kontrolerima koji izvršavaju poslovnu logiku i koji pristupaju sloju podataka.

3. Predložena arhitektura

Kratak pregled ahitekture i funkcionalnosti podsistema

Ciljni sistem će da koristi ASP.NET framework koji podržava MVC(Model, View, Controller) arhitekturu koja vrši raslojavanje na tri sloja:

1. Sloj za upravljanje podacima – strukture podataka koje se nalaze iza cijele aplikacije i reprezentovane su MySQL bazom podataka.
2. Prezentacioni sloj – korisnički interfejsi koji su reprezentovani sa CSHHTML fajlovima
3. Sloj za poslovnu logiku – „middleman“ koji povezuje view i model i omogućava razmjenu podataka između njih.

Za perzistiranje i upravljanje podacima koristiće se MySQL DBMS.

Dekompozicija sistema

Sljedeći podsistemi su identifikovani u sistemu:

1. Podsystem za upravljanje bazom podataka
2. Podsystem za poslovnu logiku
3. Podsystem za korisnički interfejs

➤ Podsystem za upravljanje bazom podataka

Podsystem za upravljanje bazom podataka sadržaće bazu podataka i MySQL DBMS koji je povezan sa ASP.NET frameworkom koji će se koristiti unutar web aplikacije. Baza podataka će biti mjesto gdje će se čuvati i održavati svi podaci o pozorištu. Pristup bazi podataka biće ograničen na korisničke upite. Neke ključne funkcije podsistema za upravljanje bazom podataka uključuju:

- Upravljanje korištenja tabele baze podataka iz drugih podsistema
- Vraćanje ispravnih podataka podsistemu za poslovnu logiku
- Vraćanje ispravnih podataka podsistemu za korisnički interfejs
- Ažuriranje tabela baza podataka
- Čuvanje podataka na sigurnom.

➤ Podsystem za poslovnu logiku

Aplikacija će koristiti upite za dobijanje podataka od podsistema za upravljanje bazom podataka kako bi mogao da ih analizira i izvrši manipulaciju nad njima, te u skladu sa zahtjevom proslijedi podsistemu za korisnički interfejs. Neki od glavnih zadataka ovog podsistema su:

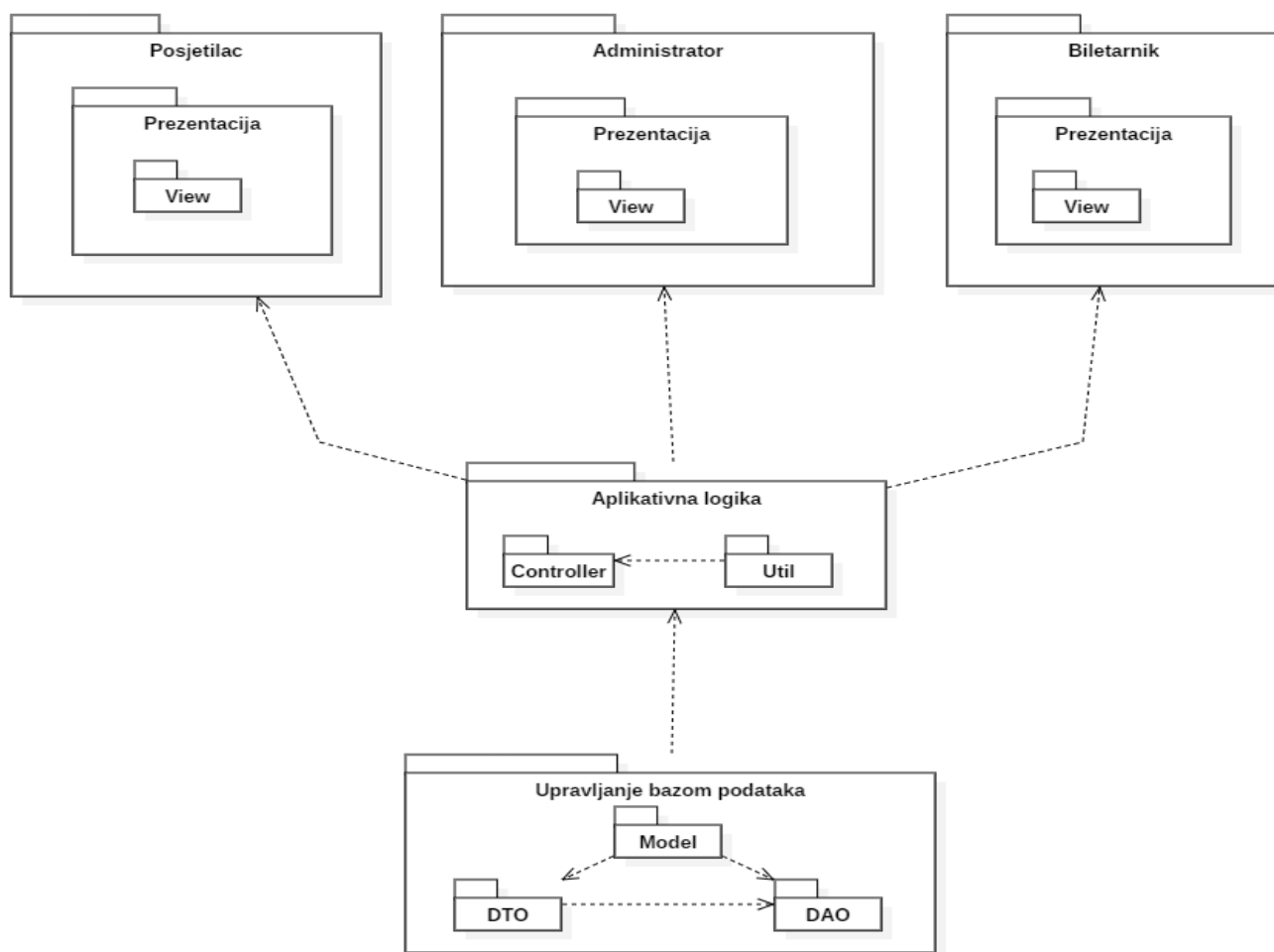
- Dohvatanje informacija o predstavama
- Dohvatanje informacija o određenoj predstavi
- Dohvatanje informacija o glumcima
- Dohvatanje informacija o rezervacijama itd.

➤ Podsystem za korisnički interfejs

Koristi se kako bi se ostvarila komunikacija sa korisnikom i kako bi se korisniku prikazali podaci koji se čuvaju u bazi podataka. Pošto korisnici u zavisnosti od toga kojoj grupi pripadaju imaju različite mogućnosti i imaju različite poglede i zbog toga ovaj podsistem se sastoji iz tri podsistema u zavisnosti od pogleda. Neke od funkcija koje obavlja ovaj podsistem su:

- Predstavljanje podataka koji se nalaze u bazi podataka na korisniku prihvatljiv način
- Pregledanje predstavljenih podataka

Na sljedećoj slici je dat dijagram paketa koji grafički predstavlja podsisteme ciljnog sistema.



HW/SW mapiranje

Hardverska konfiguracija sistema koristiće serversku mašinu koja će biti host za podsistem za upravljanje bazom podataka, mašinu koja će biti host za podsistem za poslovnu logiku i klijentsku mašinu koja će biti host za web browser odnosno za prezentaciju.

Različiti čvorovi će biti povezani kroz TCP/IP protokolom HTTP.

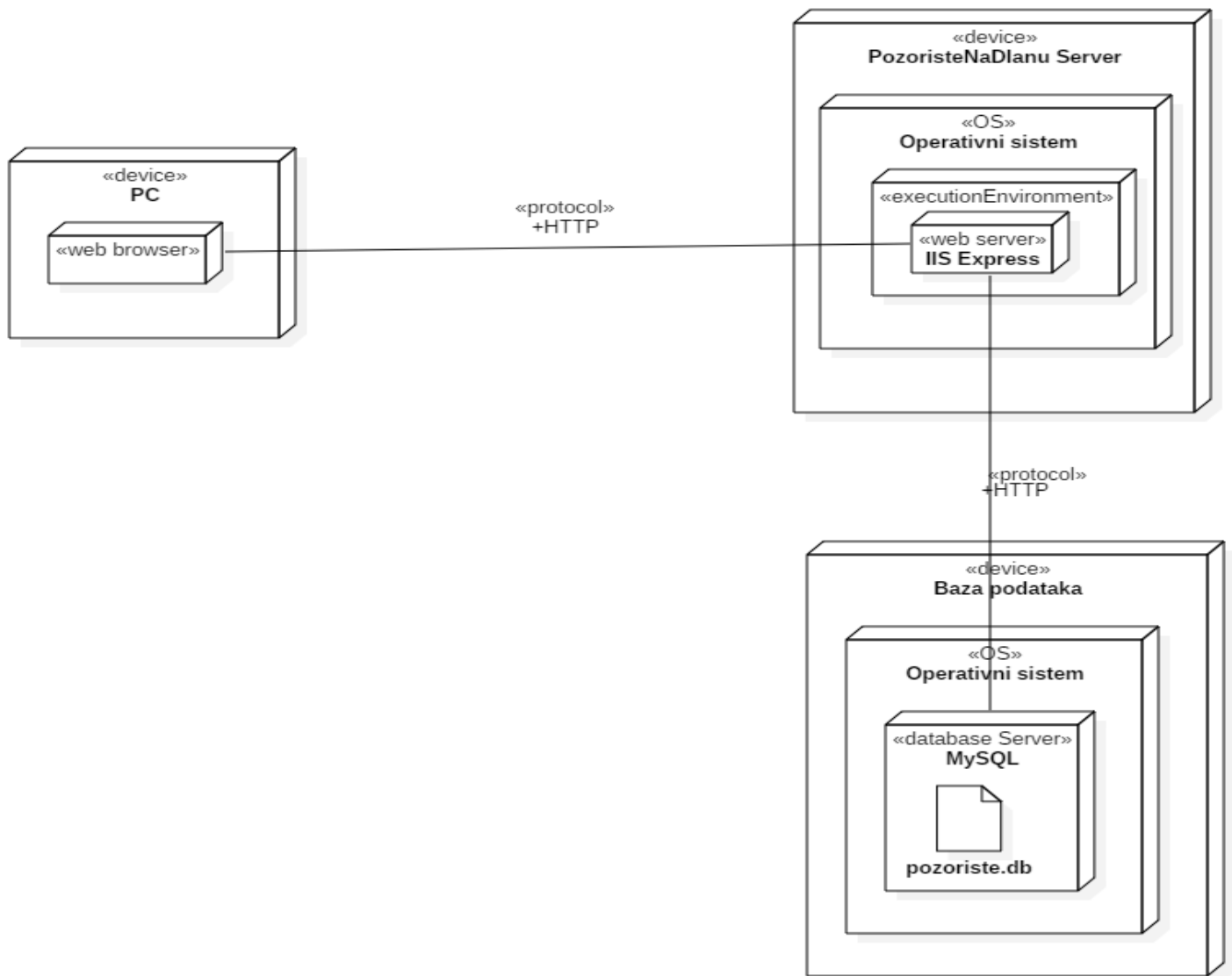
Na klijentskoj strani koristiće se gotove komponente koje uključuju web browser poput Google Chrome, Mozilla Firefox ili Opera ili koji mogu da se koriste i na Linux i na Windows operativnom sistemu.

Na serverskoj mašini nalaziće se aplikacija koja ustvari predstavlja aplikativnu logiku sistema I koja će koristiti upite za dobijanje podataka od podsistema za upravljanje bazom podataka kako bi server mogao da ih analizira i izvrši manipulaciju nad njima

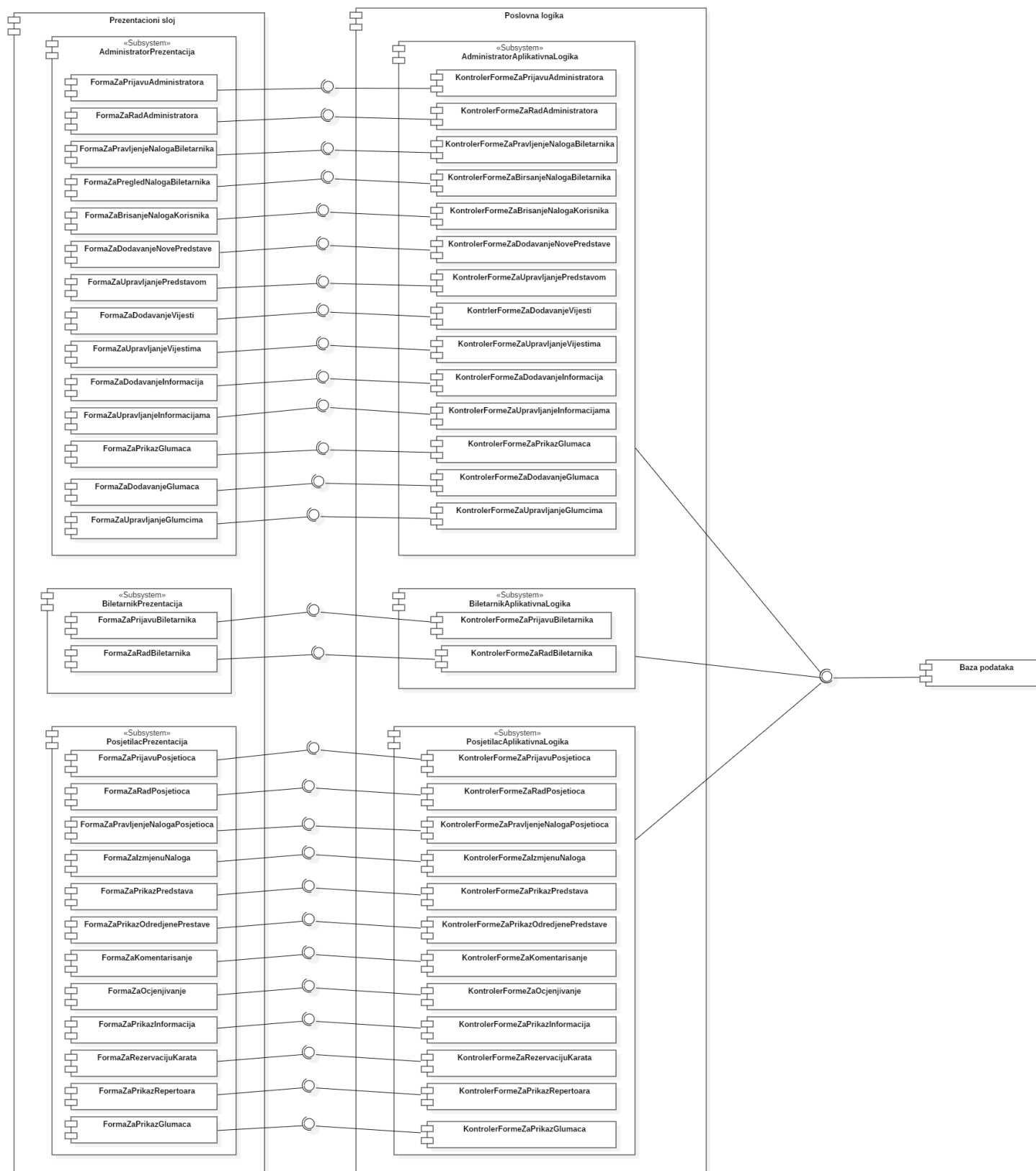
Uskladu sa predloženom arhitekturom podsistemi od kojih će se sastojati buduća aplikacija su sljedeći: prezentacioni sloj (sloj interakcije sa korisnicima), sloj za poslovnu logiku koja će biti smještena na web serveru i sloj za upravljanje podacima. U slučaju povećanog broja korisnika, aplikativni server može biti distribuiran na više uređaja u nekoj od sljedećih faza razvoja .

Dijagramom komponenata dat je razmještaj komponenata budućeg sistema. Komponenta prezentacionog sloja se sastoji iz tri podkomponente: komponenta za administratora, za biletarnika i za posjetioca. Svaka od ove tri komponente sadrži forme koje su dostupne korisnicima u zavisnosti od toga kojoj grupi korisnika pripadaju, administrator ima svoje forme, biletarnik ima svoje forme i isto tako posjetilac ima svoje forme. Svaka forma, bez obzira na to za koju grupu korisnika je namijenjena ima svoj kontoler s kojim je uparena. Kontroleri forme se nalaze u komponenti koja predstavlja aplikativnu logiku samog sistema i koja se nalazi na aplikativnom server. Ovi kontroleri forme imaju za zadatak da u zavisnosti od toga kako korisnik koristi koju formu odgovore na zahtjev korisnika. Perzistentni sloj predstavlja zasebnu komponentu i služi za čuvanje podataka koji su od značaja za ciljni sistem za upravljanje čuvanja tih podataka.

Na sljedećim slikama dati su dijagram razmještaja i dijagram komponenata.



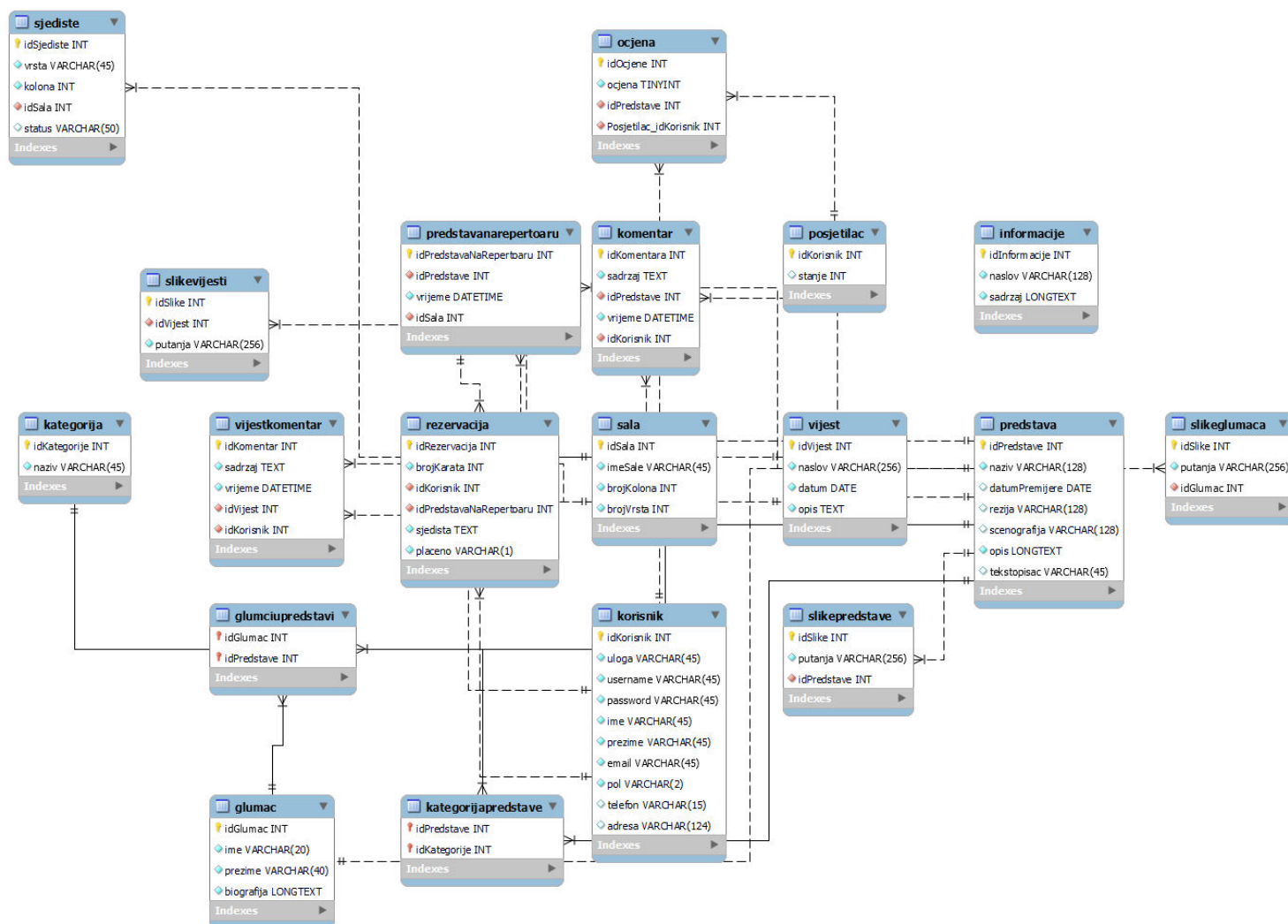
Dijagram razmještaja



Dijagram komponentata

Perzistentni sloj

Javno dostupne informacije o pozorištu ažuriraće se svaki put kada administrator izvrši dodavanje novih predstava, informacija, vijesti i tome slično. Zbog toga za čuvanje podataka koji su od značaja za posmatrani sistem koristiće se baza podataka. Za upravljanje podacima koristićemo MySQL DBMS. Sistem će primjenjivati određeni mehanizam za potvrdu identiteta za pristup bazi podataka u slučaju kada se vrši izmjena podataka koji se čuvaju. Naša baza podataka biće potpuno relacionalna. Dat je konceptualni model:



Kontrola prava pristupa i sigurnost

S obzirom na to da u sistemu postoje tri vrste korisnika i da svaki od njih ima različitu namjenu imaju i različite kontrole pristupa. Administrator ima najveće privilegije, on ima pravo pristupa svim podacima koji se čuvaju u bazi podataka. Podatke koji se nalaze u bazi može da mijenja, može da dodaje nove i da briše postojeće. Posjetilac, ukoliko nije registrovan, ima mogućnost pregleda informacija koji se nalaze u bazi podataka, kroz aplikaciju, nema pravo direktnog pristupa samoj bazi podataka i nema mogućnost mijenjanja podataka u bazi podataka. Ukoliko je korisnik registrovan, onda ima mogućnost da vrši promjene koje su vezane za nalog koji posjeduje, a te izmjene se tiču rezervacija koje se čuvaju za taj nalog, da pravi nove rezervacije i da mijenja postojeće, da ostavlja komentare na predstave i da ocjenjuje predstave. Biletarnik ima mogućnost da vrši potvrdu i poništavanje rezervacija, pa tako ima mogućnost da pristupa samo rezervacijama dok ostalim podacima ne može da pristupa. Da bi vršili izmjene u bazi podataka svaki od korisnika mora da ima korisnički nalog, korisničko ime i lozinkom kojom se autentikuje. Aplikacija će imati statičku kontrolu pristupa i u nastavku su date tabele koje to ilustruju:

Učesnici	Klase				
	Nalog	Predstava	Vijest	Repertoar	Rezervacija
Administrator	+napravi()	+dodaj()	+dodaj()	+dodaj()	
		+izmijeni()	+izmijeni()	+izmijeni()	-
	+obrisi()	+obrisi()	+obrisi()	+obrisi()	
		+pregledaj()	+pregledaj()	+pregledaj()	
Posjetilac	+napravi()	+pregledaj()	+pregledaj()	+pregledaj()	+dodaj() +izmijeni() +obrisi()
Biletarnik					+potvrdi()
	-	-	-	-	+ponisti()
					+dodaj()
					+pregledaj()

Učesnici	Klase		
	Komentar	Glumac	Informacije
Administrator		+dodaj()	+dodaj()
	+obrisi()	+izmijeni()	+izmijeni()
	+dodaj()	+obriši()	+obriši()
		+pregledaj()	+pregledaj()
Biletarnik	-	-	-
Posjetilac	+dodaj()	+pregledaj()	+pregledaj()

Sistem će podržati konkurentan pristup više korisnika za resurse na kojima postoji potreba za tim.

Jedina osoba koja može da mijenja sadržaj sadržaj same aplikacije je administrator, a ostali korisnici mogu konkurentno da pregledaju sadržaj aplikacije. Svaki posjetilac će imati mogućnost da mijenja samo podatke koji su vezani za njegov nalog tako da to ne utiče na rad ostalih korisnika.

Biletarnik će imati mogućnost da vrši potvrdu i poništavanje rezervacija u zavisnosti od toga da li je karta kupljena, kako više biletarnika može u isto vrijeme da prodaje karte ovaj konkurentan pristup biće obrađen pomoću DBMS-a, pa će sadržaj aplikacije da bude konzistentan.

Kontrola toka

Glavni zadatak ASP.NET-a je da obradi zahtjev koji je primio server, a zatim poziva funkcije i daje odgovor na taj zahtjev. Kada primi prvi zahtjev, u datoteci Global.asax, objekti rute se dodaju u tabelu rute. Tabela rute ima zadatak da preslika URL-ove u odgovarajuće rukovaoce. U suštini rutiranje je sistem

za određivanje koji rukovaoc se podudara sa kojim URL-om. Ukoliko je rukovalac koji odgovara URL, ili URL nije pronađen u tabeli rute, vraća se 404 HTTP statusni kod. Ukoliko je rukovalac pronađen dalji tok se preusmjerava na IRouteHandler. RouteHandler ima zadatak da odredi HTTP rukovalac koji će služiti zahtjevu, i to radi na osnovu zahtjeva koji je primio u RequestContext. RouteHandler kreira instancu MVCHandler i proslijedi RequestContext instancu tom rukovaocu. RequestContext instanca se koristi da bi se identifikovao IControllerFactory objekat koji se koristi da se kreira kontroler. Zatim MVCHandler metoda poziva kontrolerovu Execute metodu. Kontroler definiše koja metoda će da se izvrši. Nakon što se metoda izvrši, metoda vraća rezultat u vidu objekta klase View.

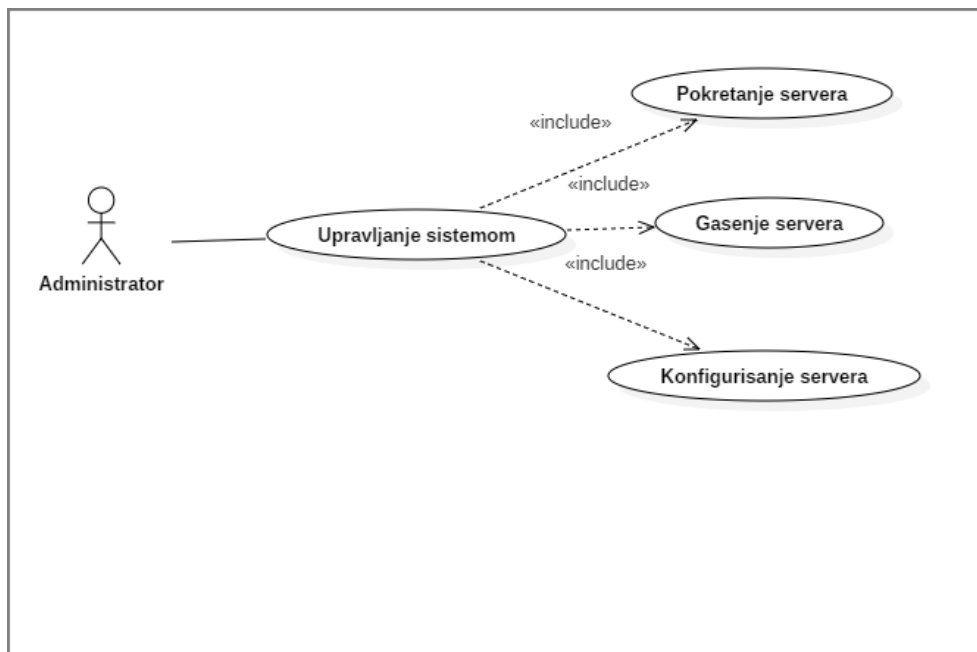
Granična stanja sistema

Inicijalizacija sistema

Pri pokretanju aplikacije, administrator i biletarniku se prikazuje forma za prijavu na sistem. Administrator i biletarnik treba da unesu kredencijale na osnovu kojih se vrši njihova autentikacija. U slučaju uspješne autentikacije administratoru i biletarniku se prikazuju forme za rad.

U slučaju posjetioca, pri pokretanju aplikacije prikazuje mu se početna forma.

Sistem će biti pokrenut na namjenskom server. Sistem je dizajniran tako da radi neprekidno. Ponovna pokretanja treba da izvršava samo administrator server. Sve promjene potrebne na server treba da izvršava samo administrator. Terminiranje sistema treba da budu rijetka i nisu poželjna.



Otkazi

Način za praćenje izuzetaka biće obrađen na nivou aplikacije. Ovo će omogućiti objedinjavanje dijelova za evidentiranje i obavješćavanje o izuzecima na jednom prikladnom mjestu. Globalni rukovalac izuzecima moći će da obradi kako specifične izuzetke napisane za potrebe aplikacije tako i generičke izuzetke kojima ne rukuje. Nakon što globalni rukovalac izuzecima obradi izuterak, korisnik će se preusmjeriti na web stranicu koja pokazuje da je došlo do greške. Svi neobrađeni izuzeci će se evidentirati u log datoteku sa nekim detaljima o prirodi kvara.

Ciljni sistem se u potpunosti oslanja na ispravno funkcionisanje server sa bazom podataka. Ukoliko se desi da padne čvor sa bazom podataka korisniku će biti prikazana informacija o grešci. Ukoliko aplikativni server ne bude dostupan, bilo da je došlo do greške u radu ili je nestalo napajanja korisniku će biti prikazana informacija o grešci.