Dizajn sistema

**KarteBL**

****

Stefan Savić

Igor Todić

Ognjen Milekić

Nemanja Janković

Damjan Milekić

# Uvod

Pored toga što ovaj dokument služi kao dopuna specifikacije korisničkih zahtjeva, njegova svrha je detaljan opis dizajna KarteBl sistema.

## Namjena sistema

KarteBL je web aplikacija. Koriste je osobe, koje žele digitalno kupiti kartu za koncert. Svrha aplikacije je brza i jednostavna kupovina karata, takodje aplikacija nudi jednostavan prikaz događaja, informacija o događajima i broj dostupnih karata za odabrani događaj. Postoje tri vrste ucesnika : administrator, korisnik i organizator. Korisnik se sam registruje na sistem dok se organizatorski nalozi kreiraju od strane admina.

## Projektni ciljevi

Projektni ciljevi u softverskom rješenju su referentna tačka za svaku realizaciju tog rješenja. Kada su dobro definisani, čine uspješniju realizaciju softverskog rješenja. Oni se poklapaju sa nefunkcionalnim zahtjevima navedenim i detaljno opisanim u specifikaciji sofverskih zahtjeva. Ovde ćemo ih navesti i ukratko opisati.

Izrada intuitivne web aplikacije, koja je oblikovana s ciljem da bude veoma pristupačna. Omogućava jednostavno korištenje čak i korisnicima bez naprednog informatičkog znanja. Informativan prikaz koncerata te informacija o koncertima.

Za izvršavanje sistema biće neophodna internet konekcija.

## Definicije i skraćenice

|  |  |
| --- | --- |
| DBMS | Database Management System(Sistem za upravljanje bazom podataka) predstavlja softversko rješenje, koje omogućava rad sa bazom podataka. |
| Mobilna aplikacija | Aplikacija za skeniranje karata na ulazu na koncerte. |
| Server | Softver koji prima zahtjeve od klijenata i salje odgovore. |
| SRS | Software requirement specification (Specifikacija softverskih zahtjeva). |
| Korisnik | Osoba, koja ima svoj nalog na sistemu i može mu  pristupiti pomoću prijave. |
| Spring boot | Radni okvir za platformu Java, koji obezbjeđuje  infrastrukturu za razvoj backend aplikacije. |
| API | Application Programming Interface |
| MVC | Model-View-Controller. Odvaja prikaz informacija od interakcije korisnika sa istima. |
| Klijent-Server arhitektura | Arhitekturni stil, koji se dijeli na dva dijela : 1. Dio, koji se izvodi na serveru-vraća izvještaje klijentu.  2. Dio, koji se izvodi na korisničkom računaru-  klijentu-šalje zahtjeve serveru. |
| UML | Unified Modeling Language |

## Referentni dokument

Referentni dokument je specifikacija korisničih zahtjeva KarteBL sistema.

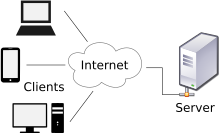
## Kratak pregled dokumenta

Ovaj document opisuje dizajna sistema, koji je detaljnije opisan u specifikaciji softverskih zahtjeva. Navedena je namjena sistema, projektni ciljevi, definicije i skraćenice (terminologija sistema) i reference. U ovom dokumentu su prikazani dijagrami komponenti i dijagrami razmještaja i konceptualni model za ciljni sistem. Takođe, biće prikazana kontrola toka (redoslijed izvršavanja akcija).

# Arhitektura sistema

Sistem je zasnovan na klijent-server arhitekturi. Jedan ili više servera obrađuje servise klijentima.

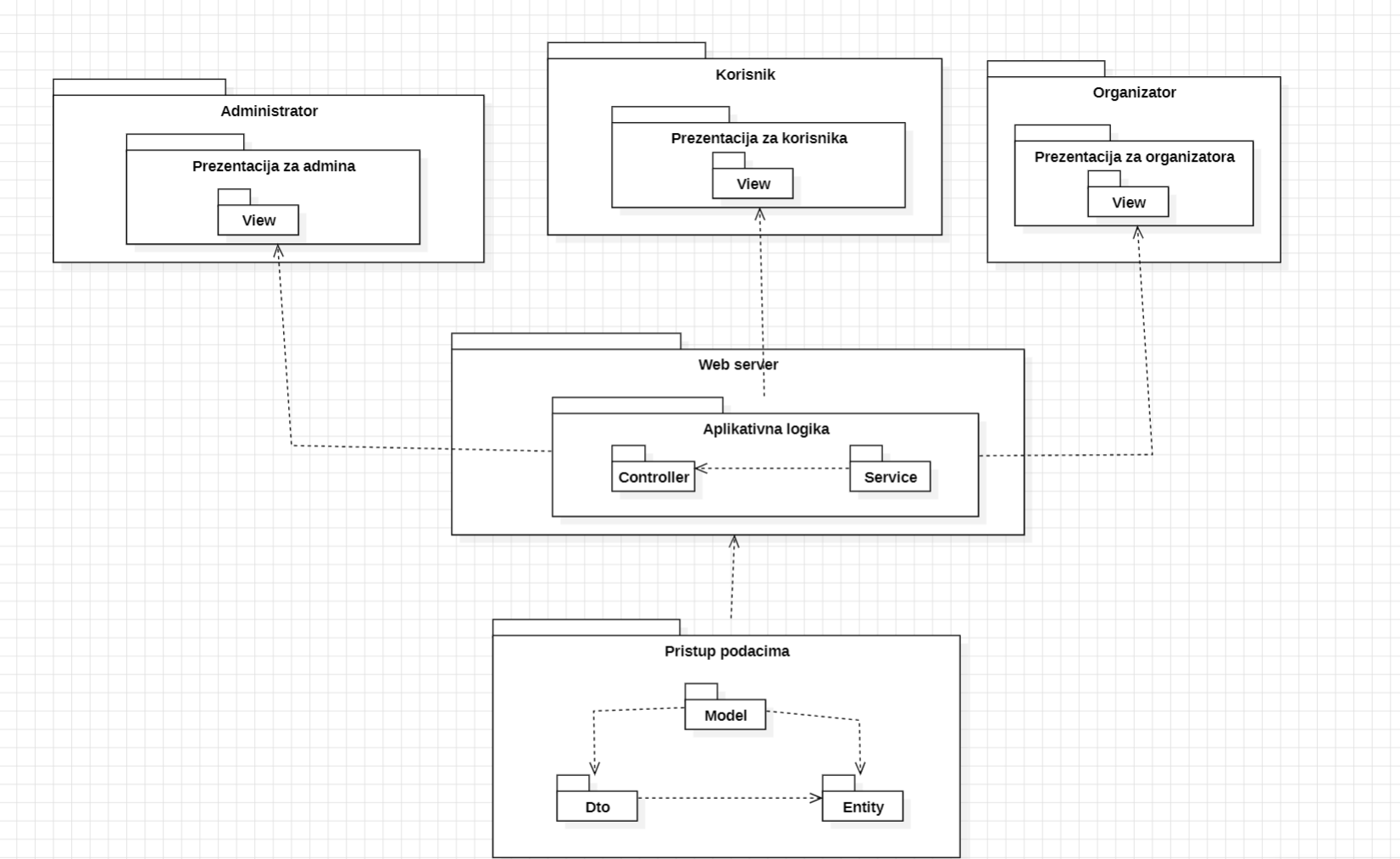
Klijent-server arhitektura je razvijena kao višenamjenska i modularna infrastruktura. Zasnovana je na slanju i primanju zahtjeva. Cilj ove arhitekture je fleksibilnost, unapređenje upotrebljivosti .



Slika 1 : Klijent – Server arhitektura

## Dekompozicija sistema

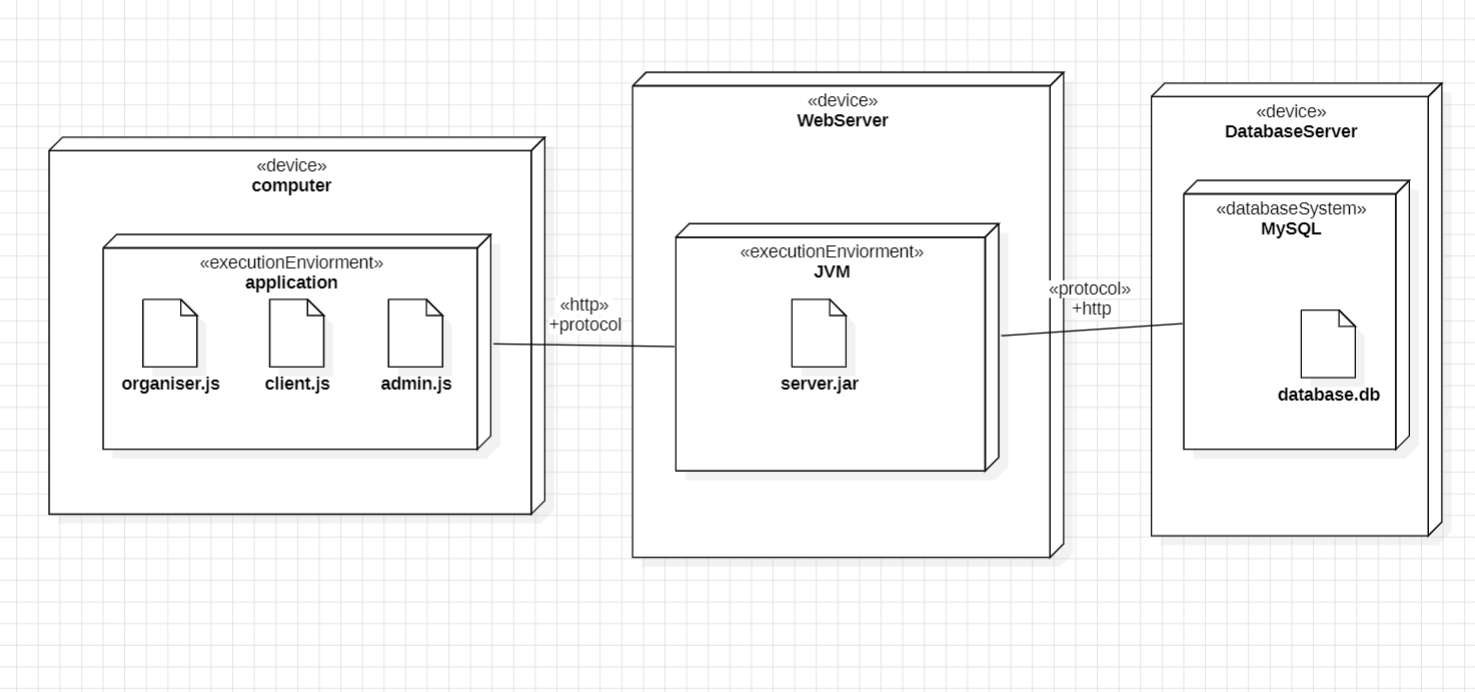
Dekompozicija sistema je postupak, kojim se sistem reprezentuje kao kolekcija podsistema. Na Slici 2 je prikazan UML dijagram paketa (Package Diagram), kojim je predstavljena dekompozicija sistema.



Slika 2 : Dekompozicija sistema (Package diagram)

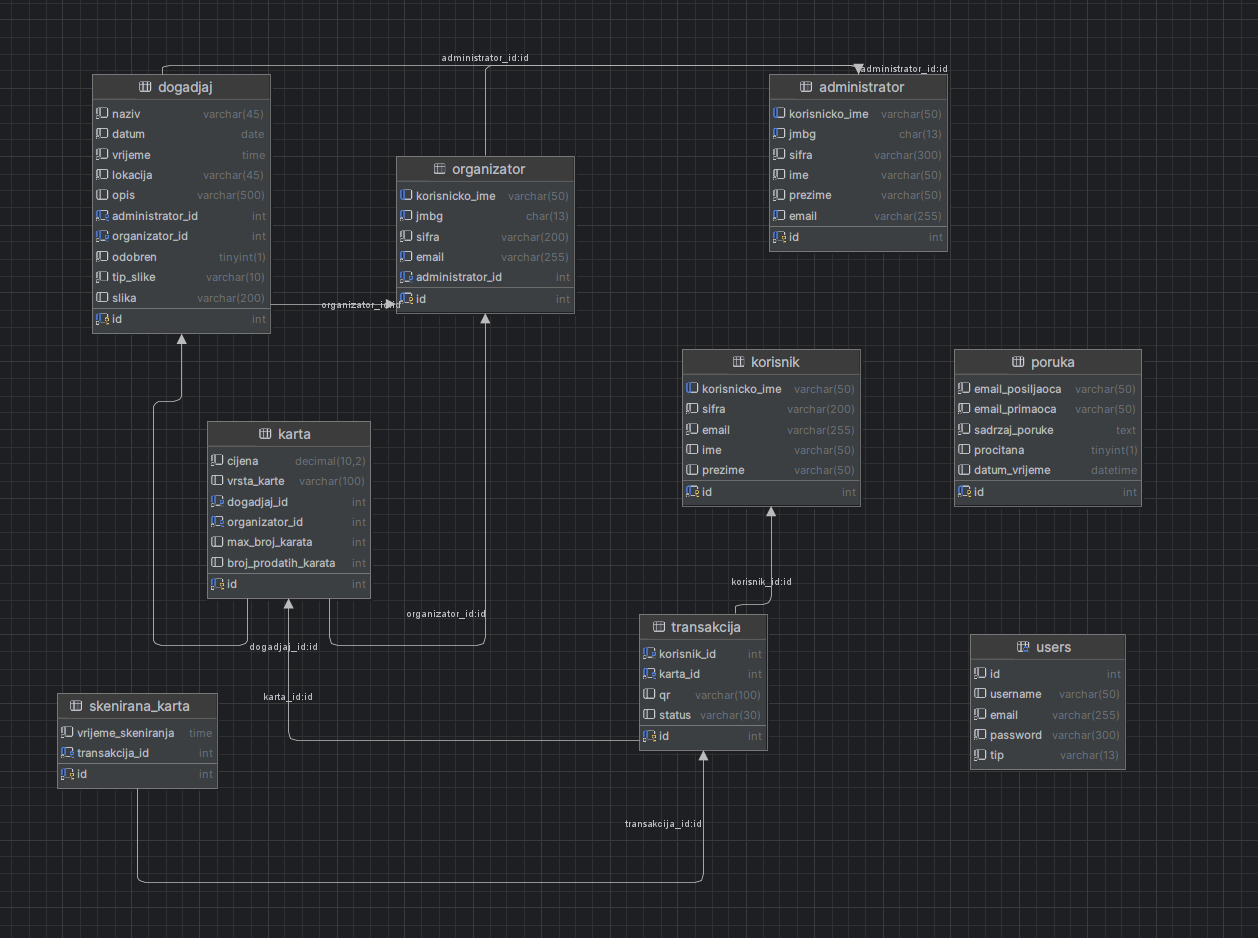
## Dijagram razmjestaja

Dijagram razmještaja(deployment diagram) je strukturni UML dijagram za modelovanje rasporeda komponenata u eksploataciji sistema (run time).



Slika: 3 Dijagram razmjestaja

## Perzistentni sloj



Slika 4: perzistentni sloj

## Prava pristupa

U sledećoj tabeli je prikazana kontrola prava pristupa korisnika sistema.

|  |  |
| --- | --- |
| Korisnik | * Kupovina karte * Login * Pregled licnih podataka * Listanje događaja * Pregled informacija o događaju |
| Organizator | * Kreiranje događaja * Kreiranje karata * Dodjeljivanje maksimalnog broja karata |
| Administrator | * Kreiranje organizatorskog naloga * Odobravanje događaja * Brisanje događaja * Pregled transakcija |

U tabeli nisu navedene sve operacije koje ucesnici mogu vrsiti nego one najbitnije. Ostale operacije su navedene u specifikaciji softverskih zahtjeva.

## Kontrola toka

Kontrola toka predstavlja redoslijed izvršavanja radnji odnosno metoda u objektno-orjentisanom sistemu. Podrazumijeva obezbjeđivanje izvršavanja operacija korisnika kad god je to potrebno. Zavisi od događaja, koje generiše korisnik.

Kontrola toka može biti centralizovana ili decentralizovana (distribuirana). Ovaj sistem ima centralizovanu kontrolu toka, koja ima event-driven mehanizam odnosno kontrola upravljanja događajima.

Izvršavanje akcija/naredbi može biti sekvencijalno (sljedeća akcija se izvršava tek po završetku prethodne) ili konkuretno (neke akcije se mogu izršavati istovremeno). Ovaj sistem omogućava korištenje više učesnika istovremeno. Njegov rad se zasniva na radu sa bazom podataka. Sistem za upravljanje bazom podataka(DBMS) omogućava konkurentan pristup, zahvaljujući modulu za upravljanje konkurentnim izvršavanjem, tako da je konkurentnost obezbjeđena na nivou DBMS-a.

Contents

[Uvod 2](#_Toc196135812)

[Namjena sistema 2](#_Toc196135813)

[Projektni ciljevi 2](#_Toc196135814)

[Definicije i skraćenice 2](#_Toc196135815)

[Referentni dokument 3](#_Toc196135816)

[Kratak pregled dokumenta 3](#_Toc196135817)

[Arhitektura sistema 3](#_Toc196135818)

[Dekompozicija sistema 4](#_Toc196135819)

[Dijagram razmjestaja 5](#_Toc196135820)

[Perzistentni sloj 6](#_Toc196135821)

[Prava pristupa 7](#_Toc196135822)

[Kontrola toka 7](#_Toc196135823)