# Arhitekturni dizajn softverskog sistema

# GeoTaxi

Branislav Jovičić 15660

Nikola Ristić 15854

Milica Martinović 15714

GeoTaxi predstavlja softverski sistem koji omogućava lakšu i bolju komunikaciju između taksista i njihovih klijenata. Treba da omogući pregled dostupnih taksi vozila preko Google mapa kao i ugovaranje vožnje.

# Arhitekturni zahtevi

## Funkcionalni zahtevi

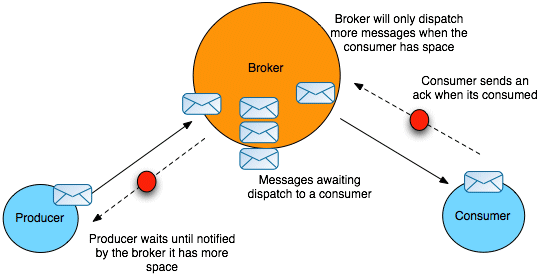
* Povezivanje klijenta i taksista preko serverske aplikacije
* Klijentu je omogućen pregled svih vozila (dostupnih i nedopstupnih), kao i slanje zahteva taksisti preko servera za novu vožnju.
* Server vodi računa o skladištenju podataka i o komunikaciji između taksista i krajnjih korisnika
* Klijent ima mogučnost ocene vozača koja se pamti na serveru (u bazi)
* Server pamti i ostale podatke o taksisti i njegovim vožnjama
* Logovanje taksiste na sistem

## Nefunkcionalni zahtevi i ograničenja

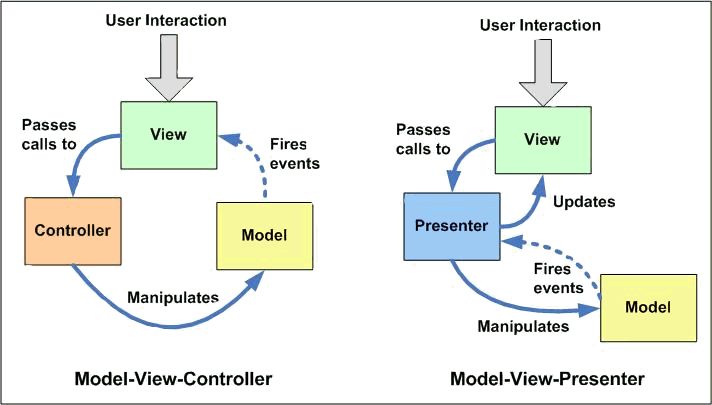
* *Proširljivost* 
  + Arhitektura aplikacije mora da bude takva da omogući lako dodavanje novih funkcionalnosti na bazi trenutne aplikacije
* *Reupotrebljivost*
  + Komponente aplikacije treba da budu projektovane tako da mogu lako da se iskoriste u budućim projektima slične funkcionalnosti.
* *Pouzdanost*
  + Ne sme doći do gubljenja poruka koje se koriste između klijenta i servera.
* *Performanse*
  + Razmena poruka između klijenta i servera treba da bude brza.
* *Lakoća korišćenja*
  + Korisnički interfejs treba da bude intuitivan i lak za korišćenje.
* *Lakoća održavanja*
  + Sistem treba da bude takav da se u budućnosti sa lakoćom mogu integrisati novi članovi tima u razvojni proces.
* *Dostupnost*
  + Sistem treba da bude dostupan za korišćenje 24/7.
* *Skalabilnost*
  + Sistem treba da podrži veći broj korisnika.

# Arhitekturni obrasci

1. Klijent-server (broker)



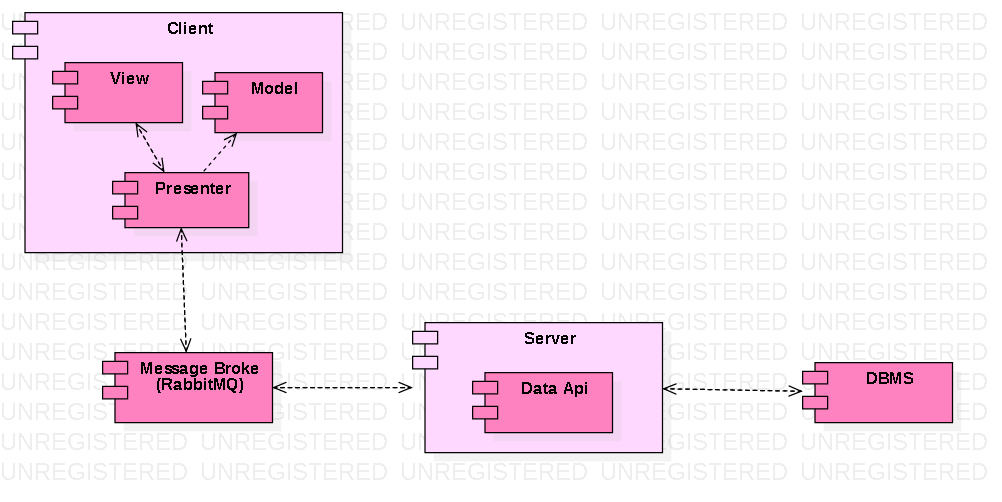
1. MVP (Model View Presenter)



# Arhitekturni dizajn

## Strukturni pogled

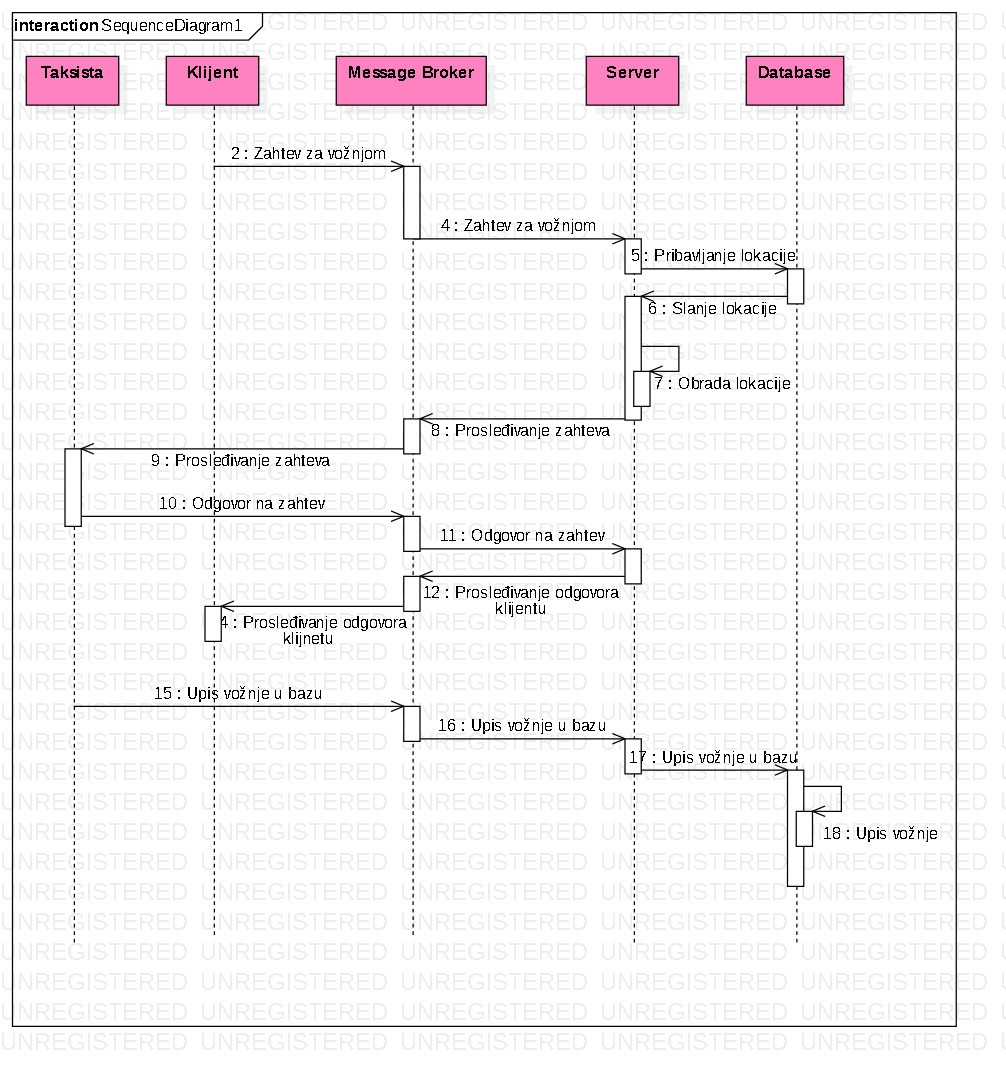
Prikazuje strukturu sistema navodeći komponente sistema. Klijent koristi MVP arhitekturni obrazac.Klijent komunicira sa serverom preko Presentera koristeći Message Broker. Serverska aplikacija predstavlja posrednika između baze podataka i ostatka sistema.



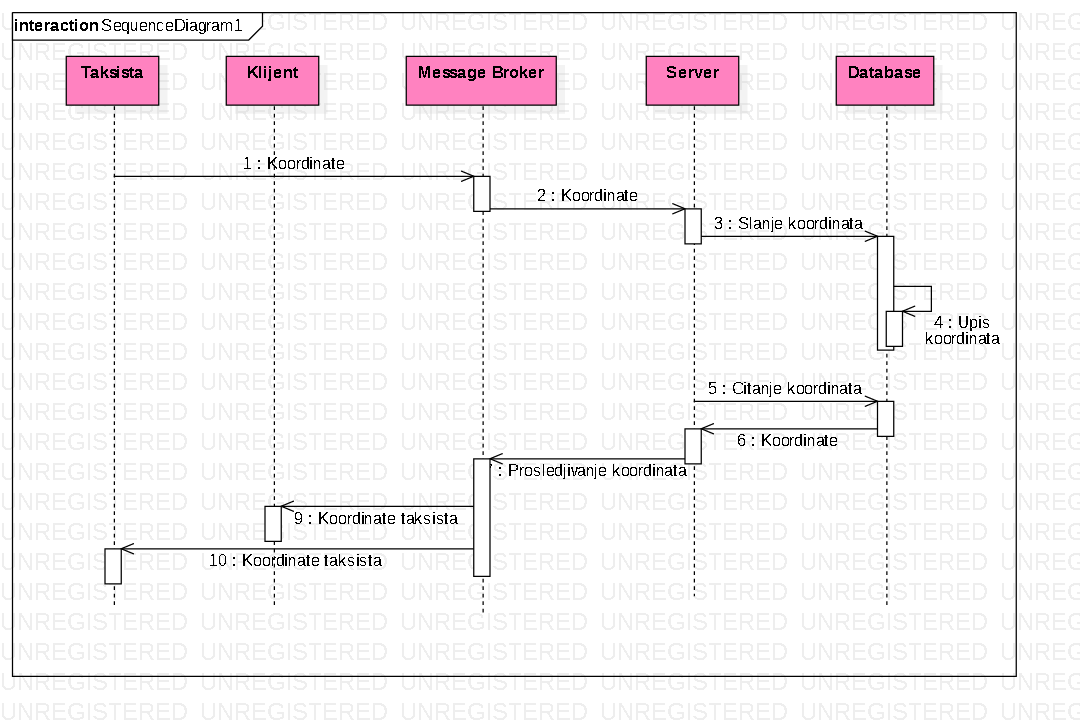
## Bihevioralni pogled

Bihevioralni pogled opisuje interakciju između komponenti sistema.

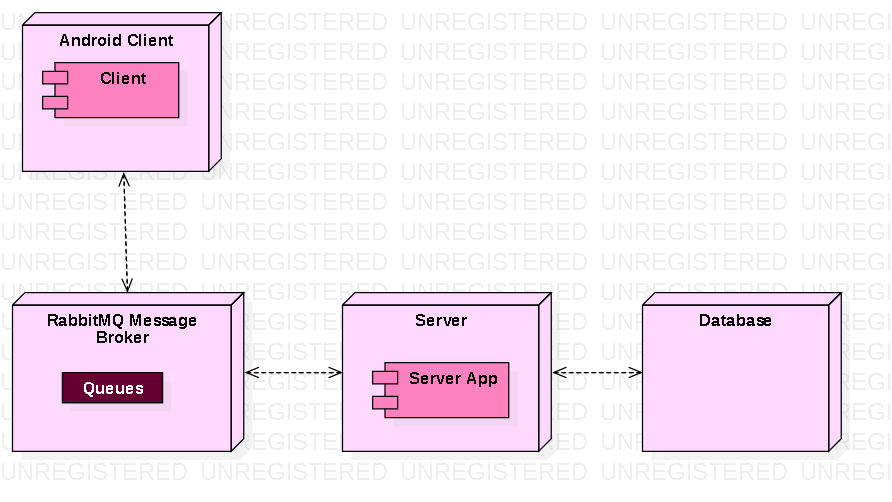
Opsti pregled interkacije kompontenti:



Pregled interakcije komponenti pri upisu lokacije taksiste u bazu podataka:



## Alokacioni pogled



# Frameworks and libraries

* **Android** **Studio** **Framework** – za razvoj klijentske aplikacije
* **Node.js** – za razvoj serverske aplikacije
* **MySql**– server baze podataka
* **Sequelize** – ORM za node.js
* **AMQP-client** – biblioteka za povezivanje sa RabbitMQ brokerom iz Androida
* **RabbitMQ** message broker
* **amqp.node** – biblioteka za povezivanje sa RabbitMQ brokerom iz node.js-a