****

**Projektovanje informacionih sistema**

**Projektni zadatak**

**Tema: Lokalna Televizija**

**Profesor: Student:**

**Prof. dr. Vladislav Miškovic Aleksandar Bogdanović  
 210090/2018**

# Uvod

Sa razvojem tehnologije u 21. vijeku gotovo je nemoguće zamisliti dom bez televizije. Kako stara, tako i mlađa populacija se informiše i zabavlja ispred takozvanih malih ekrana. Svaki od tih ekrana nudi raznolike programe kako za zabavu, sport, vijesti i ostale.

## Cilj razvoja

Cilj razvoja aplikacije je u dobijanju što više registrovanih korisnika i za uzvrat pružanje im usluga u vidu informisanja i zabave.

## Obim sistema

Funkcije sistema, odnosno šta sve aplikacija može da radi?

Funkcija sistema TV aplikacije je informisanje o ranijim događajima korisnika.   
Aplikacija u sebi sadrži sve osnovne informacije za korisnika svrtstane u kategorije.

## Rečnik

Izdvojiti sve reči i skraćenice koje nisu poznate ljudima koji nisu iz struke. Npr.

**IS** – *Informacioni sistem* je integrisani skup komponenti za sakupljanje, snimanje, čuvanje, obradu i prenošenje informacija.

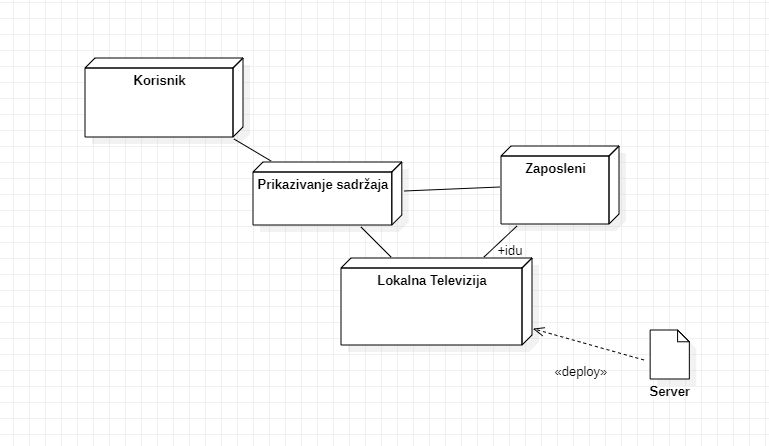
**Server** – *Server* je računarski sistem koji pruža usluge drugim operativnim računarskim sistemima – klijentima.

**Switch** – *Switch* je uređaj koji upravlja protokom podataka između delova lokalne mreže (LAN).

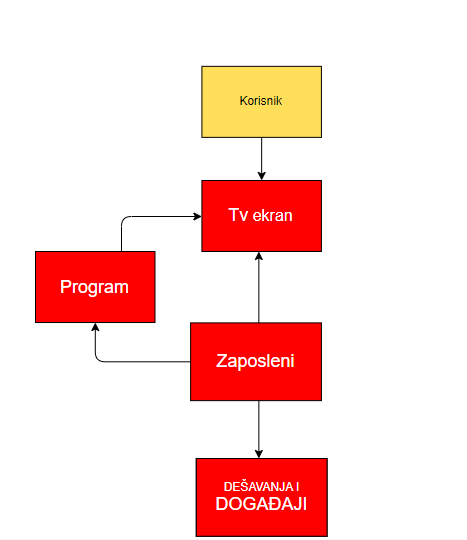
## Reference

<https://www.rtvbn.com/>

# Dijagram isporuke (Deployment Diagram)

Korisnik može da pristupi samo emitovanju sadržaja, dok zaposleni mogu da uređuju, pišu i objavljuju taj isti sadržaj. 

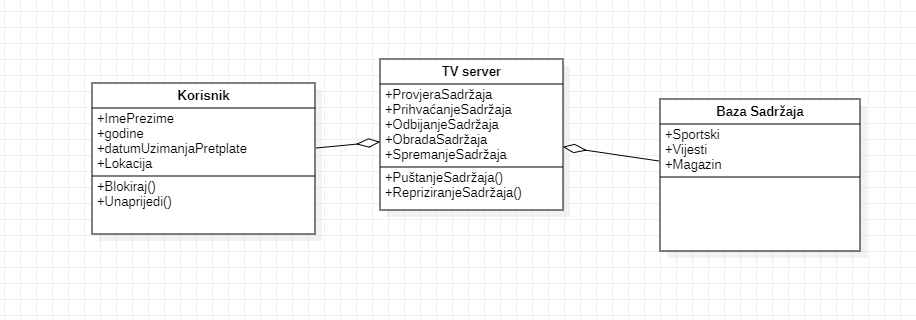
# Projekt arhitekture sistema (Architectural Design)



Korisnik dok gleda određeni proram, zaposleni su nekoliko dana prije spremili program/emisiju/vijest. Zaposleni su prenijeli dešavanja i događaje tako ih i obradili i pustili u etar.

# Projekt strukture podataka (Data Structure Design)

Korisnik sa svim svojim atributima pristupa dole navedenom TV serveru (tv programu). TV Server obrađuje sadržaj i sprema ga na duži vremenski period. Tv Server može da više puta ponovi emitovanje istog sadržaja, koji uzima iz njegove baze.



# Projekt komponenti sistema (Use Case Realizations)

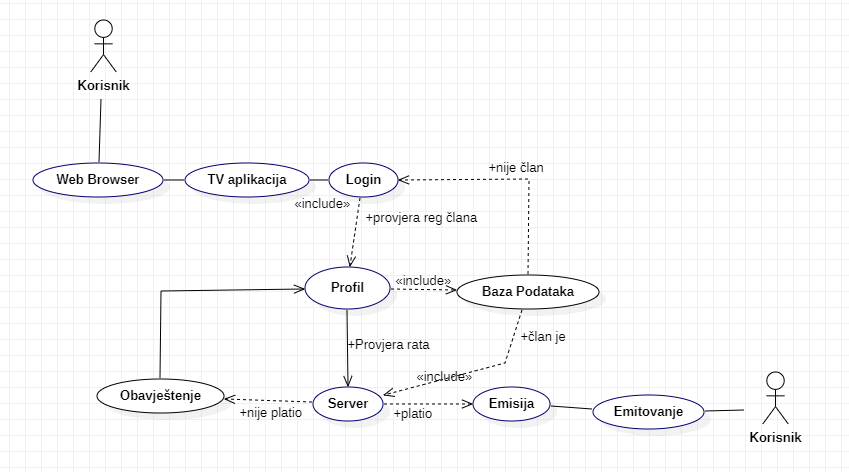
Dodati kompletni UML **Use Case dijagram** koji uključuje sve slučajeve korištenja i sve aktore. Pored toga, u nastavku navesti svaki slučaj korištenja posebno.

Svaki slučaj korištenja treba da sadrži:

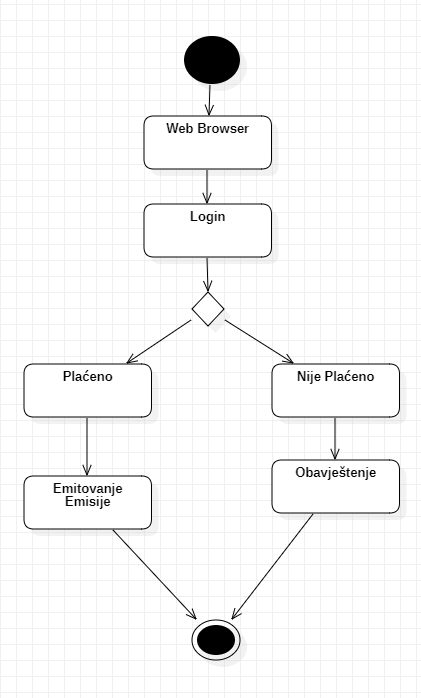
* **Scenario** – naziv slučaja korištenja i po stavkama navesti kako se odvija tok slučaja korištenja,
* **Dijagram aktivnosti** slučaja korištenja,
* **Dijagram sekvenci** slučaja korištenja.

**Napomena\*** Svaki projektni zadatak treba da sadrži minimalno jedan a maksimalno tri slučaja korištenja.

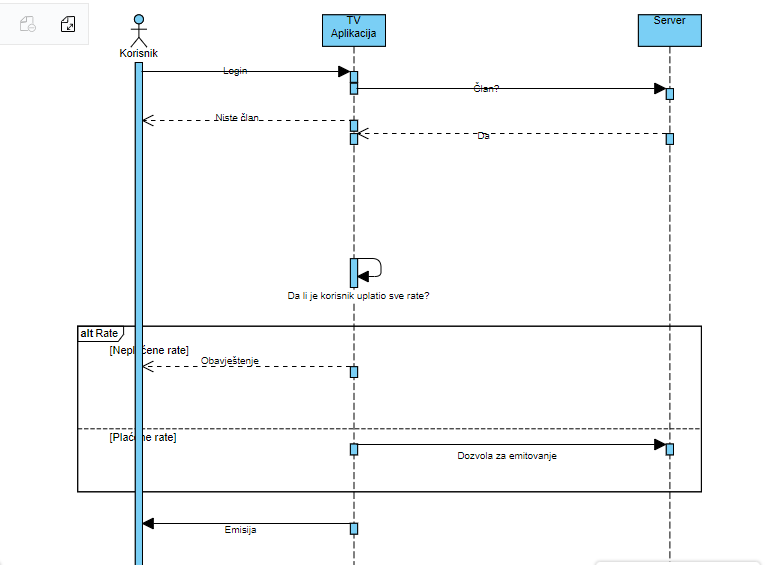
**Scenario**: Korisnik je propustio svoju omiljenu emisiju nedjeljom, ulazi na Web Browser i pronalazi našu televijsku aplikaciju da bi pogledao istu. Takođe, korisnik da bi pogledao aplikaciju mora da bude registrovan član u našoj bazi podataka i da je ispunio sve redovne uslove plaćanja. Nakon što server provjeri da je korisnik sve uredno uplatio daje mu mogućnost da pogleda propuštenu emisiju. Korisnik koji nije uplatio poslednji mjesec dana, iskače mu prozor sa obavještenjem i odbijen je za gledanje propuštene emisije.



*Use Case Diagram*

**

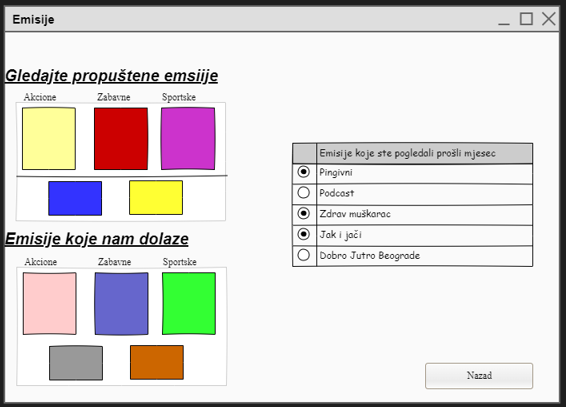
*Activity Diagram*

**

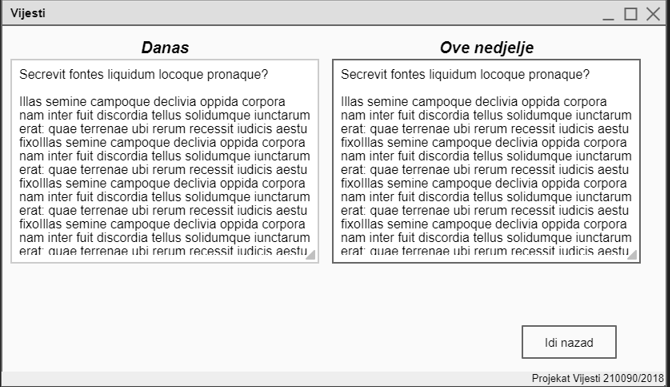
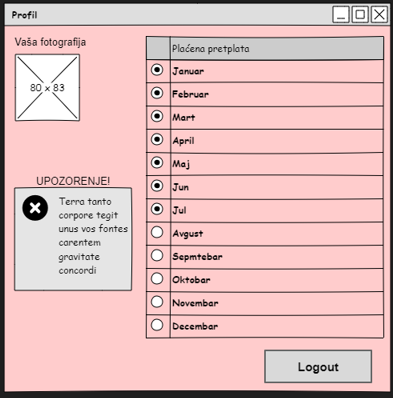
*Sequence Diagram*

# Projekt korisničkog interfejsa (User Interface Design)

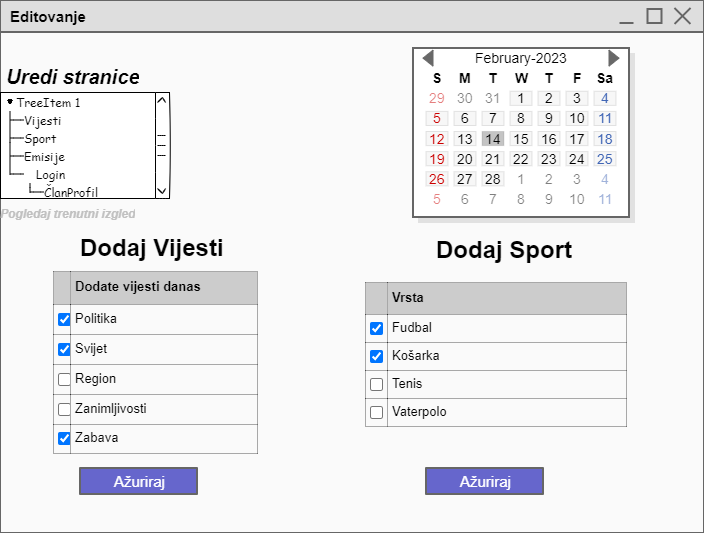
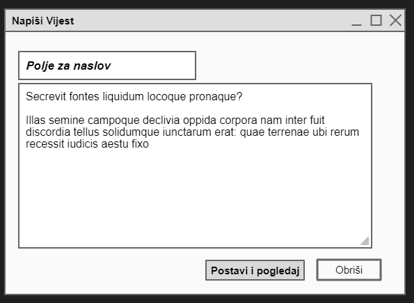
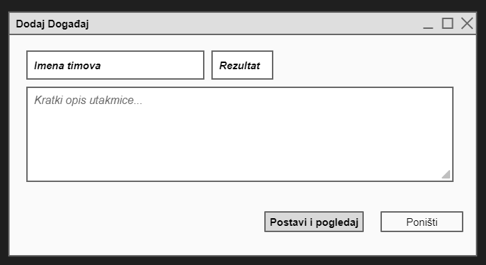
***Kako korisnici vide web stranicu.***





**

***Kako zaposleni vide web aplikaciju***

**

# Ostali elementi sistema

Infromacioni sistem zahtjeva konstanto ažuriranje i didavanje novih informacija u cilju redovnog informisanja korisnika koji redovno posjećuju sajt. Ovaj *IS* mora da obezbjedi tačnosts, raspoloživost i dostupnost u svakom momentu.

**Prilog: *Izvorni kod softverskog sistema***

***Generisan kod klase server***

import java.util.\*;

/\*\*

 \*

 \*/

public class TV server {

    /\*\*

     \* Default constructor

     \*/

    public TV server() {

    }

    /\*\*

     \*

     \*/

    public void ProvjeraSadržaja;

    /\*\*

     \*

     \*/

    public void PrihvaćanjeSadržaja;

    /\*\*

     \*

     \*/

    public void OdbijanjeSadržaja;

    /\*\*

     \*

     \*/

    public void ObradaSadržaja;

    /\*\*

     \*

     \*/

    public void SpremanjeSadržaja;

    /\*\*

     \*

     \*/

    public void PuštanjeSadržaja() {

        // TODO implement here

    }

    /\*\*

     \*

     \*/

    public void RepriziranjeSadržaja() {

        // TODO implement here

    }

}

Generisan kod klase Korisnik

import java.util.\*;

/\*\*

 \*

 \*/

public class Korisnik {

    /\*\*

     \* Default constructor

     \*/

    public Korisnik() {

    }

    /\*\*

     \*

     \*/

    public void ImePrezime;

    /\*\*

     \*

     \*/

    public void godine;

    /\*\*

     \*

     \*/

    public void datumUzimanjaPretplate;

    /\*\*

     \*

     \*/

    public void Lokacija;

    /\*\*

     \*

     \*/

    public void Blokiraj() {

        // TODO implement here

    }

    /\*\*

     \*

     \*/

    public void Unaprijedi() {

        // TODO implement here

    }

}

Generisan kod Baze Sadržaja

import java.util.\*;

/\*\*

 \*

 \*/

public class Baza Sadržaja {

    /\*\*

     \* Default constructor

     \*/

    public Baza Sadržaja() {

    }

    /\*\*

     \*

     \*/

    public void Sportski;

    /\*\*

     \*

     \*/

    public void Vijesti;

    /\*\*

     \*

     \*/

    public void Magazin;

    /\*\*

     \*

     \*/

    public void Operation1() {

        // TODO implement here

    }

}