



Oblikovanje programske potpore

Završni ispit

27. siječanj 2017.



JMBAG

Ime i prezime

Vlastoručni potpis

Minimum za prolaz završnog ispita je 12 bodova, maksimum 36 bodova

GRUPA A

1. **(1 bod)** Navedite na koje se sve načine mogu izraziti zahtjevi sustava.

- Rj. strukturiranim prirodnim jezikom, specijalnim jezikom za opis oblikovanja (npr. SDL), grafičkom notacijom (npr. UML) i matematičkom specifikacijom (FSM, teorija skupova, logika).

2. **(1 bod)** Navedite generičke aktivnosti inženjerstva zahtjeva.

- Rj. studija izvedivosti, izlučivanje zahtjeva, analiza i specifikacija zahtjeva, validacija zahtjeva, upravljanje zahtjevima

3. **(1 bod)** Navedite barem dvije značajke metodologije ubrzanog razvoja (engl. *agile methodology*) programske potpore.

- Rj. Iterativni razvoj, mali inkrementi, kontinuirano poboljšanje PP, naglasak na ljude i suradnju, uključenost korisnika u proces razvoja...

4. **(1 bod)** Koja je temeljna značajka vodopadnog modela (engl. *Waterfall model*) razvoja programske potpore?

- Rj. Prethodna faza treba se završiti prije prelaska na novu fazu.

5. **(1 bod)** Navedite na koje ste sve načine u projektnoj dokumentaciji izrazili korisničke zahtjeve?

- UCovi s pripadajućim dijagramima (funkcionalni zahtjevi), sekv. dijagrami, lista nefunkcionalnih zahtjeva

6. **(2 boda)** Razredi Lik2D te razredi Kvadrat i Pravokutnik, koji ga nasljeđuju, opisani su sljedećim kodom:

```
/**
 * Razred Lik2D
 */
public abstract class Lik2D {

    public abstract int Povrsina (int parametri[] );

    public void IdentificirajSe()
    {
        Ispis("Ja sam Lik 2D"); //metoda za ispis na konzolu
    }
}

/**
 * Razred Kvadrat nasljedjuje Lik2D
 */
```

```

public class Kvadrat extends Lik2D{

    public int Povrsina(int parametri[]){
        return parametri[0]*parametri[0];
    }

    public void IdentificirajSe(){
        Ispis ("Ja sam Kvadrat");
    }
}

/**
 * Razred Pravokutnik nasljedjuje Lik2D
 */
public class Pravokutnik extends Lik2D{

    public int Povrsina (int parametri[]){
        return parametri[0]*parametri[1];
    }
}

```

a) (1 bod) Obrazložite ispravnost/neispravnost sljedećeg retka u metodi `main()`:

```
Kvadrat k = new Lik2D();
```

- Rj. Nije ispravan, budući da se ne može instancirati primjerak apstraktnog razreda Lik2D.

b) (1 bod) Što će se ispisati u konzoli prilikom izvođenja metode `main()` čiji se kod nalazi u nastavku zadatka?

```

public static void main(String[] args) {

    Lik2D k = new Kvadrat();
    Lik2D p = new Pravokutnik();

    int []parametri_k = new int[1];
    int []parametri_p = new int[2];

    parametri_k[0]=5;
    parametri_p[0]=2;
    parametri_p[1]=3;

    Ispis(k.Povrsina(parametri_k));
    Ispis(p.Povrsina(parametri_p));

    k.IdentificirajSe();
    p.IdentificirajSe();
}

```

- Rj.
- 25
- 6
- Ja sam Kvadrat
- Ja sam Lik2D

7. (1 bod) Prema donjem dijagramu razreda, odgovorite: koliko najviše putnika može svako dizalo podići te mora li putnik uopće putovati dizalom?

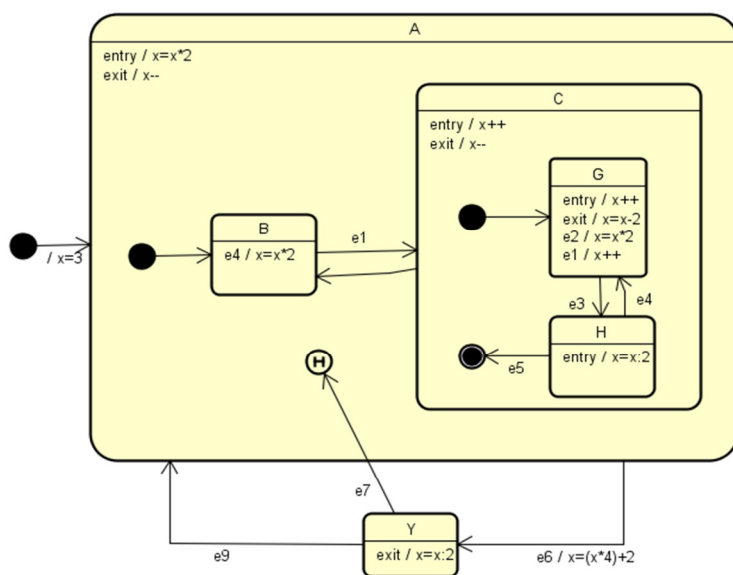


- Rj. Svako dizalo može podići najviše 8 putnika, a putnik ne mora putovati dizalom.

8. **(1 bod)** Navedite sve temeljne elemente sekvencijskog dijagrama i označite ih na skiciranom hipotetskom primjeru.

Rj. Aktori (ili objekti), životne linije, aktivnosti, poruke (može i petlje, grananja, sinkrone/asinkrone/create/destroy poruke)
<<neka skica ovdje>>...

9. **(2 boda)** Za zadani UML-dijagram stanja odredite vrijednost varijable x po završetku izvođenja događaja (ulazna vrijednost je $x = 3$), ako je redoslijed događaja koji se izvode: **e1, e3, e6, e7**.



Rj. $x =$ (izmjene redom): 6, 7, 8, 6, 3, 2, 1, 6, 3, 6, 7, 8 (**konačno**).

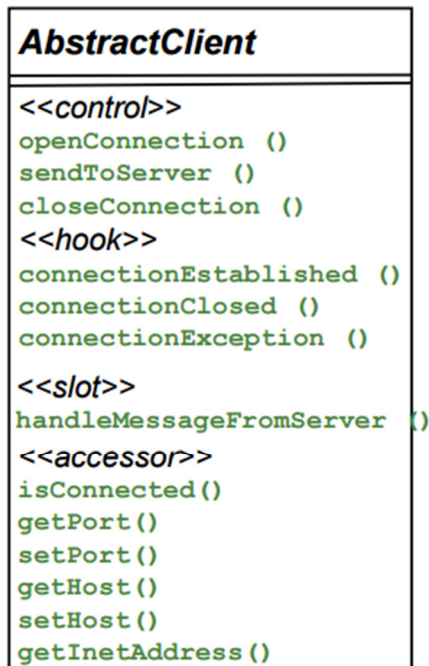
10. **(1 bod)** Kako je organiziran primjenski program u arhitekturi zasnovanoj na uslugama?

- Rj. Uslužno usmjerena arhitektura organizira primjenski program (cjelovitu aplikaciju) kao kolekciju usluga koje međusobno komuniciraju uporabom dobro definiranih sučelja.

11. **(1 bod)** Koji su minimalni elementi kojima se opisuju oblikovni obrasci (engl. *design pattern*)? Naputak: rješenje prikazite tablično.

Naziv	Razumljivo ime
Opis	Opis problema
Rješenje	Opis predložka koji može biti upotrijebljen na različite načine
Posljedice	Rezultati i pogodnosti primjene obrasca uporabe

12. **(1 bod)** Navedite kojoj skupini (ili kojim sve skupinama) označenom stereotipom pripadaju metode u razredu AbstractClient arhitekture OCSF koje se ne mogu redefinirati u podrazredima.



- Rješenje: `<<control>>` i `<<accessor>>`

13. **(1 bod)** Za slučaj arhitekture klijent-poslužitelj, kada postoji n spojenih klijenata, izračunajte minimalan broj dretvi pri radu poslužitelja implementiranog objektno usmjerenim radnim okvirom OCSF (NAPOMENA: zanemarite administracijske dretve, dretve OS-a, VM,...).

Rješenje: $n+1$

14. **(1 bod)** Program koji kao ulaze prima tri dvoznamenkasta cijela broja unesena tipkovnicom potrebno je ispitati primjenom tehnike kombinacijskog ispitivanja (engl. *Combination testing*). Izračunajte broj potrebnih ispitnih slučajeva.

- Rješenje: ulazne vrijednosti -99..99
- $199 \cdot 199 \cdot 199 = 7\,880\,599$
- bilo bi izvrsno da studenti prokomentiraju i neispravne vrijednosti ulaza.

15. **(1 bod)** Objasnite postupak inkrementalnog integracijskog ispitivanja odozgo na dolje (engl. *Top down integration*)

- Razviti kostur sustava i postepeno ga popuniti komponentama, uz inkrementalno ispitivanje
- Nije potrebno razvijati upravljačke programe, ali treba razvijati prividne komponente

16.(2 boda) Za sljedeću funkciju:

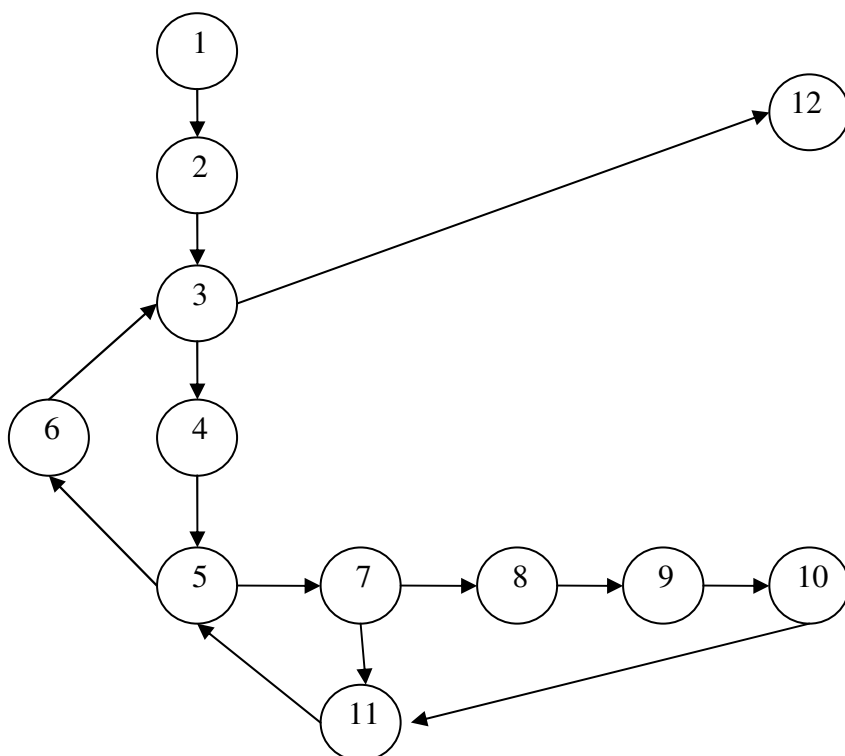
- (1 bod) nacrtajte graf tijeka programa,
- (1 bod) odredite njezinu ciklomatsku složenost. Navedite formulu za izračun.

```
public void bubbleSort(int [] array) {  
    int c, d, swap, n;  
    n = array.length;  
    for (c = 0 ; c < ( n - 1 ); c++)  
    {  
        for (d = 0 ; d < n - c - 1; d++)  
        {  
            if (array[d] > array[d+1])  
            {  
                swap      = array[d];  
                array[d]   = array[d+1];  
                array[d+1] = swap;  
            }  
        }  
    }  
    return;  
}
```

Rj.

a)

```
int c, n, swap;  
n = array.length; (1)  
for (c = 0 (2); c < ( n - 1 ) (3); c++ (6))  
{  
    for (d = 0 (4); d < n - c - 1 (5); d++ (11))  
    {  
        if (array[d] > array[d+1]) (7)  
        {  
            swap      = array[d]; (8)  
            array[d]   = array[d+1]; (9)  
            array[d+1] = swap; (10)  
        }  
    }  
}  
return; (12)
```



b) $CV(G) = 14 - 12 + 2 = 4$

17. **(1 bod)** U timskom razvoju programske potpore pojedine dijelove istodobno razvija više članova tima. Tim koristi GitLab sustav za upravljanje inačicama datoteka (engl. *version control*, *revision control*, *source control*). Gdje su smješteni podaci izvornog koda koje mijenjaju članovi tima tijekom programiranja (obrazložite odgovor)? Da li je moguć istodoban rad više članova tima na izvornom kodu?

- Rj. u lokalnim repozitorijima kod članova tima, moguć je istodoban rad.

18. **(1 bod)** Definirajte potrebne predikate i konstante te preslikajte rečenicu u dobro definiranu formulu predikatne logike prvoga reda:

"Dora proučava samo jednu veliku knjigu."

Rj.

$VK(x)$ = x je velika knjiga

$P(x,y)$ = x proučava y

$=(x,y)$ = x je jednako y

Dora = konst.

$\exists x (VK(x) \wedge P(Dora,x) \wedge \neg \exists y (VK(y) \wedge P(Dora,y) \wedge \neg =(x,y)))$

alternativno:

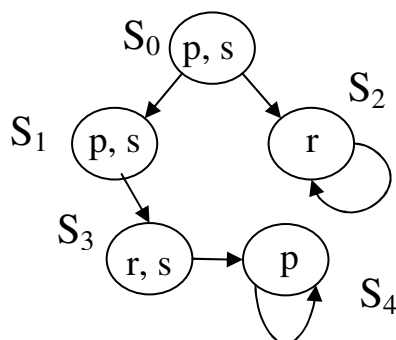
$\exists x (VK(x) \wedge P(Dora,x) \wedge \forall y ((VK(y) \wedge P(Dora,y)) \Rightarrow =(x,y)))$

19. **(1 bod)** Što mora biti zadovoljeno za interpretacije skupa formula G, tako da možemo tvrditi da vrijedi $G \models P$ tj. da je formula P logička posljedica skupa G?

Rj. Svaka interpretacija koja za skup G daje istinitost mora i za P dati istinitost.

20. **(2 boda)** Za zadani model implementacije Kripke strukturom M prema slici potrebno je:

- Odrediti S (skup stanja), R (relaciju prijelaza), L (funkciju označavanja).
- Odrediti sva stanja koja zadovoljavaju formulu \mathbf{A} (s U p).



Rj.

a) $S = \{S_0, S_1, S_2, S_3, S_4\}$

$R = \{(S_0, S_1), (S_0, S_2), (S_2, S_2), (S_1, S_3), (S_3, S_4), (S_4, S_4)\}$.

$L(S_0) = \{p, s\}$; $L(S_1) = \{p, s\}$; $L(S_2) = \{r\}$; $L(S_3) = \{r, s\}$; $L(S_4) = \{p\}$

b) S_0 - da, S_1 - da, S_2 - ne, S_3 - da, S_4 - da.

Problemski zadatak: sustav za praćenje kvalitete zraka

Korisnici pristupaju web aplikaciji sustava za praćenje kvalitete zraka putem web preglednika. Web aplikacija, sakupljač mjerenja i analizator mjerenja nalaze se na istom računalu poslužitelju. Sakupljač mjerenja prikuplja mjerenja s više udaljenih osjetnika onečišćivača putem TCP/IP protokola. Postoje tri vrste osjetnika onečišćivača: dušikovog dioksida (NO_2), ugljikovog monoksida (CO) i ozona (O_3). Nakon što prikupi mjerenja, sakupljač ih pohranjuje u bazu podataka na udaljenom računalu putem HTTPS protokola. Analizator mjerenja preuzima podatke od sakupljača, obrađuje ih te rezultate obrade šalje web aplikaciji. Web aplikacija čita podatke o mjeranjima iz baze podataka na udaljenom računalu. Komunikacija između korisničkog računala i web aplikacije odvija se HTTPS protokolom.

Web aplikaciji mogu pristupiti dvije vrste korisnika: djelatnici sustava ili klijenti. Djelatnici sustava prijavljuju se s glavne stranice. Nakon prijave, djelatnici se preusmjeravaju na stranicu aktualnih mjerenja. Djelatnici mogu pristupiti stranici oglašavanja upozorenja. Po prikazu stranice za oglašavanje upozorenja, djelatnik može ručno napisati novu obavijest upozorenja i oglasiti upozorenje. Ako upozorenje već postoji, na ovoj stranici djelatnik ga može i ukinuti. Nakon oglašavanja ili ukidanja upozorenja, paralelno se provodi obavještavanje sakupljača mjerenja i evidentiranje promjena u udaljenoj bazi podataka. Nakon provođenja tih aktivnosti, djelatniku se prikazuje stranica s aktualnim upozorenjima, a koju vide i klijenti.

Sustav u redovitom radu prikuplja mjerenja svakih sat vremena. Prekoračenjem zadanih vrijednosti mjerenja sustav prelazi u izvanredni rad pri čemu se povećava učestalost očitavanja mjerenja. U izvanrednom radu razlikujemo dvije situacije:

- jedan onečišćivač prelazi zadane vrijednosti te se u pojačanom radu mjerenja prikupljaju svakih 15 minuta;
- više onečišćivača prelazi zadane vrijednosti te se u kritičnom radu mjerenja prikupljaju svaku minutu.

Sustav automatski mijenja stanje temeljem informacija analizatora o razini dozvoljenog onečišćenja. Pri ulasku u izvanredni rad, web aplikacija evidentira vrijeme ulaska i razine svih onečišćivača u dnevnik pohranjen u udaljenoj bazi podataka, dok pri izlasku evidentira vrijeme izlaska. U izvanrednom radu, svim korisnicima se na vrhu stranice s aktualnim upozorenjima ispisuje poruka: "Prekoračenje dozvoljene razine onečišćivača", dok se na stranici s aktualnim mjeranjima sva prekoračenja označuju crvenom bojom. U slučaju kvara poslužitelja, pri oporavku, sustav se automatski vraća u isti rad u kojem se dogodila pogreška.

21. (4 boda) Dijagramom stanja modelirajte načine rada sustava za praćenje kvalitete zraka.

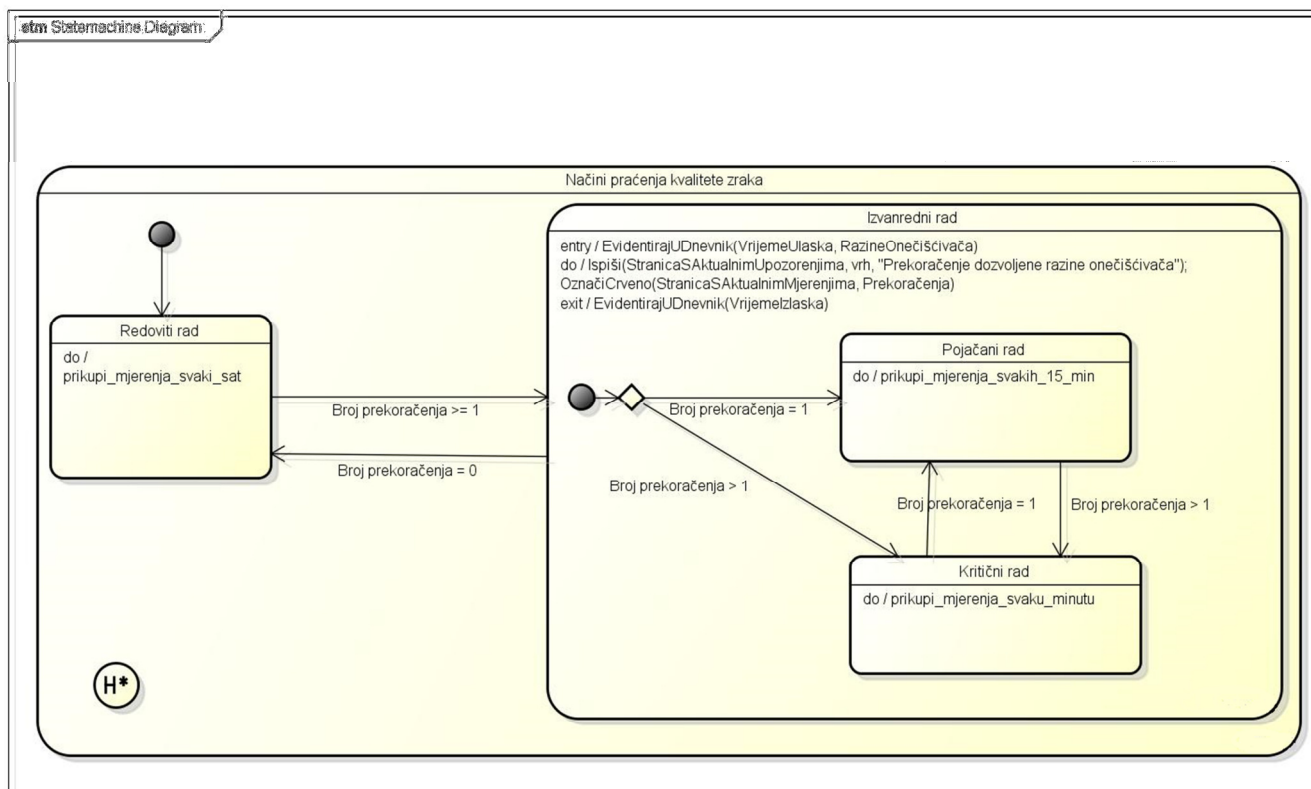
22. (4 boda) Dijagramom aktivnosti modelirajte aktivnosti oglašavanja i provedbe upozorenja, uključivši prijavu djelatnika u sustav.

23. (4 boda) Dijagramom razmještaja modelirajte cjelokupni sustav za praćenje kvalitete zraka na razini instanci. Potrebne nazive čvorova i komponenti odaberite proizvoljno.

Napomena za sva tri dijagrama: moguće pogreške nije potrebno modelirati.

Rješenja:

20. Dijagram stanja



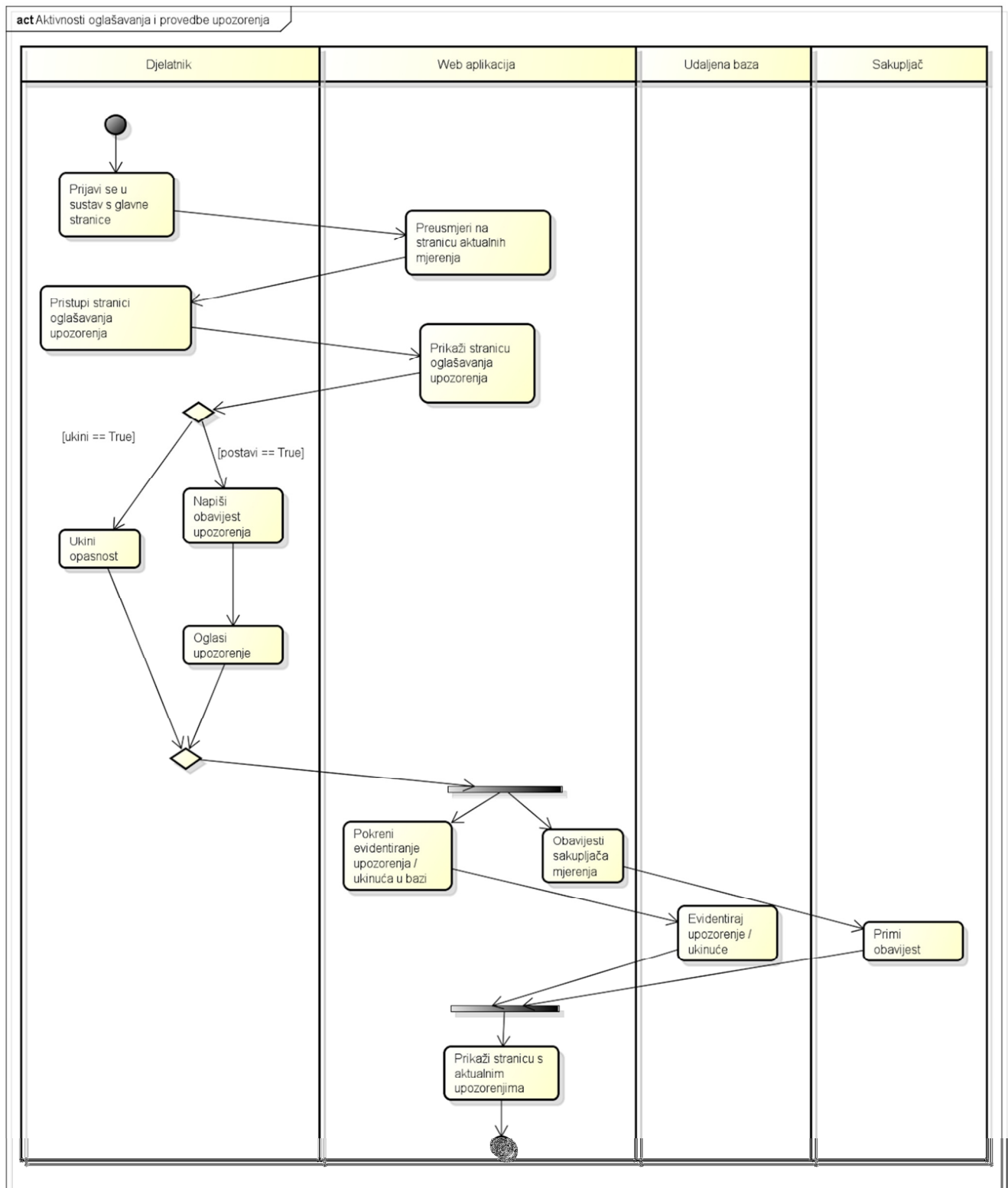
Kriteriji bodovanja:

- 1.5 bod za točno uočavanje i prikaz stanja i pseudostanja
- 0.5 bod za točan unos svih "do" aktivnosti
- 0.5 bod za "entry" i "exit" akcije
- 1 bod za točan prikaz događaja (okidača)
- 0.5 boda za uvođenje oznake H*

Napomene:

- osim romb (izbor) pseudostanja, priznavat će se rješenje i s dva prijelaza koja počinju iz početnog pseudostanja u Izvanrednom radu
- priznaju se razne varijacije prikaza "do" aktivnosti te akcija "entry" i "exit", dokle ispravno pokrivaju tekst zadatka

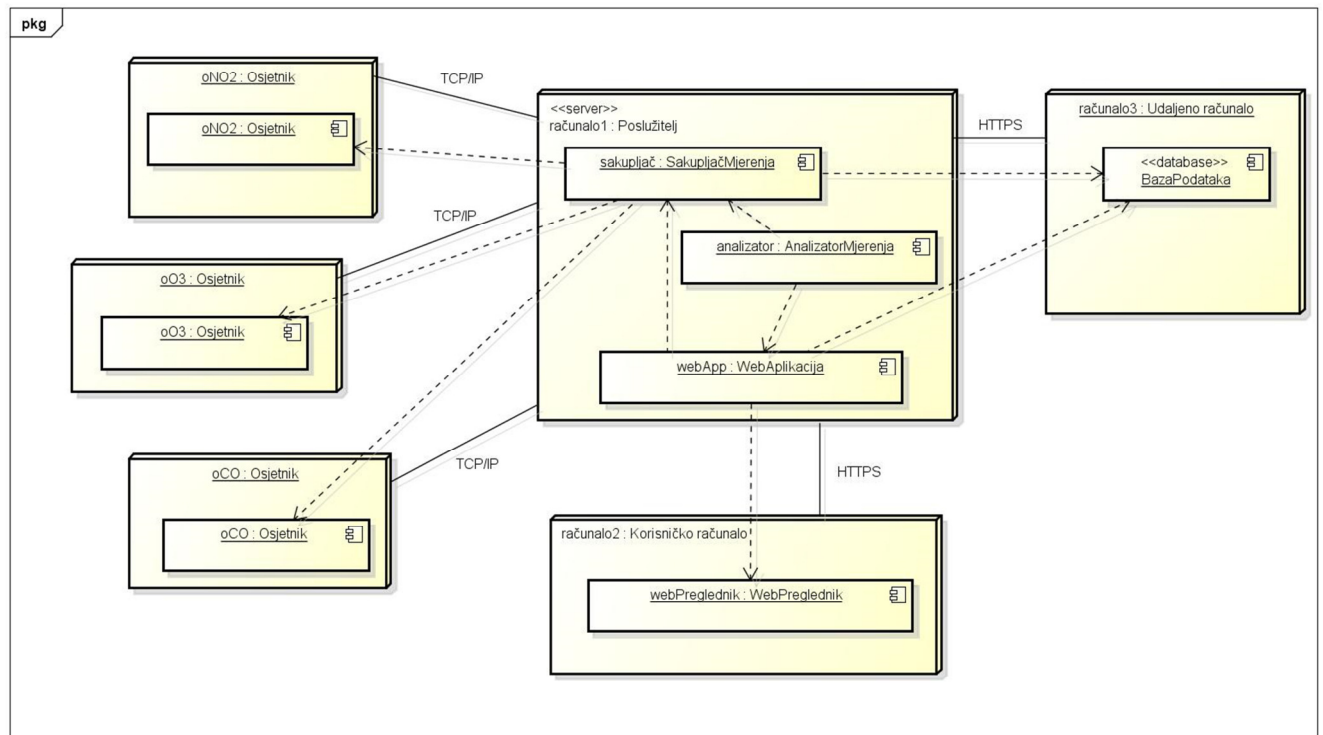
21. Dijagram aktivnosti



Kriteriji bodovanja:

- 1 bod za ispravnu podjelu plivačkih staza
- 1 bod za uspješno uočavanje redoslijeda aktivnosti
- 1 bod za prikaz grananja na stranici oglašavanja upozorenja, ovisno o želji djelatnika
- 1 bod za točan prikaz paralelnih aktivnosti vezanih uz evidentiranje i obavješćivanje

22. Dijagram razmješataja



Kriteriji bodovanja:

- 1 bod za uspješno uočavanje svih čvorova
- 1 bod za točan prikaz veza između čvorova
- 1 bod za točan prikaz svih komponenata
- 1 bod za ispravno povezivanje svih komponenata

Napomene:

- smjerovi ovisnosti između komponenti unutar poslužitelja nisu ključni pri ocjenjivanju