Programsko inženjerstvo

Završni ispit

2. veljače 2021.

GRUPA B

Izjavljujem da tijekom izrade ove zadaće neću od drugoga primiti niti drugome pružiti pomoć, te da se neću koristiti nedopuštenim sredstvima. Ove su radnje teška povreda Kodeksa ponašanja te mogu uzrokovati i trajno isključenje s Fakulteta. Također izjavljujem da mi zdravstveno stanje dozvoljava pisanje ove zadaće.

_____	_____	_____
JMBAG	*Ime i prezime*	*Vlastoručni potpis*

Završni ispit nosi ukupno 36 bodova, a minimum za prolaz završnog ispita je 12 bodova.

1. (1 bod) Navedite četiri dionika na Vašem projektu.
   Razvojni tim, asistenti, korisnici aplikacije, administratori sustava ...
2. (1 bod) Navedite klasifikaciju zahtjeva s obzirom na sadržaj.
   Funkcionalni, nefunkcionalni, domene primjene
3. (1 bod) Navedite UML dijagrame pomoću kojih ste u projektnoj dokumentaciji opisali arhitekturu programske potpore.
   Dijagram razreda, komponenti.
4. (1 bod) Kakvu ulogu imaju obrasci uporabe u Unificiranom procesu (engl. *Unified process*) razvoja programske potpore?
   Obrasci uporabe su pokretači iteracija/ .aktivnosti u životnom ciklusu i sinkroniziraju sadržaj različitih modela.
5. (1 bod) Usporedite razinu detalja UML dijagrama razreda u prvoj i drugoj inačici projektne dokumentacije.
   U prvoj prevladavaju konceptualni i specifikacijski, a u drugoj implementacijski.
6. (1 bod) Što je još uz vizualizaciju rada glavna praksa Kanban metodologije?
   Ograničavanje količine rada u tijeku.
7. (1 bod) Navedite jedan od 5 SOLID principa koji ste najviše primjenjivali na projektu i ukratko obrazložite način na koji ste ga koristili.
   Single responsibility, Open-closed, Liskov, Interface segregation, Dependency inversion
8. (2 boda) Navedite i opišite minimalne elemente oblikovnog obrasca (engl. *design pattern*). Navedite barem tri primjera oblikovnih obrazaca.
   Naziv - razumljivo ime, opis - opis problema, rješenje - opis predloška koji može biti upotrijebljen na različite načine, posljedice - rezultati i pogodnosti primjene obrasca uporabe. Npr. singleton, observer, delegation, adapter, facade, immutable, read-only, factory,...
9. (1 bod) Opišite osnovna svojstva koncepcijske arhitekture programske potpore (engl. *Conceptual Architecture*).
   Najviša razina apstrakcje,usmjerena na pogodnu dekompoziciju sustava, usmjerena na strukturno oblikovanje bez implementacijskih detalja.
10. (1 bod) Navedite dijelove općenite trorazinske arhitekture (engl. *three-tier architecture*).

Korisnička razina (engl. user; presentation; client tier), logička ili poslovna razina (engl. business; logic; middle tier), podatkovna razina (engl. data tier).

11. (1 bod) Što sve opisuje jezik WSDL (engl. Web Service Description Language)?

    Opisuje što radi usluga (metode koje pruža), način pristupa usluzi (format podataka i opis protokola) i smještaj/lokaciju usluge (URL na kojem se nalazi).

12. (1 bod) Kako se praktično ostvaruje smanjenje međuovisnosti između slojeva kod višeslojne arhitekture?

    Apstrahiranjem programskih entiteta koji povezuju slojeve, što se najčešće radi korištenjem sučelja.

13. (1 bod) Objasnite važnost specifikacije sučelja Jakarta Persistence (JPA) za radne okvire za web i navedite naziv sučelja u radnom okviru Spring Data koji se koristi za ostvarenje JPA.

    JPA omogućuje provođenje perzistencije podataka između poslužiteljske aplikacije i relacijske baze podataka bez potrebe za pisanjem SQL upita. Sučelje se naziva JpaRepository.

14. (2 boda) Što sve treba biti ispunjeno na UML dijagramu aktivnosti da započne izvođenje nekog čvora akcije i što se sve događa nakon izvođenja tog čvora?

    Prije izvođenja: da postoji odgovarajući broj znački na svim ulazima i da su zadovoljeni svi lokalni preduvjeti (engl. precondition). Nakon izvođenja: provjerava se jesu li zadovoljeni svi lokalni izlazni uvjeti (engl. postcondition) i prosljeđuju se značake na sve izlaze.

15. (1 bod) Navedite tri tipa poveznica koje se mogu crtati na UML dijagramu komponenti (nije potrebno crtati primjere).

    1. Spojnica (assembly connector, ball-and-socket, lollipop)
    2. Delegacija (engl. delegation connector)
    3. Ovisnost (engl. dependency)

16. (1 bod) Opišite osnovna svojstva i ponašanje programskog defekta (često upotrebljavani sinonim engl. error, fault, failure, bug) radi čega program ne udovoljava zahtjevima ili specifikacijama, klasificiranog kao *schroedinbug*.

    Defekt se očituje u ispravnom radu programa nakon što primijete da taj kod uopće nije trebao raditi.

17. (2 boda) Primjenom tehnika ispitivanja standarda ISO/IEC 29119 dvovrijednosne granica (engl. *two value boundary testing*) i ekvivalentnih particija (engl. *equivalence partitioning*) odredite ispitne slučajeve (engl. test case) za godišnji obračun potrošnje vode uporabom pametnog brojila koje može zatvoriti protok vode uz uvjete: potrošnja do 10 m^3 je besplatna, potrošnja do 799 m3 obračunava se 5 kn/m^3. Potrošnja iznad 799 m3 automatski zatvara protok vode. Brojilo očitava cjelobrojne vrijednosti. Odredite ekvivalentne particije i granične vrijednosti. Tablično prikažite konkretne ispitne slučajeve.

    E1: 10<= p <799 G:-1,0;9,10; 799, 800
    E3: p<0
    E4: 0<=p<=10
    E5: p>799

TC	Ulaz	Očekivani izlaz	Rezultat
TC1	-10	Poruka greške	
TC2	0	0	
TC3	1	0	
TC4	6	0	
TC5	9	0	
TC6	10	50	
TC7	799	3995	

TC8	500	2500	
TC9	800	Zatvoren ventil	
TC10	1200	Zatvoren ventil	

18. (1 bod) Kako nazivamo način razvoj programske potpore koji zahtjeva pisanje funkcijskog ispitivanja prije programiranja koda?

    Razvoj upravljan ispitivanjem (Test Driven Development, Test First Development)

19. (1 bod) Za općeniti model sustava zadan Kripke strukturom M = (S, R, L), interpretirajte formulu **M, $s_0 \models \neg$ AX (q $\wedge$ r).**

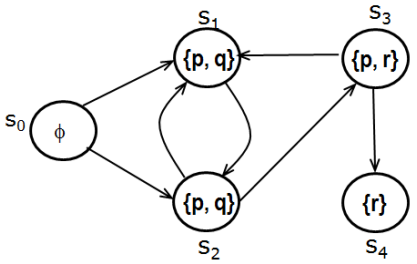
    Postoji barem jedan put od početnog stanja s0 na kojem ne vrijedi u sljedećem stanju (q $\wedge$ r) ili
    Nije istina da na svim putevima od početnog stanja s0 u sljedećem stanju vrijedi (q $\wedge$ r)

20. (2 boda) Za zadanu Kripke strukturu odredite skup stanja koja zadovoljavaju formule. Nije potrebno obrazlagati postupak.

    A. φ=EF(EG p),

    B. φ =E(p U q)

Odgovor: A. S0, S1, S2, S3 ; B. S1, S2, S3



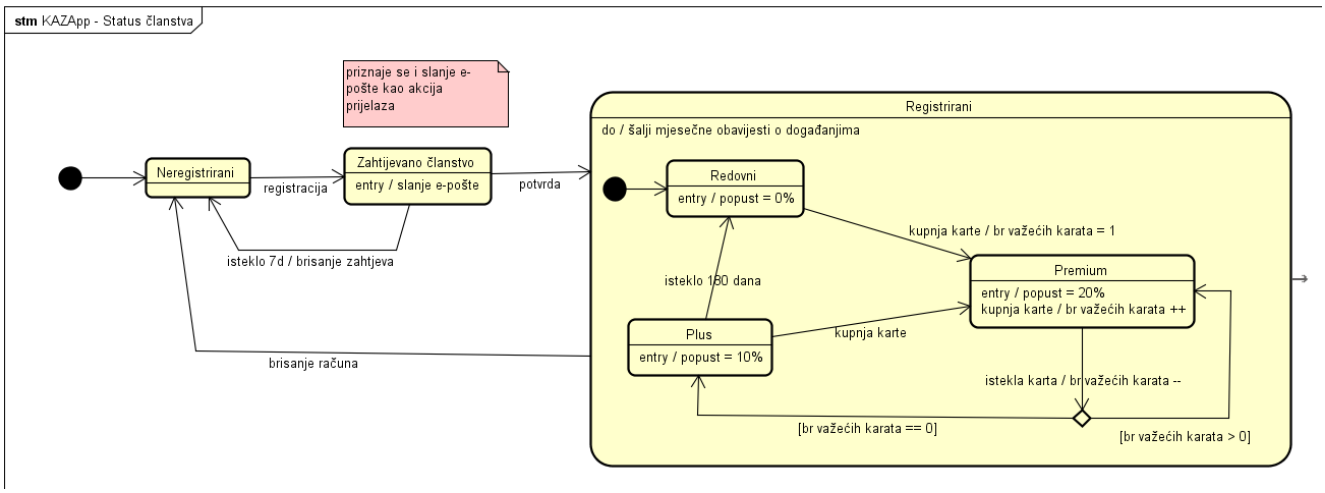
Problemski dio

Potrebno je razviti mobilnu aplikaciju KAZApp za kazališta grada Zagreba putem koje će korisnici dobivati informacije o događanjima. Za korištenje aplikacije potrebna je registracija adresom e-pošte i ostalim osobnim podacima. Prilikom registracije na upisanu adresu e-pošte šalje se zahtjev s poveznicom za potvrdu registracije. Ukoliko korisnik ne aktivira poveznicu unutar 7 dana, zahtjev za registraciju se briše. Korisnik može u svakom trenutku zatražiti brisanje korisničkog računa čime prestaje članstvo u aplikaciji i račun se trajno briše iz baze podataka.

Korisnici aplikacije na početku mjeseca dobivaju obavijesti o događanjima za taj mjesec. Putem aplikacije moguće je kupiti jednu ili više mjesečnih karata što omogućava neograničeni broj posjeta kazališnim predstavama. Karta vrijedi 30 dana od uplate, a za provedbu plaćanja aplikacija se spaja na vanjski servis za plaćanje *ENovcic* koji kupcima omogućava sigurno plaćanje kreditnim karticama. U jednoj kupnji moguće je kupiti samo jednu kartu. Kupnjom godišnje karte korisnik dobiva status Premium člana i ostvaruje popust od 20 % na kupnju nove mjesečne karte. Nakon proteklih 30 dana od zadnje kupnje mjesečne karte, korisnik prelazi u status Plus i sljedećih 180 dana može ostvariti popust od 10% na kupnju mjesečne karte. Ako niti u tom periodu ne kupi novu mjesečnu kartu, korisnik se vraća u status običnog člana.

Mobilna aplikacija, podržana isključivo na operativnom sustavu Android Marshmallow ili novijem, dohvaća podatke o kazalištima i korisnicima spajanjem na web aplikaciju KAZAWebApp koja se izvodi na web poslužitelju Apache Tomcat 9. Web aplikacija sve podatke trajno pohranjuje u MySQL bazu podataka. Web aplikacija je pokrenuta na virtualnom stroju u CARNET oblaku na kojem je instaliran operacijski sustav CentOS 8, a baza podataka na računalu na kojem je instaliran Windows Server 2016. Korišteni komunikacijski protokol između mobilne i web aplikacije te mobilne aplikacije i servisa za plaćanje je HTTPS, a između web aplikacije i baze podataka je TCP/IP.

21. (4 boda) UML dijagramom stanja modelirajte status članstva u mobilnoj aplikaciji.



Bodovanje:

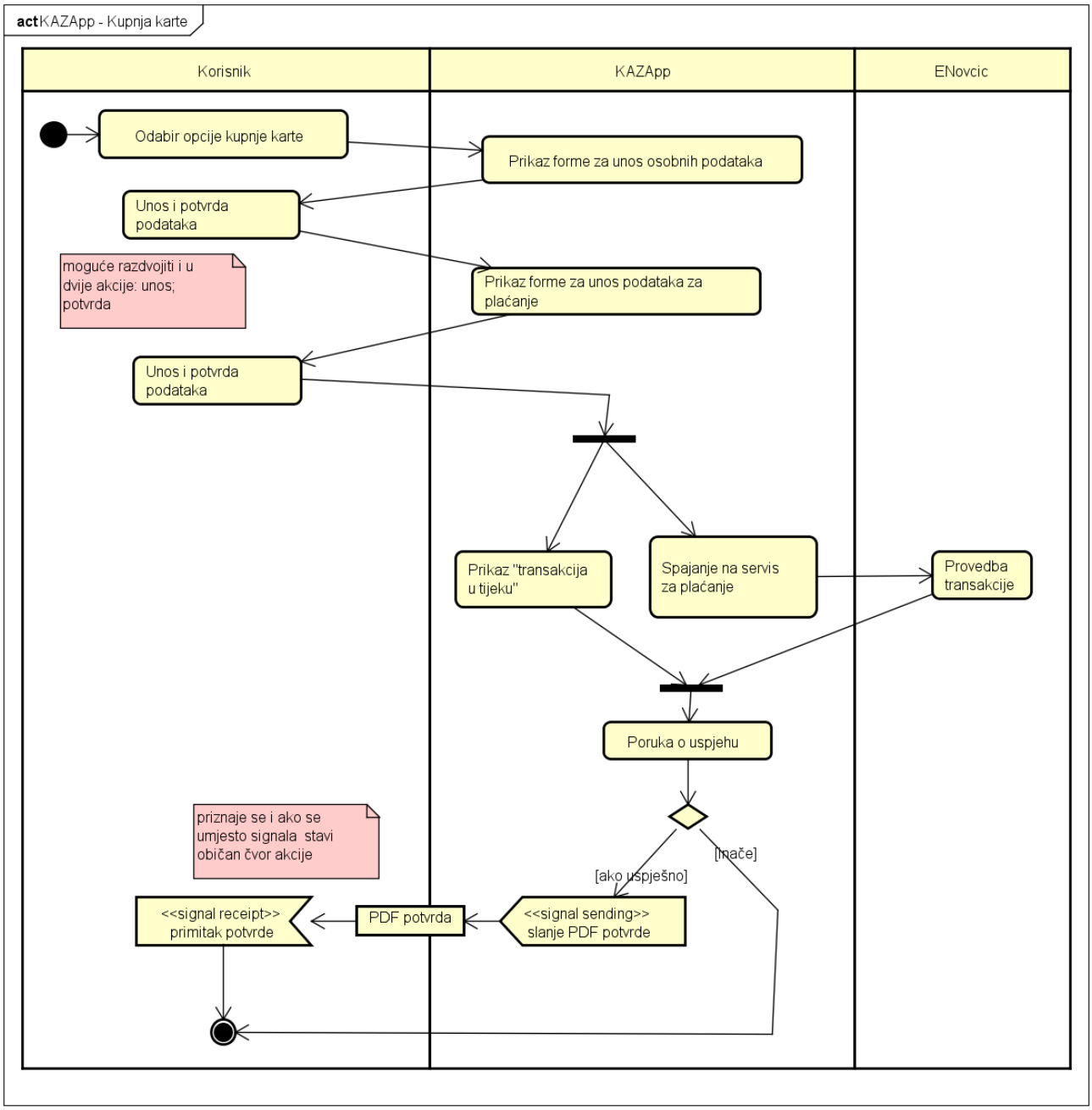
- 0.75 za navedena „vanjska" stanja: Neregistrirani, Zahtijevano članstvo (ili neki sličan naziv) i Registrirani
  - 0.25 po točno navedenom stanju
- 0.5 za prijelaze između „vanjskih" stanja: registracija, potvrda, istek 24h, brisanje računa
  - semantički ispravno, sintaksno neispravno (zagrade...) - 0.25
- 0.75 za podstanja od stanja Registrirani: Redovni, Premium, Plus
  - bodovanje: 0.25 po točno navedenom stanju
- 0.5 za prijelaze unutar stanja Registrirani, osim prijelaza vezanih uz odluku
  - semantički ispravno, sintaksno neispravno (zagrade...) - 0.25
- 0.5 za prijelaz vezan uz odluku
  - čvor odluke ne mora nužno biti prikazan, ali onda moraju postojati 2 uvjetovana prijelaza
- 0.25 za akcije slanja e-pošte i brisanja računa
  - bodovanje: sve ili ništa
  - priznaju se moguće manje razlike u rješenjima: npr. brisanje zahtjeva može biti i exit akcija od stanja Registrirani i sl.
- 0.5 za varijable stanja:
  - popust 0.25, br karata 0.25
- 0.25 za „do/slanje mjesečnih obavijesti"

- ako nema hijerarhije stanja, nego su sva stanja na istoj razini boduju se navedena stanja i prijelazi (kao da su u hijerarhiji) i onda se od toga oduzima –0.25 (takvo rješenje je moguće, ali je nezgrapno).

- stanja i prijelazi viška:
  - ako ne mijenjaju značenje dijagrama – priznati
  - ako nemaju smisla: oduzeto ukupno od -0.25 do -1 po procjeni ispravljača

22. (4 boda) UML dijagramom aktivnosti modelirajte proces kupnje jedne karte. Uz osnovni tekst zadatka uzmite i sljedeće informacije u obzir.

Proces kupnje karte započinje tako što korisnik odabere opciju kupnje karte i zatim mu se otvara forma za unos podataka o osobi na koju će glasiti karta. Nakon potvrde upisanih podataka korisnik unosi podatke o kreditnoj kartici s kojom će platiti kartu i potvrđuje kupnju. Mobilna aplikacija se spaja na servis za plaćanje i pokušava provesti transakciju, a za to vrijeme korisniku na zaslonu prikazuje da je transakcija u tijeku. Nakon završetka transakcije korisnika se obavještava o uspješnosti i ako je transakcija bila uspješna, šalje se potvrda o kupovini u PDF formatu na adresu e-pošte.

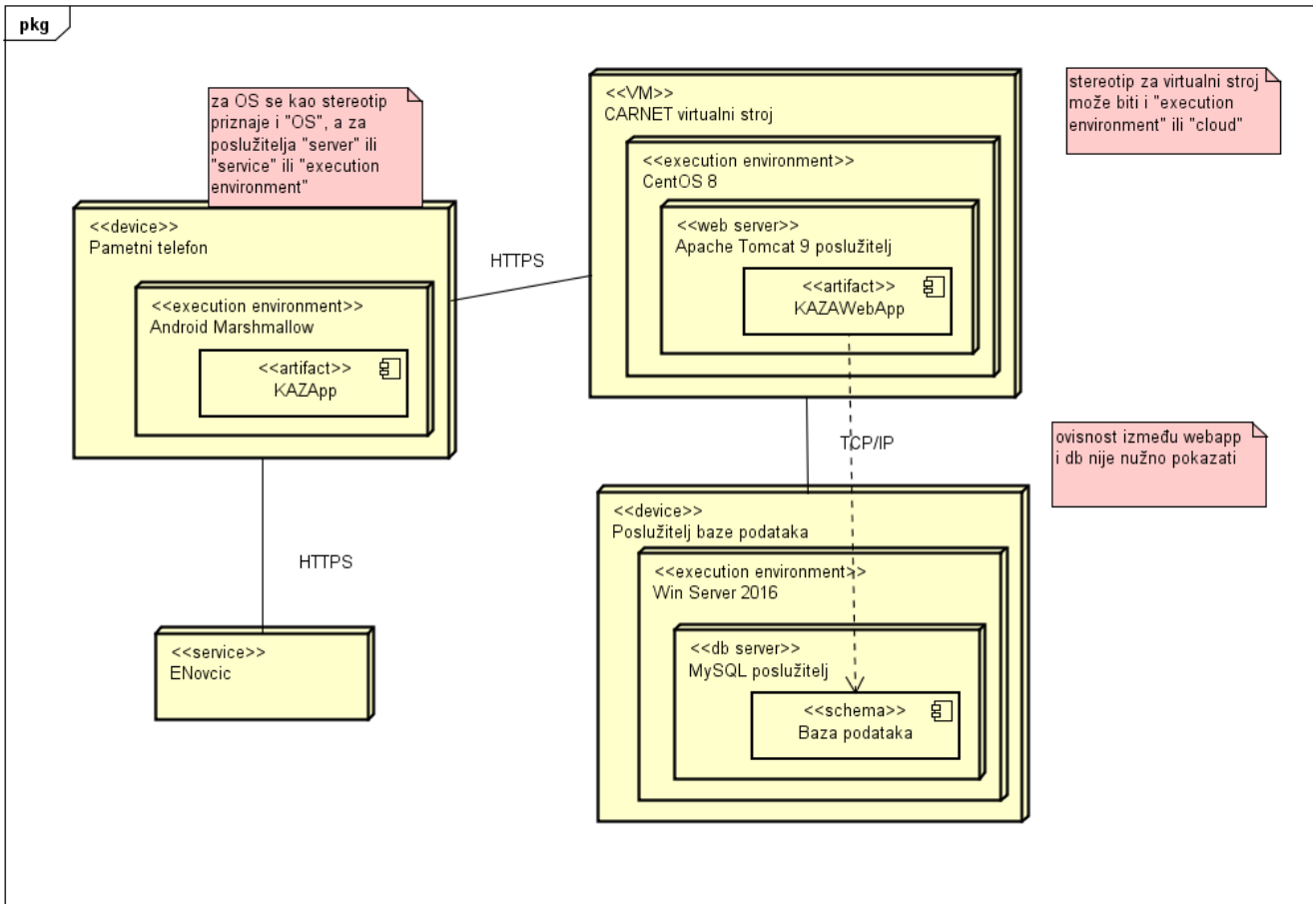


Bodovanje:

- 0.75 particije:
  - 0.25 po točno navedenoj particiji
- 0.5 početni i završni čvor:
  - 0.25 početni, 0.25 završni
- 1 ispravan slijed akcija za unos osobnih i podataka za plaćanje
  - ovaj slijed akcija moguće je modelirati s više ili manje čvorova akcije, ali mora ostati isto značenje (tijek aktivnosti) i mora biti sintaktički točno
  - ukoliko jedan dio ovog slijeda nije ispravno modeliran - 0 do 0.75 po procjeni ispravljača
- 0.75 fork/join + akcije između
  - bodovanje: 0.25 ako je prikazano kao jedan tok bez paralelizma i navedene su sve akcije
- 0.5 odluka
  - ako sintaktički nije točno: 0b
  - priznaje se ako se grana "inače" umjesto na kraj toka, vraća na neku smislenu prethodnu akciju (odabir opcije kupnje karte, potvrda kupnje..)

- 0.5 ispravno prikazano slanje potvrde
    - Priznaje se ako su umjesto signala korišteni čvorovi akcija

23. (4 boda) UML dijagramom razmještaja modelirajte cijeli sustav.



Bodovanje:

- 0.25 po ispravno prikazanoj komponenti/čvoru (ukupno 12*0.25 = 3)
    - ispravno prikazanom komponentom se smatra ona koja ima ispravan simbol (kvadar/pravokutnik) i naziv
    - u slučaju da su čvorovi prikazani dvodimenzionalno - 0.1 bod po čvoru
- 0.5 za sve ispravno definirane stereotipove
    - bodovanje sve ili ništa
    - priznaju se mogući različiti stereotipovi koji se smatraju jednako ispravnima tj. točno opisuju čvor/komponentu u kojoj je riječ:
        - za operacijski sustav (Android, RHEL, WIN..): <<execution environment>>, <<OS>>
        - za poslužitelj (web i db): <<execution environment>>, <<service>>, <<server>>, <web server>>, <<db server>>…
        - za komponentu: <<artifact>>, <schema>>, <<executable>>…
- 0.5 za sve ispravne veze između čvorova
    - bodovanje sve ili ništa