

Datum: 20.03.2018.

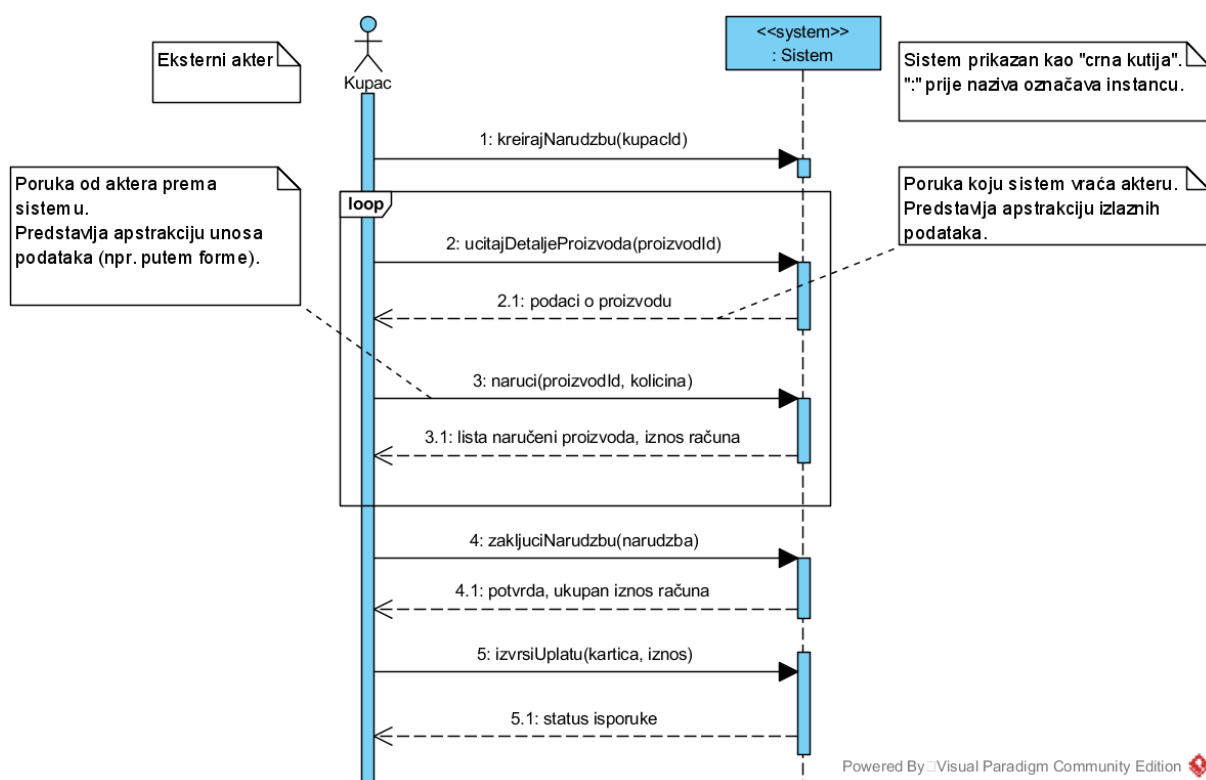
"System Sequence Diagrams" Sistemske dijagrami sekvenci

Sistemske dijagram sekvenci (System Sequence Diagram – SSD) je dijagram UML-a koji prikazuje *interakciju eksternih aktera sa sistemom*. SSD se kreira na osnovu opisa use-case dijagrama. Osim toga, SSD se koristi kao ulazni podatak za kreiranje ugovora operacija.

SSD pokazuje određeni događaj (sekvencu) u okviru jednog "use-case"-a i aktere koji direktno koriste funkcionalnosti sistema. Sistem se u ovom momentu promatra kao "*crna kutija*", tj. prikazuju se **događaji koje generiše akter i rezultati koje daje sistem**, bez opisa načina na koji je sistem došao do tih rezultata. Vrijeme na dijagramu se prikazuje na vertikalnoj liniji života objekta, tj. prvi događaj je na vrhu dijagrama, a događaji koji slijede nakon njega su prikazani ispod i to po redoslijedu dešavanja.

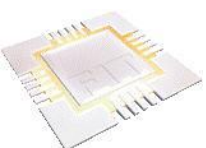
Za prikazivanje SSD-a koristit ćemo **notaciju sekvencijalnih dijagrama**.

Sljedeća slika daje pojednostavljen primjer jednog sistemskog dijagrama sekvenci kojim je predstavljena kupovina proizvoda.



Slika 1: SSD Kupovina proizvoda

Obično se na sistemskom dijagramu sekvenci prikazuje **samo glavni tok izvršavanja** "use-case"-a. Ukoliko je neki od alternativnih tokova interesantan, **može** se prikazati na **posebnoj dijagramu**.



Šta su sistemski dijagrami sekvenci?

Use-case opisuje način na koji eksterni akteri komuniciraju sa sistemom koji želimo kreirati. Za vrijeme te interakcije akter generiše **sistemske događaje** (što zahtijeva da neka **sistemska operacija** obradi taj sistemski događaj, ili drugim riječima, sistemski događaj uzrokuje izvršavanje odgovarajuće sistemske operacije).

Sistemski dijagram sekvenci je "slika" koja za jedan scenarij (**glavni tok**) prikazuje sistemske događaje koje generiše akter, redoslijed kojim se ti događaji izvršavaju, kao i događaje između sistema. Svi sistemi se tretiraju kao "zatvorene kutije" jer se žele istaknuti događaji koji prelaze granice sistema, od aktera prema sistemu.

Postavlja se pitanje zašto crtati SSD? Jako je bitno precizno odrediti koji vanjski događaji utiču na sistem. Krajnji cilj je: dizajniranje sistema koji obrađuje te vanjske događaje i generiše izlaze kao rezultate tih događaja.

Ukratko, softverski sistem izvršava akcije uslijed:

1. eksternih događaja koje generiše akter,
2. vremenskih događaja (generišu se u određeno vrijeme, periodično, itd.),
3. pogreške.

Prije detaljnog dizajna softverske aplikacije potrebno je specificirati **šta sistem treba da radi, a ne kako to radi** (ponašanje sistema). Crtanjem sistemskih dijagrama sekvenci upravo se fokusiramo na to pitanje: Koje eksterne događaje sistem treba da obradi?".

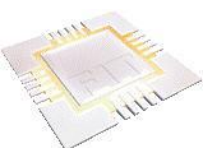
Treba naglasiti da UML **ne definira pojam "sistemskih dijagrama sekvenci", nego jednostavno dijagrama sekvenci** ("sequence diagrams"). Kada dijagram sekvenci koristimo **da opišemo interakciju između korisnika i sistema, tada govorimo o sistemskim dijagramima sekvenci**. Kasnije ćemo dijagram sekvenci koristiti da opišemo interakciju softverskih objekata, tj. koristit ćemo isti alat koji nam UML stavlja na raspolaganje, ali u drugom kontekstu.

Sistemski dijagrami sekvenci se mogu koristiti da bi se ilustrirale interakcije između različitih sistema. **Na primjer, sistem studentske službe bi mogao slati upite računovodstvenom sistemu univerziteta radi provjere izvršenih uplata (recimo kod prijave ispita, ako se želi onemogućiti prijava ispita ukoliko student nije izvršio sve obaveze plaćanja).**

Imenovanje sistemskih događaja i operacija

Sistemski događaji bi trebali biti izraženi u općem obliku, bez naznačavanja konkretnih fizičkih ulaznih uređaja. Na primjer **"unosPodataka"** je primjerenije od **"utipkavanjePodataka"** jer se ovim drugim implicira korištenje **tipkovnice** za unos iako bi se unos mogao riješiti i na drugi način (npr. skeniranjem bar-koda ili slično). Pored toga, kada je sistemski događaj izražen u općem obliku, fokusira se na **namjeru** događaja, a ne na tehniku **generisanja** događaja.

Također, radi pojašnjenja, sistemski događaji bi trebali **počinjati glagolom** (npr. dodaj..., unesi..., kreiraj...) čime se naglašava da se u stvari radi o **komandama upućenim sistemu**.



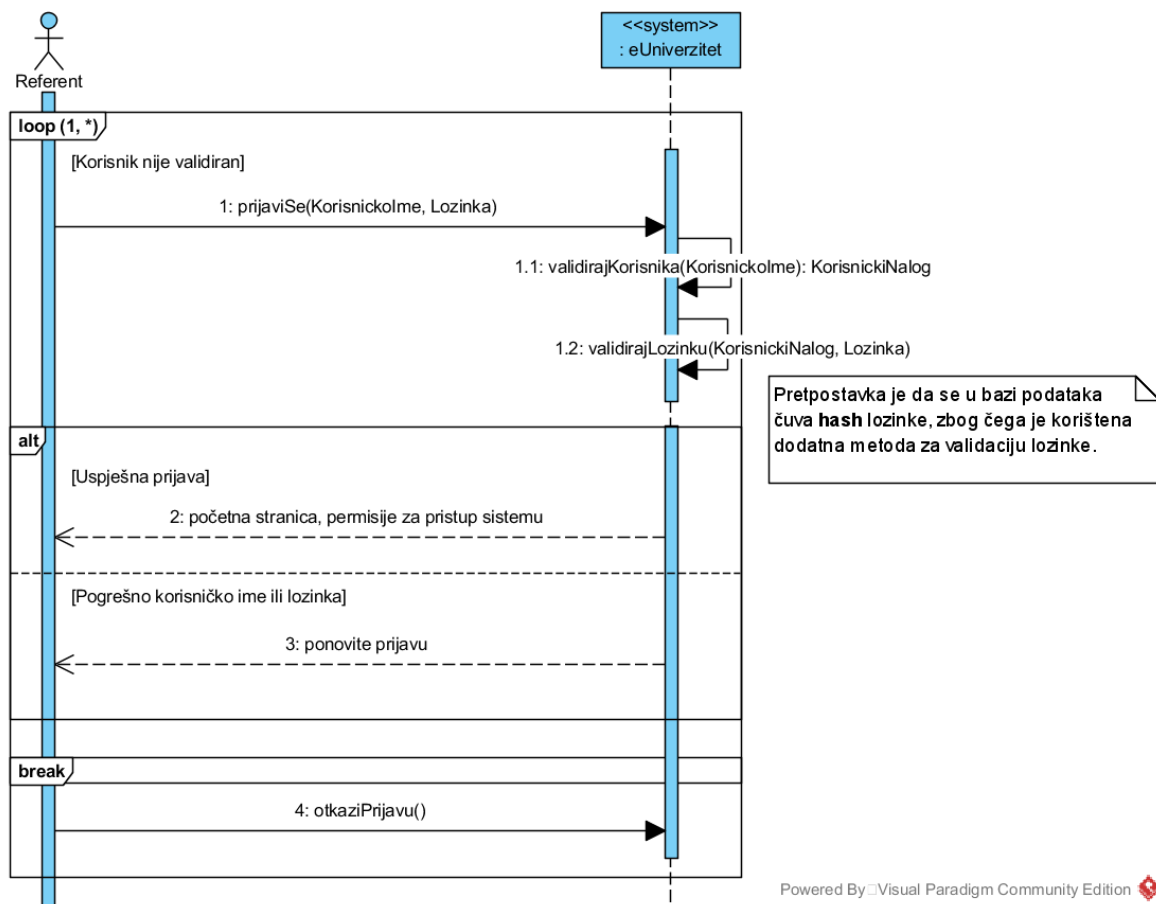
Zadaci

Za modeliranje interakcije aktera i sistema u projektu *eUniverzitet* primarno je odabran slučaj korištenja „Procesiranje zahtjeva za uvjerenjem“. Također, dat je i jednostavniji dijagram za prikaz komunikacije referenta i sistema **prilikom prijave** na sistem.

Zadatak 1: Prijava na sistem

Napisati specifikaciju slučaja korištenja „Prijava na sistem“, te na osnovu nje izraditi sistem sekvencijalni dijagram za prikaz interakcije referenta studentske službe i sistema *eUniverzitet* prilikom prijave.

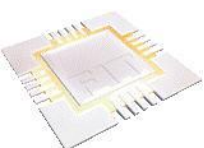
Prijedlog rješenja dijagrama:



Zadatak 2: Procesiranje zahtjeva za uvjerenjem

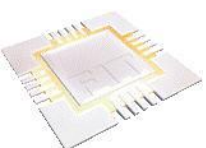
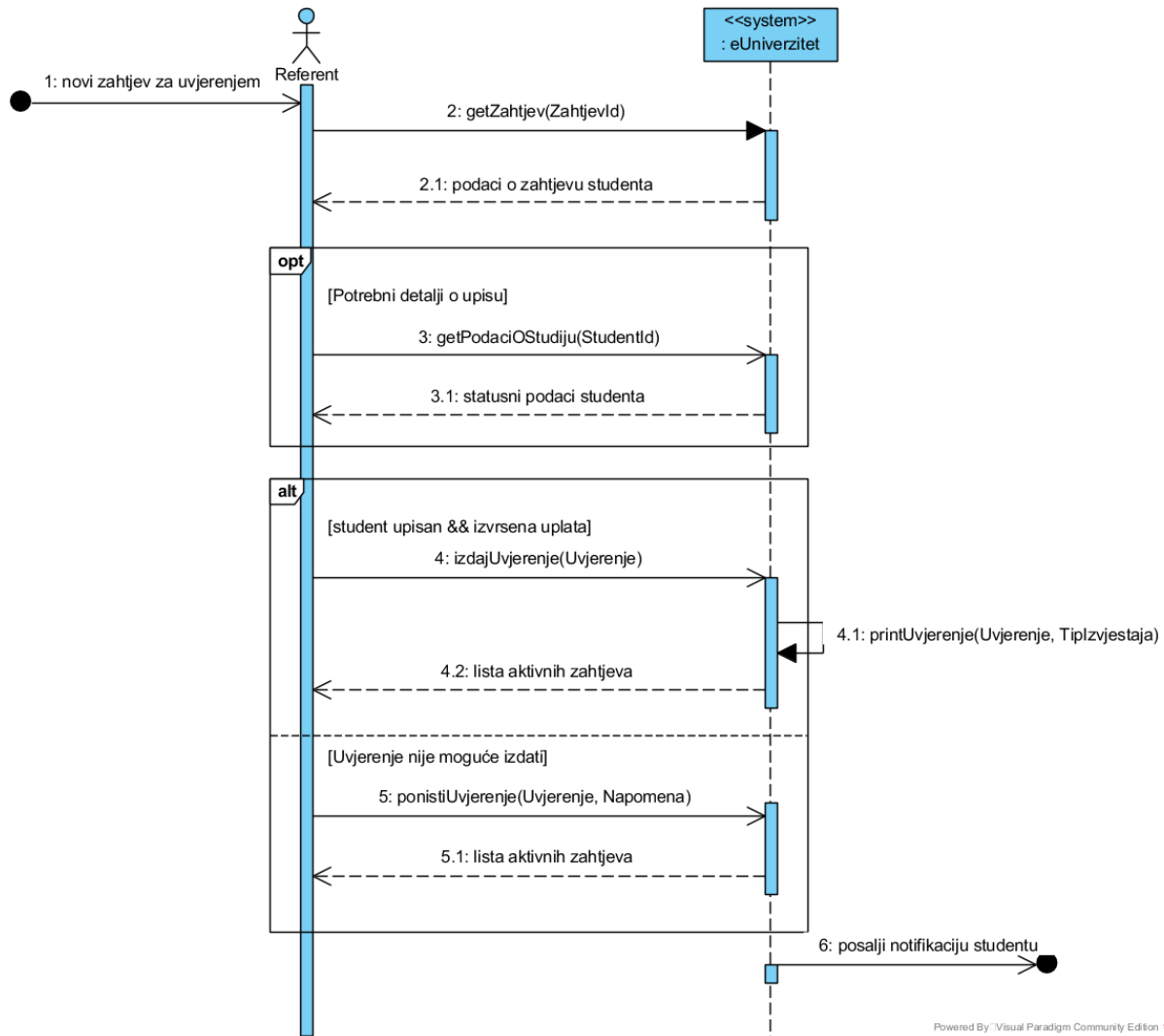
Referent studentske službe o svakom novom zahtjevu za uvjerenjem dobiva notifikaciju putem sistema. U nastavku je opisan postupak procesiranja jednog zahtjeva.

Prvobitno se preuzimaju detalji novog zahtjeva. U slučaju da referent ne može na osnovu podataka iz zahtjeva zaključiti da li je uvjerenje moguće izdati, on od sistema traži detaljniji prikaz podataka o studiju studenta koji je poslao zahtjev. Ukoliko su ispunjeni preduslovi za izdavanje uvjerenja (student je upisan u trenutni semestar u trenutnoj akademskoj godini, izvršena je uplata za procesiranje uvjerenja), referent izdaje uvjerenje, nakon čega sistem printa odgovarajući izvještaj. U slučaju da uvjerenje nije



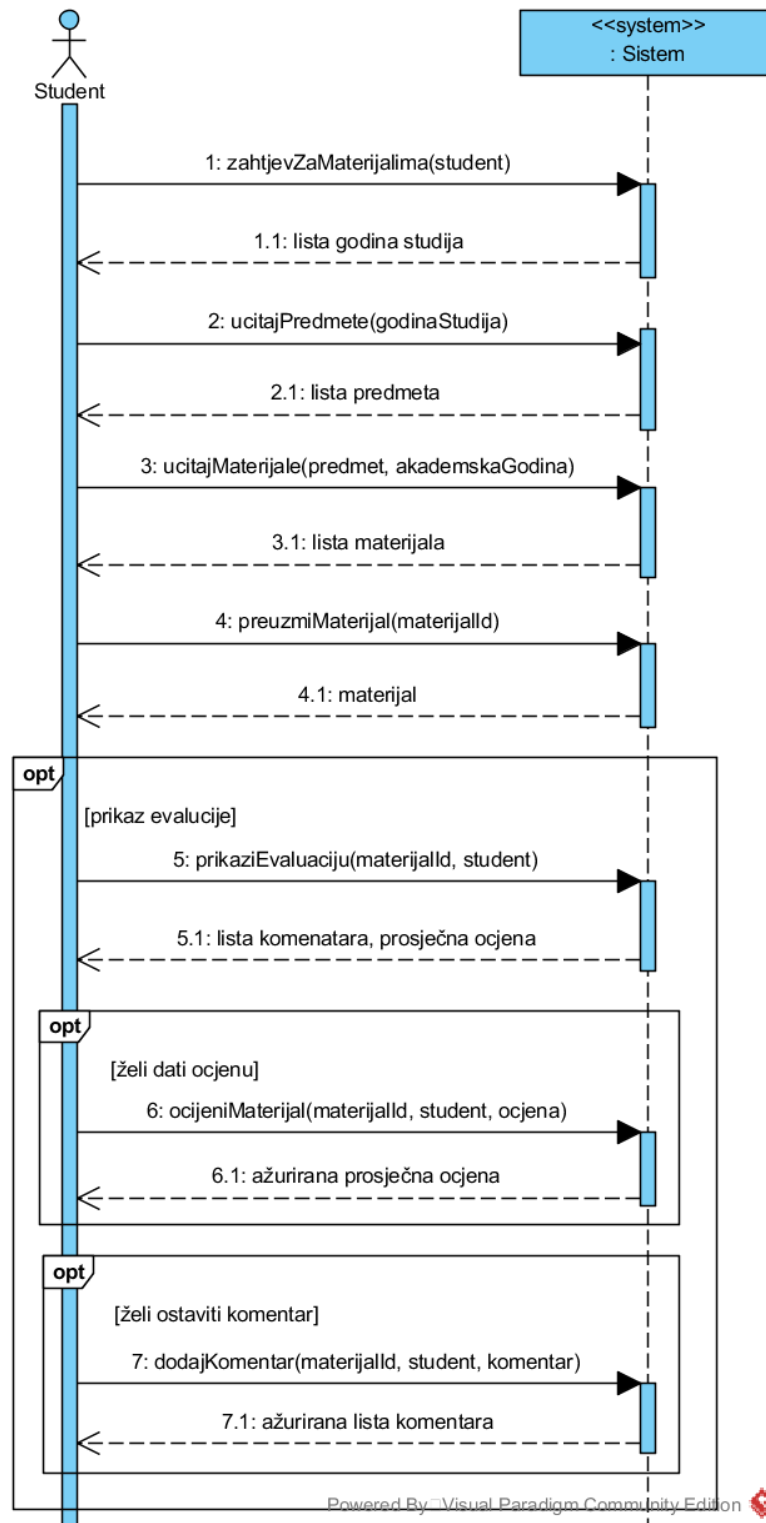
moguće izdati, referent odbija zahtjev studenta uz odgovarajuću napomenu koja sadrži razlog zbog kojeg uvjerenje nije moguće izdati. U oba slučaja sistem po završetku procesiranja uvjerenja referentu vraća listu preostalih aktivnih zahtjeva studenata. Također, studentu se šalje notifikacija da je zahtjev procesiran kako bi mogao da preuzme uvjerenje ili pročita napomenu u slučaju da je zahtjev odbijen.

Prijedlog rješenja:

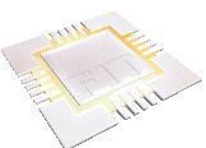


Dodatni primjeri:

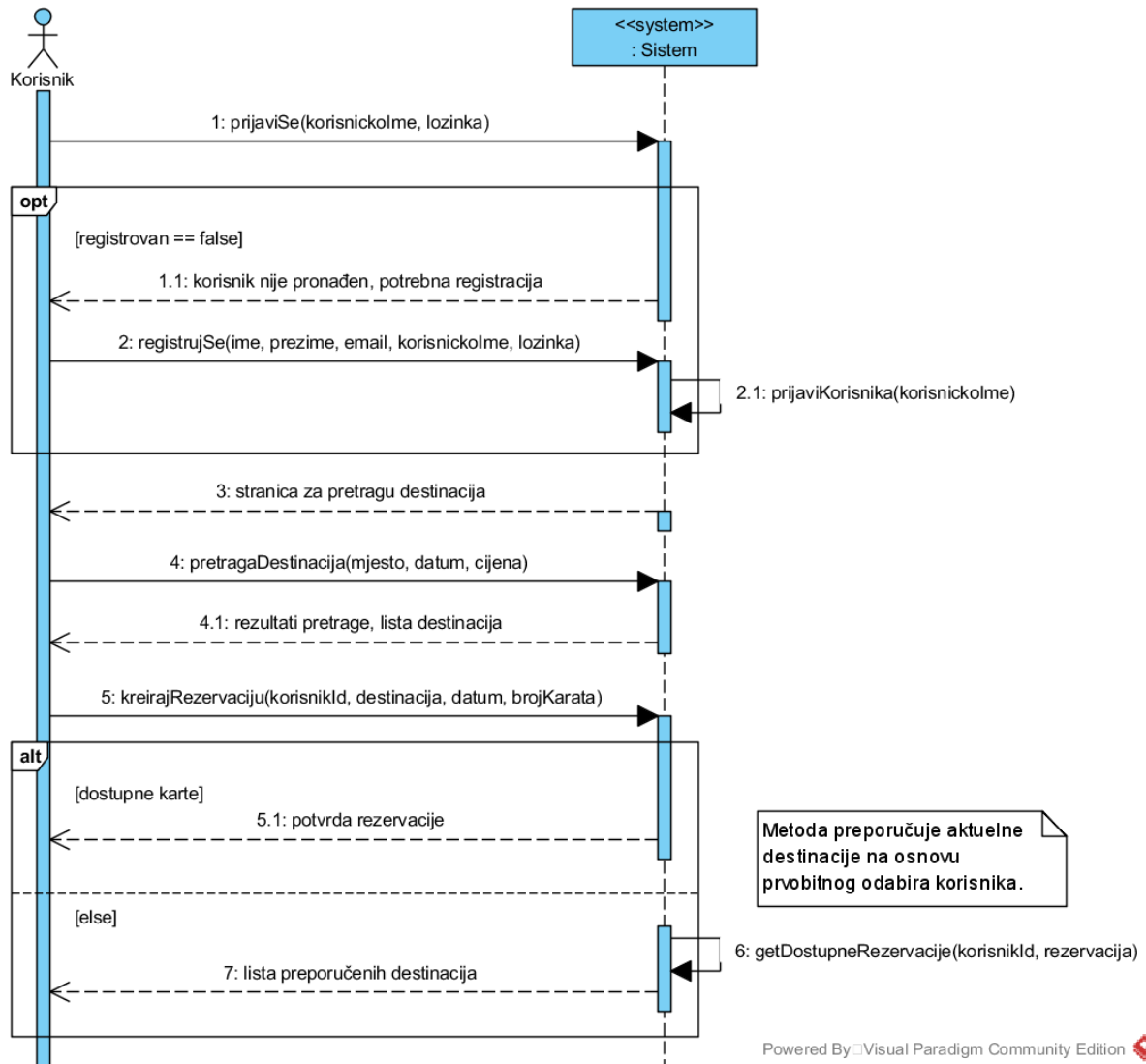
SSD Preuzimanje i ocjena materijala: U nastavku je dat primjer sistemskog dijagrama sekvenci za preuzimanje i ocjenu materijala putem DL sistema (obuhvaćen je osnovni scenarij).



Powered By Visual Paradigm Community Edition



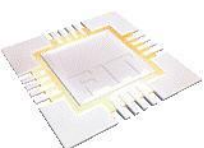
SSD Kupovina karte: U procesu on-line rezervacije korisnik se prijavi na sistem (ili registruje prije prijave, ukoliko nije prethodno registrovan). Nakon toga korisnik pretražuje dostupne destinacije. Odabirom željene destinacije i datuma započinje se proces rezervacije karte. Korisnik može rezervisati više od jedne karte. Sistem provjerava dostupnost željene rezervacije i potvrđuje rezervaciju ili informiše klijenta o drugim dostupnim rezervacijama.



Ispitni zadatak 1:

Potrebno je izraditi aplikaciju za upravljanje redom vožnje i prodajom autobuskih karata. Aplikacija je namijenjena isključivo uposlenicima određene autobuske stanice. Izdvojeni su sljedeći funkcionalni zahtjevi:

- Administrator sistema vodi evidenciju o svim uposlenicima (ime, prezime, datum_zaposlenja, radno_iskustvo) kompanije. Uposlenici kompanije mogu biti šalterski radnici, kondukteri i vozači. On se također brine za ažuriranje reda vožnje. Red vožnje treba da sadrži podatke o autobuskim linijama. Za svaku liniju se pohranjuju podaci o autobusu (br_autobusa, godište, br_sjedišta, klima, tv), broju perona, relaciji, vremenu polaska i kašnjenju izraženom u minutama. Za



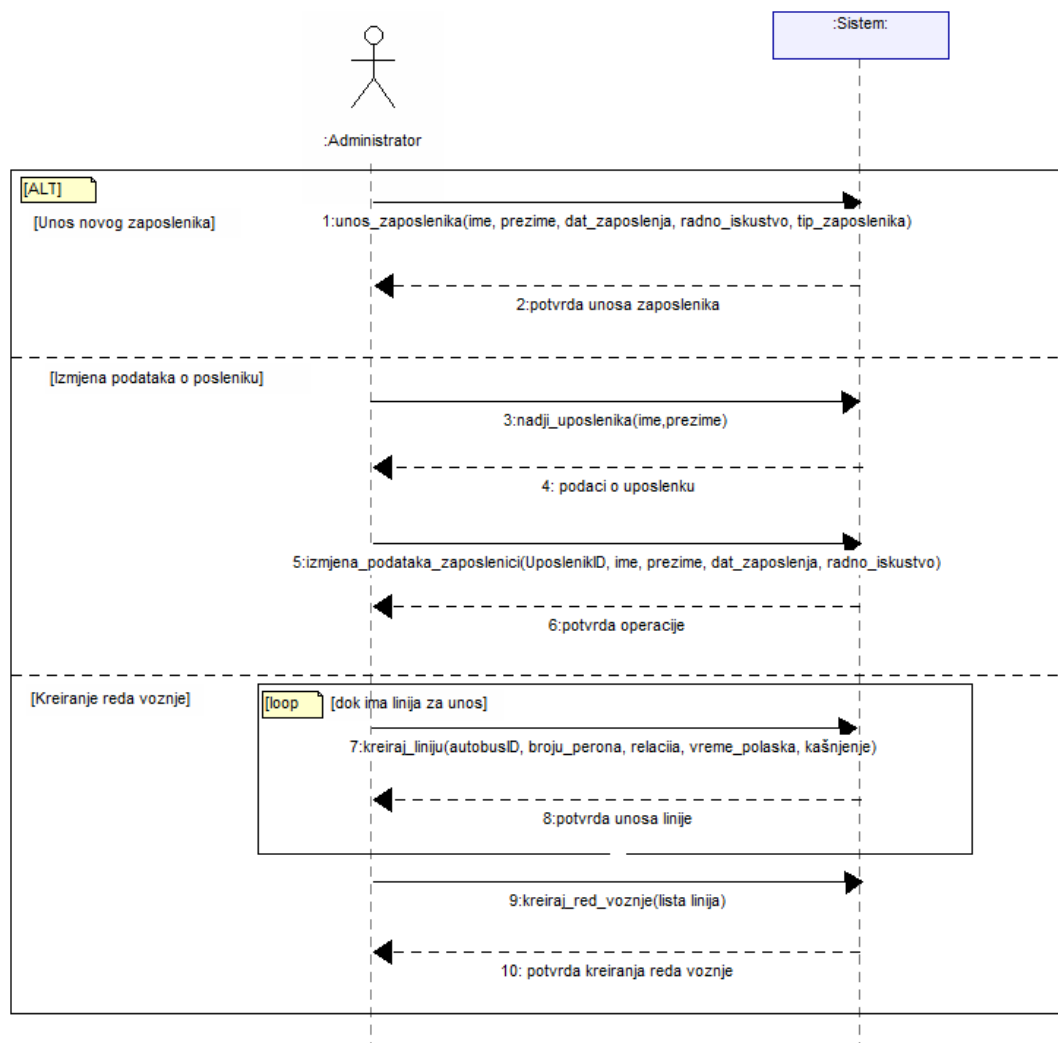
svaku autobusku liniju je u određenom periodu (sedmica u mjesecu) zadužen jedan vozač. Jedan vozač može imati dodijeljeno i više linija, ali u različitom vremenskom periodu.

- Vozači i kondukteri putem sistema mogu pregledati red vožnje. Prijavom na sistem, svaki vozač i kondukter ima označene autobuske linije na kojima je trenutno zadužen (vlastiti raspored). Oni također imaju mogućnost prijave kvara određenog autobusa, nakon čega sistem obavještava administratora kako bi se izvršile potrebne izmjene u redu vožnje. Vozači imaju licence i vozačke dozvole, te se pohranjuju podaci: br_licence, br_vozačke, datum_isteka_licence i datum_isteka_vozačke.
- Šalterski radnici obavljaju prodaju i naplatu autobuskih karata putnicima. Pri tome oni u sistemu imaju mogućnost pretrage reda vožnje po relaciji (polazište i/ili odredište) i datumu polaska. Odabirom željene autobuske linije sistem vrši provjeru slobodnog broja mjesta, te se nudi opcija izdavanja željenog broja karata putniku.

Zadatak 1: Modelirati interakciju administratora sistema sa sistemom.

Rješenje studenta:

Ispod je prijedlog rješenja studenta koji je osvojio maksimalan broj bodova (20 od mogućih 20). Dijagram je kreiran u alatu Open ModelSphere.



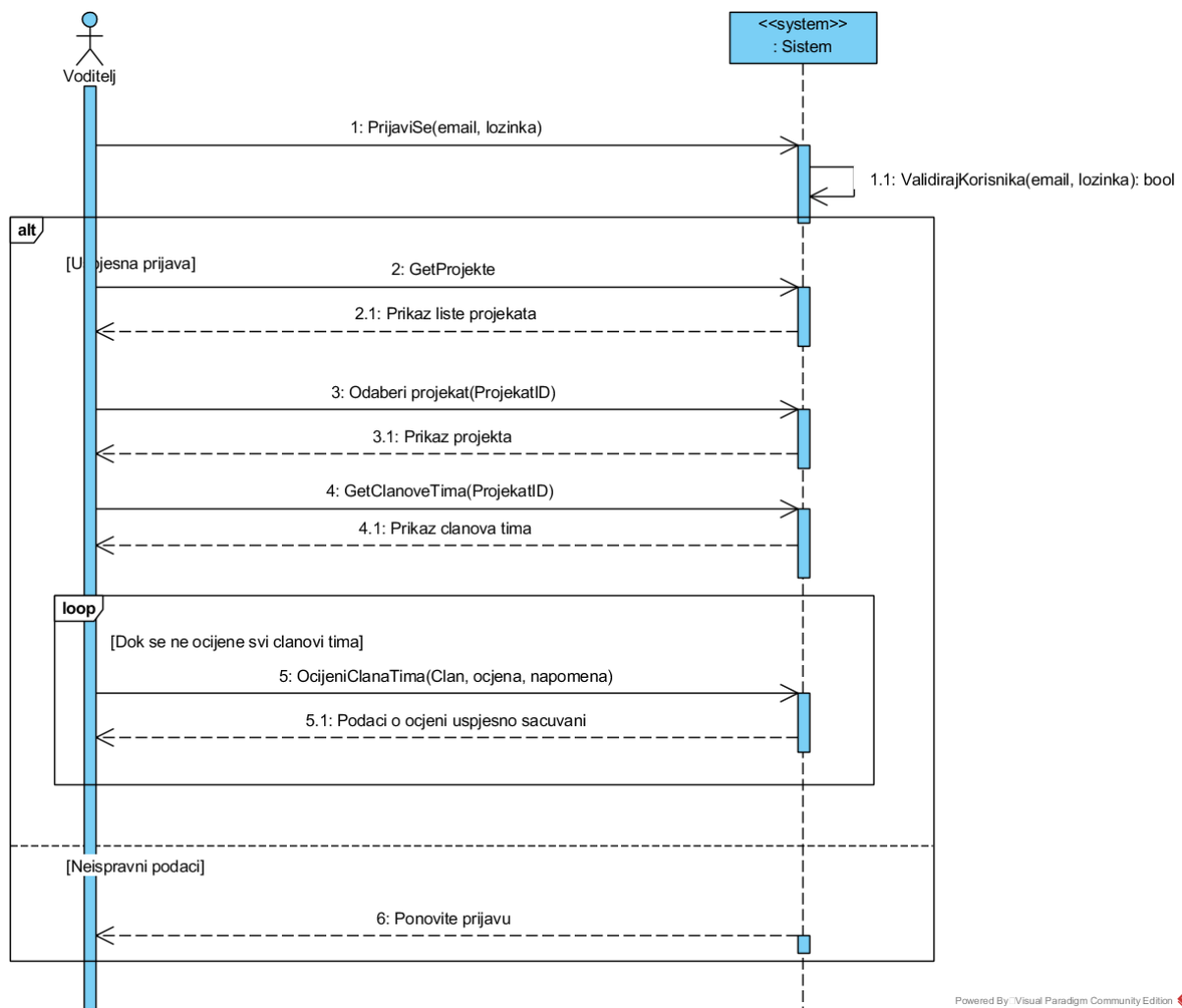
Ispitni zadatak 2:

Tekst ispitnog zadatka je prethodno objavljen u okviru materijala Use-case model.

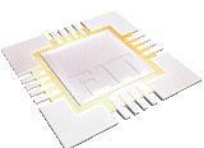
(10 bodova) Modelirati sistem sekvencijalni dijagram za slučaj korištenja „Ocjena članova tima“.

Rješenje studenta:

Ispod je prijedlog rješenja studenta koji je na ispitu osvojio maksimalan broj bodova za dijagram (10b od mogućih 10b). Dijagram je kreiran u alatu Visual Paradigm. U rješenje nije trebalo uključiti proceduru prijave na sistem, kao ni provjeru validnosti pristupnih podataka.



Powered By Visual Paradigm Community Edition



Prateće video lekcije možete pogledati u sklopu YouTube kanala na datim linkovima:

- [Sistem sekvencijalni dijagram 2017_8 - Predavanje](#)
- [Sistem sekvencijalni dijagram - Vježbe](#) (Visual Paradigm 14.0)
- [Sistem sekvencijalni dijagram](#) (Open ModelSphere 3.2)
- [Sistem sekvencijalni dijagram - Ispitni zadatak](#) (Open ModelSphere 3.2)

Literatura

1. Martin Fowler, UML Distilled Third Edition: A brief guide to the standard object modeling language, 2004. [Download](#) (Dostupno: 20.03.2018.).
2. Craig Larman, Applying UML and patterns, Prentice Hall, 2004.
3. <https://www.visual-paradigm.com/>
4. <https://www.youtube.com/user/VisualParadigm/>
5. <http://www.uml.org/>

