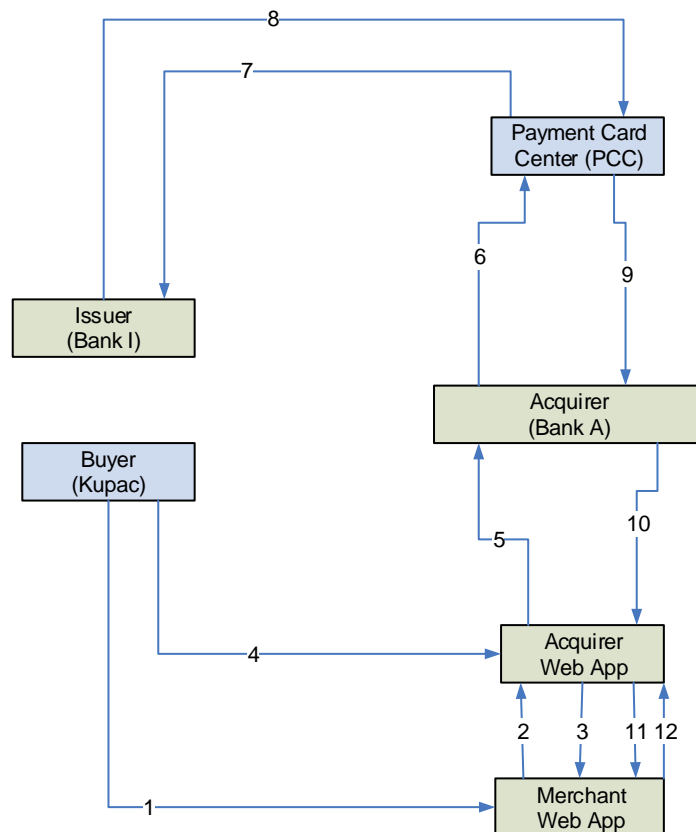


Projektni zadatak za predmet
Sistemi elektronskog plaćanja
verzija 1.0
02.11.2016.

Scenario u kome se odvija komunikacija između različitih informacionih sistema obuhvata sledeće tipove učesnika:

1. Kupac (*Buyer*)
2. Prodavac (*Merchant*)
3. Banka prodavca (*Acquirer*)
4. Banka kupca (*Issuer*)
5. Centar za platne kartice (*PCC - Payment Card Center*)

Za potrebe projektnog zadatka razmatra se ukupno pet učesnika: *Kupac* koji ima otvoren račun u *banci I* vrši neku kupovinu međunarodnog putnog osiguranja preko web-a kod *prodavca* i plaća platnom karticom posredstvom web aplikacije *Acquirer*-a za online naplatu. Račun prodavca nalazi se u *banci A*. Komunikacija između banke kupca (*Bank I*) i banke prodavca (*Bank A*) vrši se posredstvom *PCC*-a.



1. Komunikacija između učesnika

1. Korisnik posredstvom web aplikacije prodavca odabere osiguranje koje želi kupiti. Potvrda kupovine treba da redirektuje korisnika na web aplikaciju *Acquirer*-a za naplatu (na adresu *PAYMENT URL*) kao i da prilikom redirekcije prosledi *PAYMENT ID* parametar. Podatke

PAYMENT URL i PAYMENT ID web shop aplikacija će dobiti u koracima 2. i 3.

2. Merchant Wep App formira zahtev za dobijanje PAYMENT URL i PAYMENT ID parametara koji putem web servisa prosleđuje Acquirer Wep App. Podaci koji se prosleđuju su:
 - MERCAHNT ID – ID trgovca koji se dobije od banke prilikom registracije prodavca za online prodaju (tip String(30))
 - MERCHANT PASSWORD- Lozinka koja se dobije od banke prilikom registracije prodavca za online prodaju (tip String(100))
 - AMOUNT – Iznos transakcije (tip Decimal(10, 2))
 - MERCAHNT ORDER ID – Prodavčev ID transakcije (tip Number(10))
 - MERCHANT TIMESTAMP – Prodavčev timespamp transakcije (tip DateTime)
 - ERROR URL – URL web shop aplikacije na koji će Acquirer Wep App redirektovati korisnika ako se desi bilo kakva greška
3. Acquirer Wep App proverava da li je zahtev dobijen u koraku 2 ispravan i ako jeste generiše PAYMENT URL (tip URL – String(256)) i PAYMENT ID (tip Number(10)) i prosleđuje ih nazad Merchant Wep App.
4. Kada korisnik klikne pa potvrdu kupovine Merchant Wep App će redirektovati kupca na sajt za plaćanje (Acquirer Wep App) i to na adresu PAYMENT URL sa parametrom PAYMENT ID. Na ovom sajtu korisnik unosi PAN, security code, card holder name i datum do kada kartica važi.
5. Acquirer Wep App generiše ACQUIRER ORDER ID (bančin ID transakcije - tip Number(10)) i ACQUIRER TIMESTAMP. Ova dva podatka zajedno sa PAN, security code, card holder name, datum do kada kartica važi i iznosom transakcije se prosleđuju Acquirer-u (njegovom web servisu). Ovaj zahtev u stvari predstavlja zahtev za autentifikaciju kartice i autorizaciju transakcije.
6. Pristigli zahtev Acquirer prosleđuje PCC-u, takođe putem web servisa.
7. Pristigli zahtev PCC prosleđuje Issuer-u, takođe putem web servisa. PCC pronalazi Issuer-a preko PAN-a.
8. Issuer proverava pristigli zahtev i ako je ispravan i kupac ima dovoljno novca vrši se rezervacija sredstava. Issuer prosleđuje odgovor o rezultatu autentifikaciju kartice i autorizaciju transakcije nazad PCC-u (kao return vrednost poziva metode web servisa). Odgovor pored rezultata transakcije treba da sadrži i ACQUIRER ORDER ID, ACQUIRER TIMESTAMP, ISSUER ORDER ID (id transakcije issuer-a) i ISSUER TIMESTAMP podatke.
9. PCC pristigli odgovor prosleđuje Acquirer-u (kao return vrednost poziva metode web servisa).
10. Acquirer pristigli odgovor prosleđuje Acquirer Wep App (kao return vrednost poziva metode web servisa).

11. Acquirer Wep App poziva odgovarajuću metodu web servisa Merchant Wep App da bi javio rezultat transakcije. U tom pozivu pored rezultata transakcije treba da se prosledi i MERCHANT ORDER ID, ACQUIRER ORDER ID, ACQUIRER TIMESTAMP i PAYMENT ID.
12. Merchant Wep App će vratiti URL na koji korisnik treba da se redirektuje. Obično to bude jedan od dva URL-a: SUCCESS URL (ako je plaćanje uspešno) i FAILED URL (ako plaćanje nije uspešno). Acquirer Wep App će redirektovati korisnika na dobijeni URL. Ako je u komunikaciji došlo do greške i iz bilo kog razloga Merchant Wep App nema SUCCESS URL i FAILED URL onda korisnika treba da redirektuje na ERROR URL. Takođe, Merchant Wep App treba da pošalje korisniku mail o uspešnoj/neuspešnoj transakciji.

2. Web sajt za prodaju osiguranja

Međunarodno putno osiguranje podrazumeva prodaju putnog osiguranja korisnicima koji putuju u inostranstvo.

Sledeći podaci su potrebni za određivanje cene osiguranja:

1. Trajanje osiguranja
2. Region u koji se putuje (Evropa, Azija, Amerika, Japan,)
3. Broj osoba
4. Starost osoba: do 18 god., od 18-60 god., preko 60 god. starosti
5. Da li će se korisnik baviti sportom i kojim
6. Do kog iznosa je korisnik osiguran (npr. 10000eur, 30000eura; nekoliko fiksnih opcija)

Za svakog korisnika se unose sledeći podaci (ne utiču na cenu osiguranja):

1. Ime i prezime
2. JMBG
3. Broj pasoša
4. Adresa
5. Broj telefona

Jedan korisnik mora biti označen kao ugovarač osiguranja. Za njega je potrebno uneti i email adresu kako bi dobio poruku o uspešnoj/neuspešnoj kupovini osiguranja.

Osim putnog osiguranja korisniku se dodatno može ponuditi osiguranje stana/kuće dok je na putu. Na cenu osiguranja stana utiču sledeći elementi:

1. Trajanje osiguranja
2. Površina stana
3. Starost stana

4. Procenjene vrednosti stana
5. Od čega se osigurava (poplava, krađa, požar,)

Za stan se unose i sledeći podaci (koji ne utiču na cenu):

1. Adresa
2. Ime i prezime vlasnika
3. JMBG vlasnika

Osim putnog osiguranja korisniku se dodatno može ponuditi osiguranje pomoć na putu. Na cenu osiguranja pomoć na putu utiču:

1. Trajanje osiguranja
2. Paket koji se želi kupiti
 1. Šlepovanje do određene kilometraže
 2. Popravka do određene cene
 3. Smeštaj u hotelu do određenog broja dana
 4. Alternativni prevoz

Za vozilo se unose i sledeći podaci (koji ne utiču na cenu):

1. Marka i tip vozila
2. Godina proizvodnje
3. Broj registarske tablice
4. Broj šasije
5. Ime i prezime vlasnika
6. JMBG vlasnika

3. Formiranje cene osiguranja

Za formiranje cene osiguranja (kako koji podatak i na koji način utiče na ukupnu cenu osiguranja) potrebno je realizovati sistem baziran na pravilima. Cilj ovakvog sistema je da omogući relativno jednostavnu izmenu načina kako se formira cena osiguranja, bez izmena u programskom kodu. Za implementaciju pravila može se koristiti i neka od gotovih biblioteka (npr. JBoss Drools <https://www.drools.org/>)

4. Tehničke karakteristike servisa

4.1 Stil web servisa

Svi web servisi koji se realizuju u okviru ovog projektnog zadatka treba da se realizuju koristeći REST pristup.

4.2 Skladištenje podataka

Kao skladište podataka može se koristiti relacionala baza podataka ili NoSQL baza. Svaki učesnik treba da ima svoju odvojenu bazu.

5. Web aplikacije

Web aplikacije za online kupovinu i naplatu realizovati kao dve odvojene aplikacije – AngularJS frontend aplikaciju koja putem REST-a komunicira sa serverskom aplikacijom koja može biti Spring (Java), Node.js i Express (Javascript), Django (Python) ili ASP .NET aplikacija. AngularJS aplikacija treba da sadrži sve elemente pokrivene na vežbama: bower i gulpfile, strukturu projekta urađenu po trenutnim najboljim praksama, upotrebu popularnih Angular modula, frontend validaciju podataka i dobro napisane unit testove. Aplikacija treba da bude lokalizovana na srpski i engleski.

6. Bezbednost

Za bezbednu komunikaciju koristiti HTTPS. Cela aplikacija treba da bude otporna na napade definisane na OWASP Top 10 listi. Posebnu pažnju obratiti na A9, A8 i A2.

7. Kontrola verzija

Za timski rad je obavezno koristiti git (GitLab, GitHub) i da projekat bude javan. Projekat na repozitorijumu treba da bude održavan i dobro dokumentovan. Ovo uključuje:

- Formiranje smislenih commit-ova (jedan commit treba da bude jedna logička celina izmena, uz jasnu i konciznu poruku koja opisuje izmene u datom commit-u);
- Pisanje wiki stranica koje opisuju svrhu sistema i potrebne alate da bi se pokrenuli (i dalje razvijali) različiti delovi sistema;
- Koncizan i lepo formatiran README.md, sa svim relevantnim podacima, od autora do prikladne licence;
- Skripta (gulp task, gradle job, bash skripta, itd.) za „1 click deployment“;
- Release gotovog projekta, koji uključuje „dist“ uz skriptu za „1 click deployment“.

8. Propratna dokumentacija

Uz projekat je potrebno imati formiran model podataka celog sistema (prikazan kroz više dijagram), kao i DFD kreiran putem Microsoft TMT 2016 alata.

Bonus: Napraviti potpuno model pretnji sistema.

9. Test scenario

Testirati slučajeve kada je kupovina uspešna, kada je kartica nevalidna, kada nema dovoljno novca.

Frontend aplikaciju istestirati pisanjem prikladnih unit testova.

Bonus: Napisati unit testove za svaki od servisa.

ALTERNATIVA 1. PROJETKNOG ZADATKA

Plaćanje realizovati preko PayPal sistema.

ALTERNATIVA 2. PROJETNOG ZADATKA

Aplikaciju realizovati kao mobilnu (Android/iOS) aplikaciju i plaćanje preko PayPal mobile sistem.