

UNIVERZITET U BEOGRADU
FAKULTET ORGANIZACIONIH NAUKA
LABORATORIJA ZA SOFTVERSKO INŽENJERSTVO



PROJEKTOVANJE SOFTVERA

Projektni rad

Softverski sistem za prodaju video igara

Profesor:

Prof.dr. Siniša Vlajić

Student:

Pavle Dujkić 172/16

Beograd, 2020.

Sadržaj:

1. Korisnički zahtevi	4
1.1 Verbalni opis	4
1.2 Slučajevi korišćenja	5
1.2.1 Slučaj korišćenja 1: Unos video igre.....	6
1.2.2 Slučaj korišćenja 2: Izmena video igre	6
1.2.3 Slučaj korišćenja 3: Brisanje video igre	7
1.2.4 Slučaj korišćenja 4: Pretraga video igara	8
1.2.5 Slučaj korišćenja 5: Kreiranje računa	8
1.2.6 Slučaj korišćenja 6: Izmena računa.....	9
1.2.7 Slučaj korišćenja 7: Storniranje računa.....	10
1.2.8 Slučaj korišćenja 8: Pretraga računa	11
2. Analiza	12
2.1. Ponašanje softverskog sistema – Sistemski dijagrami sekvenci	12
2.1.1. Dijagram sekvenci slučaja korišćenja 1: Unos video igre	12
2.1.2. Dijagram sekvenci slučaja korišćenja 2: Izmena video igre	13
2.1.3. Dijagram sekvenci slučaja korišćenja 3: Brisanje video igre	15
2.1.4. Dijagram sekvenci slučaja korišćenja 4: Pretraga video igara	17
2.1.5. Dijagram sekvenci slučaja korišćenja 5: Kreiranje računa	19
2.1.5. Dijagram sekvenci slučaja korišćenja 6: Izmena računa	21
2.1.7. Dijagram sekvenci slučaja korišćenja 7: Storniranje računa.....	24
2.1.8. Dijagram sekvenci slučaja korišćenja 8: Pretraga računa	26
2.1.9. Rezultat analize sistemskog dijagrama sekvenci.....	28
2.2 Ponašanje softverskog sistema – Definisanje ugovora o sistemskim operacijama	28
2.2.1. Ugovor UG1 : Unos video igre.....	28
2.2.2. Ugovor UG2 : Učitavanje liste platformi.....	29
2.2.3. Ugovor UG3 : Pretraga video igara	29
2.2.4. Ugovor UG4 : Izmena video igre	29
2.2.5. Ugovor UG5 : Prikaz video igre	29
2.2.6. Ugovor UG6 : Brisanje video igre	29
2.2.7. Ugovor UG7 : Obrada računa.....	30
2.2.8. Ugovor UG8 : Čuvanje računa	30
2.2.9. Ugovor UG9 : Učitavanje liste video igara	30
2.2.10. Ugovor UG10 : Pretraga računa.....	30
2.2.11. Ugovor UG11 : Prikaz računa.....	31

2.2.12. Ugovor UG12 : Izmena računa	31
2.2.13. Ugovor UG13 : Storniranje računa	31
2.3. Struktura softverskog sistema – Konceptualni model.....	32
2.4. Struktura softverskog sistema – Relacioni model.....	32
3. Projektovanje	35
3.1. Arhitektura softverskog sistema	35
3.2. Projektovanje korisničkog interfejsa	36
3.3. Projektovanje ekranskih formi	36
3.3.1. Slučaj korišćenja 1: Unos video igre	36
3.3.2. Slučaj korišćenja 2: Izmena video igre	38
3.3.3. Slučaj korišćenja 3: Brisanje video igre	41
3.3.4. Slučaj korišćenja 4: Pretraga video igara	44
3.3.5. Slučaj korišćenja 5: Kreiranje računa	46
3.3.6. Slučaj korišćenja 6: Izmena računa	48
3.3.7. Slučaj korišćenja 7: Storniranje računa	52
3.3.8. Slučaj korišćenja 8: Pretraga računa	55
3.4. Projektovanje kontrolera korisničkog interfejsa	59
3.5. Projektovanje aplikacione logike	60
3.5.1. Kontroler aplikacione logike.....	60
3.5.2. Sistemske operacije	63
3.5.3. Domenske klase	72
3.5.4. Broker baze podataka	73
3.5.5. Projektovanje skladišta podataka	74
4. Implementacija.....	78
5. Testiranje	79
6. Literatura.....	80

1. Korisnički zahtevi

1.1 Verbalni opis

Potrebno je napraviti softverski sistem za podršku rada prodavnice video igara. Kupac dolazi u prodavnicu i bira konkretnu video igru iz liste video igara koje prodavnica nudi. Prodavac izdaje konkretnu video igru koju je kupac tražio. Takođe, prodavac evidentira račun za određenu video igru koju je kupac tražio.

Prodavnica vodi evidenciju svojih video igara kao i računa. Video igre karakteriše njihov id, naslov, opis i cena na osnovu koje se računa ukupan iznos koji se naplaćuje kupcu. Za video igru se vezuje i platforma sa kojom je data video igra kompatibilna. Platformu karakteriše id i naziv platforme. Evidencija računa sadrži video igre u obliku svojih stavki gde svaka stavka računa sadrži redni broj, količinu i iznos stavke. Kada se pomnoži cena date video igre i kolicina, dobija se iznos za naplatu. Račun karakteriše id, datum računa i iznos za naplatu. Za svaki račun se vezuje i prodavac koji ga je evidentirao. Prodavca karakteriše id, ime i prezime.

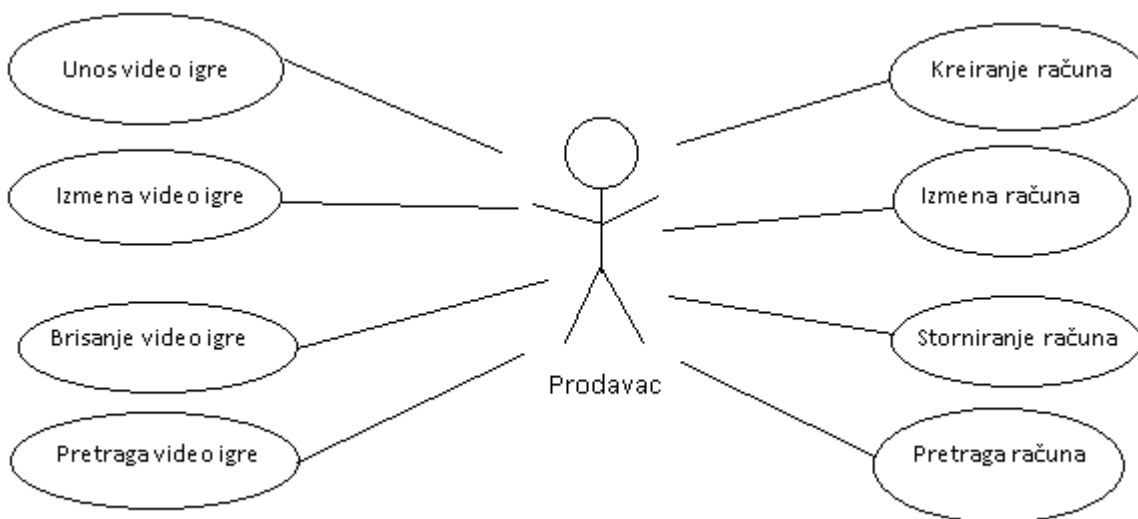
1.2 Slučajevi korišćenja

Model slučajeva korišćenja se sastoji od skupa slučajeva korišćenja, aktora i veza između slučajeva korišćenja i aktora. Slučaj korišćenja opisuje skup scenarija, odnosno skup željenih korišćenja sistema od strane aktora. Iz toga proizilazi da scenario opisuje jedno željeno korišćenje sistema od strane aktora. Scenario je opisan preko: sekvence akcija i interakcija između aktora i sistema. Slučaj korišćenja se sastoji iz glavnog i alternativnih scenarija. Scenario definišu željene fukcije sistema koje, kada se izvršavaju, pozivaju po određenom redosledu osnovne funkcije sistema.

U konkretnom slučaju identifikovani su sledeći slučajevi korišćenja:

- 1) Unos video igre
- 2) Izmena video igre
- 3) Brisanje video igre
- 4) Pretraga video igre
- 5) Kreiranje računa (*složen SK*)
- 6) Izmena računa (*složen SK*)
- 7) Storniranje računa
- 8) Pretraga računa

Dijagram slučajeva korišćenja



Slika 1 - Dijagram slučajeva korišćenja

1.2.1 Slučaj korišćenja 1: Unos video igre

Naziv SK: Unos **video igre**

Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **video igrom**. Učitane su **platforme**.

Osnovni scenario SK:

1. **Prodavac** unosi podatke o **video igri**. (*APUSO*)
2. **Prodavac** kontroliše da li je ispravno uneo podatke o **video igri**. (*ANSO*)
3. **Prodavac** poziva sistem da unese **video igru**. (*APSO*)
4. **Sistem** unosi **video igru**. (*SO*)
5. **Sistem** obaveštava prodavca da je uspešno uneo **video igru**: "Sistem je zapamtio video igru!". (*IA*)

Alternativna scenarija SK:

- 5.1 Ukoliko sistem ne može da unese **video igru**, obaveštava prodavca o tome: "Sistem ne može da zapamti video igru!". (*IA*)

1.2.2 Slučaj korišćenja 2: Izmena video igre

Naziv SK: Izmena **video igre**

Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **video igrom**. Učitane su **platforme**.

Osnovni scenario SK:

1. **Prodavac** unosi vrednost po kojoj vrši pretragu **video igara**. (*APUSO*)
2. **Prodavac** poziva sistem da izvrši pretragu **video igara** po zadatoj vrednosti. (*APSO*)
3. **Sistem** izvršava pretragu **video igara** po zadatoj vrednosti. (*SO*)
4. **Sistem** prikazuje prodavcu podatke o **video igrama** i poruku: "Sistem je uspesno ucitao video igre!". (*IA*)
5. **Prodavac** bira **video igru** koju želi da izmeni. (*APUSO*)
6. **Prodavac** poziva sistem da uči **video igru**. (*APSO*)

7. **Sistem** pronalazi **video igru**. (SO)
8. **Sistem** prikazuje podatke o **video igri** i poruku: "Sistem je uspesno učitao video igru!". (IA)
9. **Prodavac** unosi (menja) podatke o **video igri**. (APUSO)
10. **Prodavac** kontroliše da li je ispravno izmenio podatke o **video igri**. (ANSO)
11. **Prodavac** poziva sistem da izmeni **video igru**. (APSO)
12. **Sistem** menja **video igru**. (SO)
13. **Sistem** obaveštava prodavca da je uspešno izmenio **video igru**: "Sistem je zapamtio video igru!". (IA)

Alternativna scenarija SK:

- 4.1 Ukoliko sistem ne može da pronađe **video igre**, obaveštava prodavca o tome: "Nema video igre za uneti kriterijum!". Prekida se izvršavanje scenarija. (IA)
- 8.1 Ukoliko sistem ne može da prikaže konkretnu **video igru**, obaveštava prodavca o tome: "Sistem ne može da učitao video igru!". Prekida se izvršavanje scenarija. (IA)
- 13.1 Ukoliko sistem ne može da izmeni **video igru**, obaveštava prodavca o tome: "Sistem ne može da zapamti video igru!". (IA)

1.2.3 Slučaj korišćenja 3: Brisanje video igre

Naziv SK: Brisanje **video igre**

Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **video igrom**. Učitane su **platforme**.

Osnovni scenario SK:

1. **Prodavac** unosi vrednost po kojoj vrši pretragu **video igara**. (APUSO)
2. **Prodavac** poziva sistem da izvrši pretragu **video igara** po zadatoj vrednosti. (APSO)
3. **Sistem** izvršava pretragu **video igara** po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje prodavcu podatke o **video igrima** i poruku: "Sistem je uspesno učitao video igre!". (IA)
5. **Prodavac** bira **video igru** koji želi da obriše. (APUSO)
6. **Prodavac** poziva sistem da obriše **video igru**. (APSO)
7. **Sistem** briše **video igru**. (SO)

8. **Sistem** obaveštava prodavca da je uspešno obrisao **video igru**: “Sistem je obrisao video igru!”. (IA)

Alternativna scenarija SK:

- 4.1 Ukoliko sistem ne može da pronađe **video igru**, obaveštava prodavca o tome: “Nema video igre za uneti kriterijum!”. Prekida se izvršavanje scenarija. (IA)
- 8.1 Ukoliko sistem ne može da obriše **video igru**, obaveštava prodavca o tome: “Sistem ne može da obriše video igru!”. (IA)

1.2.4 Slučaj korišćenja 4: Pretraga video igara

Naziv SK: Pretraga **video igara**

Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **video igrom**. Učitane su **platforme**.

Osnovni scenario SK:

1. **Prodavac** unosi vrednost po kojoj vrši pretragu **video igara**. (APUSO)
2. **Prodavac** poziva sistem da izvrši pretragu **video igara** po zadatoj vrednosti. (APSO)
3. **Sistem** izvršava pretragu **video igara** po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje prodavcu podatke o **video igrama** i poruku: “Sistem je uspesno učitao video igre!”. (IA)
5. **Prodavac** bira **video igru** čije podatke želi da prikaže. (APUSO)
6. **Prodavac** poziva sistem da prikaže podatke o **video igri**. (APSO)
7. **Sistem** traži podatke o **video igri**. (SO)
8. **Sistem** prikazuje podatke o izabranoj **video igri** i poruku: “Sistem je uspesno učitao video igru!”. (IA)

Alternativna scenarija SK:

- 4.1 Ukoliko sistem ne može da pronađe **video igre**, obaveštava prodavca o tome: “Nema video igre za uneti kriterijum!”. Prekida se izvršavanje scenarija. (IA)
- 8.1 Ukoliko sistem ne može da prikaže podatke o **video igri** on prikazuje radniku poruku: “Sistem ne može da učitao video igru!”. (IA)

1.2.5 Slučaj korišćenja 5: Kreiranje računa

Naziv SK: Kreiranje **računa**

Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **računom**. Učitane su **video igre**.

Osnovni scenario SK:

1. **Prodavac** unosi podatke o **računu**. (APUSO)
2. **Prodavac** kontroliše da li je ispravno uneo podatke o **računu**. (ANSO)
3. **Prodavac** poziva sistem da zapamti podatke o **računu**. (APSO)
4. **Sistem** pamti podatke o **računu**. (SO)
5. **Sistem** obaveštava prodavca da je uspešno zapamćen **račun** i poruku: "Sistem je zapamtio prodaju!". (IA)
6. **Prodavac** poziva sistem da obradi **račun**. (APSO)
7. **Sistem** obrađuje **račun**. (SO)
8. **Sistem** prikazuje radniku obrađen **račun** i poruku: "Sistem je zapamtio prodaju!". (IA)

Alternativna scenarija SK:

- 5.1 Ukoliko sistem ne može da unese **račun**, obaveštava prodavca o tome: "Sistem ne može da zapamti prodaju!". Prekida se izvršavanje scenarija. (IA)
- 8.1 Ukoliko sistem ne može da obradi **račun** on prikazuje radniku poruku: "Sistem ne može da zapamti prodaju!". (IA)

1.2.6 Slučaj korišćenja 6: Izmena računa

Naziv SK: Izmena **računa**

Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **računom**. Učitane su **video igre**.

Osnovni scenario SK:

1. **Prodavac** unosi vrednost po kojoj vrši pretragu **računa**. (APUSO)
2. **Prodavac** poziva sistem da izvrši pretragu **računa** po zadatoj vrednosti. (APSO)
3. **Sistem** izvršava pretragu **računa** po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje prodavcu podatke o **računima** i poruku: "Sistem je pronasao prodaje za izabrani datum prodaje!". (IA)

5. **Prodavac** bira **račun** koji želi da izmeni. (APUSO)
6. **Prodavac** poziva sistem da učitá **račun**. (APSO)
7. **Sistem** pronalazi **račun**. (SO)
8. **Sistem** prikazuje podatke o **računu** i poruku: "Sistem je uspesno učitao prodaju!". (IA)
9. **Prodavac** unosi (menja) podatke o **računu**. (APUSO)
10. **Prodavac** kontroliše da li je ispravno izmenio podatke o **računu**. (ANSO)
11. **Prodavac** poziva sistem da izmeni **račun**. (APSO)
12. **Sistem** menja **račun**. (SO)
13. **Sistem** obaveštava prodavca da je uspešno izmenio **račun**: "Sistem je zapamtio prodaju!". (IA)

Alternativna scenarija SK:

- 4.1 Ukoliko sistem ne može da pronađe **račune**, obaveštava prodavca o tome: "Nema prodaje za izabrani datum prodaje!". Prekida se izvršavanje scenarija. (IA)
- 8.1 Ukoliko sistem ne može da prikaže konkretni **račun**, obaveštava prodavca o tome: "Sistem ne može da učitá prodaju!". Prekida se izvršavanje scenarija. (IA)
- 13.1 Ukoliko sistem ne može da izmeni **račun**, obaveštava prodavca o tome: "Sistem ne može da zapamti prodaju!". (IA)

1.2.7 Slučaj korišćenja 7: Storniranje računa

Naziv SK: Storniranje **računa**

Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **računom**. Učitane su **video igre**.

Osnovni scenario SK:

1. **Prodavac** unosi vrednost po kojoj vrši pretragu **računa**. (APUSO)
2. **Prodavac** poziva sistem da izvrši pretragu **računa** po zadatoj vrednosti. (APSO)
3. **Sistem** izvršava pretragu **računa** po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje prodavcu podatke o **računima** i poruku: "Sistem je pronasao prodaje za izabrani datum prodaje!". (IA)
5. **Prodavac** bira **račun** čije podatke želi da prikaže. (APUSO)

6. **Prodavac** poziva sistem da prikaže podatke o izabranom **računu**. (APSO)
7. **Sistem** traži podatke o izabranom **računu**. (SO)
8. **Sistem** prikazuje podatke o izabranom **računu**. (IA)
9. **Radnik** poziva sistem da stornira zadati **račun**. (APSO)
10. **Sistem** stornira **račun**. (SO)
11. **Sistem** prikazuje radniku obrisani **račun** i poruku: "Sistem je obrisao prodaju!". (IA)

Alternativna scenarija SK:

- 4.1 Ukoliko sistem ne može da pronađe **račune**, obaveštava prodavca o tome: "Nema prodaje za izabrani datum prodaje!". Prekida se izvršavanje scenarija. (IA)
- 8.1 Ukoliko sistem ne može da prikaže podatke o **računu**, obaveštava prodavca o tome: "Sistem ne može da učitao prodaju!". Prekida se izvršenje scenarija. (IA)
- 11.1 Ukoliko sistem ne može da obriše **račun**, obaveštava prodavca o tome: "Sistem ne može da obriše prodaju!". (IA)

1.2.8 Slučaj korišćenja 8: Pretraga računa

Naziv SK: Pretraga **računa**

Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **računom**. Učitane su **video igre**.

Osnovni scenario SK:

1. **Prodavac** unosi vrednost po kojoj vrši pretragu **računa**. (APUSO)
2. **Prodavac** poziva sistem da izvrši pretragu **računa** po zadatoj vrednosti. (APSO)
3. **Sistem** izvršava pretragu **računa** po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje prodavcu podatke o **računima** i poruku: "Sistem je pronasao prodaje za izabrani datum prodaje!". (IA)
5. **Prodavac** bira **račun** čije podatke želi da prikaže. (APUSO)
6. **Prodavac** poziva sistem da prikaže podatke o izabranom **računu**. (APSO)
7. **Sistem** traži podatke o traženom **računu**. (SO)
8. **Sistem** prikazuje podatke o traženom **računu** i poruku: "Sistem je uspešno učitao prodaju!". (IA)

Alternativna scenarija SK:

- 4.1 Ukoliko sistem ne može da pronađe **račune**, obaveštava prodavca o tome: “Nema prodaje za izabrani datum prodaje!”. (IA)
- 8.1 Ukoliko sistem ne može da pronađe podatke o željenom **računu**, obaveštava prodavca o tome: “Sistem ne može da učitaj prodaju!”. (IA)

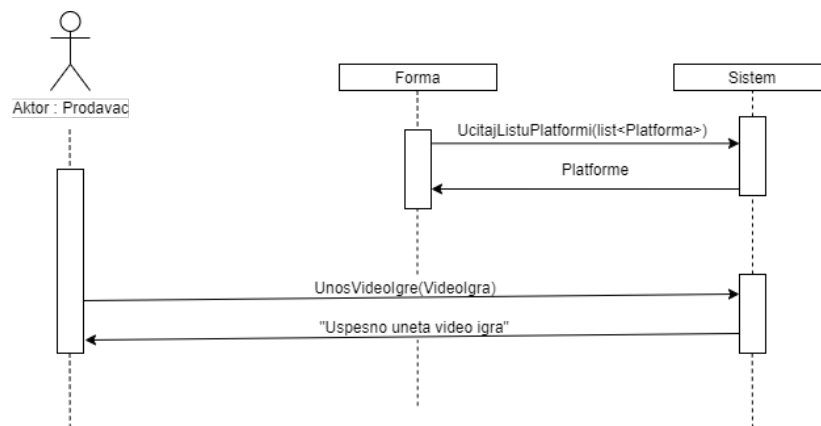
2. Analiza

2.1. Ponašanje softverskog sistema – Sistemski dijagrami sekvenci

2.1.1. Dijagram sekvenci slučaja korišćenja 1: Unos video igre

Osnovni scenario SK:

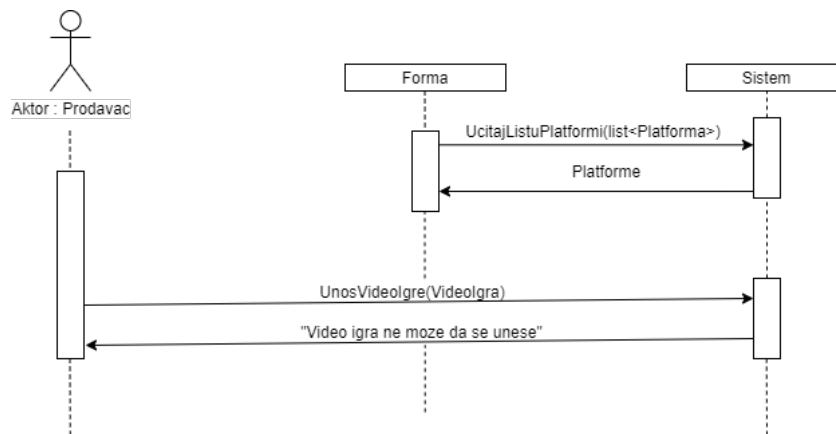
- 1. **Forma** poziva sistem da učitaj listu **platformi**. (APSO)
- 2. **Sistem** vraća formi listu **platformi**. (IA)
- 3. **Prodavac** poziva sistem da unese **video igru**. (APSO)
- 4. **Sistem** obaveštava prodavca da je uspešno uneo **video igru**: “Sistem je zapamtio video igru!”. (IA)



Slika 2 - Osnovni scenario DS1 - Unos video igre

Alternativna scenarija SK:

- 4.1. Ukoliko sistem ne može da unese **video igru**, obaveštava prodavca o tome: “ Sistem ne može da zapamti video igru! “. (IA)



Slika 3 - Alternativni scenario DS1 - Unos video igre

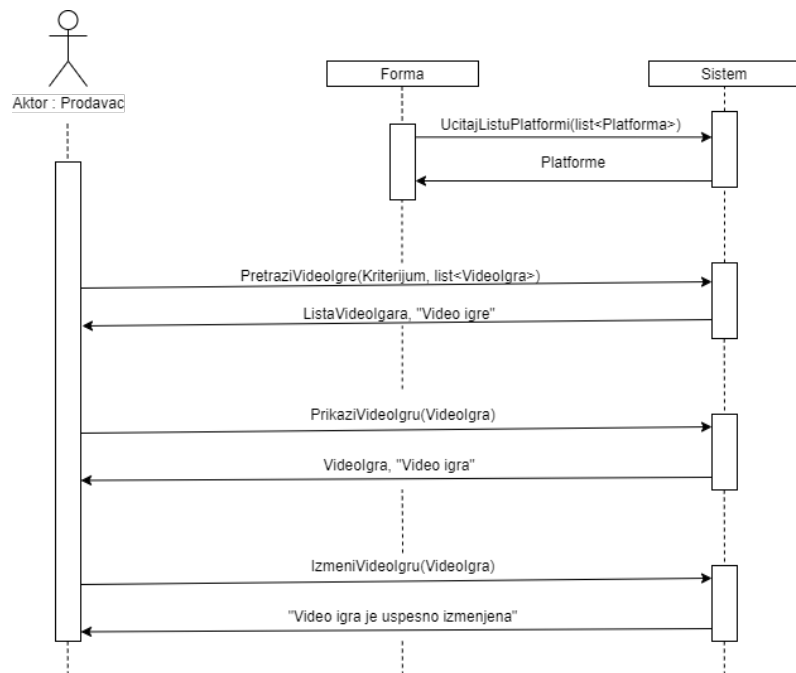
Sa navedenih dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. Signal **UnosVideolgre(Videolgra)**;
2. Signal **UcitajListuPlatformi(list <Platforma>)**.

2.1.2. Dijagram sekvenci slučaja korišćenja 2: Izmena video igre

Osnovni scenario SK:

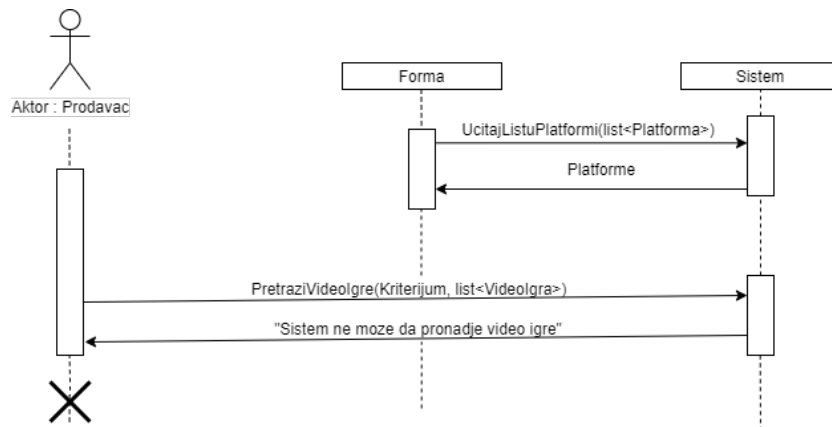
1. **Forma** poziva sistem da učitá listu **platformi**. (APSO)
2. **Sistem** vraća formi listu **platformi**. (IA)
3. **Prodavac** poziva sistem da izvrši pretragu **video igara** po zadatoj vrednosti. (APSO)
4. **Sistem** prikazuje prodavcu podatke o **video igrama** i poruku: "Sistem je uspesno ucitao video igre!". (IA)
5. **Prodavac** poziva sistem da učitá **video igru**. (APSO)
6. **Sistem** prikazuje podatke o **video igri** i poruku: "Sistem je uspesno ucitao video igru!". (IA)
7. **Prodavac** poziva sistem da izmeni **video igru**. (APSO)
8. **Sistem** obaveštava prodavca da je uspešno izmenio **video igru**: "Sistem je zapamtio video igru!". (IA)



Slika 4 - Osnovni scenario DS2 - Izmena video igre

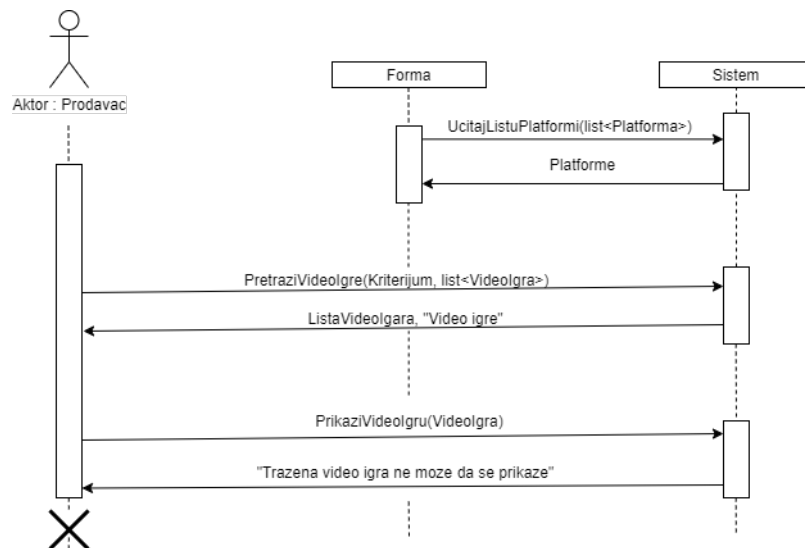
Alternativna scenarija SK:

4.1. Ukoliko sistem ne može da pronađe **video igre**, obaveštava prodavca o tome: “Nema video igre za uneti kriterijum!”. Prekida se izvršavanje scenarija. (IA)



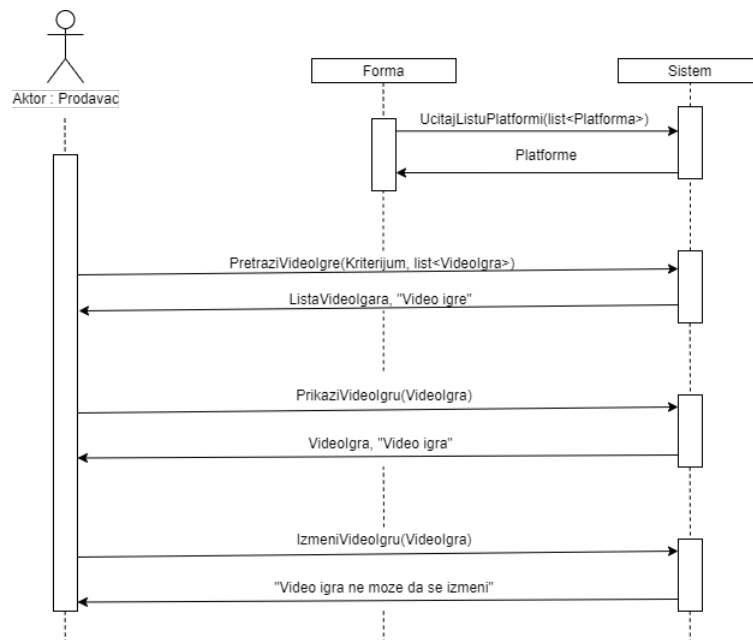
Slika 5 - Alternativni scenario DS2 - Izmena video igre

6.1. Ukoliko sistem ne može da prikaže konkretnu **video igru**, obaveštava prodavca o tome: “Sistem ne moze da ucita video igru!”. Prekida se izvršavanje scenarija. (IA)



Slika 6 - Alternativni scenarij DS2 - Izmena video igre

8.1. Ukoliko sistem ne može da izmeni **video igru**, obaveštava prodavca o tome: “Sistem ne moze da zapamti video igru!”. (IA)



Slika 7 - Alternativni scenarij DS2 - Izmena video igre

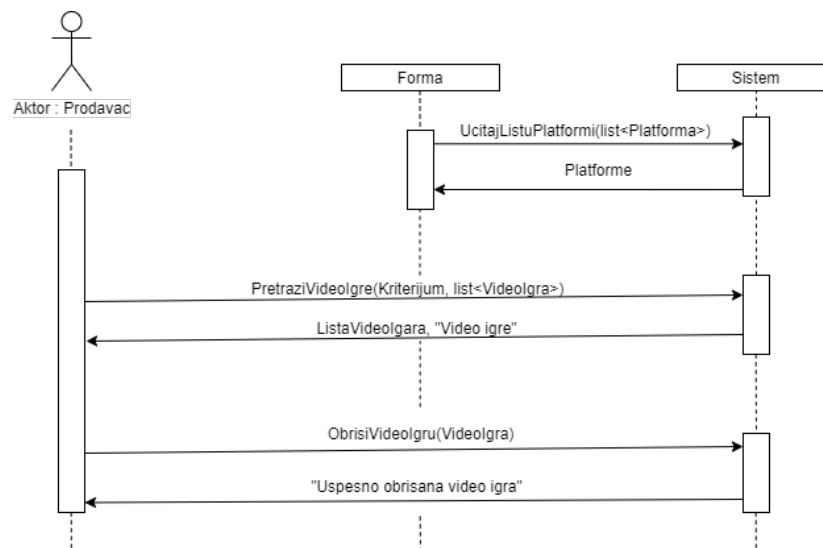
Sa navedenih dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. Signal **PretraziVideolgre**(Kriterijum, list <Videolgra>);
2. Signal **UcitajListuPlatformi**(list <Platforma>);
3. Signal **IzmeniVideolgru**(Videolgra);
4. Signal **PrikaziVideolgru**(Videolgra).

2.1.3. Dijagram sekvenci slučaja korišćenja 3: Brisanje video igre

Osnovni scenarij SK:

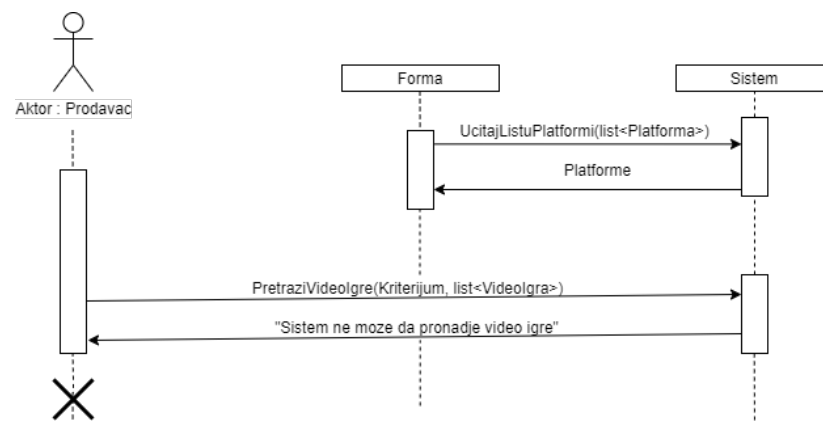
1. **Forma** poziva sistem da učitá listu **platformi**. (APSO)
2. **Sistem** vraća formi listu **platformi**. (IA)
3. **Prodavac** poziva sistem da izvrši pretragu **video igara** po zadatoj vrednosti. (APSO)
4. **Sistem** prikazuje prodavcu podatke o **video igrama** i poruku: “Sistem je uspesno ucitao video igre!”. (IA)
5. **Prodavac** poziva sistem da obriše **video igru**. (APSO)
6. **Sistem** obaveštava prodavca da je uspešno obrisao **video igru**: “Sistem je obrisao video igru!”. (IA)



Slika 8 - Osnovni scenario DS2 - Brisanje video igre

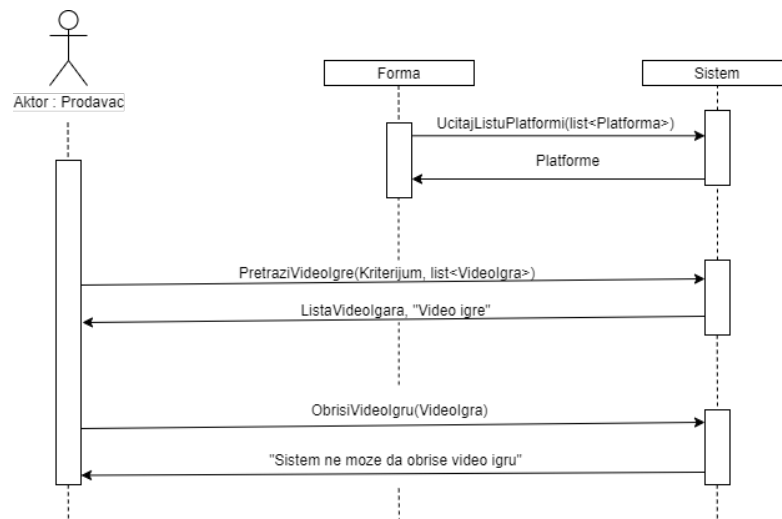
Alternativna scenarija SK:

4.1. Ukoliko sistem ne može da pronađe **video igru**, obaveštava prodavca o tome: “Nema video igre za uneti kriterijum!”. Prekida se izvršavanje scenarija. (IA)



Slika 9 - Alternativni scenario DS2 - Brisanje video igre

6.1. Ukoliko sistem ne može da obriše **video igru**, obaveštava prodavca o tome: “Sistem ne moze da obrise video igru!”. (IA)



Slika 10 - Alternativni scenario DS2 - Brisanje video igre

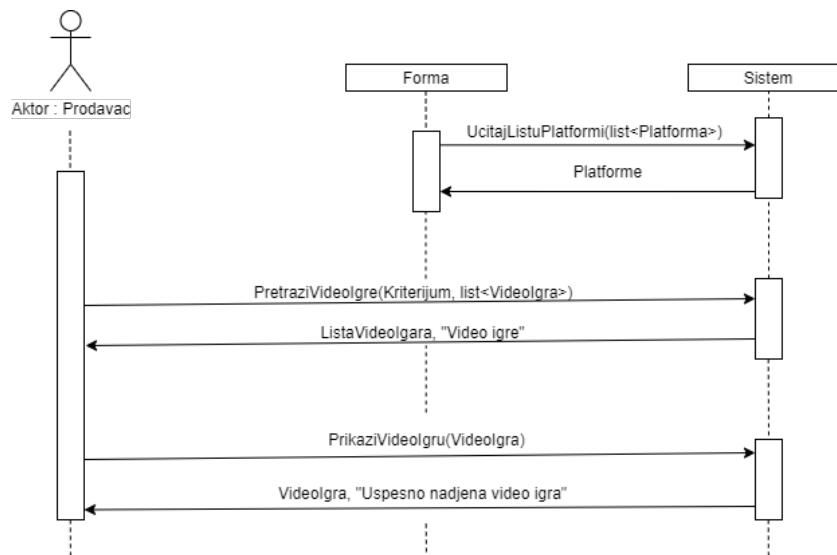
Sa navedenih dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. Signal **PretraziVideolgre**(Kriterijum, list <Videolgra>);
2. Signal **UcitajListuPlatformi**(list <Platforma>);
3. Signal **ObrisiVideolgru**(Videolgra).

2.1.4. Dijagram sekvenci slučaja korišćenja 4: Pretraga video igara

Osnovni scenario SK:

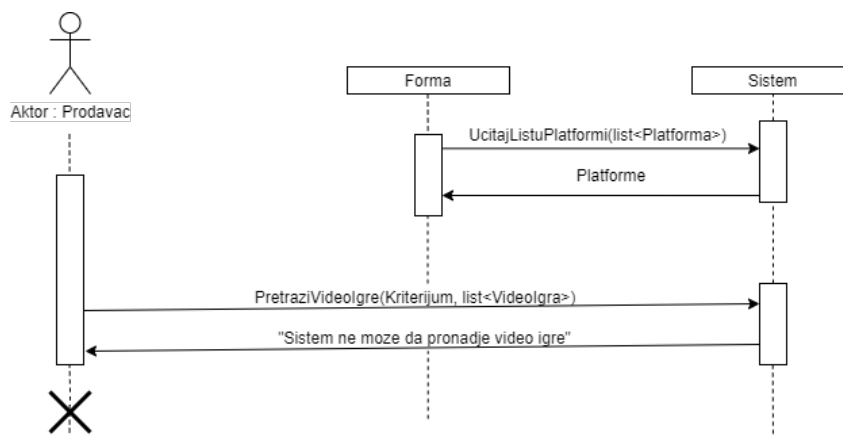
1. **Forma** poziva sistem da učitá listu **platformi**. (APSO)
2. **Sistem** vraća formi listu **platformi**. (IA)
3. **Prodavac** poziva sistem da izvrši pretragu **video igara** po zadatoj vrednosti. (APSO)
4. **Sistem** prikazuje prodavcu podatke o **video igrama** i poruku: "Sistem je uspesno učitao video igre!". (IA)
5. **Prodavac** poziva sistem da prikaže podatke o **video igri**. (APSO)
6. **Sistem** prikazuje podatke o izabranoj **video igri** i poruku: "Sistem je uspesno učitao video igru!". (IA)



Slika 11 - Osnovni scenario DS4 - Pretraga video igara

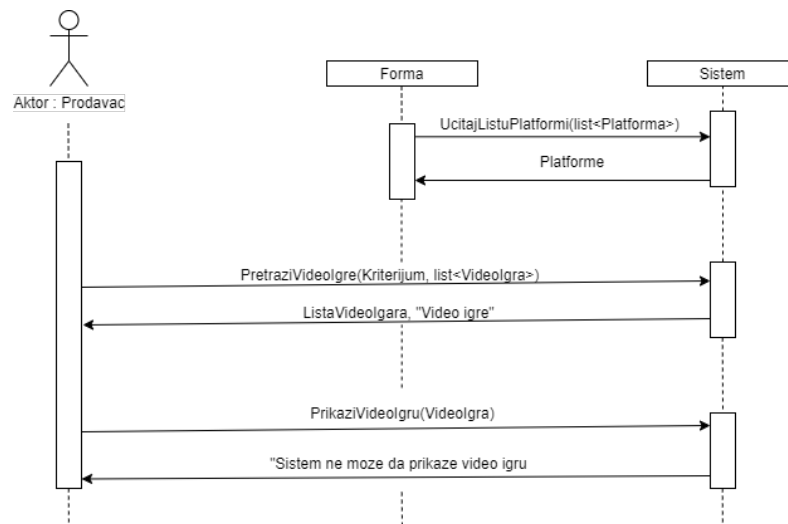
Alternativna scenarija SK:

2.1. Ukoliko sistem ne može da pronađe **video igre**, obaveštava prodavca o tome: “Nema video igre za uneti kriterijum!”. Prekida se izvršavanje scenarija. (IA)



Slika 12 - Alternativni scenario DS4 - Pretraga video igara

4.1. Ukoliko sistem ne može da prikaže podatke o **video igri** on prikazuje radniku poruku: “Sistem ne moze da ucita video igru!”. (IA)



Slika 13 - Alternativni scenario DS4 - Pretraga video igara

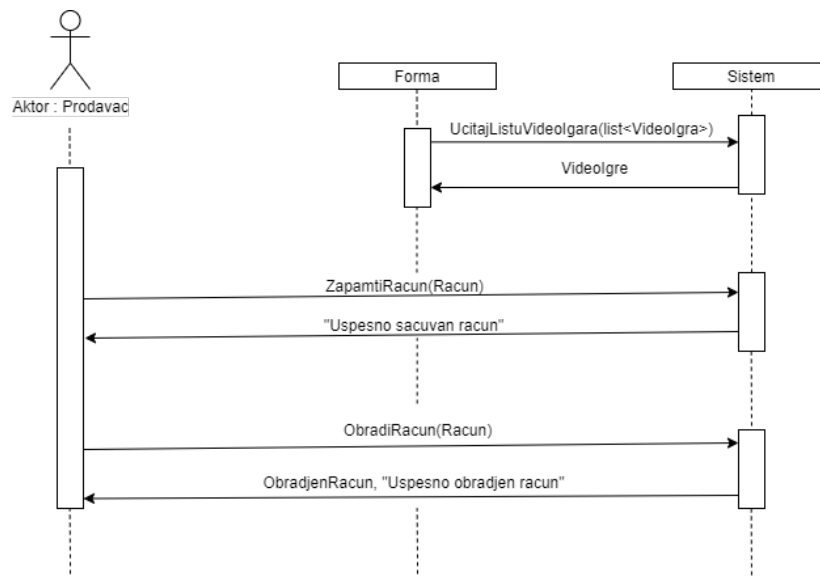
Sa navedenih dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. Signal **PretraziVideolgre**(Kriterijum, list <Videolgra>);
2. Signal **PrikaziVideolgru**(videolgra);
3. Signal **UcitajListu**(list <Platforma>).

2.1.5. Dijagram sekvenci slučaja korišćenja 5: Kreiranje računa

Osnovni scenario SK:

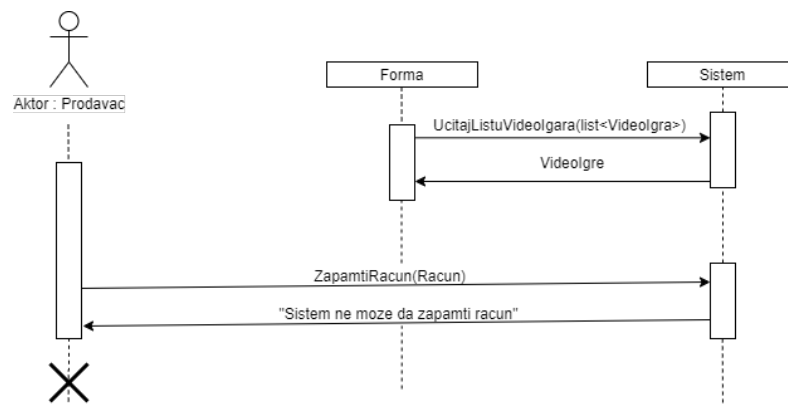
1. **Forma** poziva sistem da učitá listu **video igara**. (APSO)
2. **Sistem** vraća formi listu **video igara**. (IA)
3. **Prodavac** poziva sistem da zapamti podatke o **računu**. (APSO)
4. **Sistem** obaveštava prodavca da je uspešno zapamćen **račun** i poruku: "Sistem je zapamtio prodaju!". (IA)
5. **Prodavac** poziva sistem da obradi **račun**. (APSO)
6. **Sistem** prikazuje radniku obrađen **račun** i poruku: "Sistem je zapamtio prodaju!". (IA)



Slika 14 - Osnovni scenario DS5 - Kreiranje računa

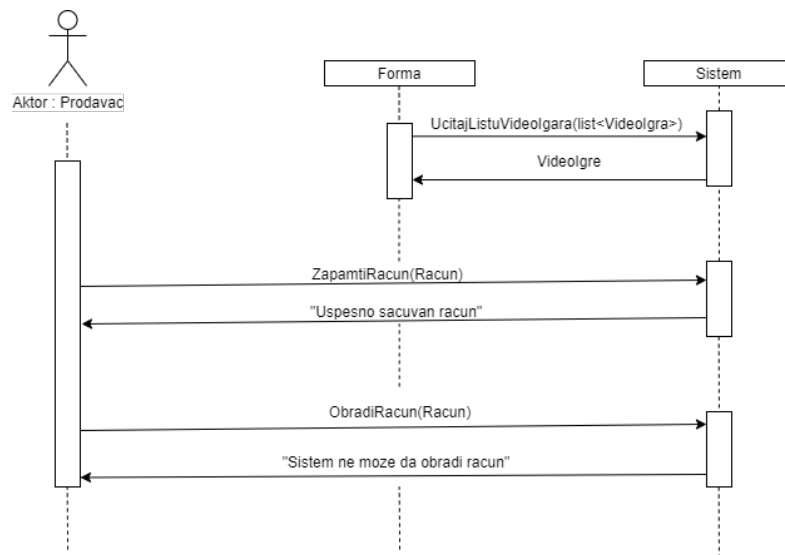
Alternativna scenarija SK:

4.1. Ukoliko sistem ne može da sačuva **račun**, obaveštava prodavca o tome: "Sistem ne moze da zapamti prodaju!". Prekida se izvršavanje scenarija. (IA)



Slika 15 - Alternativni scenario DS5 - Kreiranje računa

6.1. Ukoliko sistem ne može da obradi **račun** on prikazuje radniku poruku: "Sistem ne moze da zapamti prodaju!". (IA)



Slika 16 - Alternativni scenario DS5 - Kreiranje računa

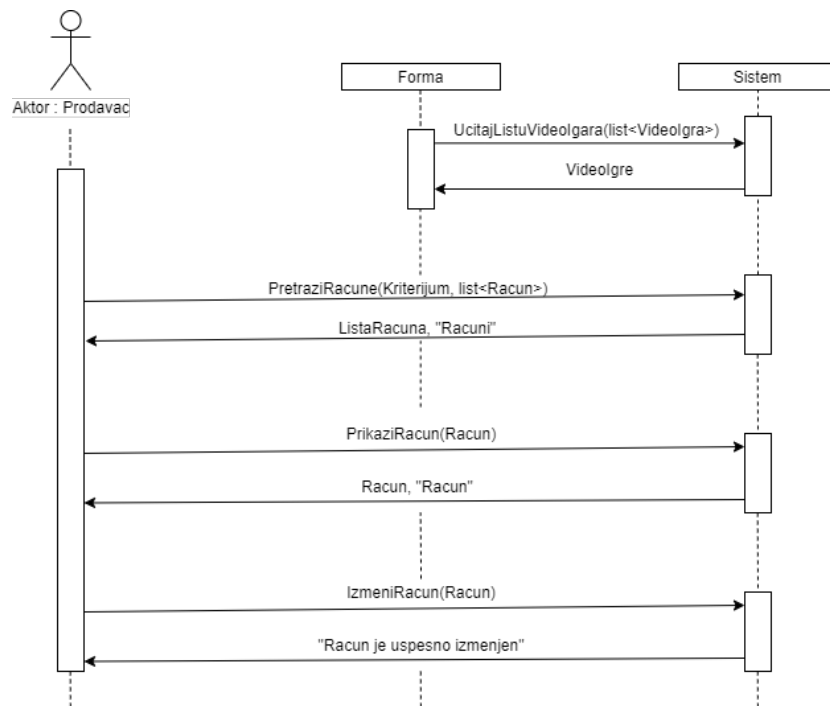
Sa navedenih dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. Signal **ObradiRacun(Racun)**;
2. Signal **ZapamtiRacun(Racun)**;
3. Signal **UcitajListuVideolgara(list <Videolgra>)**.

2.1.5. Dijagram sekvenci slučaja korišćenja 6: Izmena računa

Osnovni scenario SK:

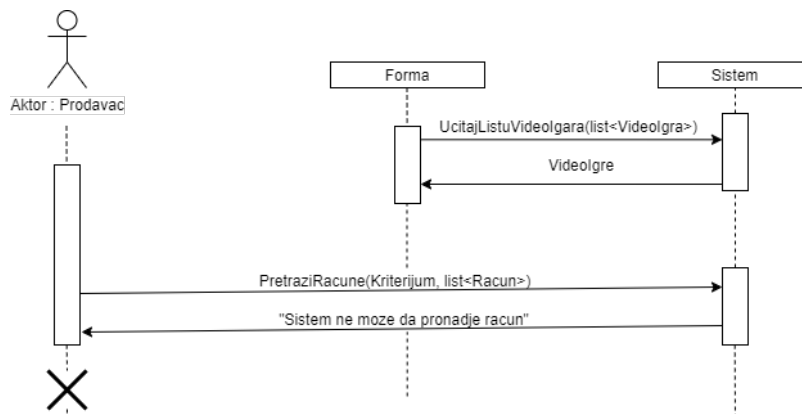
1. **Forma** poziva sistem da učitá listu **video igara**. (APSO)
2. **Sistem** vraća formi listu **video igara**. (IA)
3. **Prodavac** poziva sistem da izvrši pretragu **računa** po zadatoj vrednosti. (APSO)
4. **Sistem** prikazuje prodavcu podatke o **računima** i poruku: "Sistem je pronasao prodaje za izabrani datum prodaje!". (IA)
5. **Prodavac** poziva sistem da učitá **račun**. (APSO)
6. **Sistem** prikazuje podatke o **računu** i poruku: "Sistem je uspesno učitao prodaju!". (IA)
7. **Prodavac** poziva sistem da izmeni **račun**. (APSO)
8. **Sistem** obaveštava prodavca da je uspešno izmenio **račun**: "Sistem je zapamtio prodaju!". (IA)



Slika 17 - Osnovni scenario DS6 - Izmena računa

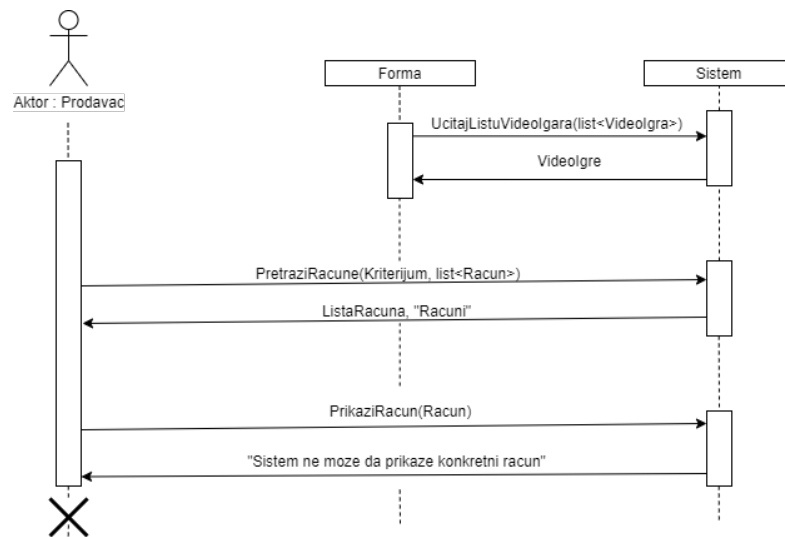
Alternativna scenarija SK:

4.1. Ukoliko sistem ne može da pronađe **račune**, obaveštava prodavca o tome: "Nema prodaje za izabrani datum prodaje!". Prekida se izvršavanje scenarija. (IA)



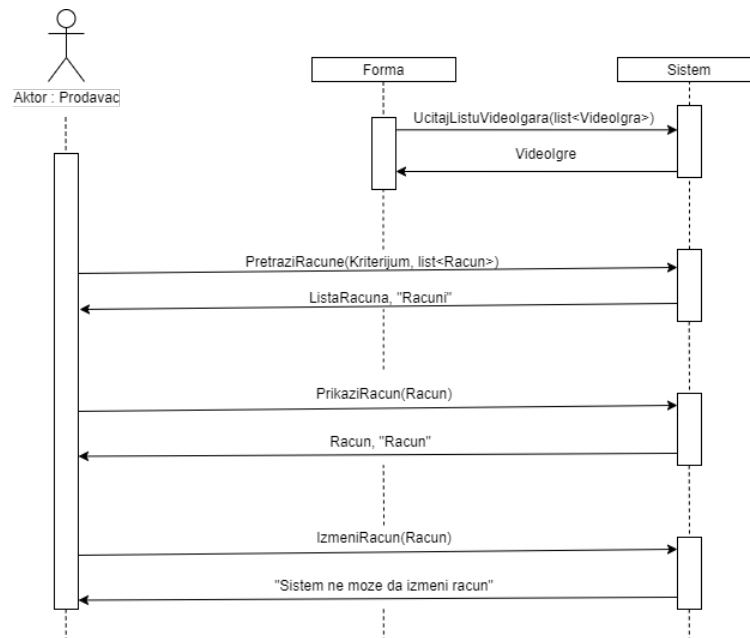
Slika 18 - Alternativni scenario DS6 - Izmena računa

6.1. Ukoliko sistem ne može da prikaže konkretni **račun**, obaveštava prodavca o tome: "Sistem ne moze da ucita prodaju!". Prekida se izvršavanje scenarija. (IA)



Slika 19 - Alternativni scenario DS6 - Izmena računa

8.1. Ukoliko sistem ne može da izmeni **račun**, obaveštava prodavca o tome: "Sistem ne moze da zapamti prodaju!". (IA)



Slika 20 - Alternativni scenario DS6 - Izmena računa

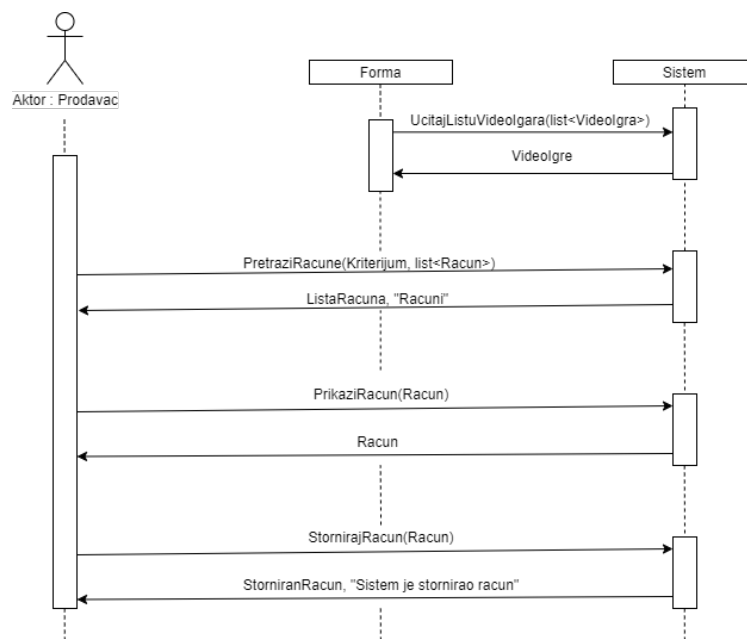
Sa navedenih dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. Signal **PretraziRacune**(Kriterijum, list <Racun>);
2. Signal **PrikaziRacun**(Racun);
3. Signal **IzmeniRacun**(Racun);
4. Signal **UcitajListuVideolgara**(list <Videolgra>).

2.1.7. Dijagram sekvenci slučaja korišćenja 7: Storniranje računa

Osnovni scenario SK:

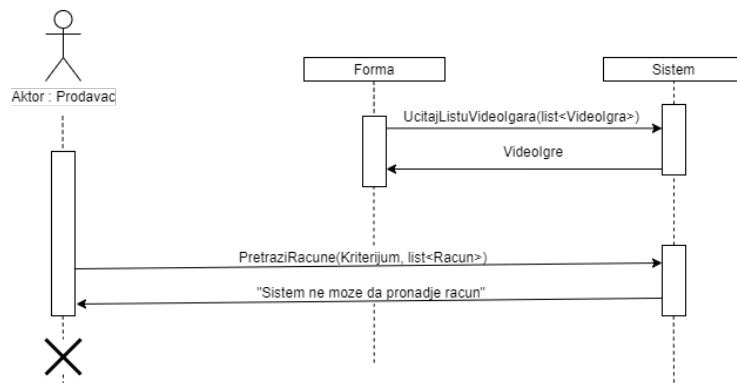
1. **Forma** poziva sistem da učitá listu **video igara**. (APSO)
2. **Sistem** vraća formi listu **video igara**. (IA)
3. **Prodavac** poziva sistem da izvrši pretragu **računa** po zadatoj vrednosti. (APSO)
4. **Sistem** prikazuje prodavcu podatke o **računima** i poruku: "Sistem je pronasao prodaje za izabrani datum prodaje!". (IA)
5. **Prodavac** poziva sistem da prikaže podatke o izabranom **računu**. (APSO)
6. **Sistem** prikazuje podatke o izabranom **računu**. (IA)
7. **Radnik** poziva sistem da stornira zadati **račun**. (APSO)
8. **Sistem** prikazuje radniku storniran **račun** i poruku: "Sistem je obrisao prodaju!". (IA)



Slika 21 - Osnovni scenario DS7 - Storniranje računa

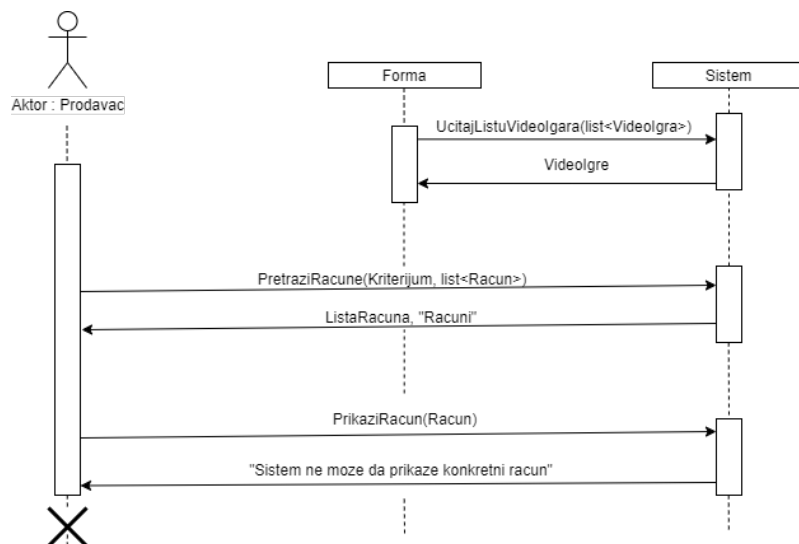
Alternativna scenarija SK:

- 4.1. Ukoliko sistem ne može da pronađe **račune**, obaveštava prodavca o tome: "Nema prodaje za izabrani datum prodaje!". Prekida se izvršavanje scenarija. (IA)



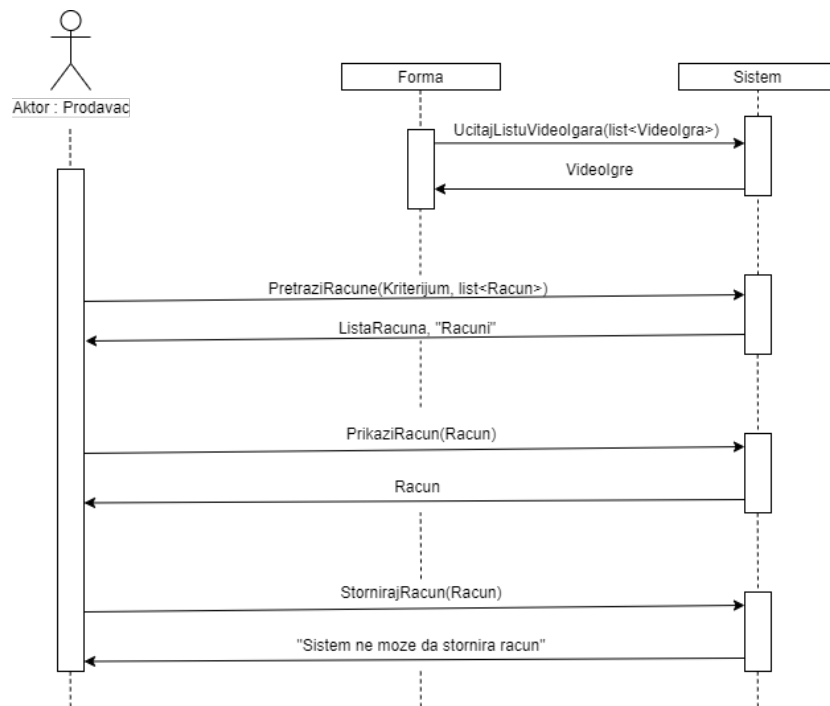
Slika 22 - Alternativni scenario D57 - Storniranje računa

6.1. Ukoliko sistem ne može da prikaže konkretni **račun**, obaveštava prodavca o tome: “Sistem ne moze da ucita prodaju!”. Prekida se izvršavanje scenarija. (IA)



Slika 23 - Alternativni scenario D57 - Storniranje računa

8.1. Ukoliko sistem ne može da obriše račun, obaveštava prodavca o tome: “ Sistem ne moze da obrise prodaju!”. (IA)



Slika 24 - Alternativni scenario DS7 - Storniranje računa

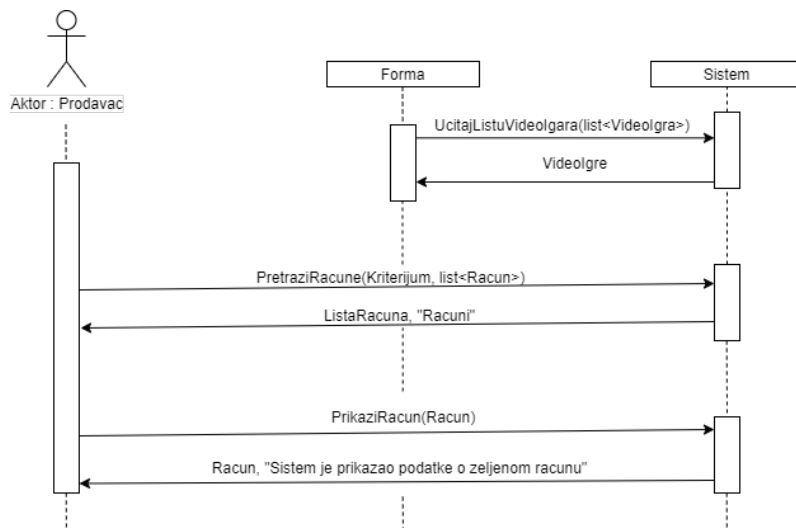
Sa navedenih dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. Signal **PretraziRacune**(Kriterijum, list <Racun>);
2. Signal **PrikaziRacun**(Racun);
3. Signal **StornirajRacun**(Racun);
4. Signal **UcitajListuVideolgara**(list <Videolgra>).

2.1.8. Dijagram sekvenci slučaja korišćenja 8: Pretraga računa

Osnovni scenario SK:

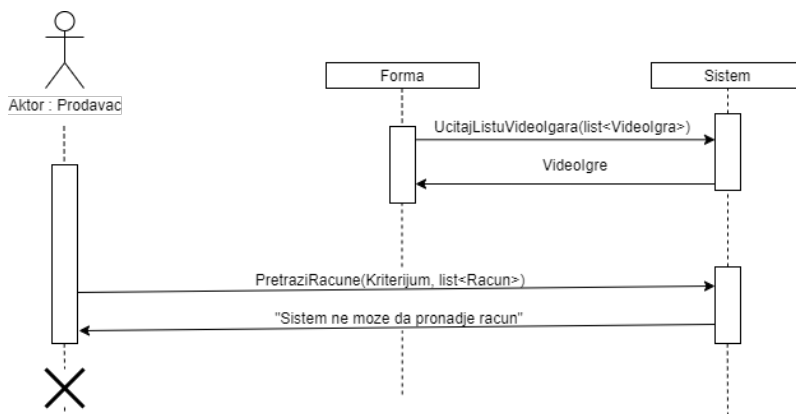
1. **Forma** poziva sistem da učitá listu **video igara**. (APSO)
2. **Sistem** vraća formi listu **video igara**. (IA)
3. **Prodavac** poziva sistem da izvrši pretragu **računa** po zadatoj vrednosti. (APSO)
4. **Sistem** prikazuje prodavcu podatke o **računima** i poruku: "Sistem je pronasao prodaje za izabrani datum prodaje!". (IA)
5. **Prodavac** poziva sistem da prikaže podatke o izabranom **računu**. (APSO)
6. **Sistem** prikazuje podatke o traženom **računu** i poruku: "Sistem je uspesno učitao prodaju! ". (IA)



Slika 25 - Osnovni scenario DS8 - Pretraga računa

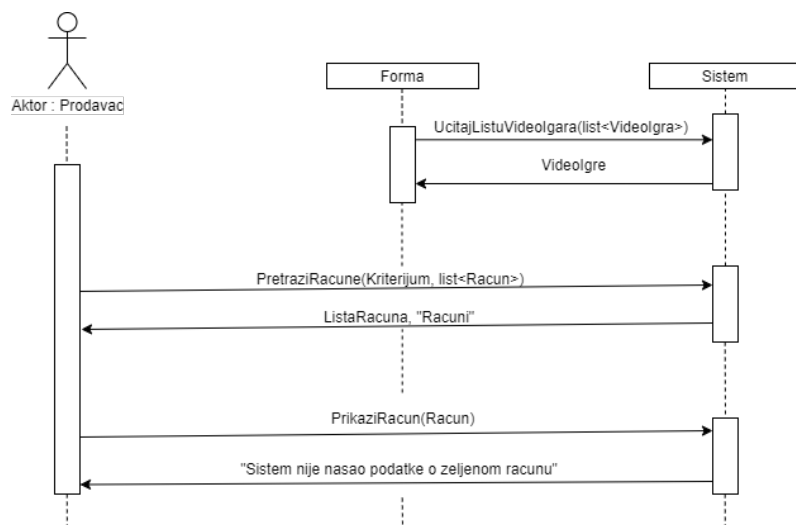
Alternativna scenarija SK:

4.1. Ukoliko sistem ne može da pronađe račune, obaveštava prodavca o tome: “Nema prodaje za izabrani datum prodaje!”. Prekida se izvršenje scenarija. (IA)



Slika 26 - Alternativni scenario DS8 - Pretraga računa

6.1. Ukoliko sistem ne može da pronađe podatke o željenom **računu**, obaveštava prodavca o tome: “Sistem ne moze da ucita prodaju!”. (IA)



Slika 27 - Alternativni scenario DS8 - Pretraga računa

Sa navedenih dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. Signal **UcitajListuVideolgara**(list <Videolgra>);
2. Signal **PretraziRacune**(Kriterijum, list<Racun>);
3. Signal **PrikaziRacun**(Racun).

2.1.9. Rezultat analize sistemskog dijagrama sekvenci

Kao rezultat analize scenarija uočavaju se sledeće sistemske operacije:

1. Signal **UnosVideolgre**(Videolgra);
2. Signal **UcitajListuPlatformi**(list<Platforma>);
3. Signal **PretraziVideolgre**(Kriterijum, list <Videolgra>);
4. Signal **IzmeniVideolgru**(Videolgra);
5. Signal **PrikaziVideolgru**(Videolgra);
6. Signal **ObrisiVideolgru**(Videolgra);
7. Signal **ObradiRacun**(Racun);
8. Signal **ZapamtiRacun**(Racun);
9. Signal **UcitajListuVideolgara** (list <Videolgra>);
10. Signal **PretraziRacune**(Kriterijum, list <Racun>);
11. Signal **PrikaziRacun**(Racun);
12. Signal **IzmeniRacun**(Racun);
13. Signal **StornirajRacun**(Racun).

2.2 Ponašanje softverskog sistema – Definisanje ugovora o sistemskim operacijama

2.2.1. Ugovor UG1 : Unos video igre

Operacija: Signal UnosVideolgre(Videolgra);

Veza sa SK: SK1

Preduslovi: Mora biti zadovoljeno prosto vrednosno I strukturno ograničenje nad objektom **Videolgra**. Vraćena lista platformi.

Postuslovi: Izvršen je unos video igre.

2.2.2. Ugovor UG2 : Učitavanje liste platformi

Operacija: Signal UcitajListuPlatformi(list<Platofrma>);

Veza sa SK: SK1, SK2, SK3, SK4

Preduslovi: /

Postuslovi: /

2.2.3. Ugovor UG3 : Pretraga video igara

Operacija: Signal PretraziVideolgre(Kriterijum, list<Videolgra>);

Veza sa SK: SK2, SK3, SK4

Preduslovi: /

Postuslovi: /

2.2.4. Ugovor UG4 : Izmena video igre

Operacija: Signal IzmeniVideolgru(Videolgra);

Veza sa SK: SK2

Preduslovi: Mora biti zadovoljeno prosto vrednosno I strukturno ograničenje nad objektom **Videolgra**. Vraćena je lista platformi.

Postuslovi: Izvršena je izmena video igre.

2.2.5. Ugovor UG5 : Prikaz video igre

Operacija: Signal PrikaziVideolgru(Videolgra);

Veza sa SK: SK2, SK4

Preduslovi: /

Postuslovi: /

2.2.6. Ugovor UG6 : Brisanje video igre

Operacija: Signal ObrisiVideolgru(Videolgra);

Veza sa SK: SK3

Preduslovi: Mora biti zadovoljeno prosto vrednosno I strukturno ograničenje nad objektom **Videolgra**.

Postuslovi: Izvršeno je brisanje video igre.

2.2.7. Ugovor UG7 : Obrada računa

Operacija: Signal ObradiRacun(Racun);

Veza sa SK: SK5

Preduslovi: /

Postuslovi: /

2.2.8. Ugovor UG8 : Čuvanje računa

Operacija: Signal ZapamtiRacun(Racun);

Veza sa SK: SK5

Preduslovi: Mora biti zadovoljeno prosto vrednosno I strukturno ograničenje nad objektom **Racun**.
Vraćena je lista video igara.

Postuslovi: Izvršen je unos računa.

2.2.9. Ugovor UG9 : Učitavanje liste video igara

Operacija: Signal UcitajListuVideolgara(list<Videolgra>);

Veza sa SK: SK5, SK6, SK7, SK8

Preduslovi: /

Postuslovi: /

2.2.10. Ugovor UG10 : Pretraga računa

Operacija: Signal PretraziRacune(Kriterijum, list<Racun>);

Veza sa SK: SK6, SK7, SK8

Preduslovi: /

Postuslovi: /

2.2.11. Ugovor UG11 : Prikaz računa

Operacija: Signal PrikaziRacun(Racun);

Veza sa SK: SK6, SK7, SK8

Preduslovi: /

Postuslovi: /

2.2.12. Ugovor UG12 : Izmena računa

Operacija: Signal IzmenaRacuna(Racun);

Veza sa SK: SK6

Preduslovi: Mora biti zadovoljeno prosto vrednosno I strukturno ograničenje nad objektom **Racun**.
Vraćena je lista video igara.

Postuslovi: Izvršena je izmena računa.

2.2.13. Ugovor UG13 : Storniranje računa

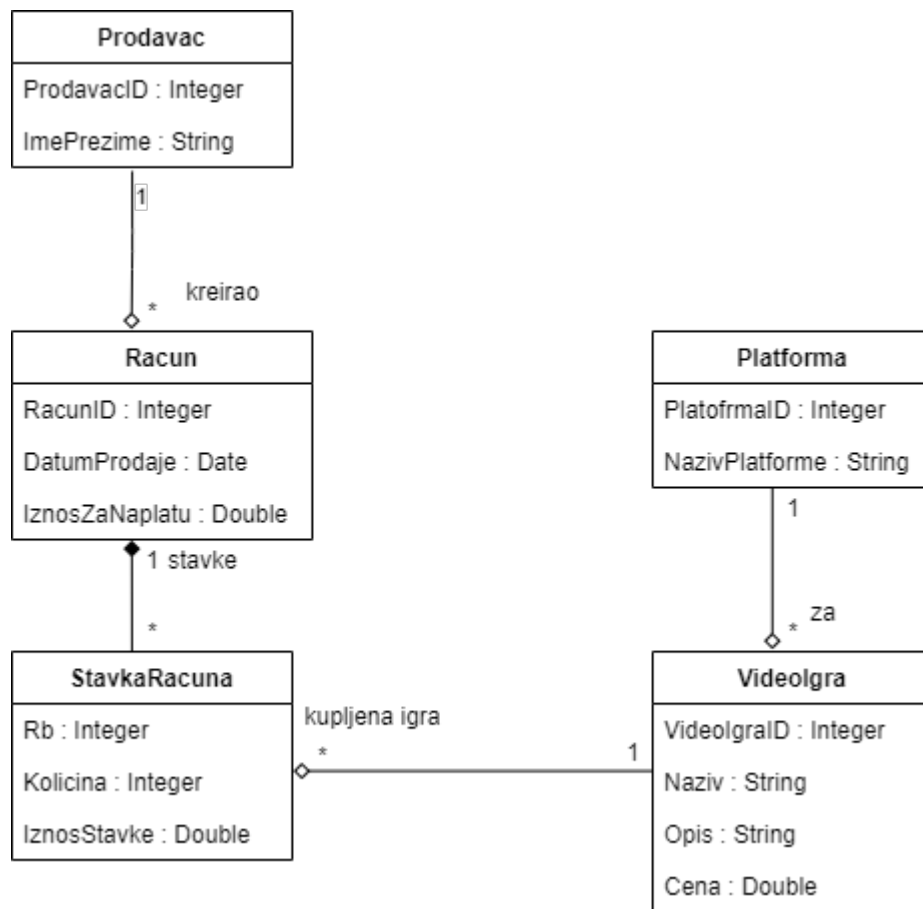
Operacija: Signal StornirajRacun(Racun);

Veza sa SK: SK7

Preduslovi: Mora biti zadovoljeno prosto vrednosno I strukturno ograničenje nad objektom **Racun**.

Postuslovi: Izvršeno je storniranje računa.

2.3. Struktura softverskog sistema – Konceptualni model



Slika 28 - Konceptualni model

2.4. Struktura softverskog sistema – Relacioni model

Prodavac (_prodavacID, imePrezime);

Racun (_racunID, datumProdaje, iznosZaNaplatsu, prodavacID);

StavkaRacuna (_racunID, _rb, kolicina, iznosStavke, videolgralD);

Videolgra (_videolgralD, naslov, opis, cena, platformalD);

Platforma (_platformalD, nazivPlatforme);

Prodavac		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT / UPDATE CASCADES Racun DELETE RESTRICTED Racun
	prodavacID	int	not null			
	imePrezime	string	not null			

Platforma		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT / UPDATE CASCADES Videolgra DELETE RESTRICTED Videolgra
	platofrmID	int	not null			
	nazivPlatofrm	string	not null			

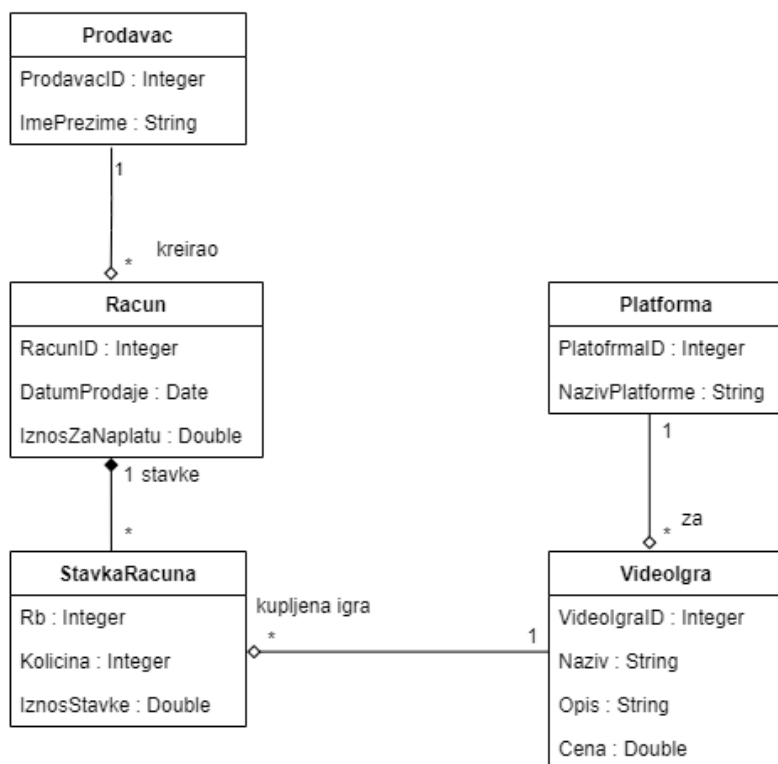
Videolgra		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED Platforma UPDATE CASCADES StavkaRacuna RESTRICTED Platforma DELETE RESTRICTED StavkaRacuna
	videolgraID	int	not null			
	naziv	string	not null			
	opis	string	not null			
	cena	double	not null			
	platformaID	int	not null			

Racun		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED Prodavac UPDATE CASCADES StavkaRacuna RESTRICTED Prodavac DELETE RESTRICTED StavkaRacuna
	racunID	int	not null			
	datumProdaje	date	not null			
	iznosZaNaplatu	double	not null		iznosZaNaplatu = sum(StavkaRacuna.IznosStavke)	
	idAdmin	int	not null			

StavkaRacuna		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED Racun, Videolgra UPDATE RESTRICTED Racun, Videolgra DELETE /
	racunID	int	not null			
	rb	int	not null			
	kolicina	int	not null			
	iznosStavke	double	not null		iznosStavke = kolicina * Videolgra.Cena	
	videolgraID	int	not null			

Kao rezultat analize I scenarija SK i pravljenja konceptualnog modela dobija se logička struktura i ponašanje softverskog sistema:

Struktura sistema



Ponasanje sistema

Sistemske operacije
UnosVideolgre(Videolgra): signal
UcitajListuPlatformi(list<Platforma>): signal
PretraziVideolgre(Kriterijum, list <Videolgra>): signal
IzmeniVideolgru(Videolgra): signal
PrikaziVideolgru(Videolgra): signal
ObrisiVideolgru(Videolgra): signal
ObradiRacun(Racun): signal
ZapamtiRacun(Racun): signal
UcitajListuVideolgara (list <Videolgra>): signal
PretraziRacune(Kriterijum, list <Racun>): signal
PrikaziRacun(Racun): signal
IzmeniRacun(Racun): signal
StornirajRacun(Racun): signal

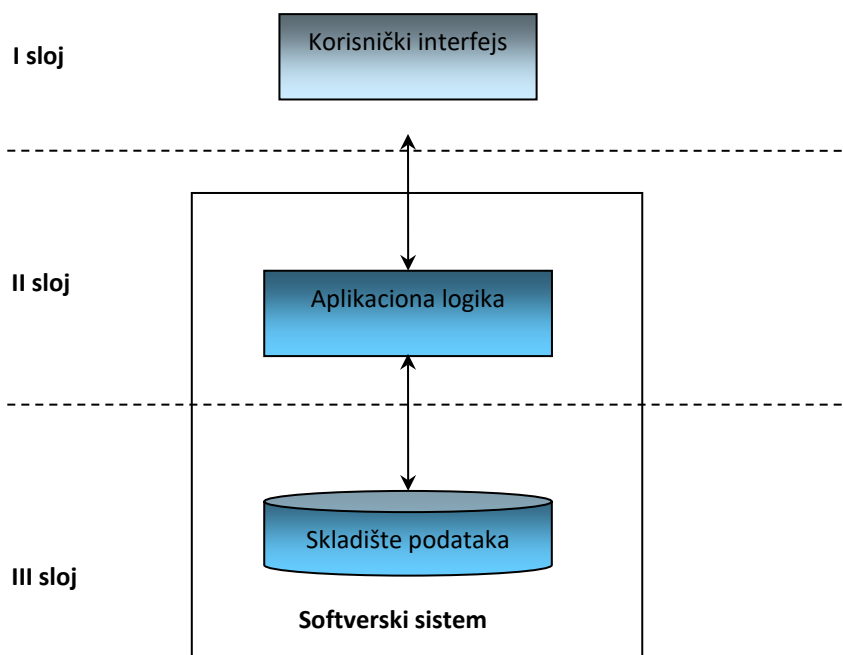
3. Projektovanje

Faza projektovanja opisuje fizičku strukturu i ponašanje softverskog sistema (arhitekturu softverskog sistema). Projektovanje arhitekture softverskog sistema obuhvata projektovanje korisničkog interfejsa, aplikacione logike i skladišta podataka. Projektovanje korisničkog interfejsa obuhvata projektovanje ekranskih formi i kontrolera korisničkog interfejsa. U okviru aplikacione logike se projektuju kontroler aplikacione logike, poslovna logika i broker baze podataka. Projektovanje poslovne logike obuhvata projektovanje logičke strukture i ponašanja softverskog sistema.

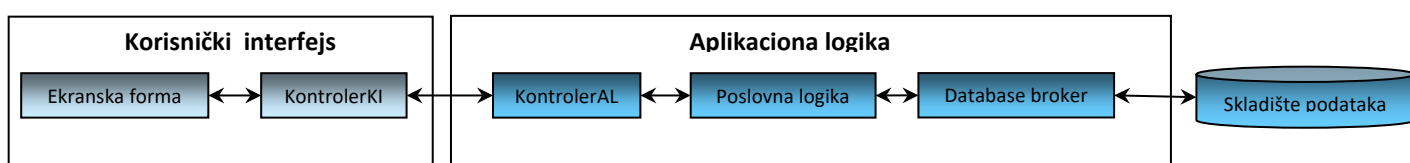
3.1. Arhitektura softverskog sistema

U okviru faze projektovanja opisuje se fizička struktura i ponašanje softverskog sistema tj. arhitektura softverskog sistema. U ovom seminarskom radu je korišćena klasična troslojna arhitektura, koja se sastoji od:

- Korisničkog interfejsa
- Aplikacione logike
- Skladišta podataka



Slika 29 - Troslojna arhitektura



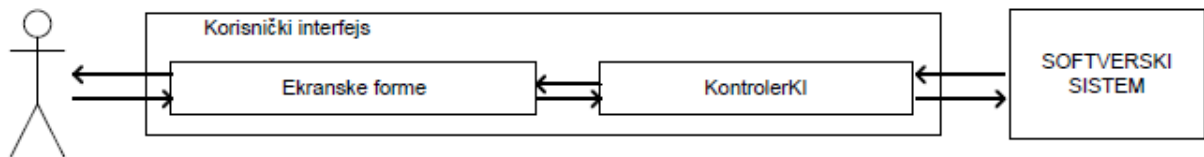
Slika 30 - Troslojna arhitektura – detaljniji prikaz

3.2. Projektovanje korisničkog interfejsa

Kontroler korisničkog interfejsa i ekranske forme ne predstavljaju softverski sistem, već realizaciju ulaza i izlaza iz softverskog sistema. Kontroler korisničkog interfejsa ima uloga da konvertuje podatke u grafičke elemente ekranskih formi i da podatke ili pošalje u softverski sistem ili da primi podatke iz softverskog sistema i iste konvertuje u elemente ekranske forme.

Korisnički interfejs se sastoji iz:

- Ekranske forme,
- Kontrolera korisničkog interfejsa.



Slika 31 - Struktura korisnickog interfejsa

3.3. Projektovanje ekranskih formi

Korisnički interfejs definisan je preko skupa ekranskih formi. Scenarija korišćenja ekranskih formi su direktno povezana sa scenarijima slučajeva korišćenja.

Preko slučajeva korišćenja projektovane su i sve ostale ekranske forme koje će aplikacija posedovati, a koje se pozivaju iz menija glavne ekranske forme klijentskog dela aplikacije.

3.3.1. Slučaj korišćenja 1: Unos video igre

Naziv SK: Unos **video igre**

Aktori SK: Prodavac

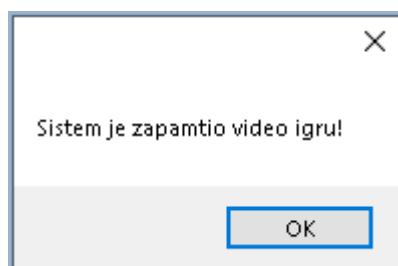
Učesnici SK: Prodavac i sistem

Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **video igrom**. Učitane su **platforme**.

Osnovni scenario SK:

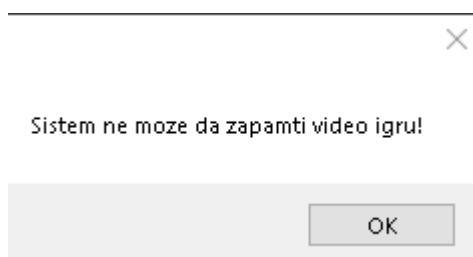
1. **Prodavac** unosi podatke o **video igri**. (APUSO)

2. **Prodavac** kontroliše da li je ispravno uneo podatke o **video igri**. (ANSO)
3. **Prodavac** poziva sistem da unese **video igru**. (APSO)
4. **Sistem** unosi **video igru**. (SO)
5. **Sistem** obaveštava prodavca da je uspešno uneo **video igru**: "Sistem je zapamtio video igru!". (IA)



Alternativna scenarija SK:

- 5.1 Ukoliko sistem ne može da unese **video igru**, obaveštava prodavca o tome: "Sistem ne može da zapamti video igru!". (IA)



3.3.2. Slučaj korišćenja 2: Izmena video igre

Naziv SK: Izmena **video igre**

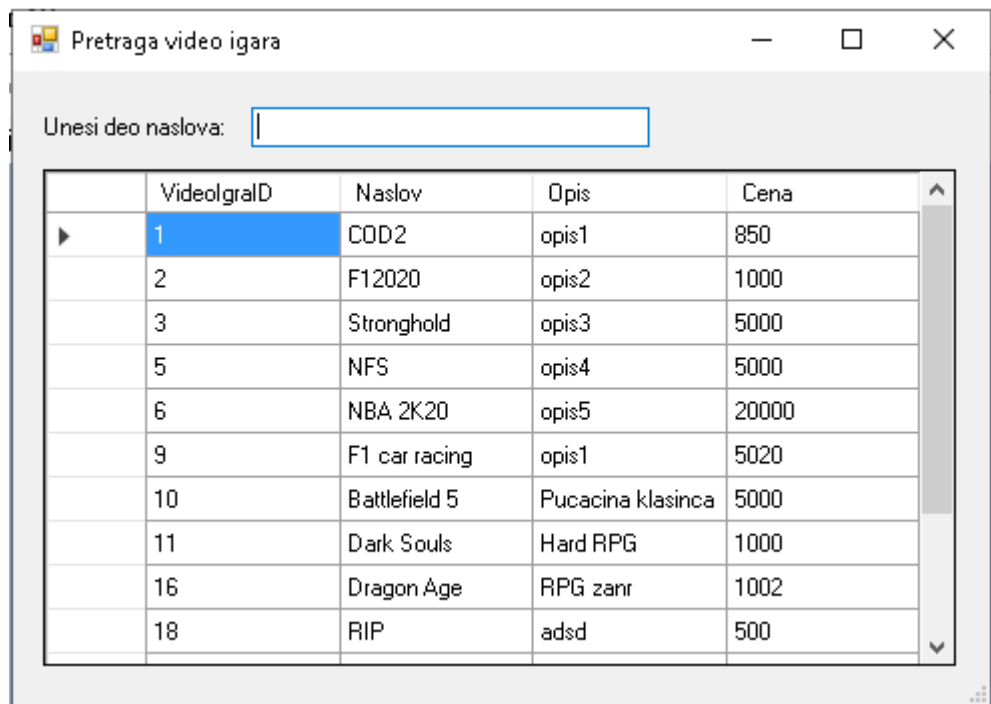
Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **video igrom**. Učitane su **platforme**.

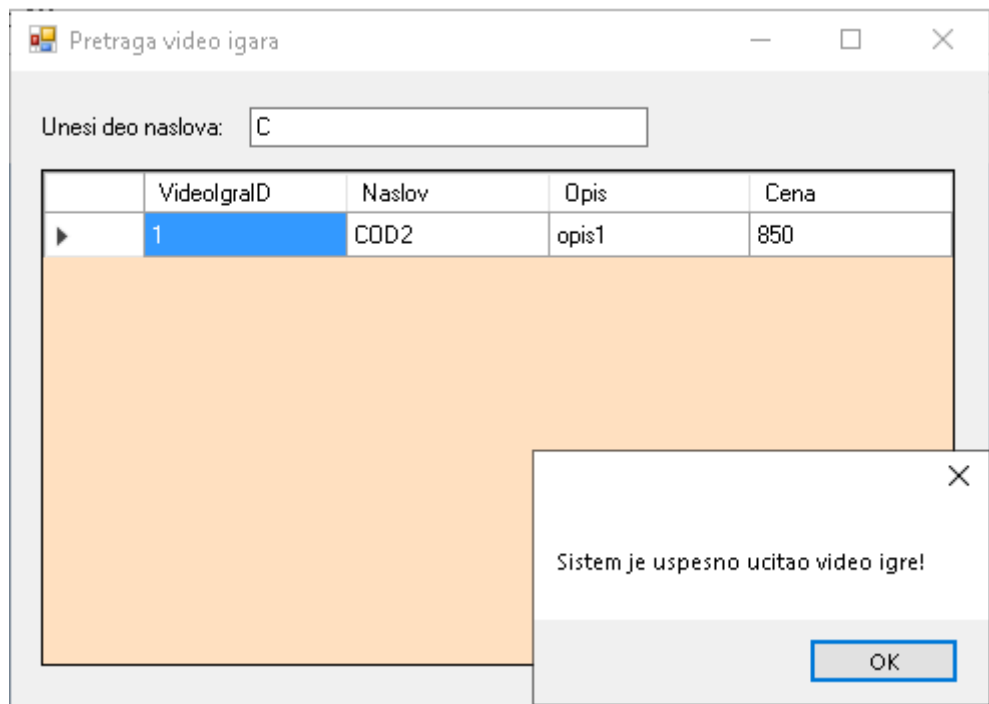
Osnovni scenario SK:

1. **Prodavac** unosi vrednost po kojoj vrši pretragu **video igara**. (APUSO)

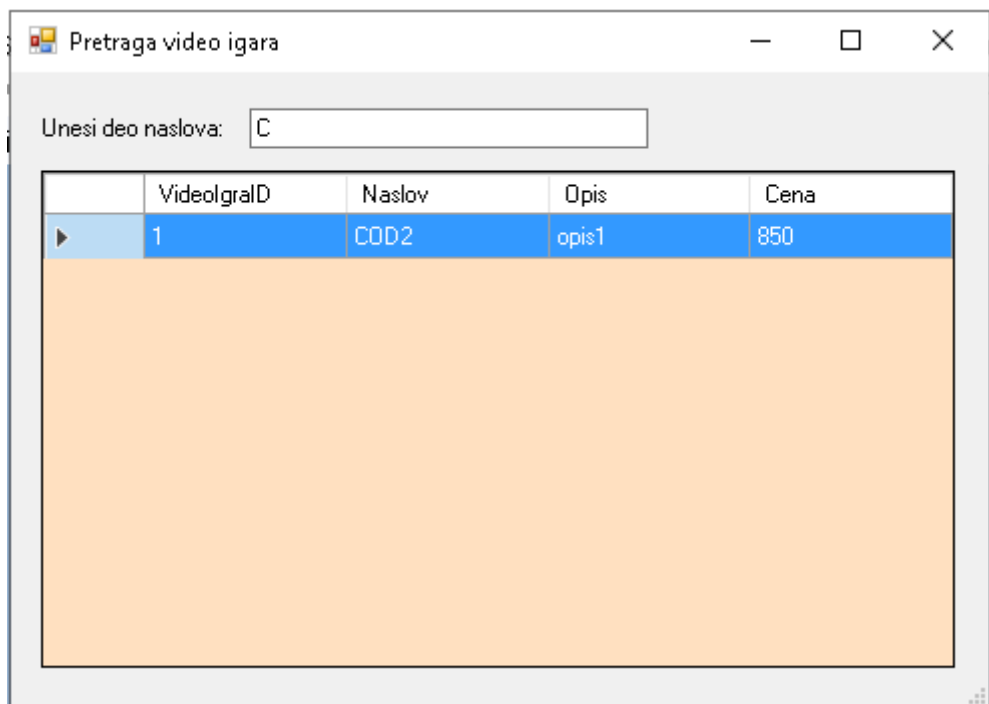


	VideogralD	Naslov	Opis	Cena
▶	1	COD2	opis1	850
	2	F12020	opis2	1000
	3	Stronghold	opis3	5000
	5	NFS	opis4	5000
	6	NBA 2K20	opis5	20000
	9	F1 car racing	opis1	5020
	10	Battlefield 5	Pucacina klasinca	5000
	11	Dark Souls	Hard RPG	1000
	16	Dragon Age	RPG zanr	1002
	18	RIP	adsd	500

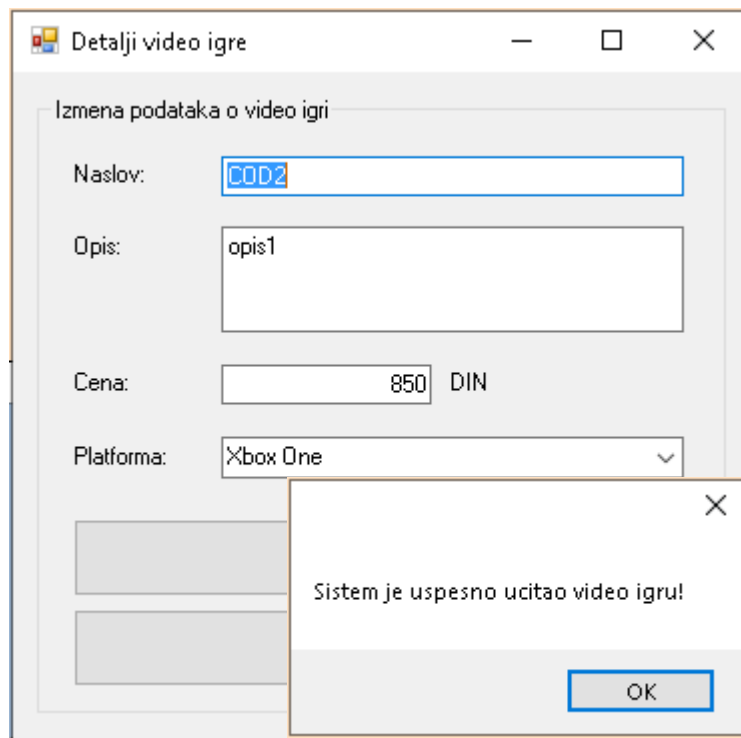
2. **Prodavac** poziva sistem da izvrši pretragu **video igara** po zadatoj vrednosti. (APSO)
3. **Sistem** izvršava pretragu **video igara** po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje prodavcu podatke o **video igrama** i poruku: "Sistem je uspesno učitao video igre!". (IA)



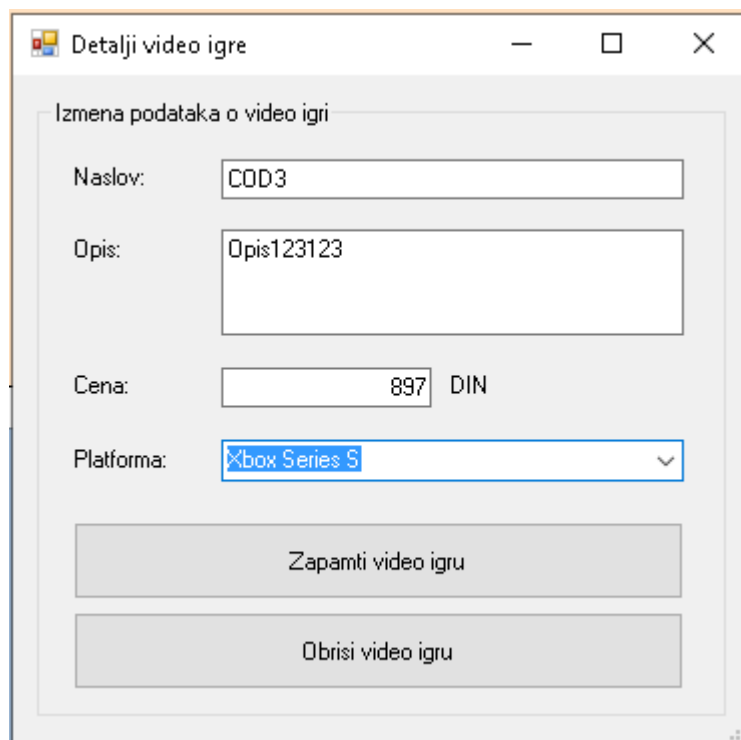
5. **Prodavac** bira **video igru** koju želi da izmeni. (APUSO)



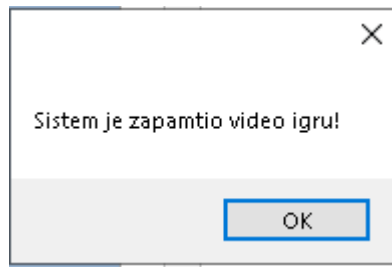
6. **Prodavac** poziva sistem da učitá **video igru**. (APSO)
7. **Sistem** pronalazi **video igru**. (SO)
8. **Sistem** prikazuje podatke o **video igri** i poruku: "Sistem je uspesno učitao video igru!". (IA)



9. **Prodavac** unosi (menja) podatke o **video igri**. (APUSO)

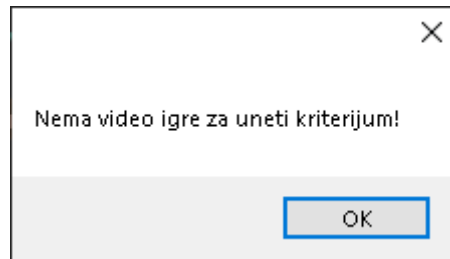


10. **Prodavac** kontroliše da li je ispravno izmenio podatke o **video igri**. (ANSO)
11. **Prodavac** poziva sistem da izmeni **video igru**. (APSO)
12. **Sistem** menja **video igru**. (SO)
13. **Sistem** obaveštava prodavca da je uspešno izmenio **video igru**: "Sistem je zapamtio video igru!". (IA)

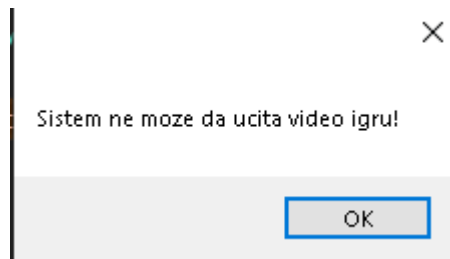


Alternativna scenarija SK:

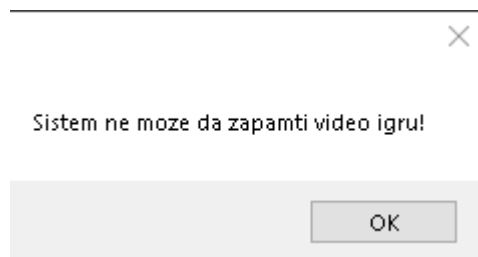
- 4.1 Ukoliko sistem ne može da pronađe **video igre**, obaveštava prodavca o tome: "Nema video igre za uneti kriterijum!". Prekida se izvršavanje scenarija. (IA)



- 8.1 Ukoliko sistem ne može da prikaže konkretnu **video igru**, obaveštava prodavca o tome: "Sistem ne moze da ucita video igru!". Prekida se izvršavanje scenarija. (IA)



- 13.1 Ukoliko sistem ne može da izmeni **video igru**, obaveštava prodavca o tome: "Sistem ne moze da zapamti video igru!". (IA)



3.3.3. Slučaj korišćenja 3: Brisanje video igre

Naziv SK: Brisanje **video igre**

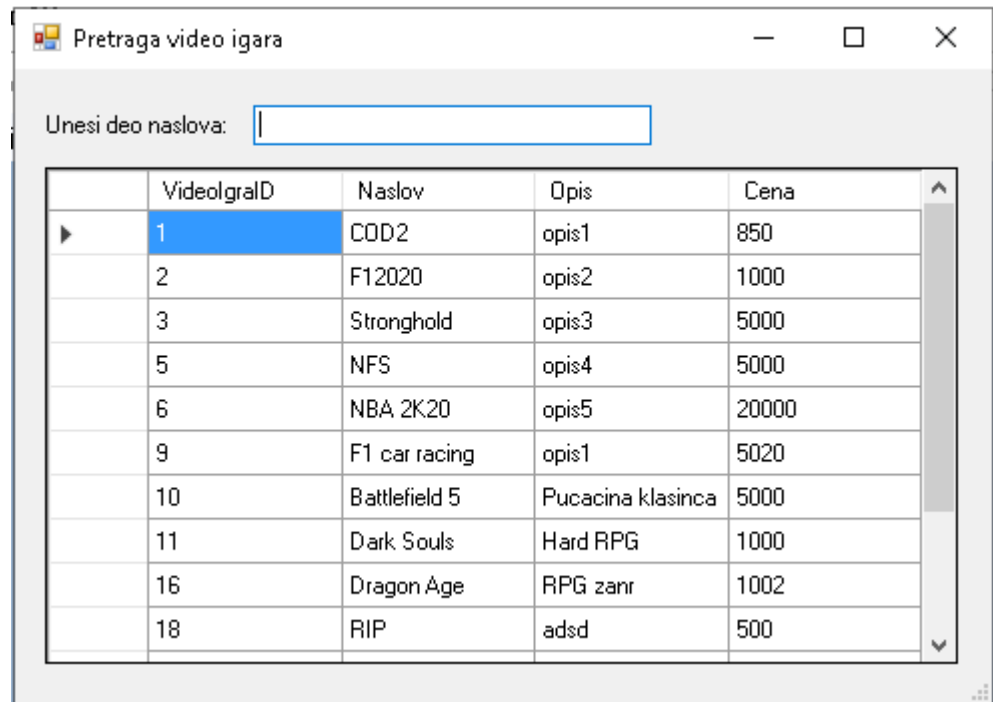
Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **video igrom**. Učitane su **platforme**.

Osnovni scenario SK:

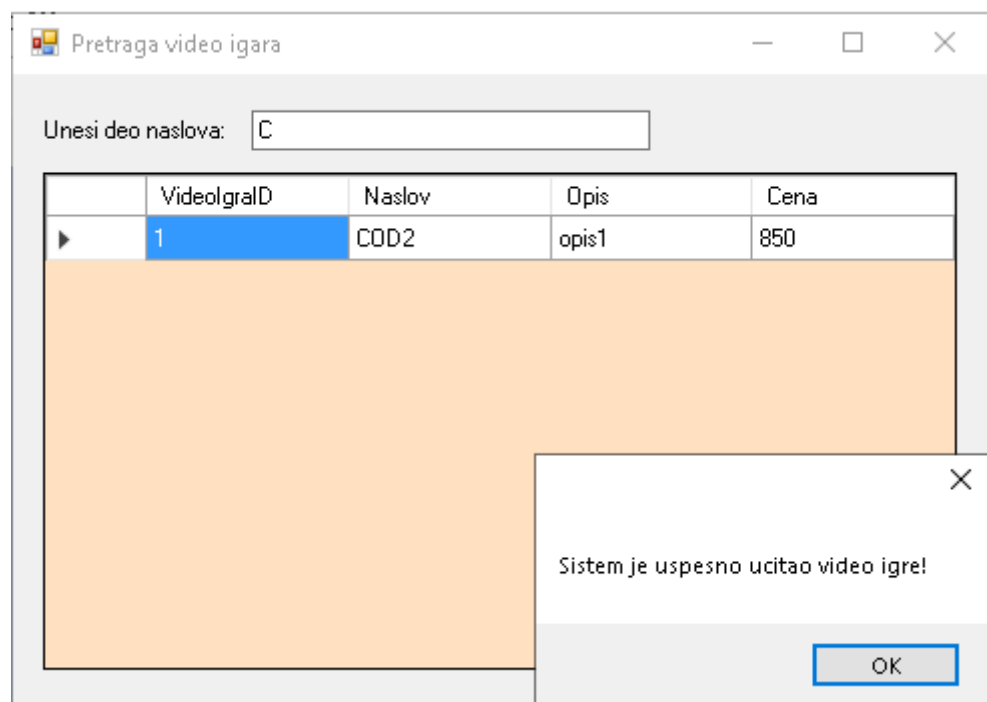
1. **Prodavac** unosi vrednost po kojoj vrši pretragu **video igara**. (APUSO)



The screenshot shows a window titled "Pretraga video igara". At the top, there is a text input field labeled "Unesi deo naslova:" which is currently empty. Below the input field is a table with the following data:

	VideolgralD	Naslov	Opis	Cena
▶	1	COD2	opis1	850
	2	F12020	opis2	1000
	3	Stronghold	opis3	5000
	5	NFS	opis4	5000
	6	NBA 2K20	opis5	20000
	9	F1 car racing	opis1	5020
	10	Battlefield 5	Pucacina klasinca	5000
	11	Dark Souls	Hard RPG	1000
	16	Dragon Age	RPG zanr	1002
	18	RIP	adsd	500

2. **Prodavac** poziva sistem da izvrši pretragu **video igara** po zadatoj vrednosti. (APSO)
3. **Sistem** izvršava pretragu **video igara** po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje prodavcu podatke o **video igrama** i poruku: "Sistem je uspesno učitao video igre!". (IA)

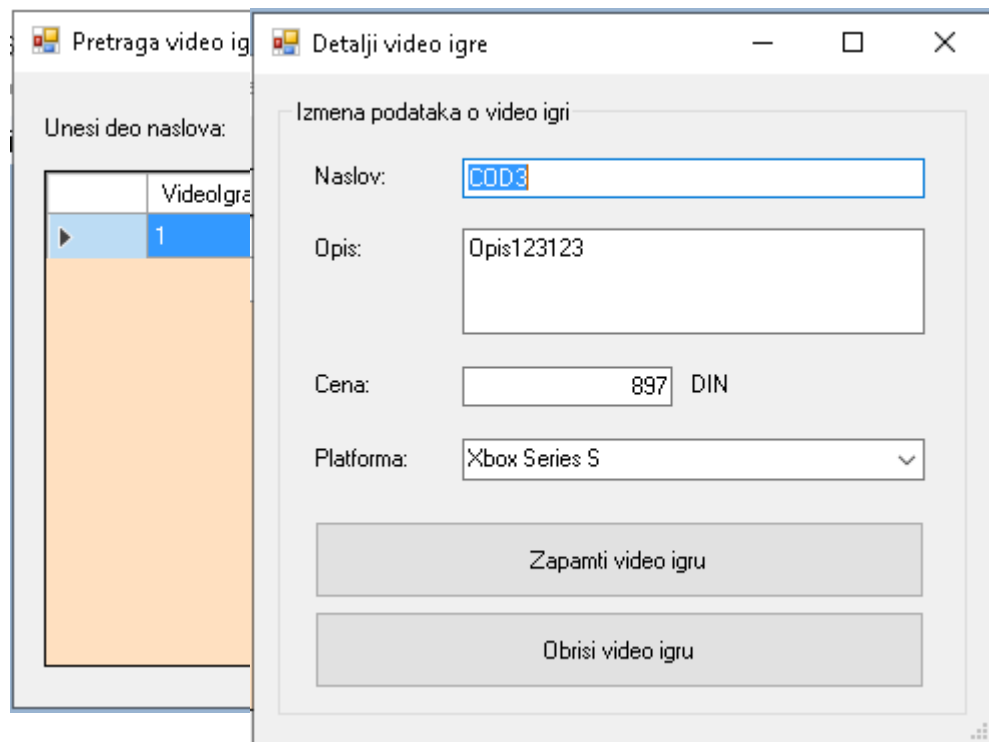


The screenshot shows the same window "Pretraga video igara". The text input field "Unesi deo naslova:" now contains the letter "C". The table below it shows only the first row of data:

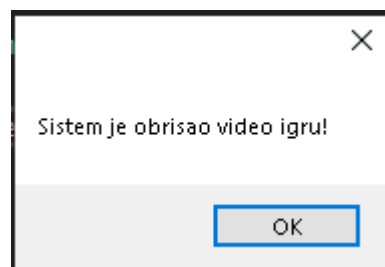
	VideolgralD	Naslov	Opis	Cena
▶	1	COD2	opis1	850

Below the table, a large orange rectangular area is visible. In the bottom right corner, a small white dialog box with a close button (X) is open. It contains the text "Sistem je uspesno učitao video igre!" and an "OK" button.

5. **Prodavac** bira **video igru** koji želi da obriše. (APUSO)

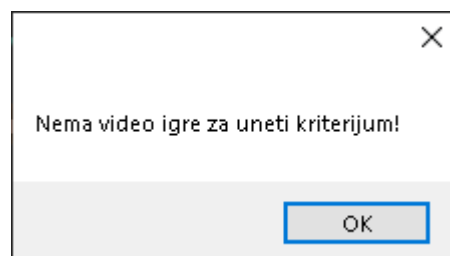


6. **Prodavac** poziva sistem da obriše **video igru**. (APSO)
7. **Sistem** briše **video igru**. (SO)
8. **Sistem** obaveštava prodavca da je uspešno obrisao **video igru**: "Sistem je obrisao video igru!". (IA)

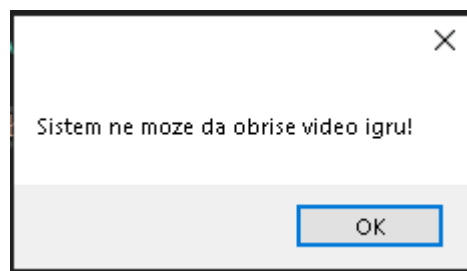


Alternativna scenarija SK:

- 4.1 Ukoliko sistem ne može da pronađe **video igru**, obaveštava prodavca o tome: "Nema video igre za uneti kriterijum!". Prekida se izvršavanje scenarija. (IA)



- 8.1 Ukoliko sistem ne može da obriše **video igru**, obaveštava prodavca o tome: "Sistem ne može da obriše video igru". (IA)



3.3.4. Slučaj korišćenja 4: Pretraga video igara

Naziv SK: Pretraga **video igara**

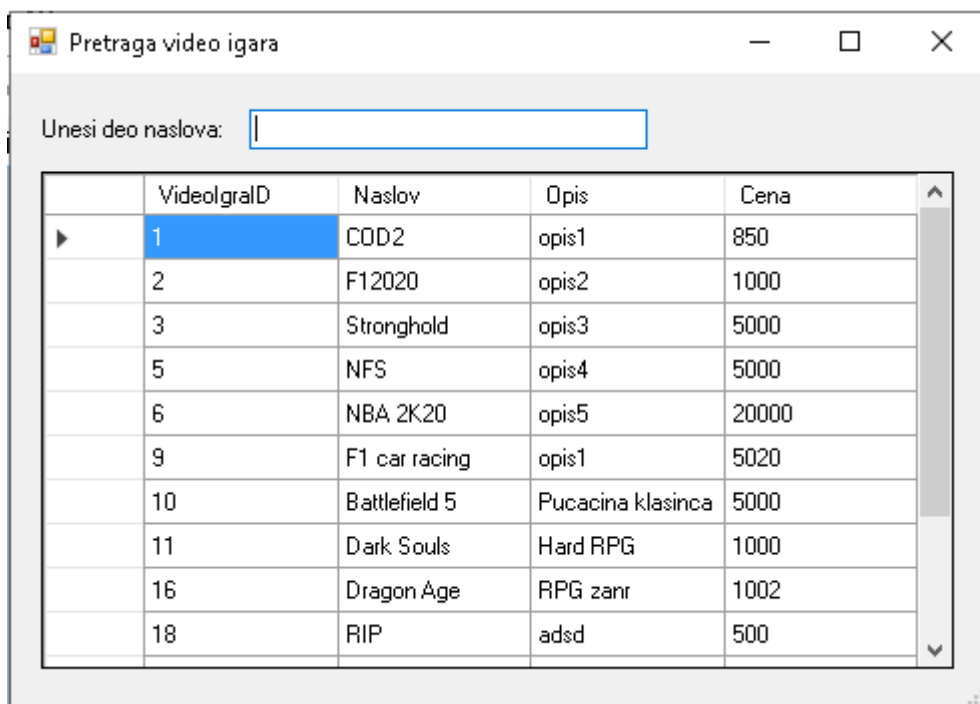
Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

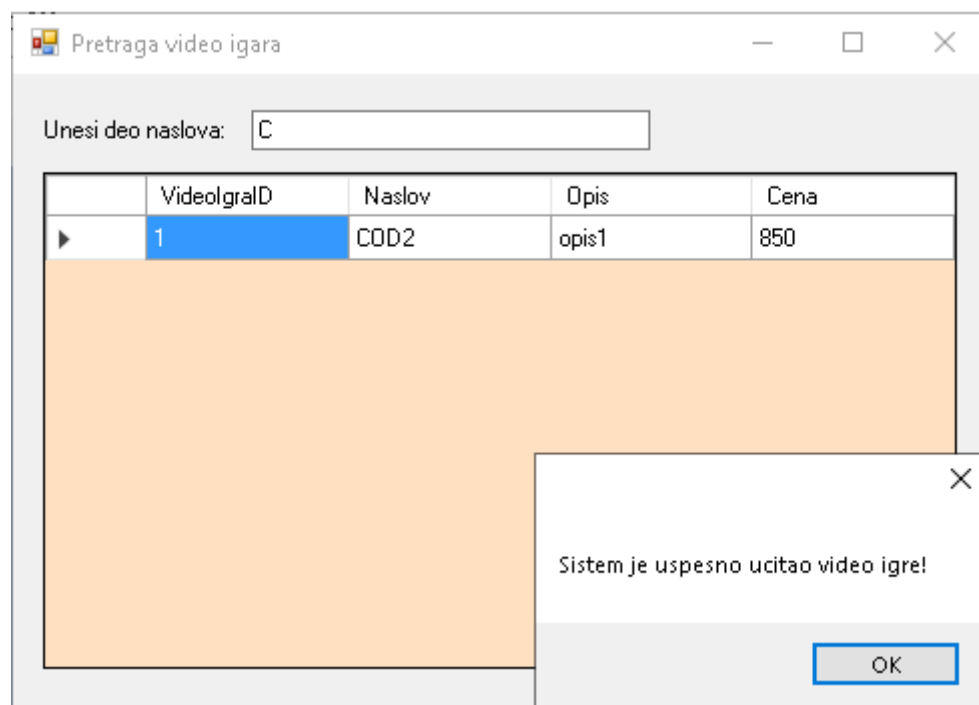
Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **video igrom**. Učitane su **platforme**.

Osnovni scenario SK:

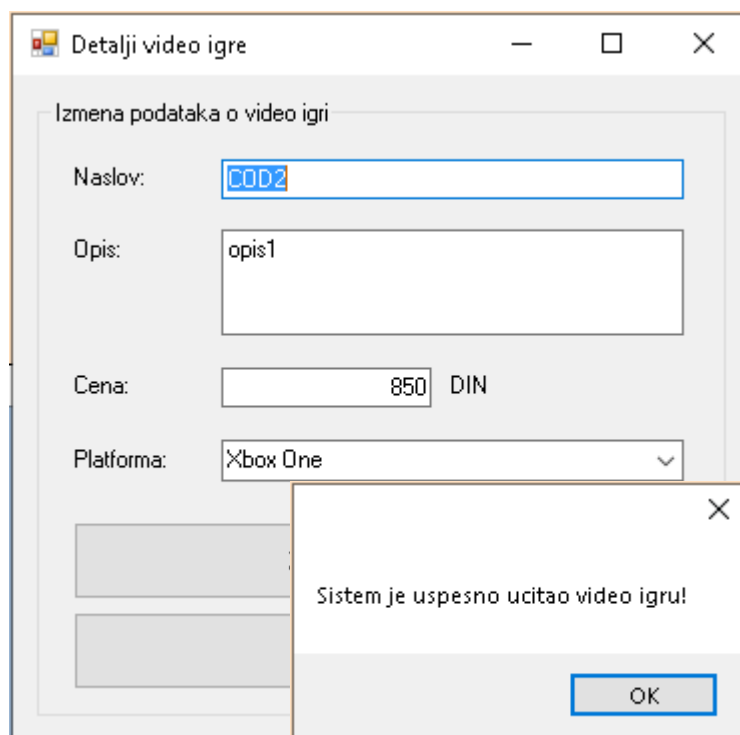
1. **Prodavac** unosi vrednost po kojoj vrši pretragu **video igara**. (APUSO)



2. **Prodavac** poziva sistem da izvrši pretragu **video igara** po zadatoj vrednosti. (APSO)
3. **Sistem** izvršava pretragu **video igara** po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje prodavcu podatke o **video igrama** i poruku: "Sistem je uspesno ucitao video igre!". (IA)



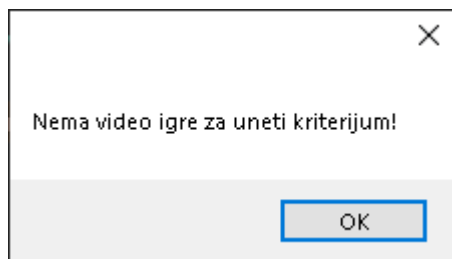
5. **Prodavac** bira **video igru** čije podatke želi da prikaže. (APUSO)
6. **Prodavac** poziva sistem da prikaže podatke o **video igri**. (APSO)
7. **Sistem** traži podatke o **video igri**. (SO)
8. **Sistem** prikazuje podatke o izabranoj **video igri** i poruku: "Sistem je uspesno učitao video igru!". (IA)



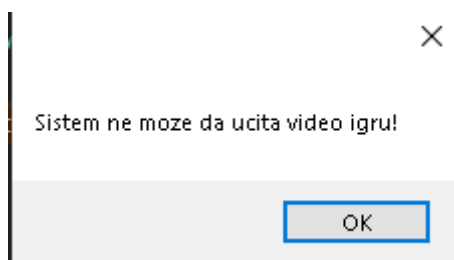
Alternativna scenarija SK:

- 4.1 Ukoliko sistem ne može da pronađe **video igre**, obaveštava prodavca o tome: "Nema video

igre za uneti kriterijum!”. Prekida se izvršavanje scenarija. (IA)



- 8.1 Ukoliko sistem ne može da prikaže podatke o **video igri** on prikazuje radniku poruku: "Sistem ne moze da ucita video igru!". (IA)



3.3.5. Slučaj korišćenja 5: Kreiranje računa

Naziv SK: Kreiranje **računa**

Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **računom**. Učitane su **video igre**.

Osnovni scenario SK:

1. **Prodavac** unosi podatke o **računu**. (APUSO)

Unos prodaje

Datum prodaje: 18. 2.2021. Prodavac: Marija Marijanovic ID 101 Kreiraj prodaju

Stavke prodaje

Video igra: NBA 2K20 Kolicina: 2

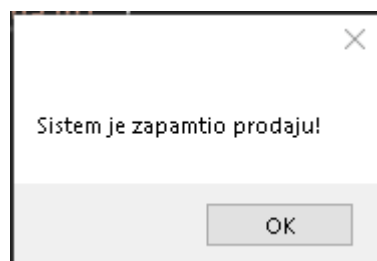
Dodaj stavku Obrisi stavku

	RedniBroj	Kolicina	IznosStavke	Videolgra
▶	1	1	897	COD3
	2	2	20000	NBA 2K20

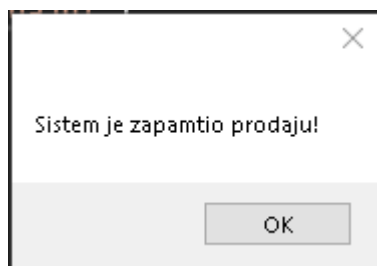
Iznos za naplatu 40897 din

Sacuvaj prodaju Odustani

2. **Prodavac** kontroliše da li je ispravno uneo podatke o **računu**. (ANSO)
3. **Prodavac** poziva sistem da zapamti podatke o **računu**. (APSO)
4. **Sistem** pamti podatke o **računu**. (SO)
5. **Sistem** obaveštava prodavca da je uspešno zapamćen **račun** i poruku: "Sistem je zapamtio prodaju!". (IA)

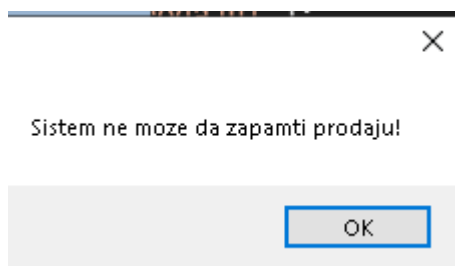


6. **Prodavac** poziva sistem da obradi **račun**. (APSO)
7. **Sistem** obrađuje **račun**. (SO)
8. **Sistem** prikazuje radniku obrađen **račun** i poruku: "Sistem je zapamtio prodaju!". (IA)

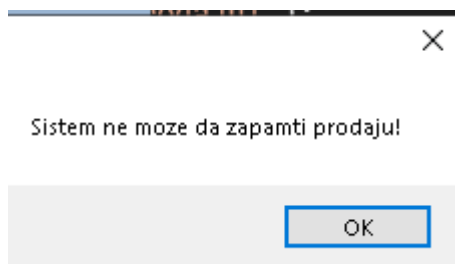


Alternativna scenarija SK:

- 5.1 Ukoliko sistem ne može da unese **račun**, obaveštava prodavca o tome: "Sistem ne može da zapamti prodaju!". Prekida se izvršavanje scenarija. (IA)



- 8.1 Ukoliko sistem ne može da obradi **račun** on prikazuje radniku poruku: "Sistem ne može da zapamti prodaju!". (IA)



3.3.6. Slučaj korišćenja 6: Izmena računa

Naziv SK: Izmena **računa**

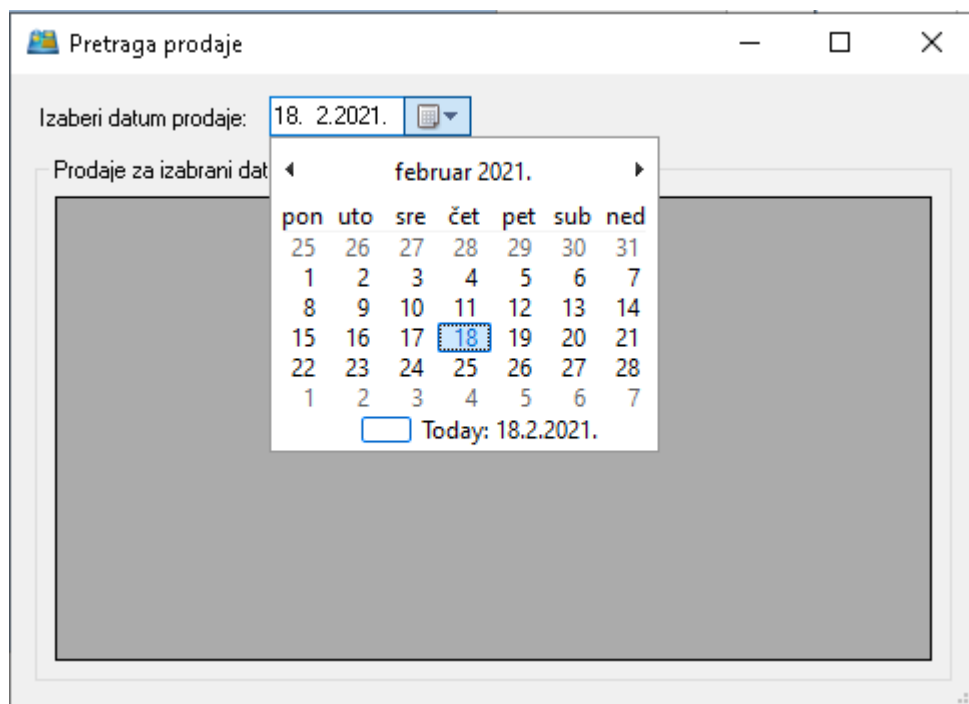
Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

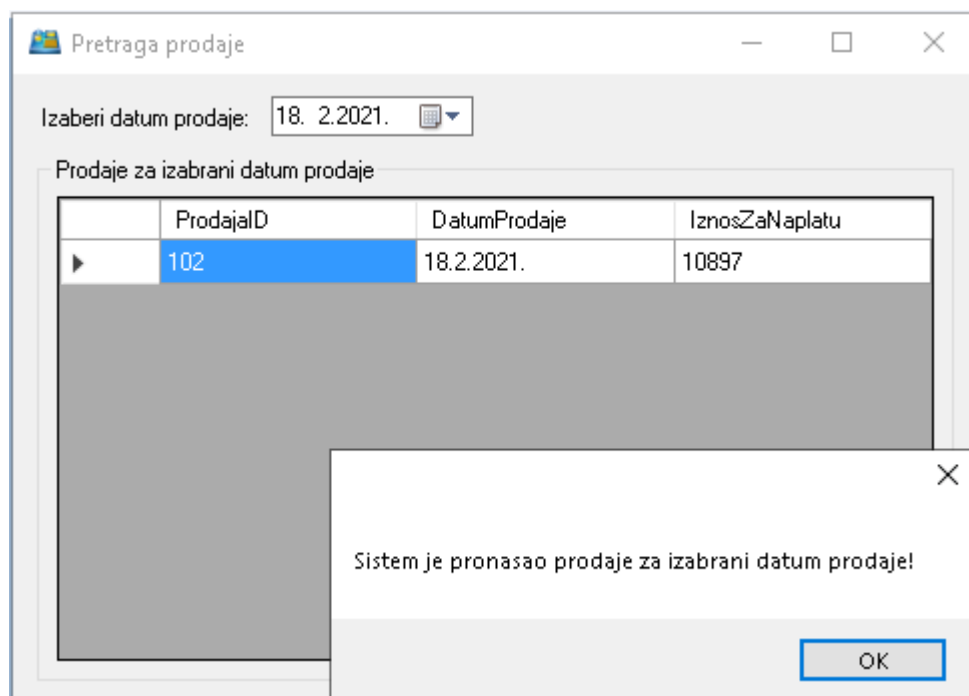
Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **računom**. Učitane su **video igre**.

Osnovni scenario SK:

1. **Prodavac** unosi vrednost po kojoj vrši pretragu **računa**. (APUSO)



2. **Prodavac** poziva sistem da izvrši pretragu **računa** po zadatoj vrednosti. (APSO)
3. **Sistem** izvršava pretragu **računa** po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje prodavcu podatke o **računima** i poruku: "Sistem je pronasao prodaje za izabrani datum prodaje!". (IA)



5. **Prodavac** bira **račun** koji želi da izmeni. (APUSO)

Pretraga prodaje

Izaberi datum prodaje: 18. 2.2021.

Prodaje za izabrani datum prodaje

	ProdajalID	DatumProdaje	IznosZaNaplatu
▶	102	18.2.2021.	10897

6. **Prodavac** poziva sistem da učitá račun. (APSO)
7. **Sistem** pronalazi račun. (SO)
8. **Sistem** prikazuje podatke o računu i poruku: "Sistem je uspesno učitao prodaju!". (IA)

Detalji prodaje

Datum prodaje: 18. 2.2021. Prodavac: Marta Martić

Stavke prodaje

Video igra: COD3 Kolicina:

Dodaj stavku Obrisi stavku

	RedniBroj	Kolicina	IznosStavke	VideoIgra
▶	1	1	897	COD3
	2	2	5000	NFS

Iznos za naplatu: 10897 din

Izmeni prodaju Obrisi prodaju

Sistem je uspesno učitao prodaju!

OK

9. **Prodavac** unosi (menja) podatke o **računu**. (APUSO)

Detalji prodaje

Datum prodaje: 18. 2.2021. Prodavac: Filip Filipovic

Stavke prodaje

Video igra: Dark Souls Kolicina: 1

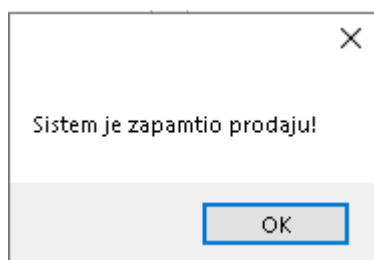
Dodaj stavku Obrisi stavku

	RedniBroj	Kolicina	IznosStavke	Videolgra
	1	1	897	COD3
	2	2	5000	NFS
▶	3	1	1000	Dark Souls

Iznos za naplatu: 11897 din

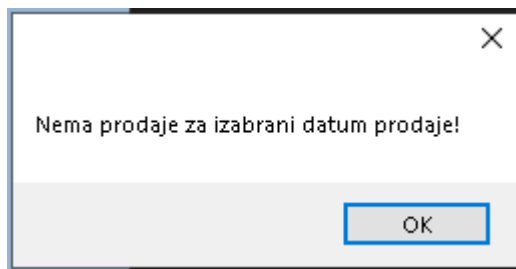
Izmeni prodaju Obrisi prodaju Odustani

10. **Prodavac** kontroliše da li je ispravno izmenio podatke o **računu**. (ANSO)
11. **Prodavac** poziva sistem da izmeni **račun**. (APSO)
12. **Sistem** menja **račun**. (SO)
13. **Sistem** obaveštava prodavca da je uspešno izmenio **račun**: "Sistem je zapamtio prodaju!". (IA)

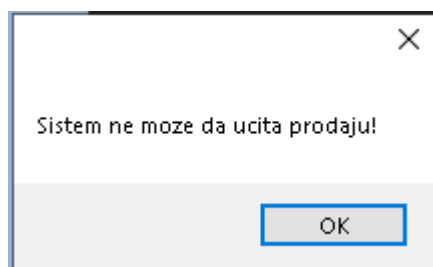


Alternativna scenarija SK:

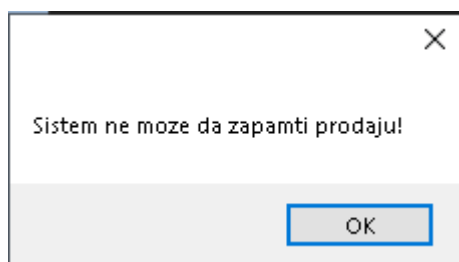
- 4.1 Ukoliko sistem ne može da pronađe **račune**, obaveštava prodavca o tome: "Nema prodaje za izabrani datum prodaje!". Prekida se izvršavanje scenarija. (IA)



- 8.1 Ukoliko sistem ne može da prikaže konkretni **račun**, obaveštava prodavca o tome: "Sistem ne može da učitaj prodaju!". Prekida se izvršavanje scenarija. (IA)



- 13.1 Ukoliko sistem ne može da izmeni **račun**, obaveštava prodavca o tome: "Sistem ne može da zapamti prodaju!". (IA)



3.3.7. Slučaj korišćenja 7: Storniranje računa

Naziv SK: Storniranje **računa**

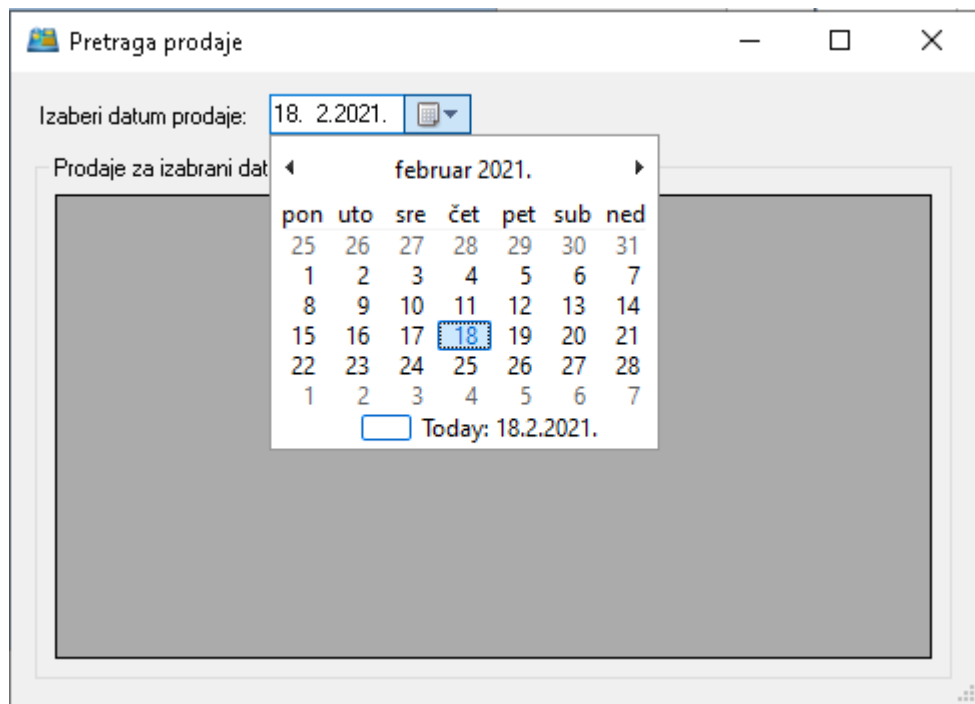
Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

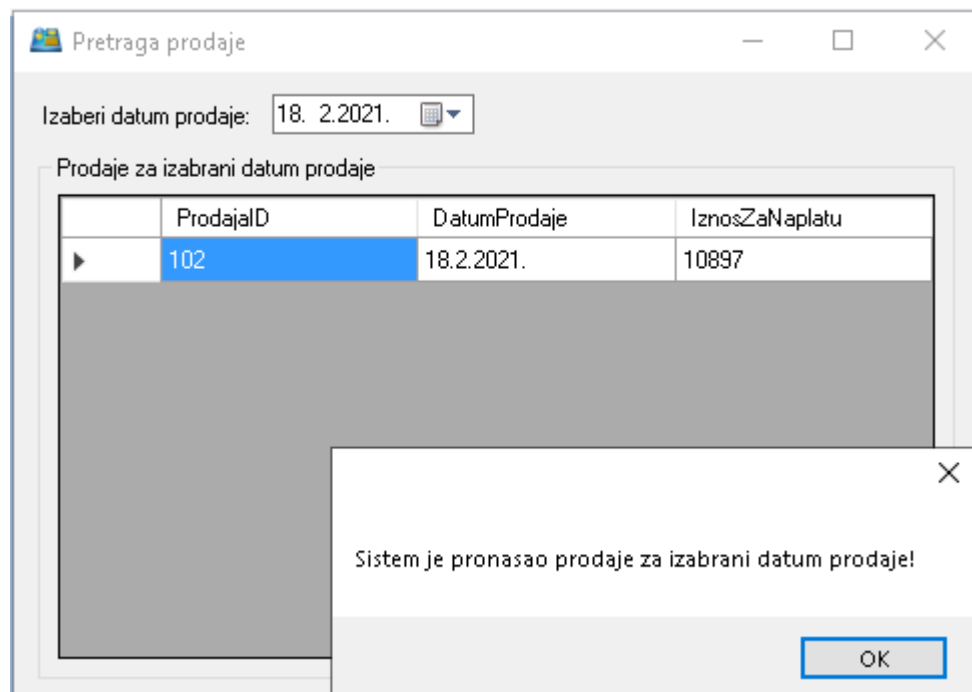
Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa **računom**. Učitane su **video igre**.

Osnovni scenario SK:

1. **Prodavac** unosi vrednost po kojoj vrši pretragu **računa**. (APUSO)



2. **Prodavac** poziva sistem da izvrši pretragu **računa** po zadatoj vrednosti. (APSO)
3. **Sistem** izvršava pretragu **računa** po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje prodavcu podatke o **računima** i poruku: "Sistem je pronasao prodaje za izabrani datum prodaje!". (IA)



5. **Prodavac** bira **račun** čije podatke želi da prikaže. (APUSO)

Pretraga prodaje

Izaberi datum prodaje: 18. 2.2021.

Prodaje za izabrani datum prodaje

	ProdajalD	DatumProdaje	IznosZaNaplatu
▶	102	18.2.2021.	10897

6. **Prodavac** poziva sistem da prikaže podatke o izabranom **računu**. (APSO)
7. **Sistem** traži podatke o izabranom **računu**. (SO)
8. **Sistem** prikazuje podatke o izabranom **računu**. (IA)

Detalji prodaje

Datum prodaje: 18. 2.2021. Prodavac: Marta Martić

Stavke prodaje

Video igra: COD3 Kolicina:

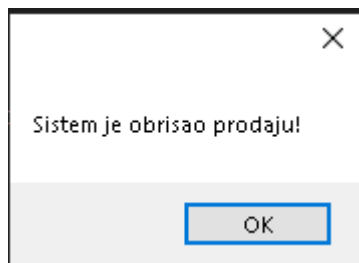
Dodaj stavku Obrisi stavku

	RedniBroj	Kolicina	IznosStavke	Videolgra
▶	1	1	897	COD3
	2	2	5000	NFS

Iznos za naplatu: 10897 din

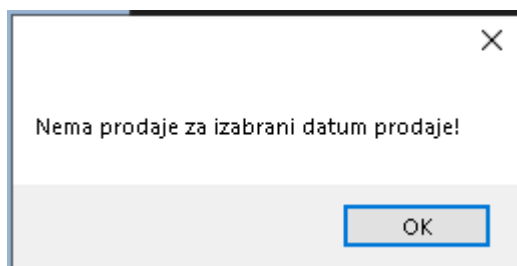
Izmeni prodaju Obrisi prodaju Odustani

9. **Radnik** poziva sistem da stornira zadati **račun**. (APSO)
10. **Sistem** stornira **račun**. (SO)
11. **Sistem** prikazuje radniku obrisani **račun** i poruku: "Sistem je obrisao prodaju!". (IA)

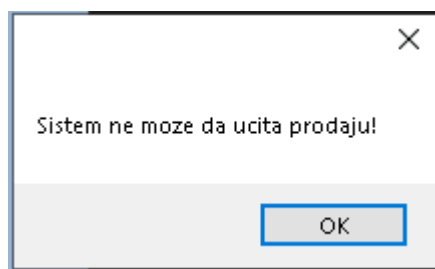


Alternativna scenarija SK:

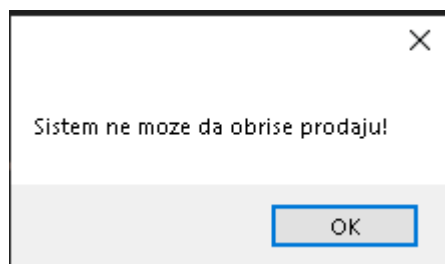
- 4.1 Ukoliko sistem ne može da pronađe **račune**, obaveštava prodavca o tome: "Nema prodaje za izabrani datum prodaje!". Prekida se izvršavanje scenarija. (IA)



- 8.1 Ukoliko sistem ne može da prikaže podatke o **računu**, obaveštava prodavca o tome: "Sistem ne moze da ucita prodaju!". Prekida se izvršenje scenarija. (IA)



- 11.1 Ukoliko sistem ne može da obriše **račun**, obaveštava prodavca o tome: "Sistem ne moze da obrise prodaju!". (IA)



3.3.8. Slučaj korišćenja 8: Pretraga računa

Naziv SK: Pretraga računa

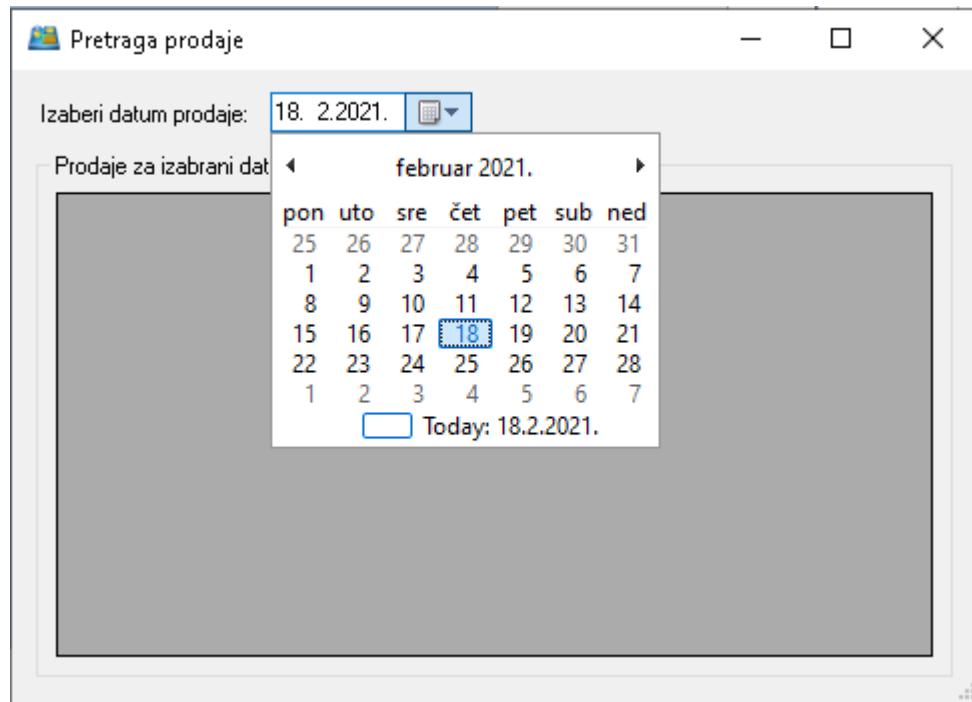
Aktori SK: Prodavac

Učesnici SK: Prodavac i sistem

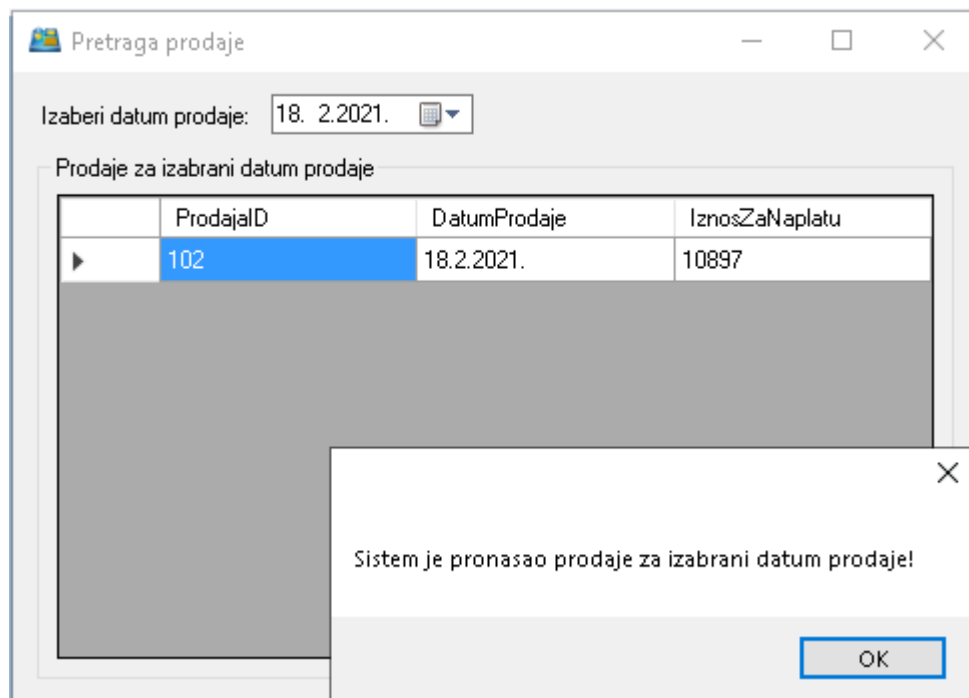
Preduslov: Sistem je uključen. Sistem prikazuje formu za rad sa računom. Učitane su video igre.

Osnovni scenario SK:

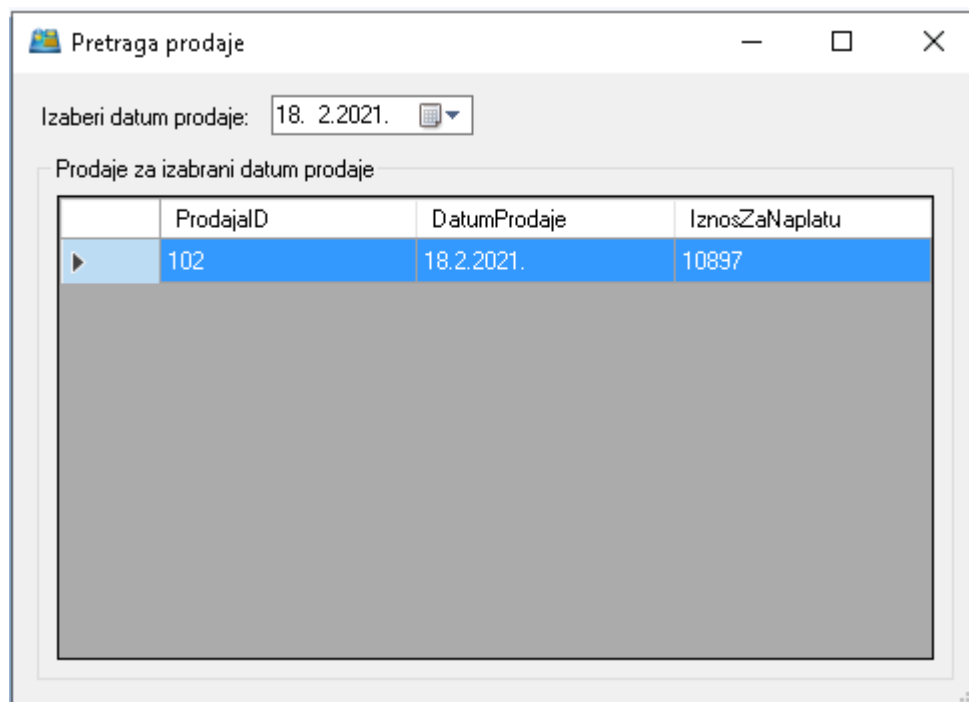
1. **Prodavac** unosi vrednost po kojoj vrši pretragu računa. (APUSO)



2. **Prodavac** poziva sistem da izvrši pretragu računa po zadatoj vrednosti. (APSO)
3. **Sistem** izvršava pretragu računa po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje prodavcu podatke o računima i poruku: "Sistem je pronasao prodaje za izabrani datum prodaje!". (IA)



5. **Prodavac** bira **račun** čije podatke želi da prikaže. (APUSO)



6. **Prodavac** poziva sistem da prikaže podatke o izabranom **računu**. (APSO)
7. **Sistem** traži podatke o traženom **računu**. (SO)
8. **Sistem** prikazuje podatke o traženom **računu** i poruku: "Sistem je uspesno učitao prodaju!". (IA)

Detalji prodaje

Datum prodaje: 18. 2.2021. Prodavac: Marta Martić

Stavke prodaje

Video igra: COD3 Kolicina:

Dodaj stavku Obrisi stavku

	RedniBroj	Kolicina	IznosStavke	Videolgra
▶	1	1	897	COD3
	2	2	5000	NFS

Iznos za naplatu: 10897 din

Izmeni prodaju Obrisi prodaju

Sistem je uspesno ucitao prodaju!

OK

Alternativna scenarija SK:

- 4.1 Ukoliko sistem ne može da pronađe **račune**, obaveštava prodavca o tome: “Nema prodaje za izabrani datum prodaje!”. Prekida se izvršenje scenarija. (IA)

Nema prodaje za izabrani datum prodaje!

OK

- 8.1 Ukoliko sistem ne može da pronađe podatke o željenom **računu**, obaveštava prodavca o tome: “Sistem ne moze da ucita prodaju!”. (IA)

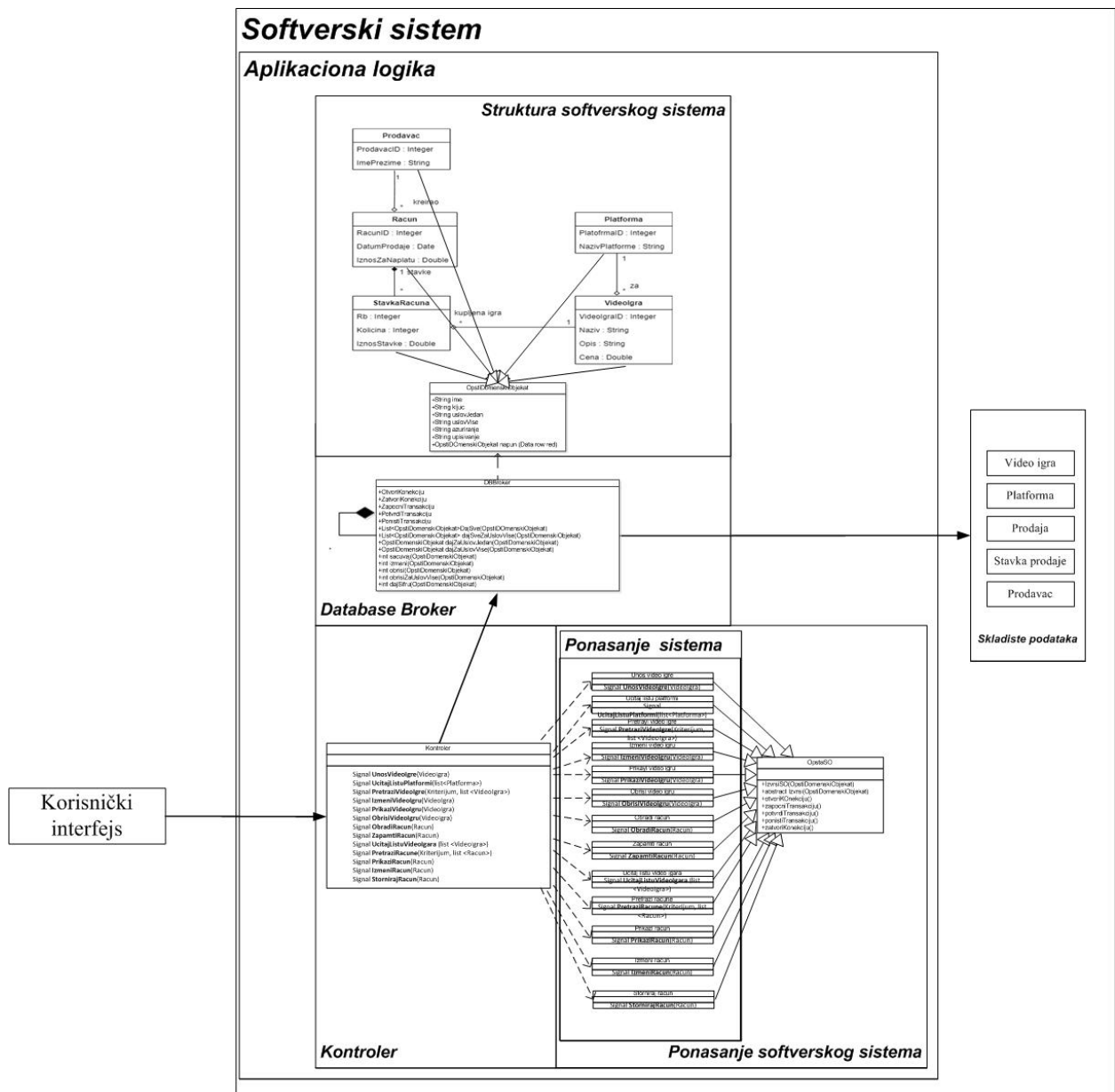
Sistem ne moze da ucita prodaju!

OK

3.4. Projektovanje kontrolera korisničkog interfejsa

Aplikaciona logika služi za opisivanje strukture i ponašanja softverskog sistema i projektuje se nezavisno od korisničkog interfejsa i obrnuto. Drugim rečima, aplikaciona logika (koja predstavlja *Model* u MVC paternu) nema znanja o tome gde se nalazi korisnički interfejs (koja predstavlja *View* u MVC paternu).

Kontroler je odgovoran da prihvati zahtev za izvršenje sistemske operacije od klijenta i da ga prosledi do poslovne logike koja je odgovorna za izvršenje sistemske operacije.



Slika 32 - Slika KKI Korisnički interfejs u kontekstu arhitekture softverskog sistema

3.5. Projektovanje aplikacione logike

3.5.1. Kontroler aplikacione logike

Kontroler aplikacione logike treba da podigne serverski soket koji će da osluškuje mrežu. Kada klijent (klijentski soket) uspostavi konekciju sa kontrolerom (serverskim soketom), tada kontroler treba da generiše nit koja će uspostaviti dvosmernu vezu sa klijentom (ulaznu i izlaznu). Slanje i primanje podataka od klijenta se ostvaruje preko soketa.

Klijent šalje zahtev za izvršenje neke od SO do odgovarajuće niti (koju smo nazvali "Obrada"), koja je povezana sa tim klijentom. "Obrada" prima zahtev i dalje ga preusmerava do klasa koje su odgovorne za izvršenje SO. Nakon izvršenja SO rezultat se vraća do aplikacione logike, odnosno do "Obrade", koja taj rezultat šalje nazad do klijenta.

Primer specifičnog kontrolera aplikacione logike

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net;
using System.Runtime.Serialization.Formatters.Binary;
using System.Net.Sockets;
using Biblioteka;

namespace Komunikacija
{
    public class Komunikacija
    {
        TcpClient klijent;
        BinaryFormatter formater;
        NetworkStream tok;

        public bool poveziSeNaServer()
        {
            try
            {
                klijent = new TcpClient("127.0.0.1", 10001);
                tok = klijent.GetStream();
                formater = new BinaryFormatter();
                return true;
            }
            catch (Exception)
            {
                return false;
            }
        }

        public void kraj()
        {
            TransferKlasa transfer = new TransferKlasa();
            transfer.Operacija = Operacije.Kraj;
            formater.Serialize(tok, transfer);
        }

        public object vratiSvePlatforme()
```

```

{
    TransferKlasa transfer = new TransferKlasa();
    transfer.Operacija = Operacije.VratiSvePlatforme;
    transfer.TransferObjekat = new Platforma();
    formater.Serialize(tok, transfer);

    transfer = formater.Deserialize(tok) as TransferKlasa;
    return transfer.Rezultat;
}

public object kreirajVideoIgru(VideoIgra vi)
{
    TransferKlasa transfer = new TransferKlasa();
    transfer.Operacija = Operacije.KreirajVideoIgru;
    transfer.TransferObjekat = vi;
    formater.Serialize(tok, transfer);

    transfer = formater.Deserialize(tok) as TransferKlasa;
    return transfer.Rezultat;
}

public object zapamtiVideoIgru(VideoIgra vi)
{
    TransferKlasa transfer = new TransferKlasa();
    transfer.Operacija = Operacije.ZapamtiVideoIgru;
    transfer.TransferObjekat = vi;
    formater.Serialize(tok, transfer);

    transfer = formater.Deserialize(tok) as TransferKlasa;
    return transfer.Rezultat;
}

public object pretraziProdaje(Prodaja p)
{
    TransferKlasa transfer = new TransferKlasa();
    transfer.Operacija = Operacije.PretraziProdaje;
    transfer.TransferObjekat = p;
    formater.Serialize(tok, transfer);

    transfer = formater.Deserialize(tok) as TransferKlasa;
    return transfer.Rezultat;
}

public object ucitajProdaju(Prodaja p)
{
    TransferKlasa transfer = new TransferKlasa();
    transfer.Operacija = Operacije.UcitajProdaju;
    transfer.TransferObjekat = p;
    formater.Serialize(tok, transfer);

    transfer = formater.Deserialize(tok) as TransferKlasa;
    return transfer.Rezultat;
}

public object obrisiProdaju(Prodaja p)
{
    TransferKlasa transfer = new TransferKlasa();
    transfer.Operacija = Operacije.ObrisiProdaju;
    transfer.TransferObjekat = p;
    formater.Serialize(tok, transfer);

    transfer = formater.Deserialize(tok) as TransferKlasa;
    return transfer.Rezultat;
}

public object Odustani(Object o)
{

```

```

        TransferKlasa transfer = new TransferKlasa();
        transfer.Operacija = Operacije.Odustani;
        transfer.TransferObjekat = o;
        formater.Serialize(tok, transfer);

        transfer = formater.Deserialize(tok) as TransferKlasa;
        return transfer.Rezultat;
    }

    public object obrisiVideoIgru(VideoIgra vi)
    {
        TransferKlasa transfer = new TransferKlasa();
        transfer.Operacija = Operacije.ObrisiVideoIgru;
        transfer.TransferObjekat = vi;
        formater.Serialize(tok, transfer);

        transfer = formater.Deserialize(tok) as TransferKlasa;
        return transfer.Rezultat;
    }

    public object ucitajVideoIgru(VideoIgra vi)
    {
        TransferKlasa transfer = new TransferKlasa();
        transfer.Operacija = Operacije.UcitajVideoIgru;
        transfer.TransferObjekat = vi;
        formater.Serialize(tok, transfer);

        transfer = formater.Deserialize(tok) as TransferKlasa;
        return transfer.Rezultat;
    }

    public object pretraziVideoIgre(VideoIgra vi)
    {
        TransferKlasa transfer = new TransferKlasa();
        transfer.Operacija = Operacije.PretraziVideoIgre;
        transfer.TransferObjekat = vi;
        formater.Serialize(tok, transfer);

        transfer = formater.Deserialize(tok) as TransferKlasa;
        return transfer.Rezultat;
    }

    public object kreirajProdaju(Prodaja p)
    {
        TransferKlasa transfer = new TransferKlasa();
        transfer.Operacija = Operacije.KreirajProdaju;
        transfer.TransferObjekat = p;
        formater.Serialize(tok, transfer);

        transfer = formater.Deserialize(tok) as TransferKlasa;
        return transfer.Rezultat;
    }

    public object zapamtiProdaju(Prodaja p)
    {
        TransferKlasa transfer = new TransferKlasa();
        transfer.Operacija = Operacije.ZapamtiProdaju;
        transfer.TransferObjekat = p;
        formater.Serialize(tok, transfer);

        transfer = formater.Deserialize(tok) as TransferKlasa;
        return transfer.Rezultat;
    }

    public object vratiSveProdavce()
    {

```

```

        TransferKlasa transfer = new TransferKlasa();
        transfer.Operacija = Operacije.VratiSveProdavce;
        transfer.TransferObjekat = new Prodavac();
        formater.Serialize(tok, transfer);

        transfer = formater.Deserialize(tok) as TransferKlasa;
        return transfer.Rezultat;
    }

    public object vratiSveVideoIgre()
    {
        TransferKlasa transfer = new TransferKlasa();
        transfer.Operacija = Operacije.VratiSveVideoIgre;
        transfer.TransferObjekat = new VideoIgra();
        formater.Serialize(tok, transfer);
        Object tmp = null;
        try
        {
            tmp = formater.Deserialize(tok);
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
            throw;
        }

        transfer = tmp as TransferKlasa;
        return transfer.Rezultat;
    }
}

```

3.5.2. Sistemske operacije

U prethodnoj fazi, fazi analize, odredili smo ugovore o sistemskim operacijama, pri čemu smo rekli da jedan ugovor opisuje ponašanje jedne sistemske operacije, tako što opisuje ŠTA operacija treba da radi, ali ne i kako. Sada se u fazi projektovanja za svaki od ugovora projektuje konceptualno rešenje (realizacija) SO. To znači da ćemo za svaku klasu odgovornu za izvršenje SO definisati KAKO će se sistemska operacija izvršiti.

Na samom početku projektovanja SO izbeći ćemo aspekte realizacije koji su vezani za konekciju sa bazom, perzistentnost i transakcije.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Biblioteka;
using Sesija;

namespace SistemskeOperacije
{
    public abstract class OpstaSO
    {
        public Object izvrsiSO(OpstiDomenskiObjekat odo)
        {
            Object rezultat = null;
            Broker.dajSesiju().otvoriKonekciju();
            Broker.dajSesiju().zapocniTransakciju();
            try

```

```

    {
        rezultat = Izvrsi(odo);
        Broker.dajSesiju().potvrdiTransakciju();
    }
    catch (Exception)
    {
        Broker.dajSesiju().ponistiTransakciju();
    }
    finally
    {
        Broker.dajSesiju().zatvoriKonekciju();
    }

    return rezultat;
}

public abstract object Izvrsi(OpstiDomenskiObjekat odo);
}
}

```

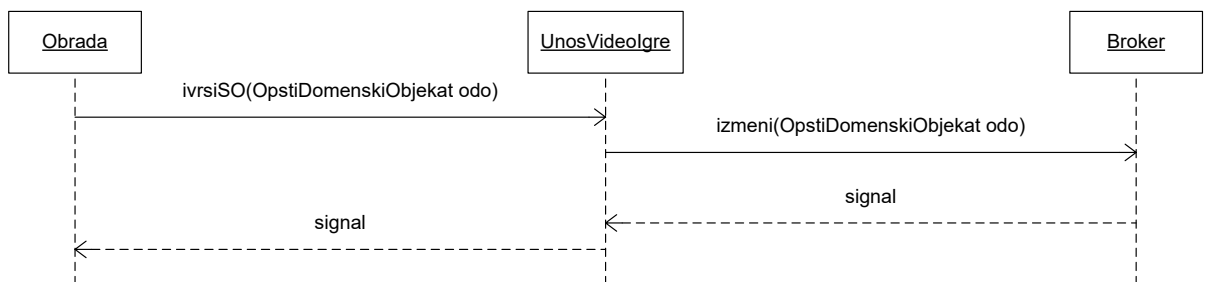
Ugovor UG1 : Unos video igre

Operacija: Signal UnosVideoIgre(Videolgra);

Veza sa SK: SK1

Preduslovi: Mora biti zadovoljeno prosto vrednosno I strukturno ograničenje nad objektom **Videolgra**. Vraćena lista platformi.

Postuslovi: Izvršen je unos video igre.



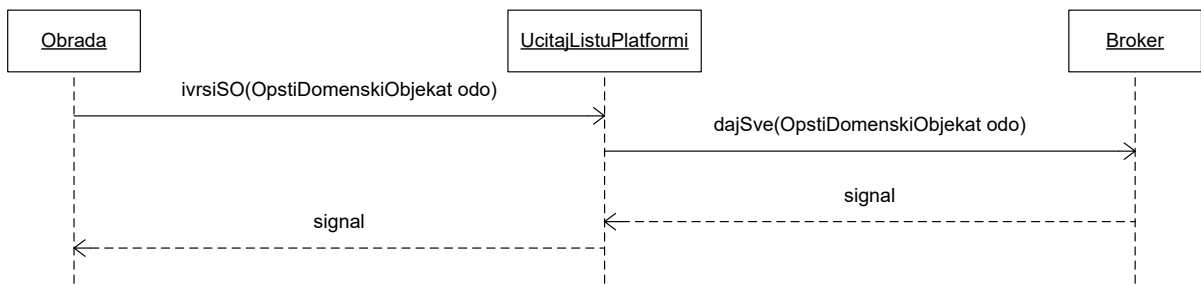
Ugovor UG2 : Učitavanje liste platformi

Operacija: Signal UcitajListuPlatformi(list<Platofrma>);

Veza sa SK: SK1, SK2, SK3, SK4

Preduslovi: /

Postuslovi: /



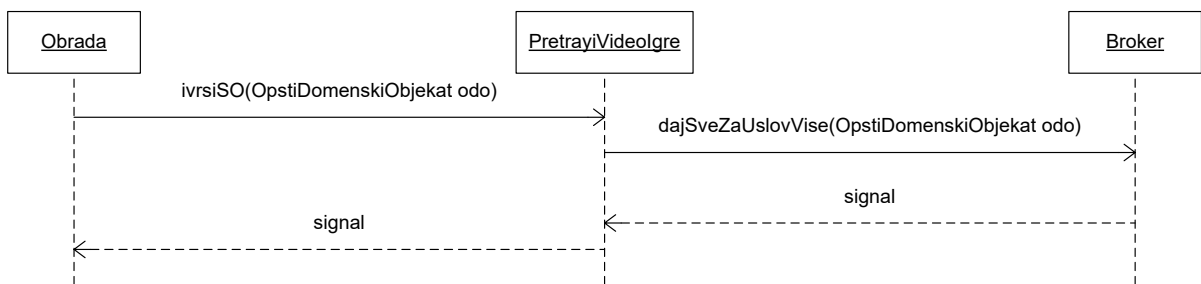
Ugovor UG3 : Pretraga video igara

Operacija: Signal PretraziVideolgre(Kriterijum, list<Videolgra>);

Veza sa SK: SK2, SK3, SK4

Preduslovi: /

Postuslovi: /



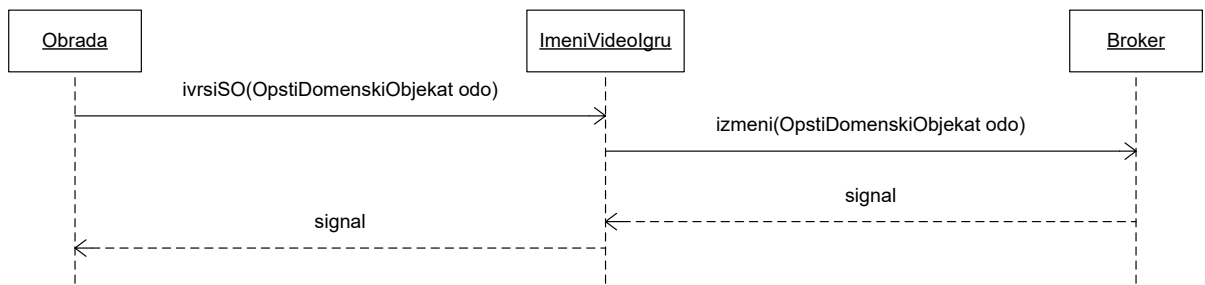
Ugovor UG4 : Izmena video igre

Operacija: Signal IzmeniVideolgru(Videolgra);

Veza sa SK: SK2

Preduslovi: Mora biti zadovoljeno prosto vrednosno I strukturno ograničenje nad objektom **Videolgra**. Vraćena je lista platformi.

Postuslovi: Izvršena je izmena video igre.



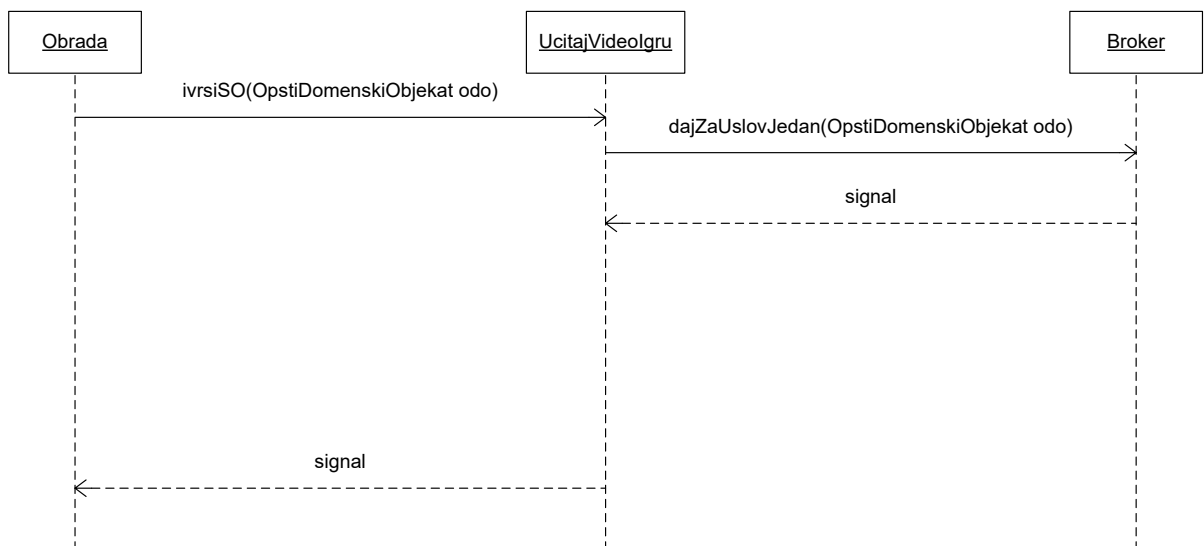
Ugovor UG5 : Prikaz video igre

Operacija: Signal PrikaziVideolgru(Videolgra);

Veza sa SK: SK2, SK4

Preduslovi: /

Postuslovi: /



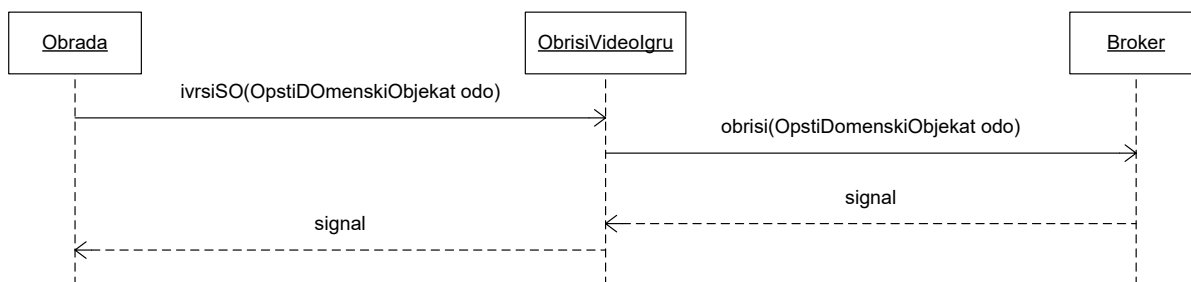
Ugovor UG6 : Brisanje video igre

Operacija: Signal ObrisiVideolgru(Videolgra);

Veza sa SK: SK3

Preduslovi: Mora biti zadovoljeno prosto vrednosno I strukturno ograničenje nad objektom **Videolgra**.

Postuslovi: Izvršeno je brisanje video igre.



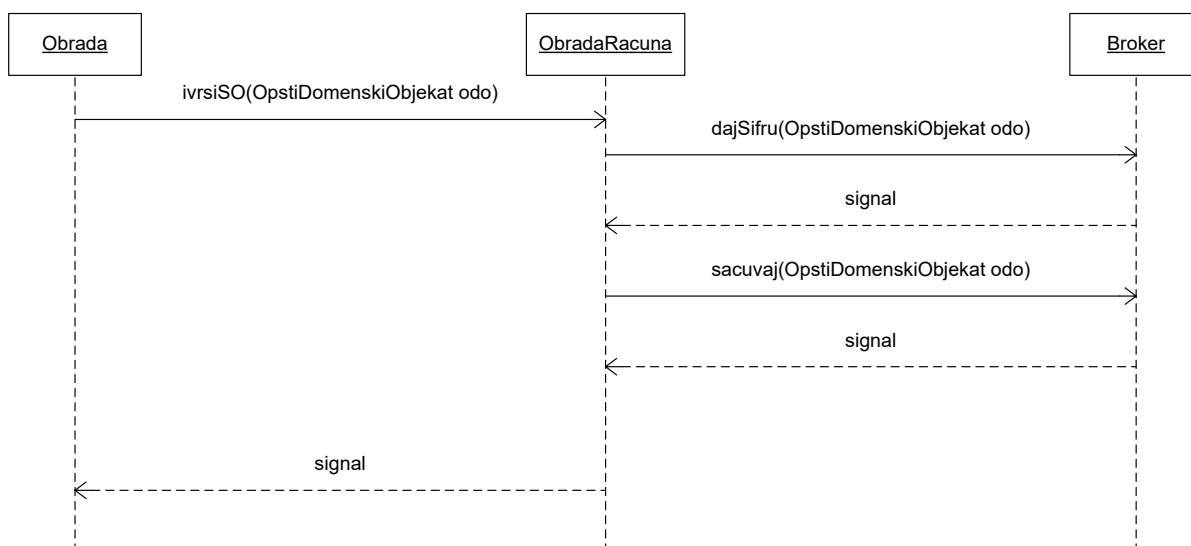
Ugovor UG7 : Obrada računa

Operacija: Signal ObradiRacun(Racun);

Veza sa SK: SK5

Preduslovi: /

Postuslovi: /



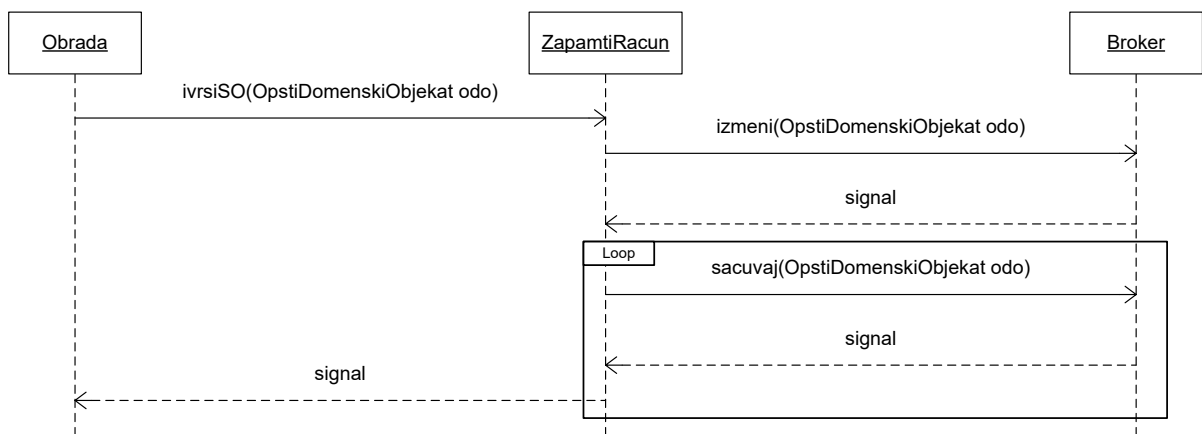
Ugovor UG8 : Čuvanje računa

Operacija: Signal ZapamtiRacun(Racun);

Veza sa SK: SK5

Preduslovi: Mora biti zadovoljeno prosto vrednosno I strukturno ograničenje nad objektom **Racun**.
Vraćena je lista video igara.

Postuslovi: Izvršen je unos računa.



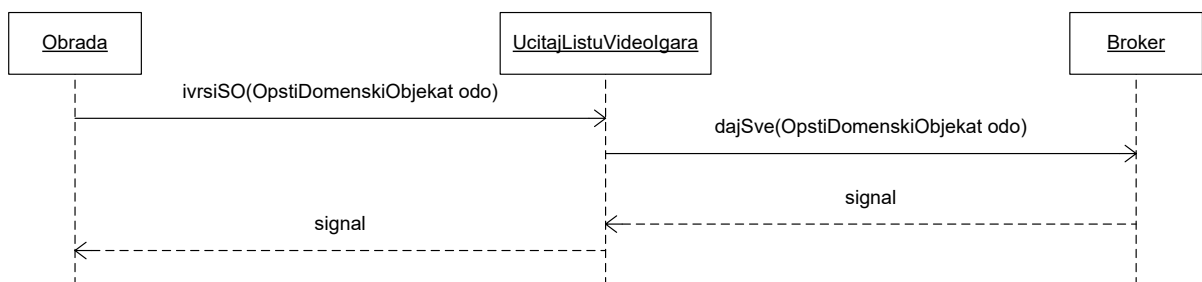
Ugovor UG9 : Učitavanje liste video igara

Operacija: Signal UcitajListuVideolgara(list<Videolgara>);

Veza sa SK: SK5, SK6, SK7, SK8

Preduslovi: /

Postuslovi: /



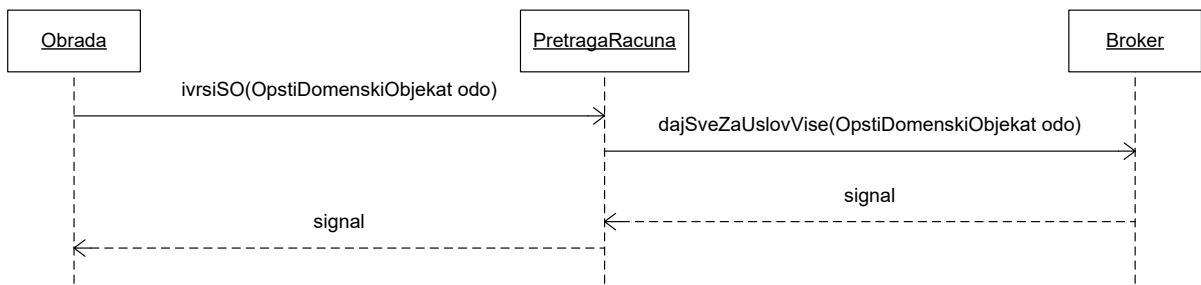
Ugovor UG10 : Pretraga računa

Operacija: Signal PretraziRacune(Kriterijum, list<Racun>);

Veza sa SK: SK6, SK7, SK8

Preduslovi: /

Postuslovi: /



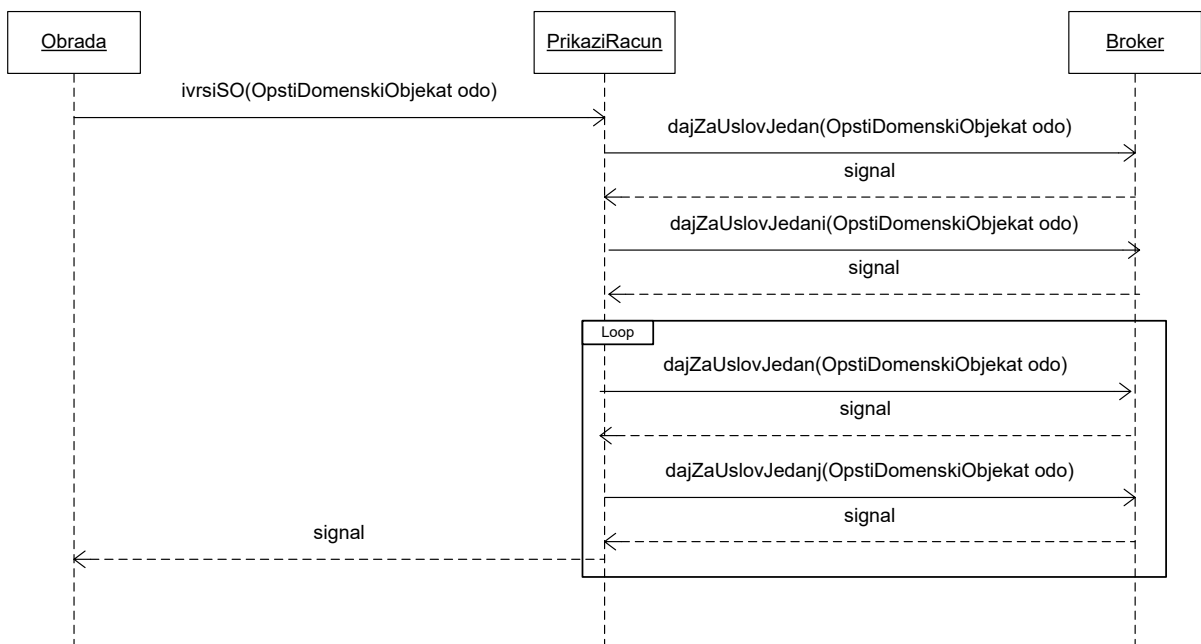
Ugovor UG11 : Prikaz računa

Operacija: Signal PrikaziRacun(Racun);

Veza sa SK: SK6, SK7, SK8

Preduslovi: /

Postuslovi: /



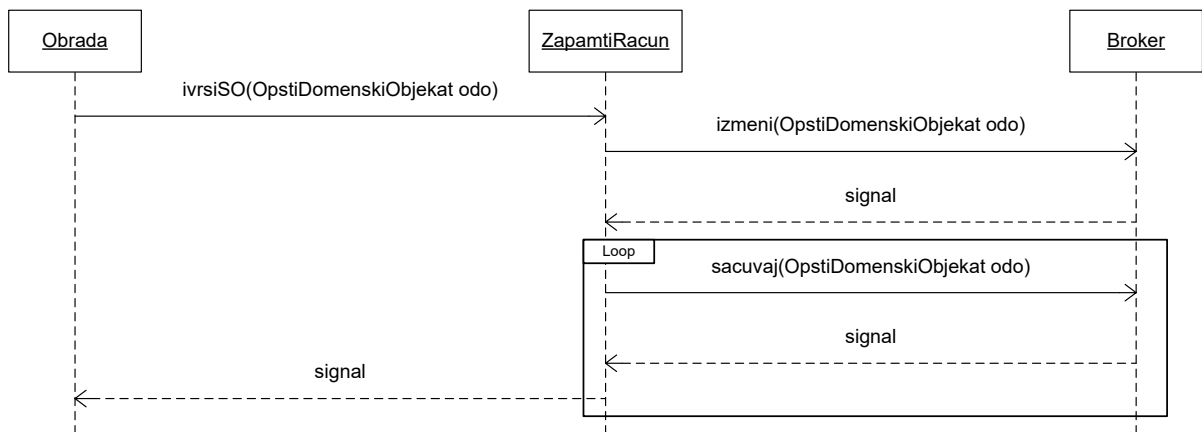
Ugovor UG12 : Izmena računa

Operacija: Signal IzmenaRacuna(Racun);

Veza sa SK: SK6

Preduslovi: Mora biti zadovoljeno prosto vrednosno I strukturno ograničenje nad objektom **Racun**.
Vraćena je lista video igara.

Postuslovi: Izvršena je izmena računa.



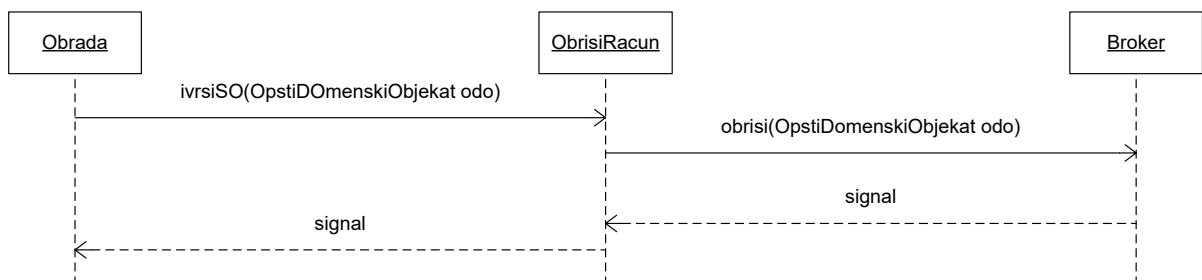
Ugovor UG13 : Storniranje računa

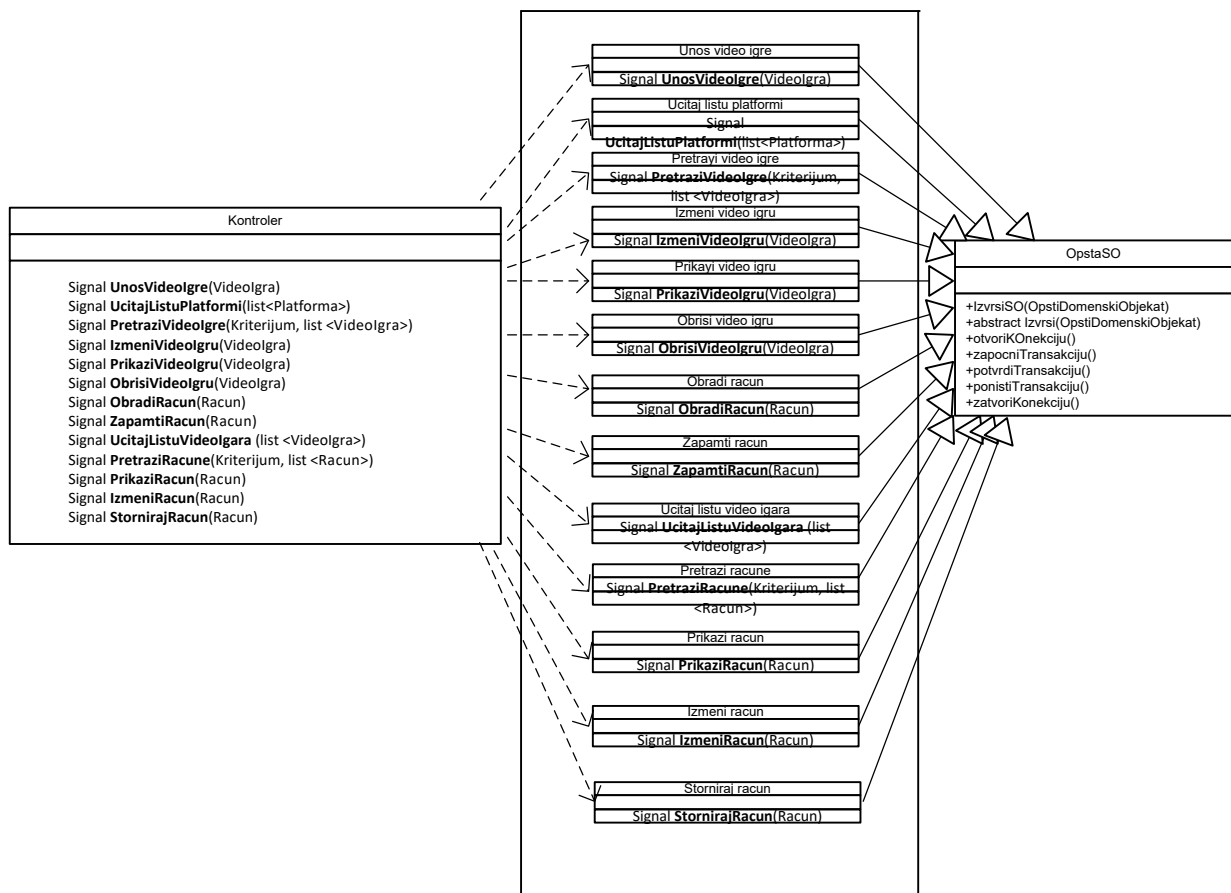
Operacija: Signal StornirajRacun(Racun);

Veza sa SK: SK7

Preduslovi: Mora biti zadovoljeno prosto vrednosno I strukturno ograničenje nad objektom **Racun**.

Postuslovi: Izvršeno je storniranje računa.





Slika 33 - Klase koje su odgovorne za SO nasleđuju klasu OpstaSO

3.5.3. Domenske klase

Na osnovi konceptualnih klasa prave se softverske klase strukture. Svaka klasa sadrži privatna polja atributa, getter-e, setter-e za iste, konstruktore (bilo bezparametrizovani bilo parametrizovani).

Domenske klase implementiraju interfejs *OpstiDomenskiObjekat* kako bi omogućili lakše implementiranje metoda *DatabaseBroker*-a. Na taj način *DatabaseBroker* prima interfejs umesto samih klasa i olakšava kreiranje generičkih upita.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;

namespace Biblioteka
{
    public interface OpstiDomenskiObjekat
    {
        string tabela
        {
            get;
        }

        string kljuc
        {
            get;
        }

        string uslovJedan
        {
            get;
        }

        string uslovVise
        {
            get;
        }

        string azuriranje
        {
            get;
        }

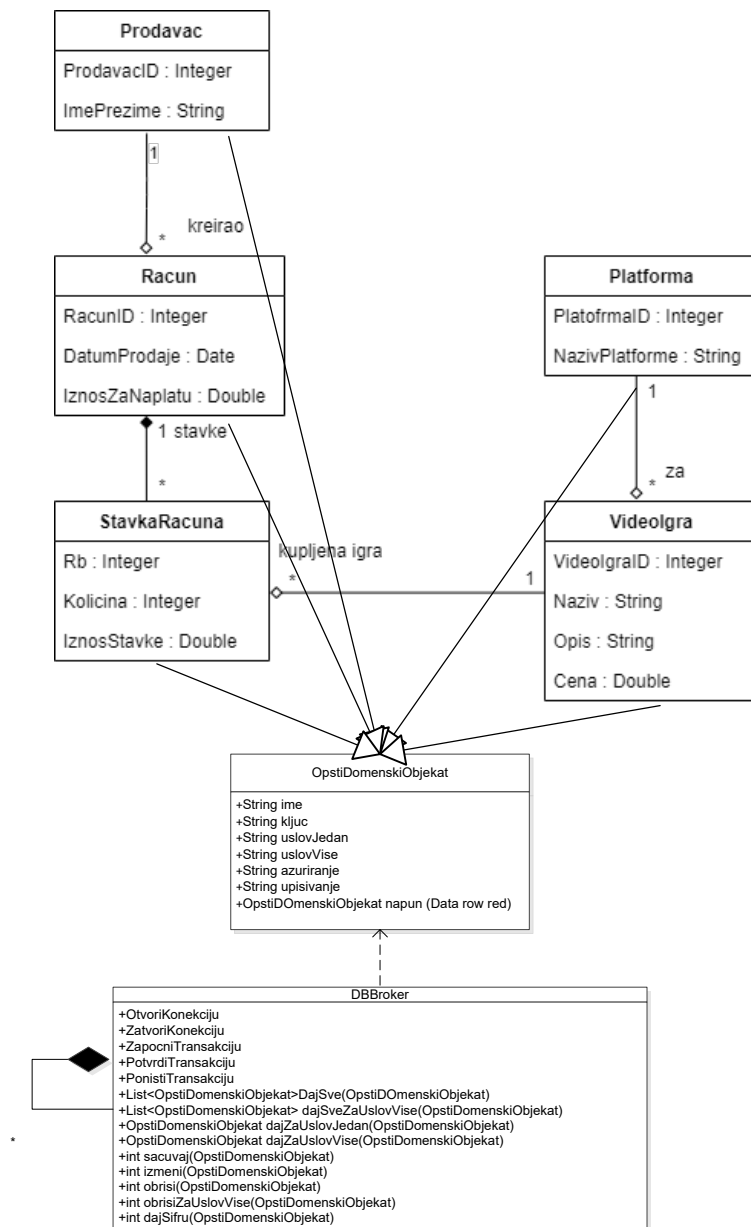
        string upisivanje
        {
            get;
        }

        OpstiDomenskiObjekat napuni(DataRow red);
    }
}
```


3.5.4. Broker baze podataka

Broker baze podataka je softverska klasa odgovorna za komunikaciju između poslovne logike i skladišta podataka. Drugim rečima, projektuje se kako bi obezbedio perzistentni servis objektima domenskih klasa koji se čuvaju u bazi podataka. Klasa *DatabaseBroker* predstavlja perzistentni okvir koji posreduje u svim operacijama nad bazom podataka i realizuje sledeće metode:

```
dajSesiju()  
dajSifru(Biblioteka.OpstiDomenskiObjekat)  
dajSve(Biblioteka.OpstiDomenskiObjekat)  
dajSveZaUslovVise(Biblioteka.OpstiDomenskiObjekat)  
dajZaUslovJedan(Biblioteka.OpstiDomenskiObjekat)  
dajZaUslovVise(Biblioteka.OpstiDomenskiObjekat)  
izmeni(Biblioteka.OpstiDomenskiObjekat)  
obrisi(Biblioteka.OpstiDomenskiObjekat)  
obrisiZaUslovVise(Biblioteka.OpstiDomenskiObjekat)  
otvoriKonekciju()  
ponistiTransakciju()  
potvrdiTransakciju()  
sacuvaj(Biblioteka.OpstiDomenskiObjekat)  
zapocniTransakciju()  
zatvoriKonekciju()  
instanca  
konekcija  
transakcija
```



Slika 34 - Database broker klasa se povezuje sa klasom OpstiDomenskiObjekat

3.5.5. Projektovanje skladišta podataka

Na osnovi strukture softverskih klasa projektovane su tabele (skladišta podataka) relacionog sistema za upravljanje bazom podataka. U ovom radu je korišćen MS Access.

Platforma			
	PlatformaID	NazivPlatforme	Click to Add
+	1	PC	
+	2	PlayStation 5	
+	3	PlayStation 4	
+	4	Nintendo Switch	
+	5	Xbox series X	
+	6	Xbox One	
+	7	Xbox Series S	
+	8	Nintendo Switch	
*			

Platforma

Prodaja					
	ProdajaID	DatumProd	IznosZaNaplatu	ProdavacID	Click to Add
+	8	3.7.2020.	300	1	
+	16	13.8.2020.	0	1	
+	51	17.8.2020.	18450	1	
+	53	17.8.2020.	0	1	
+	68	17.8.2020.	1950	1	
+	69	19.8.2020.	22500	1	
+	70	24.9.2020.	1700	1	
+	72	12.12.2020.	5000	1	
+	81	11.2.2021.	850	1	
+	82	11.2.2021.	1000	4	
+	84	,2021. 03:01:33	1700	1	
+	88	16.2.2021.	850	1	
+	94	16.2.2021.	11700	1	
+	96	16.2.2021.	0	1	
+	98	16.2.2021.	500	1	
+	99	16.2.2021.	5020	1	
*					

Prodaja

Prodavac			
	ProdavacID	ImePrezime	Click to Add
+	1	Marija Marijan	
+	2	Veljko Veljkov	
+	3	Filip Filipovic	
+	4	Marta Martic	
+	5	Aleksandar Ra	
*			

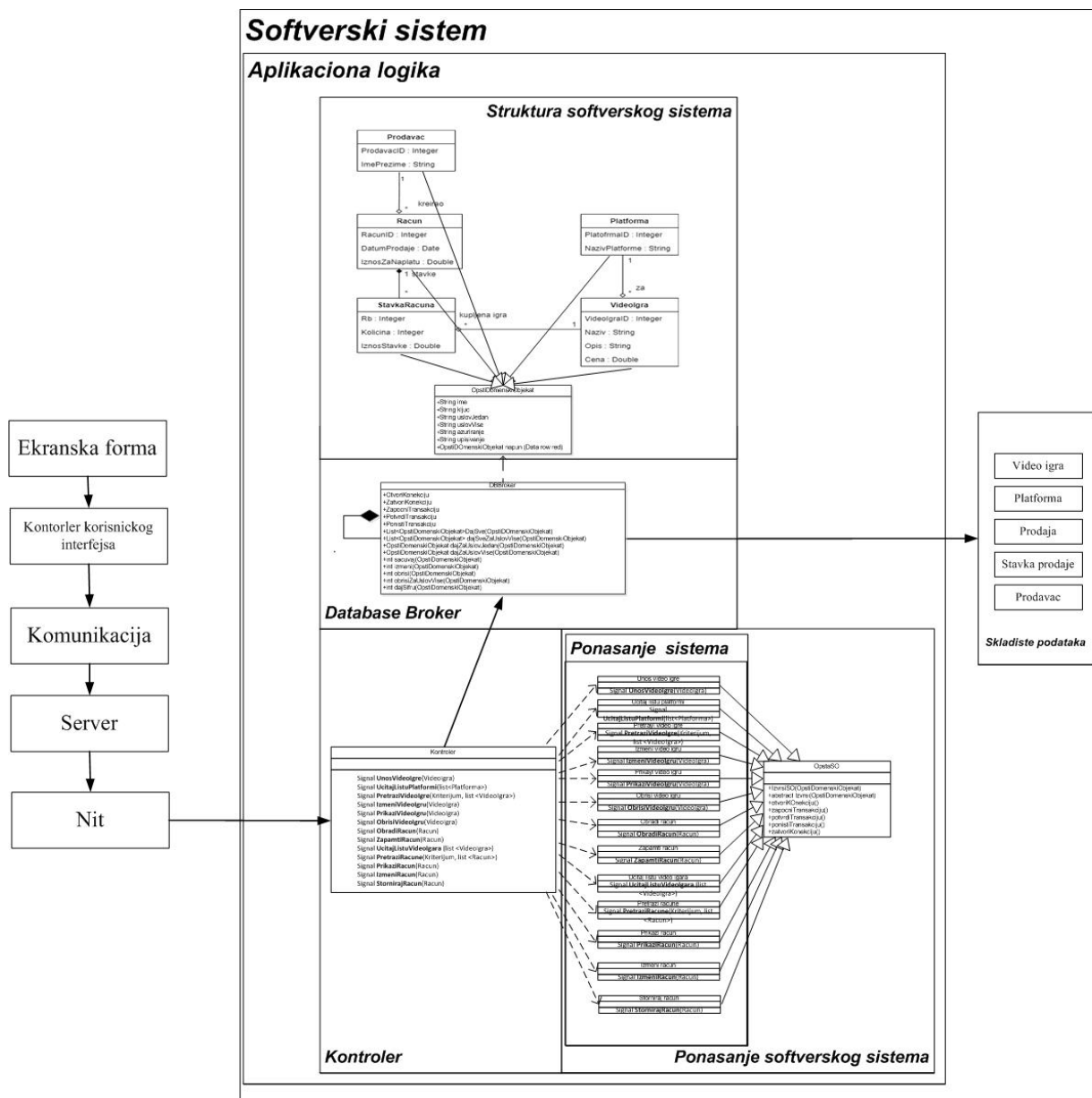
Prodavac

StavkaProdaje						
ProdajaID	RB	Kolicina	IznosStavke	VideolgraID	Click to Add	
3	1	3	300	1		
51	1	5	1450	1		
68	1	3	1950	1		
69	1	2	2500	1		
70	1	1	850	1		
70	2	1	850	1		
72	1	3	5000	3		
81	1	1	850	1		
82	1	1	1000	2		
84	1	1	850	1		
84	2	1	850	1		
88	1	1	850	1		
94	1	1	5000	3		
94	2	1	5000	3		
94	3	1	850	1		
94	4	1	850	1		
98	1	1	500	19		
99	1	1	5020	9		
*						

StavkaProdaje

Videolgra					
VideolgraID	Naslov	Opis	Cena	PlatformaID	Click to Add
1	COD2	opis1	850	6	
2	F12020	opis2	1000	2	
3	Stronghold	opis3	5000	6	
5	NFS	opis4	5000	6	
6	NBA 2K20	opis5	20000	8	
9	F1 car racing	opis1	5020	1	
10	Battlefield 5	Pucacina klasinca	5000	1	
11	Dark Souls	Hard RPG	1000	1	
16	Dragon Age	RPG zanr	1002	1	
18	RIP	adsd	500	4	
19	BLA		500	1	
20	Street Fighter	Fight Simulator	500	1	
21	Jade Empire	RPG	780	3	
*			0		

Videolgra

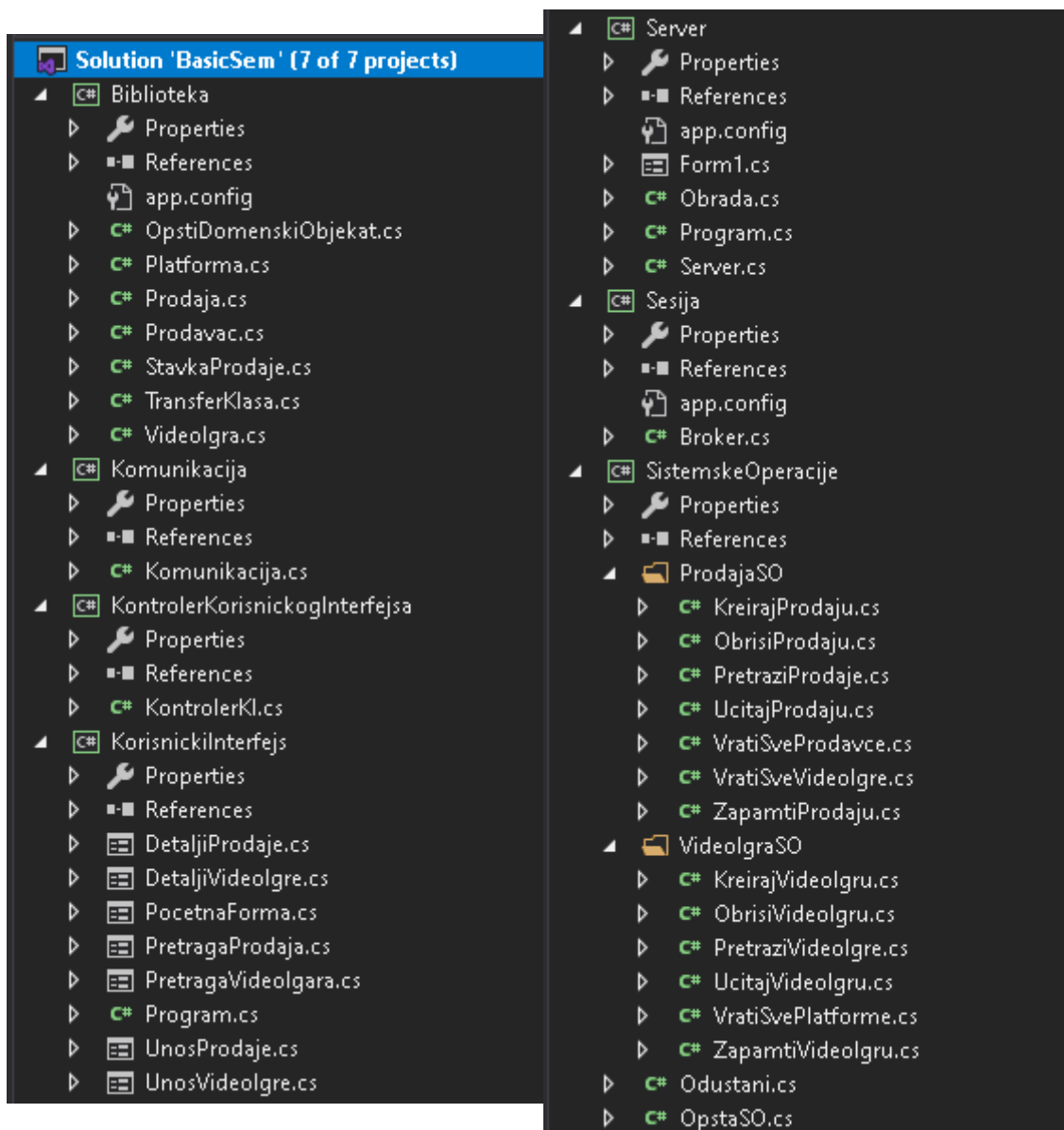


Slika 35 - ASSTBP Arhitektura soft. Sistema nakon projektovanja tabela baze podataka

4. Implementacija

Softverski sistem, rezultat ovog rada, razvijen je u programskom jeziku C#, a izvorni kod je priložen na pratećem USB-u. Sistem je projektovan kao klijent-server. Kao razvojno okruženje korišćen je Visual Studio 2019. Kao sistem za upravljanjem bazom podataka korišćen je MSACCESS.

Na osnovi arhitekture softverskog sistema dobijene su sledeće softverske klase:



5. Testiranje

Svaki od implementiranih slučajeva korišćenja je testiran. Prilikom testiranja svakog slučaja korišćenja pored unetih pravilnih podataka unošeni su i nepravilni podaci da bi se utvrdilo kakav će rezultat izvršenja biti. Na osnovi izvršenih testiranja otklonjeni su uočeni nedostaci.

6. Literatura

Projektovanje softvera – Skripta, dr Siniša Vlajić, Beograd - 2020.