



Jesenji semestar, 2017/18

PREDMET: SE201 - Uvod u softversko inženjerstvo

Online sistem za distribuciju igrice

(Projektni zadatak)

Profesor: Dragan Domazet

Asistent: Lazar Mrkela

AUTOR

Sava Jeremić 2733

sava.jeremic.2733@metropolitan.ac.rs

Sadržaj

Sadržaj.....	1
1. Uvod.....	4
1.1 Opis i svrha dokumenta.....	4
1.2 Definicije, akronimi i skraćenice.....	4
1.3 Alati i metodologija izrade.....	5
1.4 Reference.....	5
2. Opšti opis.....	6
2.1 Funkcije i korisnici sistema.....	6
2.2 Perspektiva proizvoda.....	6
2.2.1 Interfejs sistema:.....	6
2.2.2 Korisnički interfejs:.....	7
2.2.3 Hardverski interfejs:.....	7
2.2.4 Softverski interfejs:.....	7
2.2.5 Komunikacioni interfejs:.....	7
2.2.6 Memorijski zahtevi:.....	7
2.3 Zahtevi korisnika.....	7
2.4 Uslovi i ograničenja.....	7
2.5 Pretpostavke i ograničenja.....	7
3. Specifični softverski zahtevi.....	8
3.1 Funkcionalni zahtevi administratori.....	8
3.2 Administrator.....	8
3.2.1 User Database.....	8
3.2.1.1 User Viewer.....	9
3.2.1.2 New User.....	9
3.2.1.3 Edit User.....	9
3.2.1.4 Remove User.....	9
3.2.2 Game Database.....	9
3.2.2.1 Game Add.....	9
3.2.2.2 Game Update.....	9
3.2.2.3 Game Remove.....	9
3.3 Korisnik.....	10
3.3.1 Game Database.....	10
3.3.1.1 Game Viewer.....	10
3.3.1.2 Game Search.....	10

3.3.1.3 Game Genre.....	10
3.3.2 User Registration.....	10
3.3.2.1 User Profile.....	10
3.3.2.2 User Cart.....	11
3.3.2.3 User Games.....	11
3.3.3 Remote Access.....	11
3.3.3.1 Internet Access.....	11
3.2 Nefunkcionalni zahtevi.....	11
3.2.1 User Friendly Interface and Adaptive Design.....	11
3.2.2 Performances.....	12
3.2.3 Security.....	12
4. Slučajevi korišćenja sistema.....	13
4.1 Korisnik.....	13
4.2 Administrator.....	17
5. Apstraktni dijagram sekvenci(sistem sekvencijalni dijagram).....	20
5.1 Korisnik.....	20
5.1.1 Registracija.....	20
5.1.2 Pregled svih igrica.....	21
5.1.3 Pretraživanje igrica po imenu.....	22
5.1.4 Pegled svih igrica po žanru.....	23
5.1.5 Profil igrice.....	24
5.1.6 Login.....	25
5.1.7 Ažuriranje profila.....	26
5.1.8 Dodavanje igrica u korpu.....	27
5.1.9 Pregled igrica iz korpe.....	27
5.1.10 Brisanje igrica iz korpe.....	28
5.1.11 Kupovina igrica.....	28
5.1.12 Pregled kupljenih igrica.....	29
5.1.13 Logout.....	30
5.2 Administrator.....	31
5.2.1 Prijavljivanje.....	31
5.2.2 Dodavanje igrice.....	32
5.2.3 Ažuriranje igrice.....	33
5.2.4 Odjavljivanje.....	33
5.3 Sekvencijalni dijagram(nivo upotrebljenih klasa).....	34
5.3.1 Korisnik.....	35

5.3.2 Administrator.....	35
5.4 Klasni dijagram.....	36
5.5 Model baze podataka.....	37
6. Arhitektura sistema.....	39
6.1 Šablon arhitekture sistema - MVC.....	39
6.2 Infrastrukturni dijagram.....	40
6.3 Aplikacijski dijagram.....	40
7. Testiranje i implementacija.....	42
7.1 Selenium.....	42
7.2 Testiranje i prikaz softvera.....	42
8. Korisničko uputstvo.....	51
9. Zaključak.....	52

1. Uvod

Tema projektnog zadatka je online sistem za distribuciju igrica, koji će se zvati *GameDream*, odnosno online prodavnica za PC igrice, alternativno reći "*Gaming Client*".

Ovaj dokument nudi sve potrebne funkcionalne i nefunkcionalne zahteve ovog online web sistema, kako i opšti opis sistema, tako i njegovo testiranje, i predstavljanje funkcija sistema kroz dijagrame i tabele.

Ovaj projekat naravno neće imati potpunu funkciju pravog, puno-funkcionalnog gejming klijenta, u smislu pokretanja istog kako bih igrali igre, da li offline ili online, ili naravno sama instalacija igara.

1.1 Opis i svrha dokumenta

Dokument predstavlja opis online sistema koji će se razviti, kao i sve njegove funkcionalnosti. Takođe će biti opisan potreban hardver i softver za neophodan i ispravan rad i pokretanje samog sistema.

1.2 Definicije, akronimi i skraćenice

U ovom segmentu dokumenta, biće navedeni pojmovi koji će se koristiti u dokumentu, koji su bitni za dalje razumevanje i čitanje dokumenta.

U nastavku će se navesti i opisati navedene terminologije koje će se koristiti, kao i neki od termina:

- **HTML**(*Hypertext Markup Language*) - hipertekstualni markup jezik namenjen za opis, odnosno izradu veb stranica. Hipertekst dokument se stvara pomoću HTML jezika.
- **CSS**(*Cascading Style Sheets*) - jezik formatiranja pomoću kojeg se definiše izgled elemenata veb stranice.
- **PHP**(*Hypertext Preprocessor*) - skripting jezik namenjen za dinamičnu i interaktivnu izradu veb sadržaja i on se izvodi na strani servera.
- **Angular 2.4 Typescript MVC** - je front-end framework baziran na TypeScript-u. Koristiće se najnovija verzija Angulara, to je 2.4, ili samo 4, koja je bazirana na typescriptu, sličan javascriptu koji se koristio u predhodnim verzijama angular tehnologije.
- **Bootstrap** - je front-end framework za dizajniranje veb sajtova i aplikacija, baziran je na HTML, CSS i Javascript šablonima za kreiranje i stilizovanje svih potrebnih komponenti interfejsa veb stranice.
- **XAMPP**(*Cross-Platform(X), Apache, MariaDB, PHP, Perl*) - je besplatni i open-source veb server koji se sadrži od Apache HTTP servera, MariaDB baze podataka i interpretera za skripte pisane u PHPu kao i Perlu. Koristeći Apache server ćemo kreirati veb sajt, dok uz pomoć MariaDB, odnosno MySQL servera pristupati bazi podataka, tj PhpMyAdmin.
- **MySQL**(*phpmyadmin*) - koristeći phpmyadmin, besplatan open-source alat za MySQL bazu podataka, preko koje ćemo upravo ubacivati podatke, izvlačiti iz njih iste, kao i brisati.
- **Sybase PowerDesigner** - alat za modelovanje, koju je razvio Sybase. Preko ovog alata ćemo prikazivati sve dijagrame i slučajeve korišćenja.
- **Selenium** - portabilni framework za testiranje veb aplikacija.
- **Google Chrome** - Internet pretraživač.

-
- **CRUD(Create Read Update Delete)** - Kreiranje, čitanje, ažuriranje i brisanje.

1.3 Alati i metodologija izrade

Model softverskog procesa koji će se koristiti jeste vodopad(eng. *the waterfall model*), koji predstavlja planski vođeni proces, jer se ceo proces mora planirati i odrediti termini za sve aktivnosti procesa, pre nego što počne njegovo izvršenje.

Front-end veb tehnologije koje će se koristiti za kreiranje sistema, će biti odrađene u alatima kao što su HTML5, CSS3, JavaScript, PHP, kao i framework-ovi Angular 4, Bootstrap 3...

Za back-end će se koristiti PHP, putem kojeg će biti ostvarena konekcija i povezivanje sa bazom podataka MySQL(PHPMyAdmin), gde ćemo čuvati sve podatke koje će sistem sadržati, kako vezane za korisnika, tako i za igre, kupovinu, itd.

Softver korišćen za izradu modela je PowerDesigner.

Online softver za testiranje koji se koristi jeste Selenium.

1.4 Reference

[1] Angular 2.4: <https://angular.io/docs>

[2] Bootstrap: <https://getbootstrap.com/docs/4.0/getting-started/introduction/>

[3] PHP: <http://php.net/>

[4] Sybase PowerDesigner:
http://infocenter-archive.sybase.com/help/index.jsp?topic=/com.sybase.stf.powerdesigner.doc_s_12.1.0/html/bwug/bwugp12.htm

[5] Software engineering patterns: https://www.tutorialspoint.com/design_pattern/

[6] UML diagrams: <https://www.tutorialspoint.com/uml/index.htm>

[7] Selenium: <http://www.seleniumhq.org/docs/>

2. Opšti opis

Softverski sistem koji će se opisati u ovom delu dokumenta jeste web aplikacija, online sistem za distribuciju i kupovinu igrica, koja pruža mogućnost korisnicima da pre svega postanu član sistema, registrovanjem na online sistem.

Sistem će pritom pružiti korisnicima mogućnost da kupe PC Igre, koje će moći pretraživati, gledati po kategorijama, odnosno žanrovima, kao i kategorijama u smislu najpopularnijih, tek-dolazećih igara, itd.

Korisnik će moći preko svog naloga da vidi svoje kupljene igre, kao i 'wish' listu, odnosno listu želja, koja predstavlja listu igrica koje bih želeo u budućnosti da kupi. Takođe će postojati profil korisnika, u okviru kojeg će se nalaziti njegov avatar, odnosno profilna slika, kao i neke dodatne i opcione informacije o korisniku koje on može postaviti. Korisnik kupuje igrice, tako što ih pre svega stavlja u korpu, u okviru koje može da stavi koliko hoće igrica, ali koje već ne poseduje naravno, i kasnije može kada god želi, da ih ukloni iz korpe ukoliko ne želi da ih kupi, ili sve igrice iz korpe da kupi jednostavnim klikom na dugme.

Sam interfejs i veb aplikacija je adaptivna i intuitivna za korišćenje.

2.1 Funkcije i korisnici sistema

Osnovne funkcije sistema pre svega predstavljaju registrovanje i login na sistem. U okviru toga, korisnik koji nije registrovan da bi ostale funkcije sistema 'otključao', mora se registrovati i ulogovati. Prilikom logovanja, imamo korisnike, i administratora sistema.

Posetioci veb sistema, odnosno korisnici koji nemaju nalog, će moći surfovati sajtom. To podrazumeva pregled svih igrica koje postoje, njihova pretraga kako po imenu tako i žanru, i pregled samih igara po kategorijama.

Registrovani korisnici mogu pored već uključenih funkcionalnosti mogu kupovati igrice. To se obavlja dodavanjem iste u korpu, ili do više igrica, i klikom na dugme koje prebacuje igrice u posed korisničkog naloga. Dalje, korisnik može videti igrice koje on poseduje, kako to, može i stavljati igrice u svoju wishlistu, listu željenih igrica, koji će možda nekada u budućnosti poželeti da kupi. Registrovan korisnik takođe poseduje sopstveni profil, u okviru kojeg ima profilnu sliku (avatar) kao i mogućnost da unese ostale opcione informacije o njemu.

Administrator sistema će imati u okviru svoj profila, admin dužnosti. Te dužnosti podrazumevaju dodavanje novih igrica i žanrova.

Svaki unos, čitanje, ažuriranje ili brisanje, će biti zabeležen u bazi podataka.

2.2 Perspektiva proizvoda

Ovo je nezavisna aplikacija. Interfejs sistema, korisnički interfejs i hardverski interfejs su definisani sledećim redosledom:

2.2.1 Interfejs sistema:

Korisnici sistema, kako administrator ili korisnici, će moći na brz, lak i uočljiv način moći unositi parametre u potrebna polja, kako prilikom registracije ili logovanja, tako i kod dodavanje igrice nove u bazu podataka. Prilikom popunjavanja skoro svake forme, potrebno je popuniti sva polja. Iako korisnik možda pogreši u tome, biće obavešten da ista popuni.

2.2.2 Korisnički interfejs:

Korisnici sistema će imati jasan i adaptivan prikaz svih informacija iz baze, u čitljivoj formi, kako bi se tim informacijama kasnije koristili.

2.2.3 Hardverski interfejs:

Veb sistem, odnosno veb sajt je kao i svaki drugi, korisnici će moći bez problema i uz minimalne hardverske komponente će moći pokrenuti sistem.

2.2.4 Softverski interfejs:

Korišćeni jezici za izradu veb sistema: Angular 2.4 i PHP.

2.2.5 Komunikacioni interfejs:

Korisnici pristupaju pomoću LAN ili Wireless mreže mrežnog interfejsa.

2.2.6 Memorijski zahtevi:

Veb sistem je odrađen tako da nisu potrebni veliki memorijski resursi.

2.3 Zahtevi korisnika

Dizajn interfejsa sistema je odrađen jednostavno i intuitivno za navigiranje za sve korisnike, kako bih uspešno obavljali svoje zadatke. Za rad na aplikaciji nije potrebno neko dodatno znanje sa računarima. Takođe je pravilo tri klika upotrebljeno, tako da se korisnici uspešno mogu navigirati u bilo koji deo sistema gde im je želja da odu, i izvrše željeno.

2.4 Uslovi i ograničenja

Prvobitan uslov, ujedno i ograničenje ovog sistema jeste da korisnik mora imati pristup internet kako bi pristupio softverskom sistemu.

Bezbednost, samog sistem kao i baze podataka, kao i najbitniji aspekt svakog sistema, pogotovu većih, moraju biti uvedeni i pravilno implementirati. Lični podaci korisnika koje se unose u bazu moraju biti zaštićeni, tako da ukoliko nisu, bi mogli maliciozni napadači ugroziti korisnike, pa i sami sistem. O bezbednosti će biti više reči u nastavku dokumenta u okviru nefunkcionalnih zahteva.

Potrebno je popuniti sva polja. Ukoliko korisnik popunjava neke podatke, i izostavi neka polja ili unese podatke u pogrešnom formatu, sistem treba da obavesti korisnika o tome. Takođe sistem pored ovih upozorenja o greškama, treba svuda greške ispraviti gde se pojavljuju i obavestiti korisnika.

2.5 Pretpostavke i ograničenja

Korisnik već zna, ili treba da zna da koristi veb aplikacije, kao i ovu konkretnu, i da naravno ima pristup internetu. Ukoliko dođe do ažuriranja veb sajta, kako zbog ubacivanja novog sadržaja/igrica ili ažuriranjem novih cena i ponuda ili slično, korisnika treba obavestiti ukoliko uđe na sajt, i ako nije u funkciji, da ga obavesti o tome šta se dešava.

3. Specifični softverski zahtevi

U okviru specifičnih softverskih zahteva će biti navedeni i opisani svi funkcionalni i nefunkcionalni zahtevi sistema. Zahtevi za razvoj softverskih sistema predstavljaju način na koji softverski sistem bi trebalo da se razvije i šta taj sistem radi.

	Title ID	Full Description	Code	Priority	Workload	Risk	Status
1	1.	Functional Requirements	REQ_ 2	Undefined	0	Undefined	Draft
2	1.1	User Database	REQ_ 17	Undefined	0	Undefined	Draft
3	1.1.1	User Viewer	REQ_ 15	Undefined		Undefined	Draft
4	1.1.2	New User	REQ_ 20	Undefined		Undefined	Draft
5	1.1.3	Edit User	REQ_ 21	Undefined		Undefined	Draft
6	1.1.4	Remove User	REQ_ 22	Undefined		Undefined	Draft
7	1.2	Game Database	REQ_ 9	Undefined	0	Undefined	Draft
8	1.2.1	Game Viewer	REQ_ 10	Undefined		Undefined	Draft
9	1.2.2	Game Search	REQ_ 11	Undefined		Undefined	Draft
10	1.2.3	Game Genre	REQ_ 27	Undefined		Undefined	Draft
11	1.3	User Registration	REQ_ 16	Undefined	0	Undefined	Draft
12	1.3.1	User Profile	REQ_ 12	Undefined		Undefined	Draft
13	1.3.2	User Cart	REQ_ 14	Undefined		Undefined	Draft
14	1.3.3	User Games	REQ_ 18	Undefined		Undefined	Draft
15	1.4	Remote Access	REQ_ 13	Undefined	0	Undefined	Draft
16	1.4.1	Internet Access	REQ_ 25	Undefined		Undefined	Draft
17	2.	Non-Functional Requirements	REQ_ 3	Undefined	0	Undefined	Draft
18	2.1	User Friendly Interface and Adaptive Design	REQ_ 28	Undefined		Undefined	Draft
19	2.2	Performances	REQ_ 29	Undefined		Undefined	Draft
20	2.3	Security	REQ_ 30	Undefined		Undefined	Draft

Slika 1 - Pregled zahteva sistema

3.1 Funkcionalni zahtevi administratori

Funkcionalni zahtevi definišu funkcije sistema. Oni opisuju šta bi sistem trebalo da radi. Ovde ćemo razdvojiti zahteve u dva dela, u dva tipa korisnika koja postoje u sistemu, to su administrator i korisnici.

3.2 Administrator

1	1.	Functional Requirements	REQ_ 2	Undefined	0	Undefined	Draft
2	1.1	User Database	REQ_ 17	Undefined	0	Undefined	Draft
3	1.1.1	User Viewer	REQ_ 15	Undefined		Undefined	Draft
4	1.1.2	New User	REQ_ 20	Undefined		Undefined	Draft
5	1.1.3	Edit User	REQ_ 21	Undefined		Undefined	Draft
6	1.1.4	Remove User	REQ_ 22	Undefined		Undefined	Draft
7	1.2	Game Database	REQ_ 9	Undefined	0	Undefined	Draft
8	1.2.1	Game Add	REQ_ 23	Undefined		Undefined	Draft
9	1.2.2	Game Update	REQ_ 24	Undefined		Undefined	Draft
10	1.2.3	Game Remove	REQ_ 26	Undefined		Undefined	Draft

Slika 2 - Funkcionalni zahtevi - Administrator

Administrator sistema će imati potpunu CRUD kontrolu u bazi podataka nad korisnicima i igricama.

3.2.1 User Database

Sistem će sadržati podatke o svim korisnicima sistema u dobro obezbeđenoj bazi podataka. Administrator može koristiti ove podatke kako bih manipulirao njima, u vidu pregleda korisnika, dodavanju, brisanju ili ažuriranju korisnika.

3.2.1.1 User Viewer

Administrator može videti sve korisnike, nad kojima može manipulirati ukoliko je potrebno.

3.2.1.2 New User

Administrator može dodati novog korisnika, iako postoji već registracija na sistemu, iz razloga ukoliko dodaje nekog novog privilegovanog korisnika, odnosno zaposlenog, kao što je moderator veb sajta, ukoliko softverski sistem ubaci novu funkcionalost, kao što je forum, ili komentare, ili još nekog administratora.

3.2.1.3 Edit User

Sistem pruža mogućnost izmene profila korisnika, ukoliko postoje neprikladni opisi korisnika, ili imena i slično.

3.2.1.4 Remove User

Sistem može ukloniti, odnosno onemogućiti korisnike iz baze podataka, odnosno banovati ih. Time će im se korisnički nalog deaktivirati, odnosno korisnik neće moći ući na svoj profil.

3.2.2 Game Database

Administrator može videti sve igrice, i nad njima može manipulirati kao i kod korisnika.

3.2.2.1 Game Add

Administrator može dodavati nove igrice, kao što su nove koje izlaze na tržište i biti u toku sa njima.

3.2.2.2 Game Update

Sistem će ažurirati igre, ukoliko kompanija koja pravi istu izbaci neki update. Sistem će takođe sa vremena na vreme, ažurirati cene igra, a odnosno staviti popust ili slično na određeno vreme.

3.2.2.3 Game Remove

Administrator može ukoliko bude bilo potrebno nekada i da izbriše igricu iz baze podataka.

3.3 Korisnik

7	1.2	Game Database	REQ_ 9	Undefined	0	Undefined	Draft
8	1.2.1	Game Viewer	REQ_ 10	Undefined		Undefined	Draft
9	1.2.2	Game Search	REQ_ 11	Undefined		Undefined	Draft
10	1.2.3	Game Genre	REQ_ 27	Undefined		Undefined	Draft
11	1.3	User Registration	REQ_ 16	Undefined	0	Undefined	Draft
12	1.3.1	User Profile	REQ_ 12	Undefined		Undefined	Draft
13	1.3.2	User Cart	REQ_ 14	Undefined		Undefined	Draft
14	1.3.3	User Games	REQ_ 18	Undefined		Undefined	Draft
15	1.4	Remote Access	REQ_ 13	Undefined	0	Undefined	Draft
16	1.4.1	Internet Access	REQ_ 25	Undefined		Undefined	Draft

Slika 3 - Funkcionalni zahtevi - Korisnik

Korisnik sistema će moći pretraživati i gledati igrice po imenu, žanrovima, videti listu igara po žanrovima, registrovati se na sistem, posedovati uz to profil, korpu, kao i listu svojih igrica.

3.3.1 Game Database

Baza podataka veb sistema će posedovati sve podatke za igrice igrice, koje će korisniku biti na usluzi. U okviru baze podataka postoji više funkcionalnosti:

3.3.1.1 Game Viewer

Sistem će imati stranu gde će korisnik moći videti sve igrice, i videti sve njene attribute i detalje koje su potrebni. Sistem pruža korisniku i listu žanrova, koje može odabrati i po tim kriterijumima izlistati igrice.

3.3.1.2 Game Search

Korisnici će imati search bar, tekstualno polje za pretraživanje igrice po imenu, ukoliko znaju koju igricu žele da kupe, kako bi je lakše našli, ili ukoliko žele uneti ključnu reč odnosno naziv neke franšize, i žele izlistati sve delove iste.

3.3.1.3 Game Genre

Sistem će pružiti u okviru menija, sve žanrove koje postoje iz baze podataka, po kojim mogu isključivo pretraživati igrice.

3.3.2 User Registration

Sistem omogućava registrovanje svakom korisniku, besplatno i jednostavno. Prilikom registrovanja na sistem, od korisnika se traži da unese svoju elektronsku poštu, korisničko ime, kao i šifru. Nakon registracije, korisnik se automatski registruje na sistem. Korisnik se može izlogovati ili ulogovati na sistem kasnije preko svog korisničkog imena i šifre kad god želi. Registrovani korisnici imaju sledeće privilegije:

3.3.2.1 User Profile

Registrovani korisnici će imati svoj profil, u okviru kojeg će biti prikazani već uneti podaci kao što su ime i email adresa. Pored obaveznih podataka, postojaće i dodatni opisi za profil, kao što su: ime i prezime, nacionalnost, rođendan, profilna slika. Takođe će moći i ažurirati odnosno izmeniti svoje opcione podatke.

3.3.2.2 User Cart

Registrovani korisnici će moći kupovati igre. Igre kupuju tako što kliknu na igru, koja se pritom stavlja u korpu korisnika. U okviru korpe, biće prikazane sve igre, cena igre, kako zasebno tako i suma svih u korpi. Korisnik kupuje igre onda kada klikne na dugme checkout. Korisnik u svakom trenutku može dodati bilo koju igricu u korpu, kao i izbaciti je iz iste, putem remove dugmeta.

3.3.2.3 User Games

Registrovani korisnici, će moći na svom profilu da dodaju igre u listu omiljenih igra, kao i wish listu, odnosno listu želja, gde stavljaju igre koje bih nekada u budućnosti kupili. Korisnici koji su kupili igre, moći će ih pretraživati i videti na svom nalogu.

3.3.3 Remote Access

Sistem omogućava korisnicima da pristupe sistemu preko internet pretraživača, odnosno putem interneta.

3.3.3.1 Internet Access

Putem interneta će korisnici moći pretraživati igre, kupovati ih, kao i sve ostale već navedene funkcionalnosti.

3.2 Nefunkcionalni zahtevi

Nefunkcionalni zahtevi su zahtevi koji nisu direktno povezani sa servisima koje sistem treba da obezbedi svojim korisnicima, već definišu ograničenja implementacije sistema.

18	2.	Non-Functional Requirements	REQ_ 3	Undefined	0	Undefined	Draft
19	2.1	User Friendly Interface and Adaptive Design	REQ_ 28	Undefined		Undefined	Draft
20	2.2	Performances	REQ_ 29	Undefined		Undefined	Draft
21	2.3	Security	REQ_ 30	Undefined		Undefined	Draft

Slika 4 - Nefunkcionalni zahtevi

U okviru nefunkcionalnih zahteva ovog softverskog sistema, nabrojaćemo tri:

3.2.1 User Friendly Interface and Adaptive Design

Interfejs sistema mora biti user friendly i biti adaptivna. To znači jednostavno i brzo, da se korisnik može što bolje snaći u sistemu, bez mnogo dvoumljena prilikom izvršavanja njegovih akcija, kako registrovanje, tako i pretraga igara i slično. Pod adaptivnim dizajnom podrazumevamo da prilikom dizajniranja sistema, da ga moramo prilagoditi da može biti upotrebljiv na uređajima različitih veličina i performanski, kao što su mobilni uređaj, tablet, računar.

Adaptivnim dizajnom dajemo fleksibilnost našem sistemu, kao i korisnicima. Sistem mora biti adaptivan za korisnike, dakle, tako da mogu pristupiti našem sistemu sa "bilo kog" uređaja.

Ovakvu adaptaciju nam pruža, i koristiti će se Bootstrap framework, kojim ćemo dakle prilagođavati prikaz sistema na različitim uređajima različitih rezolucija.

3.2.2 Performances

Performanse sistema moraju naravno biti brze i kratke, tako da se brzina prilagodi i izvrši brzo u zavisnosti od broja korisnika koji su u istom trenutku na sistemu.

Treba prilagoditi performanse tako da korisnik ne mora da čeka na duže od 1-2 sekunde da se učita sistem. Istraživanja su pokazala da 1 u 4 posetioca sajta napuste odmah sajt ukoliko je vreme potrebno za učitavanje duže od 4 sekunde.

Performanse naravno zavise od veličine fajlova, s tim dakle i veličina slika i ostalih sadržaja, tako da i njih treba minimizovati i što bolje implementirati i kompresovati.

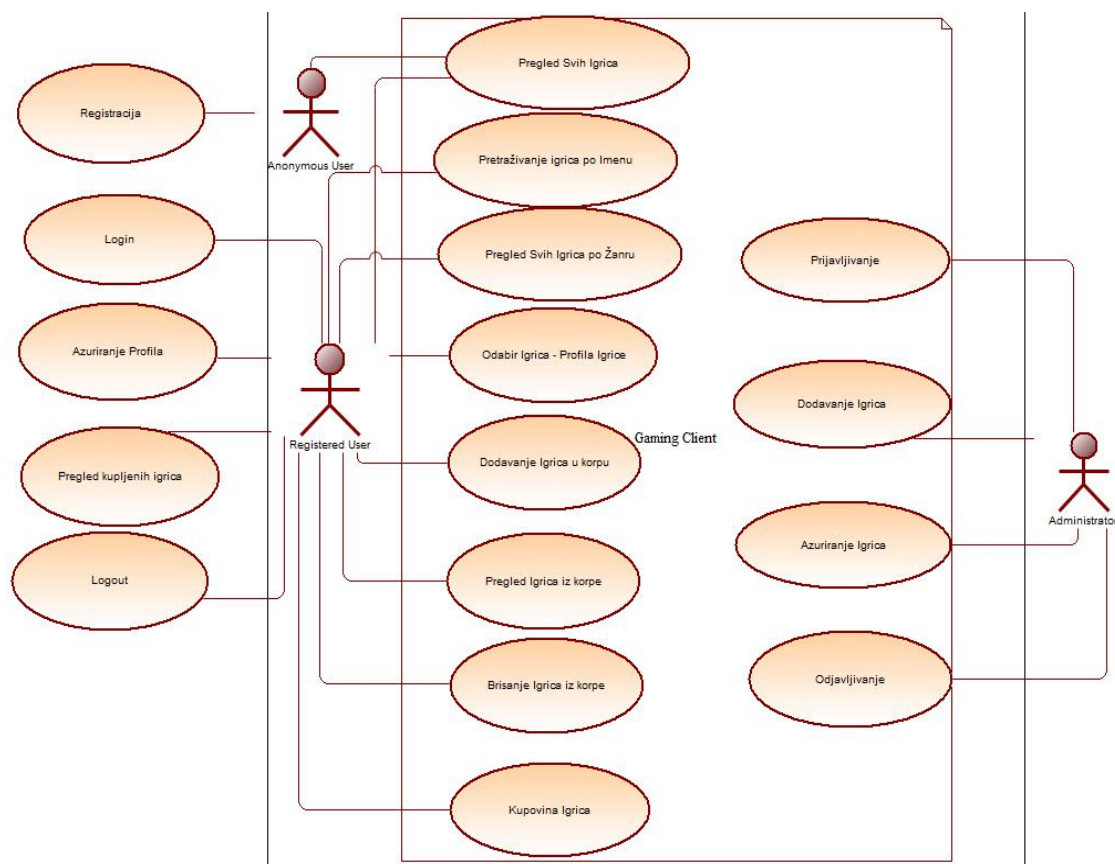
Ukoliko dođe do dužeg učitavanja, treba staviti Loading Screen, koji će sadržati neku zanimljivu animaciju ili poruku koje će držati korisniku pažnju i ostaviti ga da sačeka da se učita sistem.

3.2.3 Security

Bezbednost, kao i najbitniji aspekt svakog sistema, pogotovu većih, moraju biti uvedeni i pravilno implementirani. Lični podaci korisnika koje se unose u bazu podataka moraju biti zaštićeni, tako da ukoliko nisu, bi mogli maliciozni napadači ugroziti korisnike, pa i sami sistem.

Ovo sprečavamo time što implementiramo u naš php PDO, PHP Data Objects, ekstenzija koja nam omogućava pristup bazi podataka, prilikom pisanja koda, prepared statement su primaran i glavni način za sprečavanje i rešavanje SQL inejction napada, kojim maliciozni korisnici se mogu opslužiti kako bih naudili sistemu.

4. Slučajevi korišćenja sistema



Slika 5 - Use Case - Kompletan dijagram slučajeva korišćenja sistema

Slučaj korišćenja sistema utvrđuje sve aktere, odnosno korisnike sistema, koji učestvuju u interakciji sa softverskim sistemom, koje sve funkcije oni imaju i obavljaju koristeći isti. Postojeće dva aktera sistema i slučajevi korišćenja za njih:

4.1 Korisnik

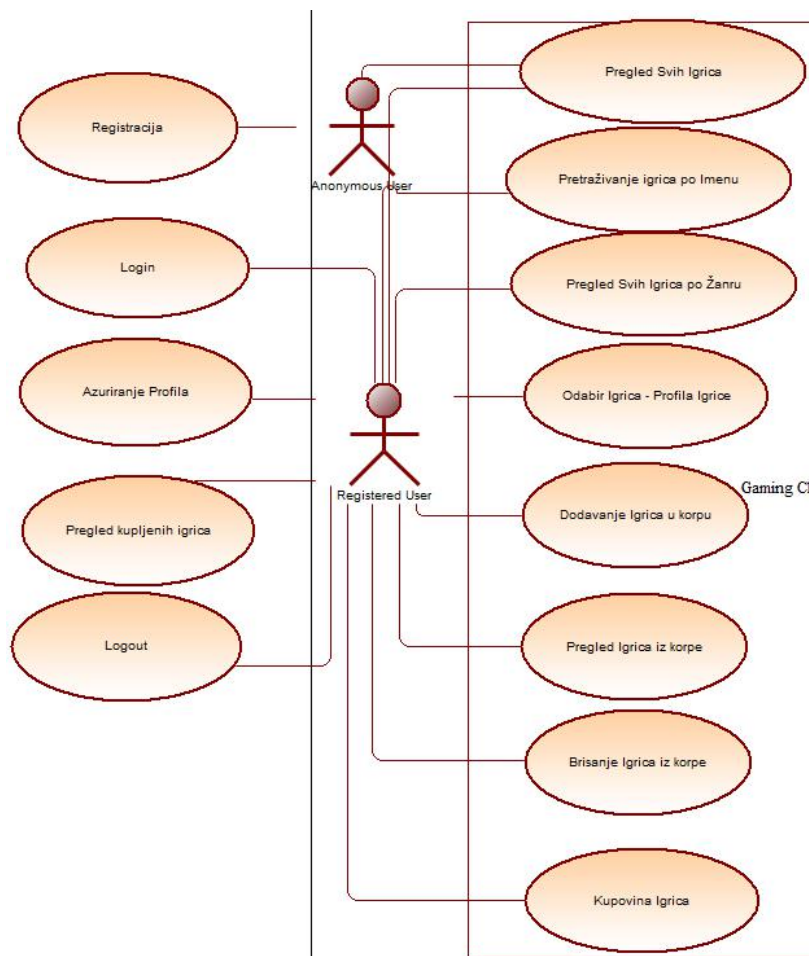
Kod slučajeva korišćenja korisnika, podelićemo korisnika u dva pod-aktera, anonimnog i registrovanog.

Anonimni korisnik će moći da se registruje, pregleda igrice sistema, pretraživanje igrica po imenu, po žanru, dok će registrovani imati iste ove funkcije kao i sve ostale.

Slučajevi korišćenja korisnika:

- Registracija - anonimni korisnik se može registrovati na sistem, tako što popunjava formu za registraciju, kako bih imao pristup beneficijama registrovanog korisnika. Od korisnika se traži username(korisničko ime), password(šifra), kao i njegova email adresa, sva ova polja su neophodna i moraju biti popunjena prilikom registrovanja.
- Pregled svih igrica - korisnik će moći pregledati sve igrice koje sistem sadrži, klikom na 'Store', u okviru menija veb sistema, gde se nalazi lista svih igrica iz baze podataka.
- Pretraživanje igrica po imenu - korisnik će moći pretraživati igrice u search baru po imenu, kako bih suzio listu igara. Ovu pretragu može obaviti na bilo kojoj stranici gde se izlistavaju igrice, kao što su Store, po žanru pretraga igrica, korisnikove igrice, igrice u korpi, i slično.

- Pregled svih igrica po žanru - korisnik će moći iz menija, u kojem su izvučeni svi žanrovi iz baze podataka, da odabere žanr, i pritom dobiti listu svih igrica iz odabranog žanra.
- Profil igrice - korisnik će moći iz bilo kojeg pregleda igrice, da klikne na igricu, gde mu se daje profil igrice, gde se nalaze sve informacije o igrici.
- Login - korisnik koji ima nalog, se može ulogovati na sistem popunjavanjem polja za korisničko ime i šifru, ova dva polja moraju biti popunjena i tačna, proverava se njihovo postojanje u bazi podataka.
- Ažuriranje profila - registrovani korisnik će moći ući na svoj profil kada je ulogovan, gde će se nalaziti njegova profilna slika, kao i opcione informacije o njemu koje može popuniti, kao što su profilna slika, ime, prezime, datum rođenja, nacionalnost.
- Dodavanje igrica u korpu - korisnik dodaje igrice u korpu tako što iz bilo kojeg pomenutog pregleda igara, odabere igru i klikne na dugme da doda istu u korpu.
- Pregled igrica iz korpe - korisnik će u svakom trenutku moći videti svoju korpu u okviru menija, gde može videti sve igrice koje se nalaze u korpi.
- Brisanje igrica iz korpe - u okviru pregleda korpe, korisnik može ukloniti igru iz korpe, ukoliko se predomisli o kupovanju igrice, ili ukoliko je slučajno postavio u istu.
- Kupovina igrica - korisnik može kupiti igrice, u okviru pregleda igrica iz korpe, gde se nalazi dugme za kupovinu.
- Pregled kupljenih igrica - korisnik može videti kasnije listu svih svojih igrica koje on poseduje i koje je kupio.
- Logout - korisnik se može izlogovati iz sistema u svakom trenutku.



Slika 6 - Use Case - Dijagram slučajeva korišćenja korisnika sistema

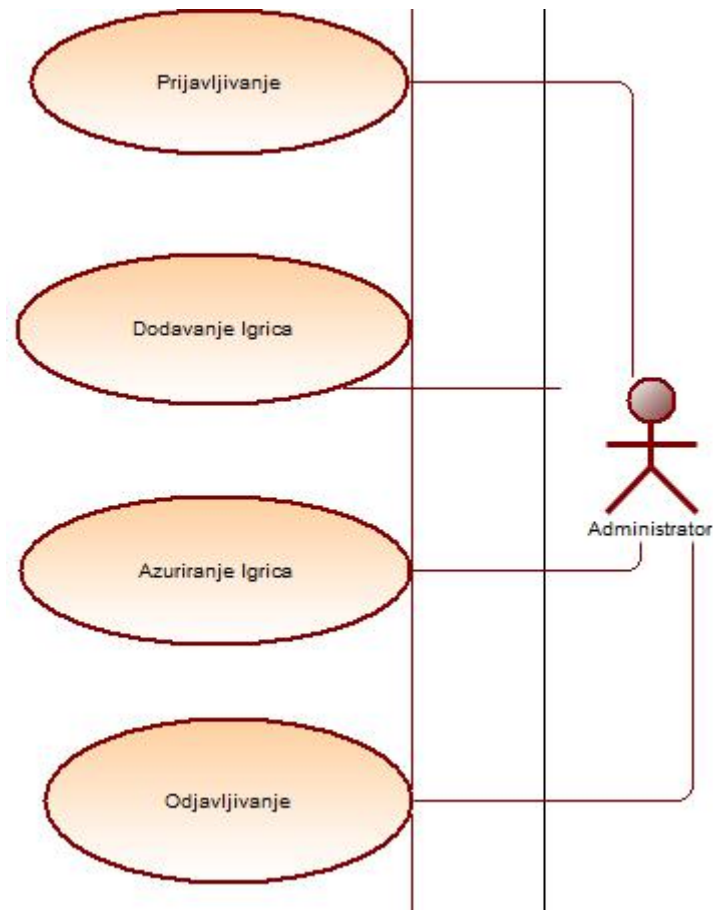
Use Case	Preduslovi	Koraci	Izuzeci	Postuslovi
Registracija	Korisnik mora imati pristup internetu, kao i svoju email adresu.	Korisnik popunjava formu za registrovanje.	Korisnik se ne može registrovati ukoliko već postoji nalog na isto korisničko ime i adresu.	Korisnik je uspešno napravio nalog, sada može koristiti sve funkcionalnosti sistema.
Pregled svih igrica	Korisnik mora imati pristup internetu i navigirati se do cilja.	Korisnik ulazi u Store u okviru menija, bira All Games, gde mu se pritom prikazuje lista igrica.		Korisnik može videti sve igrice u lepo formatiranoj tabeli, sa nekoliko informacija o igrici.
Pretraživanje igrica po imenu	Korisnik se mora navigirati do bilo kojeg pregleda liste igrica - Sve igrice, po žanru...	Korisnik se navigira do pregleda igrica, pritom klikne na tekstualno polje za pretragu, i ukuca bilo koju reč, karaktere koje će tražena igrica sadržati.		Korisnik je uspešno suzio listu igara koja mu se prikazuje, prikazujući mu željene igrice po definisanom kriterijumu za ime.
Pegled svih igrica po žanru		Korisnik može videti sve igrice po kriterijumu samo jednog odabranog žanra tako što u meniju u okviru 'Store' odaberu jedan od žanrova		Korisnik može videti sve igrice odabranog žanra u lepo formatiranoj tabeli, sa nekoliko informacija o igrici.
Profil igrice	Korisnik se mora navigirati do bilo kojeg pregleda liste igrica - Sve igrice, po žanru...	Korisnik se navigira do pregleda igrica, pritom klikne na sliku igrice ili uopšteno na jednu ćeliju reda tabele igrica.		Korisnik može videti igricu koju je odabrao, gde se nalaze sve informacije o datoj igrici.
Login	Korisnik mora imati nalog na sistemu. Potrebna je registracija.	Korisnik u okviru menija sistema bira login modul, za prikazivanje forme	Korisnik nije prijavljen na sistem, ili se njegovi unosi ne poklapaju sa	Korisnik se uspešno ulogovao na sistem.

		za login koje mora popuniti.	bazom, tj nepostojaći nalog.	
Ažuriranje profila	Korisnik mora biti prijavljen na sistem.	Korisnik bira iz menija svoj profil, identifikovan svojim korisničkim nalogom, bira na opciju profil, odatle može popunjavati opcione informacije o sebi.	Korisnik nije prijavljen na sistem.	Korisnik je uspešno uneo, ili ažurirao informacije o sebi.
Dodavanje igrica u korpu	Korisnik mora biti prijavljen na sistem. Korisnik se mora navigirati do bilo kojeg pregleda liste igrica - Sve igrice, po žanru...	Korisnik nakon navigiranja do pregleda liste igara, može videti pored svake igrice dugme za dodavanje u korpu, klikom na to dugme, korisnik dodaje igricu u korpu.	Korisnik nije prijavljen na sistem. Korisnik je već kupio igricu ili se igrice već nalazi u korpi.	Korisnik se je uspešno ubacio igricu u korpu.
Pregled igrica iz korpe	Korisnik mora biti prijavljen na sistem.	Korisnik pristupa korpi u bilo kom trenutku u okviru menija, može hoverovati ili kliknuti na nju, kako bi video listu igara koje se nalaze u njoj.	Korisnik nije prijavljen na sistem. Korisnik nema igrice u korpi.	Korisnik je uspešno otvorio korpu u kojoj može videti sve igrice, koje kasnije može kupiti, ili ih izbaciti.
Brisanje igrica iz korpe	Korisnik mora biti prijavljen na sistem. Korisnik mora imati igricu u korpi kako bi je izbacio	Korisnik se navigira do korpe, pored svake igrice, postojaće dugme ' <u>Remove</u> ', koje može kliknuti i ukloniti igricu	Korisnik nije prijavljen na sistem. Korisnik nema igrice u korpi.	Korisnik je uspešno uklonio igricu iz korpe.
Kupovina igrica	Korisnik mora biti prijavljen na sistem. Korisnik mora imati dodate igrice u korpi kako	Korisnik se navigira do korpe, odatle, videće dugme ' <u>Go to checkout</u> ' preko kojeg može kupiti	Korisnik nije prijavljen na sistem. Korisnik mora imati dodate igrice u korpi kako bi ih	Korisnik je uspešno kupio sve igrice koje su se nalazile u korpi.

	bi ih kupio.	sve igrice iz korpe.	kupio.	
Pregled kupljenih igrica	Korisnik mora biti prijavljen na sistem. Korisnik mora imati kupljene igrice kako bi ih video.	Korisnik bira iz menija svoj profil, identifikovan svojim korisničkim nalogom, bira na opciju ' <u>YOUR GAMES</u> ', gde će se potom preusmeriti na stranu za prikaz svih korisnikovih igrica.	Korisnik nije prijavljen na sistem. Korisnik mora imati kupljene igrice kako bi ih video.	Korisnik se uspešno navigirao do strane gde može videti sve igrice koje on poseduje.
Logout	Korisnik mora biti prijavljen na sistem.	Korisnik bira iz menija svoj profil, identifikovan svojim korisničkim nalogom, bira dugme ' <u>LOGOUT</u> '.		Korisnik se uspešno odjavio sa sistema.

4.2 Administrator

- Prijavljivanje - administrator sistema se prijavljuje na sistem kao i svaki drugi korisnik, putem login forme, ali uz specifično korisničko ime i šifru koje ga identifikuju kao admina.
- Dodavanje igrica - administrator sistema će moći dodavati nove igrice. Administrator ovo radi preko svog **Admin Work** menija u okviru opadajućeg menija. Odatle, administrator će imati 2 dugmeta za dodavanje, **Add Game** i **Add Rest**. Na prvom slučaju, on dodaje igrice i unosi sve osnovne informacije o igrici, dok u drugoj, kojoj pristupa kada završi prvo dodavanje, on dodaje žanr, kompaniju i jezik igrici kao završne parametre. Razlog za ovim razdvajanjem prilikom dodavanja jeste pre svega, zbog više-na-više veze u bazi podataka između ova 3 atributa i entiteta igrica. Drugi razlog je ukoliko želimo da dodamo još jezika igrici, ili žanrova, možemo je tim putem ažurirati.
- Ažuriranje igrica - administrator će moći ažurirati igrice preko opcije **Add Rest**, pomenute u prošlom slučaju, ažurirajući parametre igrice kao što su žanr, kompanija i jezik.
- Odjavljivanje - administrator sistema se preko opadajućeg menija svoj profila može izlogovati klikom na dugme 'logout'.



Slika 7 - Use Case - Dijagram slučajeva korišćenja administratora sistema

Use Case	Preduslovi	Koraci	Izuzeci	Postuslovi
Prijavljivanje	Administrator mora imati svoj privilegovani nalog.	Administrator popunjava formu za loginovanje.	Administrator nema nalog.	Administrator se uspešno prijavio na sistem.
Dodavanje igrice	Administrator mora imati svoj privilegovani nalog.	Navigacijom do 'Admin Work' menija, admin klikće na 'Add Game' dugme I nakon toga na 'Add Rest' dugme, gde pritom popunjava formu za dodavanje igrice.	Administrator nema nalog.	Administrator je uspešno dodao igricu.
Ažuriranje igrice	Administrator mora imati svoj	Navigacijom do 'Admin Work'	Administrator nema	Administrator je uspešno ažurirao

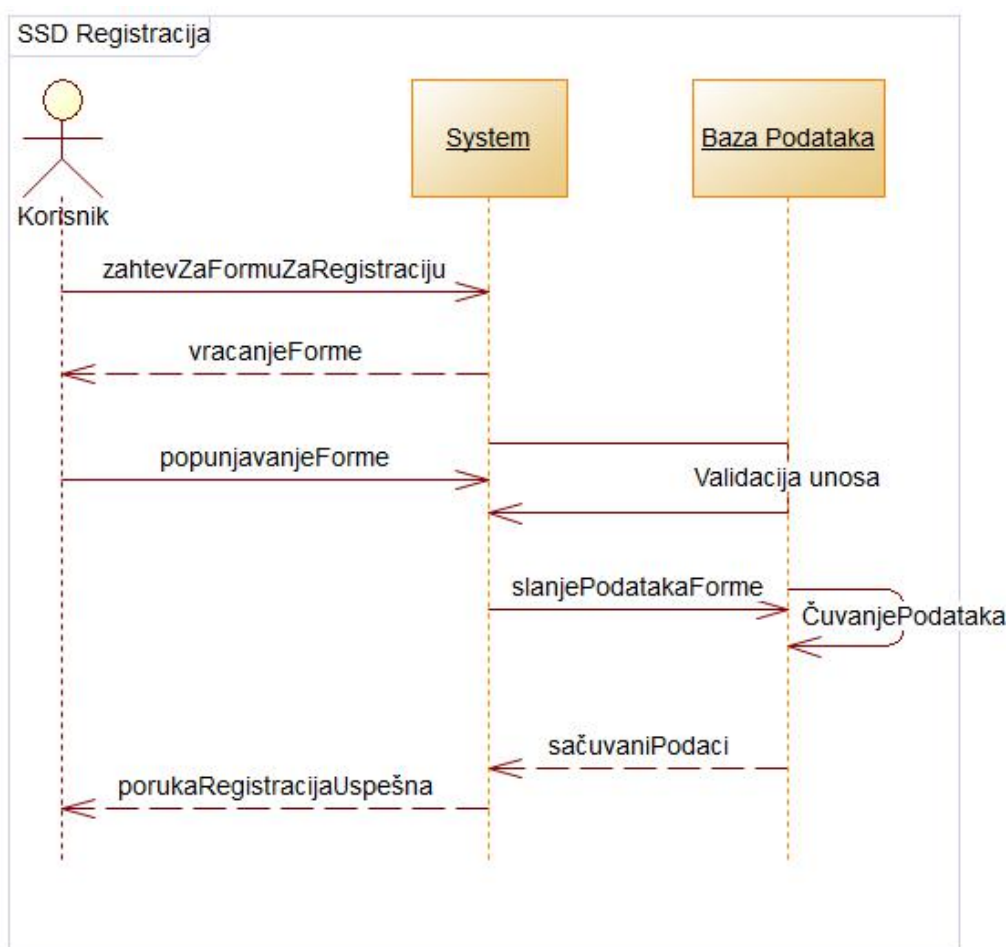
	privilegovani nalog.	menija, admin klikće na ' <u>Add Rest</u> ' dugme, gde pritom popunjava formu za dodavanje igrice.	nalog.	igricu.
Odjavljivanje	Administrator mora biti ulogovan na svom nalogu.	U okviru svog profila u meniju, administrator hoveruje preko njega i ide na dugme ' <u>Logout</u> '.	Administrator nije ulogovan.	Administrator se uspešno odjavio na sistem.

5. Apstraktni dijagram sekvenci(sistem sekvencijalni dijagram)

Sistemske dijagram sekvenci (SSD) prikazuje interakciju spoljnih aktera sa sistemom i kreira se na osnovu opisa dijagrama slučajeva korišćenja koje smo opisali u prethodnom poglavlju dokumentacije. Ovi dijagrami se kreiraju na osnovu opisa dijagrama slučajeva korišćenja. SSD prikazuje određeni događaj i aktere koje učestvuju u tom događaju, zajedno sa među-funkcionalnostima koje se dešavaju u toku.

5.1 Korisnik

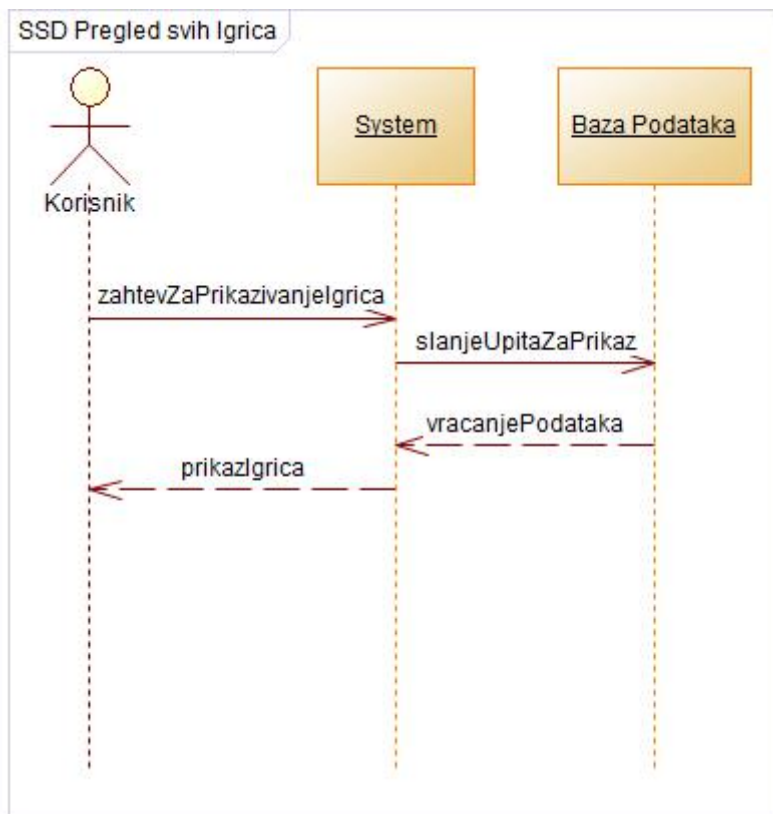
5.1.1 Registracija



Slika 8 - SSD - Registracija korisnika

Korisnik prilikom registrovanja pre svega prvo šalje zahtev za prikaz forme za registraciju, klikom na dugme za otvaranje registracije, gde sistem pritom odgovara na to vraćanjem tražene forme. Pritom korisnik može popuniti 3 ponuđena i neophodna tekstualna polja za registraciju, a to su username, password i email adresa. Pritom se unos proverava preko angular forme za dužinu reči kao i format emaila, proveru nakon toga ide dalje na proveru u bazu podataka, da se proveru postojanje korisnika u istoj. Nakon uspešnog unosa podataka forme, forma se šalje u bazu podataka gde će se podaci sačuvati, i pritom vratiti sistemu podaci, gde će se nakon toga korisnik uspešno ulogovati na sistem.

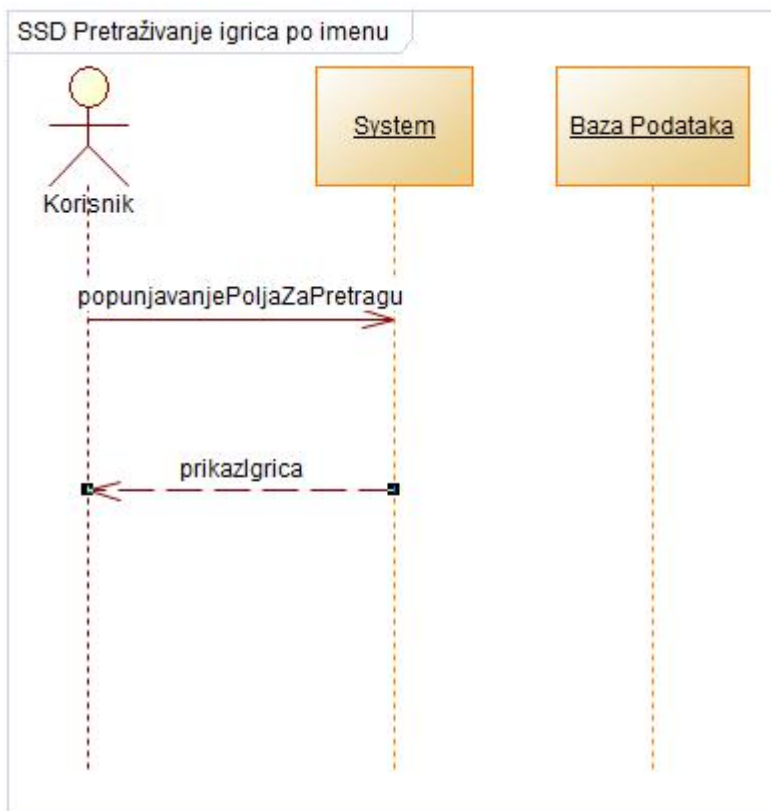
5.1.2 Pregled svih igrica



Slika 9 - SSD - Pregled svih igrica

Korisnik otvara stranu za prikaz igrica, time, traži od sistema prikaz stranice, i sistem šalje SQL upit bazi podataka za prikaz svih knjiga. Baza podataka pronalazi podatke, i šalje ih sistemu, sistem prikazuje podatke korisniku.

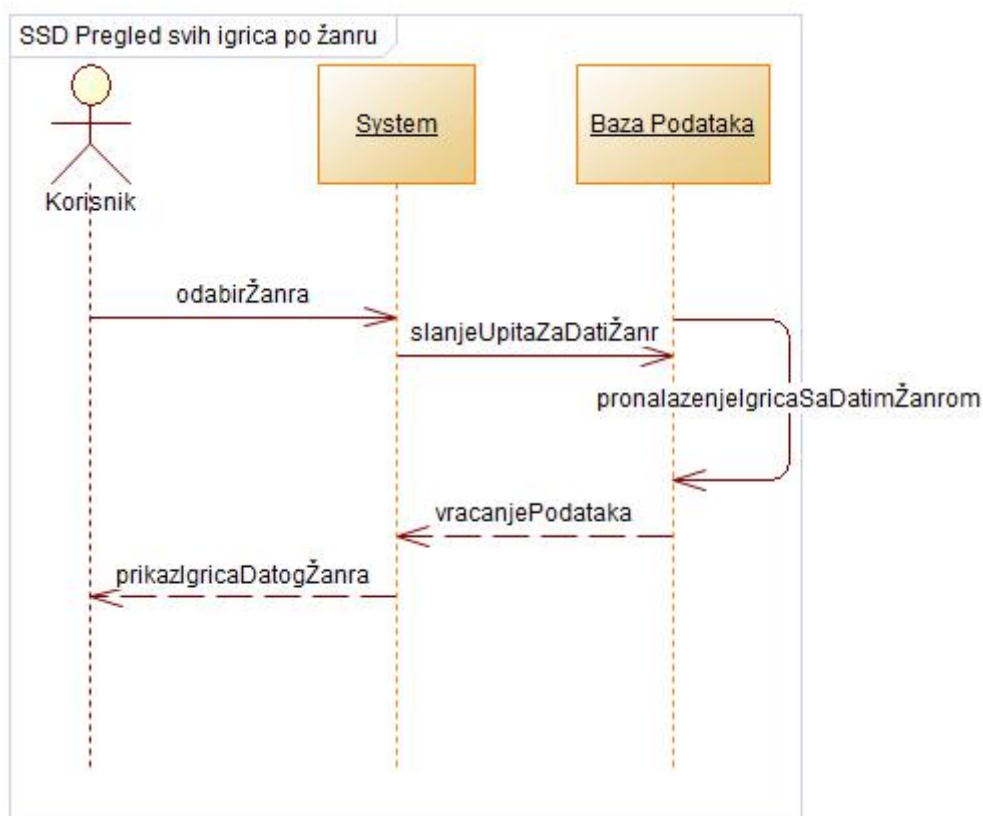
5.1.3 Pretraživanje igrica po imenu



Slika 10 - SSD - Pretraživanje igrica po imenu

U ovom slučaju nam nije potrebna baza podataka, posao obavlja angular, korisnik unosi u tekstualno polje karaktere, sistem ih čita i vraća korisniku prikaz podataka.

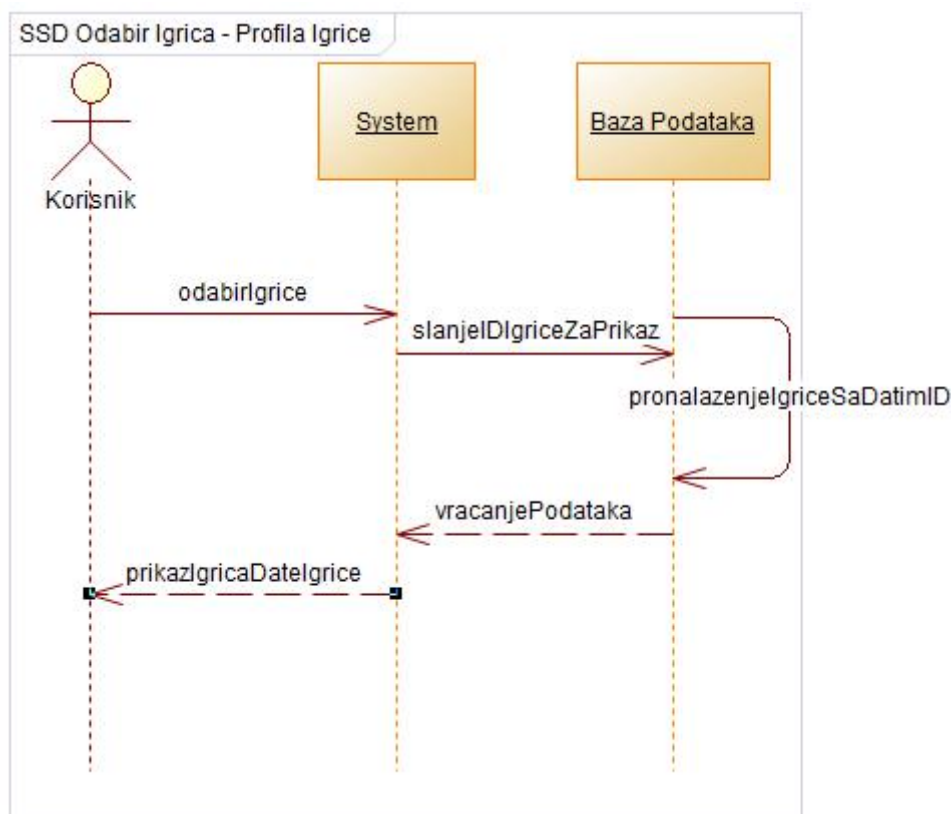
5.1.4 Pregled svih igrica po žanru



Slika 11 - SSD - Pregled svih igrica po žanru

Korisnik bira iz liste žanrova jedan žanr, bazi podataka se šalje SQL upit sa ID-jem odabranog žanra da pokupi sve igrice iz iste kategorije. Baza podataka pronalazi sve igrice, i vraća podatke sistemu, sistem prikazuje korisniku sve podatke.

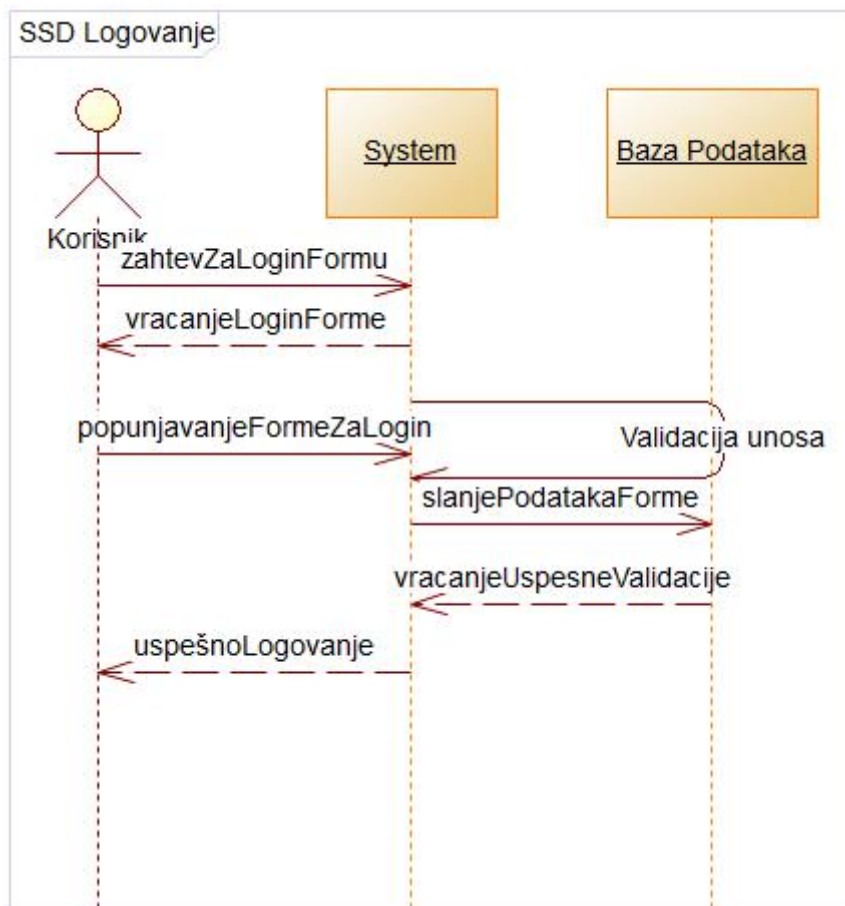
5.1.5 Profil igrice



Slika 12 - SSD - Odabir igrice - Profil igrice

Korisnik bira igricu klikom na nju ili sliku iste, pritom se šalje SQL upit za dobijanje svih informacija o igrici sa ID-jem odabrane igrice. Baza podataka pronalazi igricu sa datim ID-jem, uzima sve informacije, attribute njene, vraća podatke sistemu, sistem prikazuje korisniku podatke igrice.

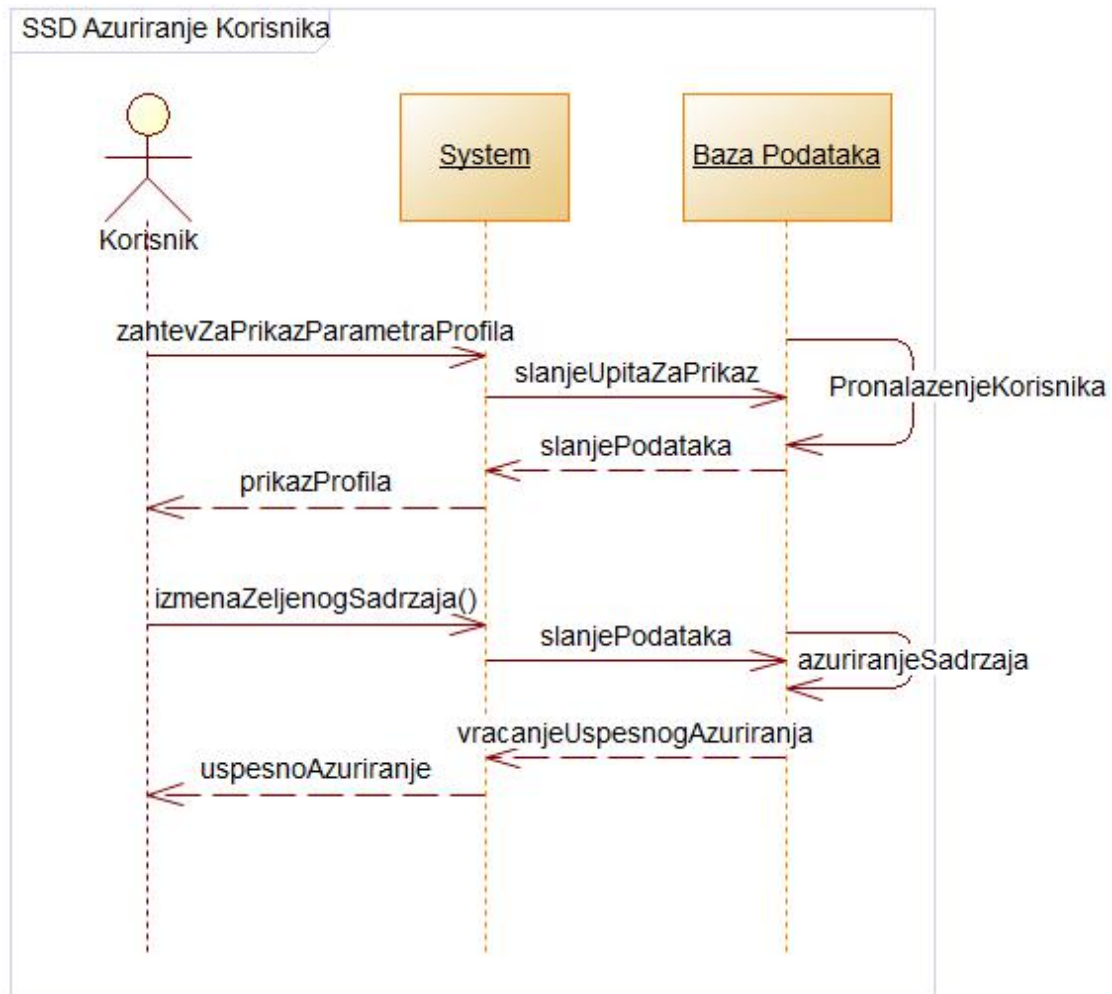
5.1.6 Login



Slika 13 - SSD - Logovanje korisnika

Korisnik otvara, odnosno zahteva pre svega formu za login, sistem vraća modal formu za istu. Pritom korisnik popunjava podatke, u tom procesu, kada se pošalju podaci bazi podataka, proverava se u njoj da li postoji korisnik sa istim korisničkim imenom i sa datom šifrom. Nakon uspešne validacije, baza podataka šalje sistemu da je korisnik uspešno ulogovan.

5.1.7 Ažuriranje profila

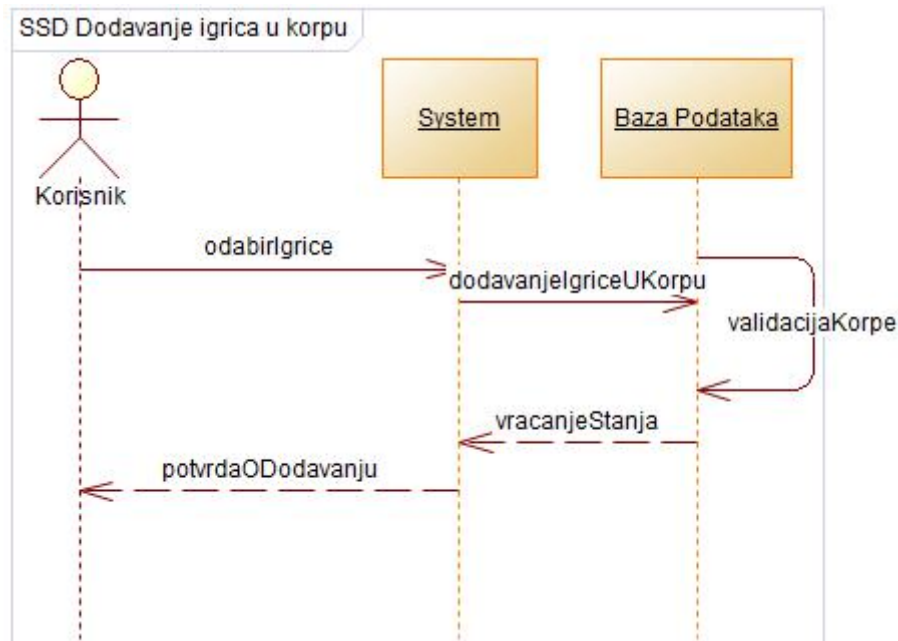


Slika 14 - SSD - Ažuriranje profila korisnika

Korisnik šalje zahtev za prikaz forme sa parametrima profila koja želi editovati ili prikazati. Sistem šalje SQL upit bazi podataka za prikaz parametra korisnika sa njegovim ID-jem. Baza podataka pronalazi podatke i šalje ih nazad sistemu, sistem prikazuje profil korisniku.

Korisnik pritom unosi sadržaj, sistem šalje podatke bazi podataka, ona ih ažurira, i vraća poruku uspešnog ažuriranja.

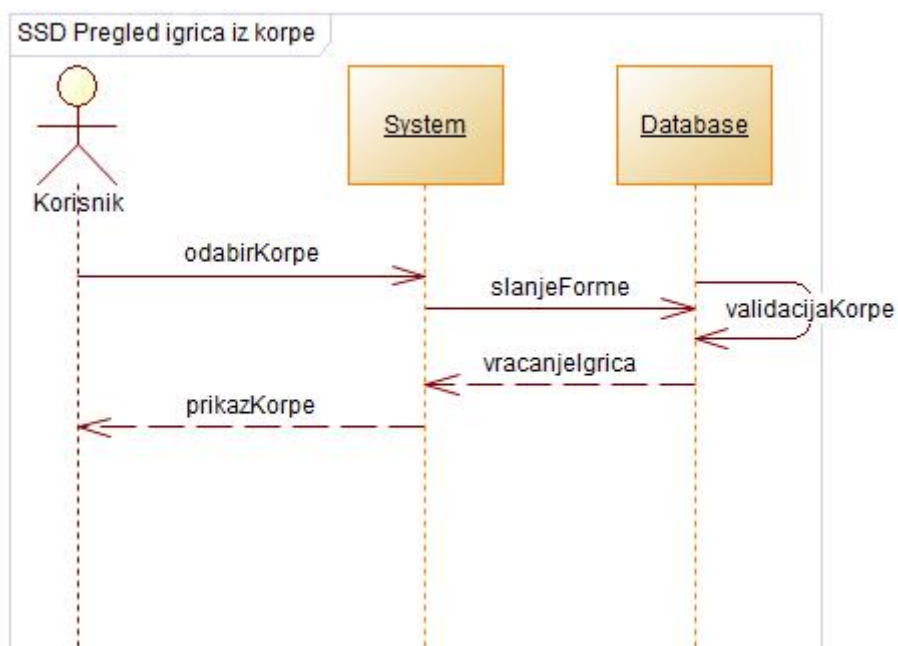
5.1.8 Dodavanje igrica u korpu



Slika 15 - SSD - Dodavanje igrica u korpu

Korisnik bira igricu koju želi dodati u korpu, sistem šalje SQL upit sa datim ID-jem igrice bazi. Baza podataka validira korpu i proverava da igrice nije već kupljena ili već postoji u korpi. Nakon uspešnog dodavanja, baza podataka šalje uspešan prolaz upita, sistem vraća korisniku poruku pritom o uspešnom dodavanju.

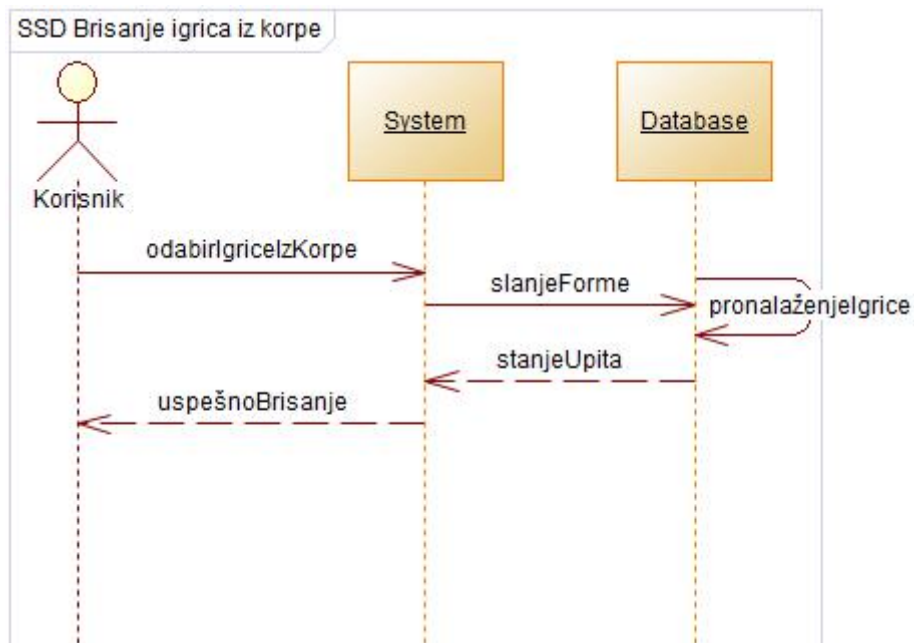
5.1.9 Pregled igrica iz korpe



Slika 16 - SSD - Pregled igrica iz korpe

Korisnik bira svoju korpu, identifikovana i vezana za datog korisnika, sistem šalje SQL upit bazi za povratak podataka za igrice date korpe i korisnika. Baza podataka pritom vraća stanje o izvršenom upitu sa podacima, sistem prikazuje pritom igrice iz korpe korisniku.

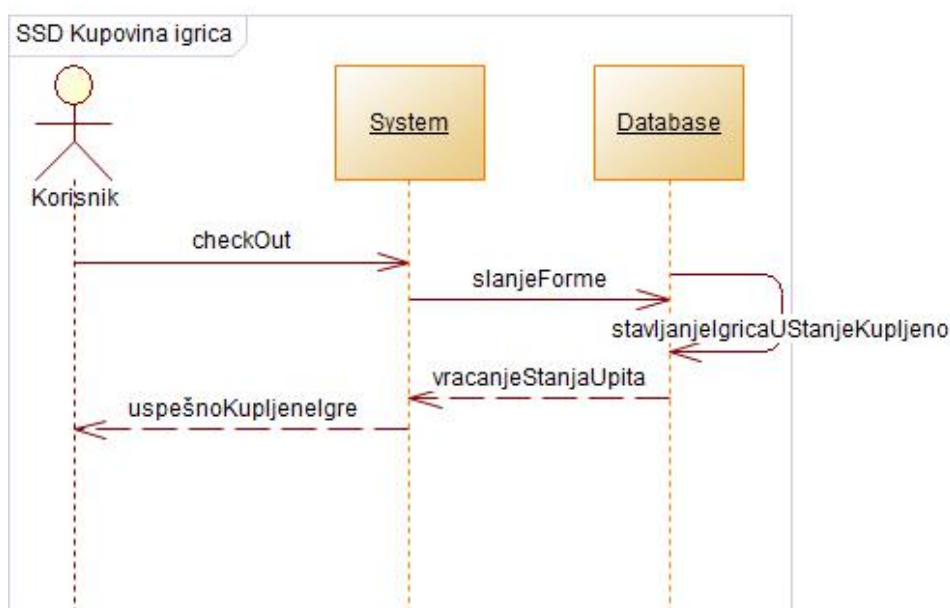
5.1.10 Brisanje igrice iz korpe



Slika 17- SSD - Brisanje igrice iz korpe

Korisnik bira igricu koju želi ukloniti iz korpe, sistem šalje bazi podataka SQL upit za brisanje igrice. Baza podataka pronalazi i briše igricu, vraća poruku sistemu o uspešnom izvršavanju upita, sistem šalje poruku korisniku o uspešnom uklanjanju igrice.

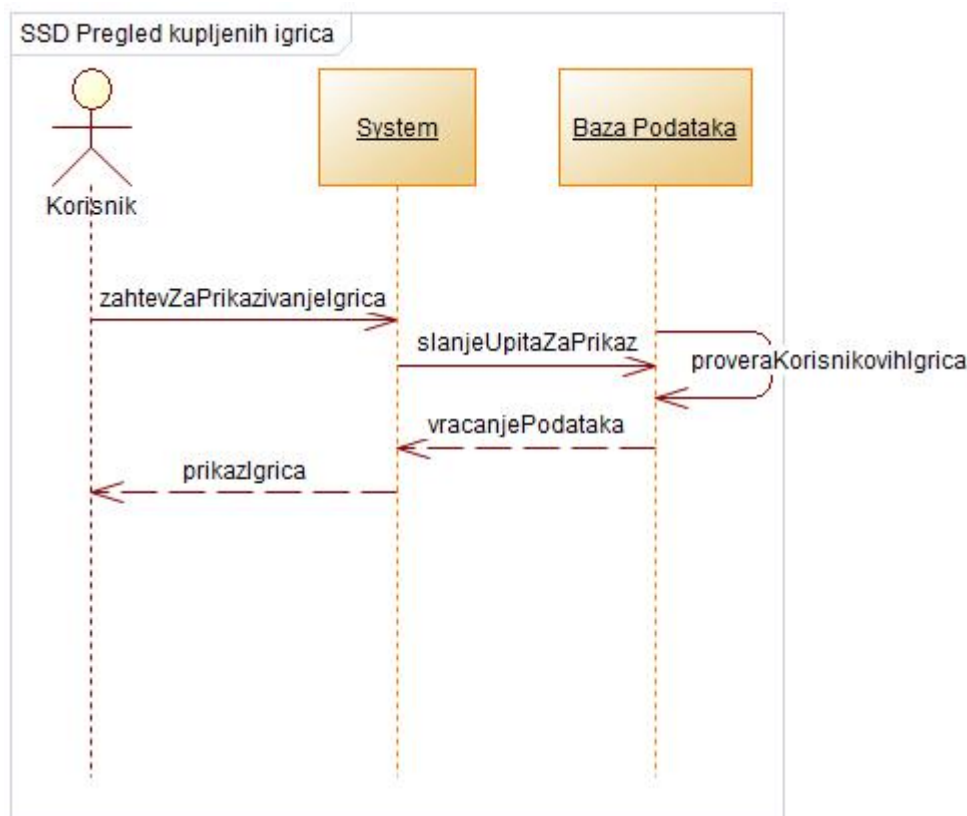
5.1.11 Kupovina igrice



Slika 18 - SSD - Kupovina igrice

Korisnik šalje sistemu preko dugmeta zahtev za kupovanje svih igrica korpe, sistem šalje bazi podataka SQL upit za kupovanje svih igrica. Baza podataka pritom sve igrice koje imaju stanje bivanja u korpi, prebacuje u stanje kupljeno, vraća poruku sistemu o uspešnoj transakciji, i na kraju sistem korisniku šalje o uspešnoj transakciji.

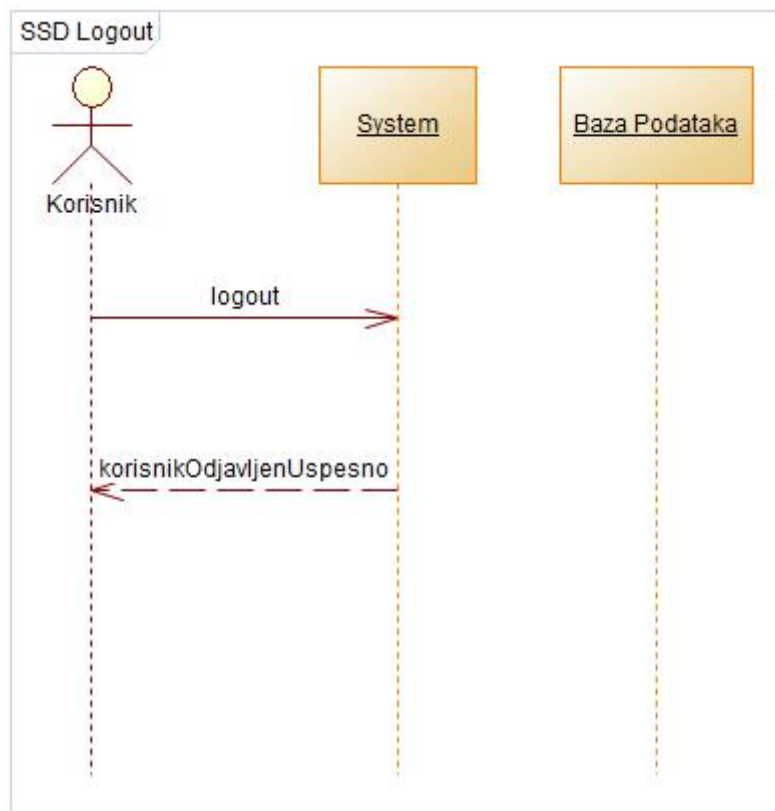
5.1.12 Pregled kupljenih igrica



Slika 19 - SSD - Pregled kupljenih igrica

Korisnik zahteva prikaz svih njegovih igrica iz baze, sistem šalje zahtev bazi podataka, odnosno SQL upit. Baza podataka proverava sve korisnikove igrice i sakuplja ih, vraća te podatke sistemu na prikazivanje za korisnika.

5.1.13 Logout

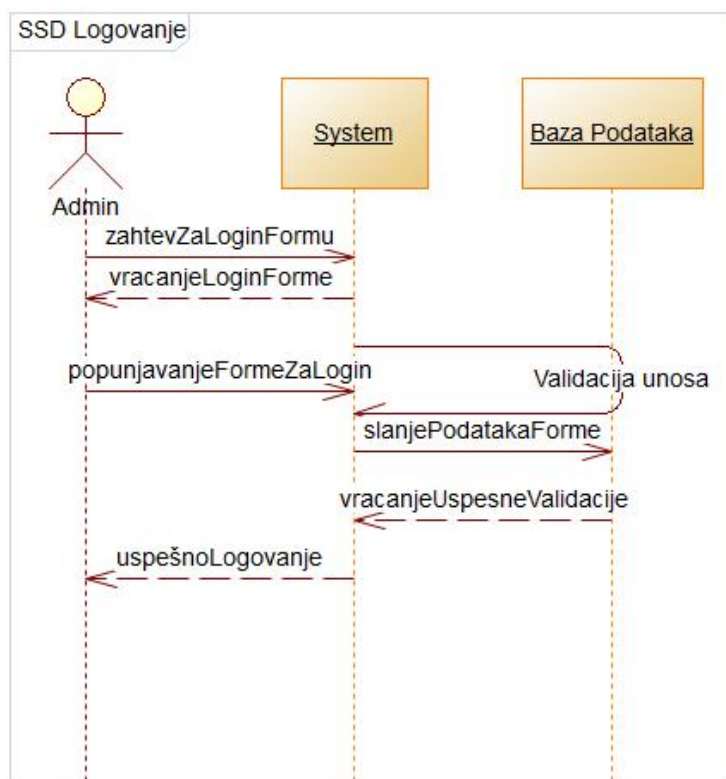


Slika 20 - SSD - Odjavljivanje korisnika

Korisnik se odjavljuje na sistem preko samog sistema, davajući mu naznaku klikom na dugme za logout, sistem vraća poruku da je korisnik uspešno izlogovan.

5.2 Administrator

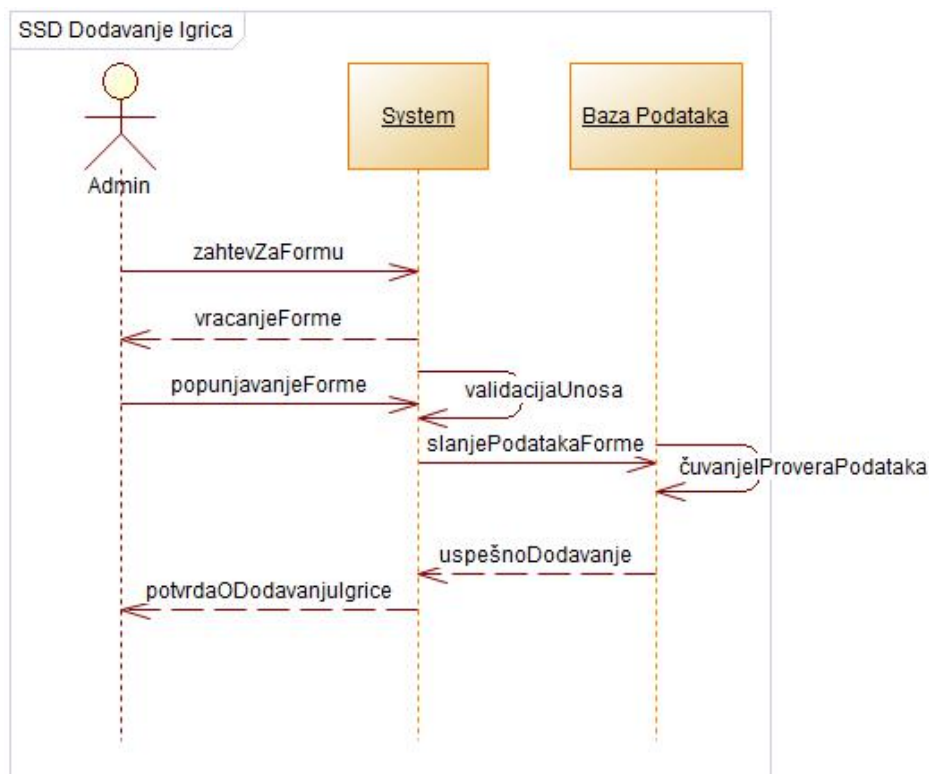
5.2.1 Prijavljivanje



Slika 21 - SSD - Prijavljivanje administratora

Postupak prijavljivanja administratora na sistem je isti kao i kod korisnika. Razlika je u povratnim informacijama koje baza podataka šalje sistemu, a to je stanje odnosno flag korisnika, a to je da je administrator, i da ga sistem prepozna kao istog i da mu dodatne privilegije.

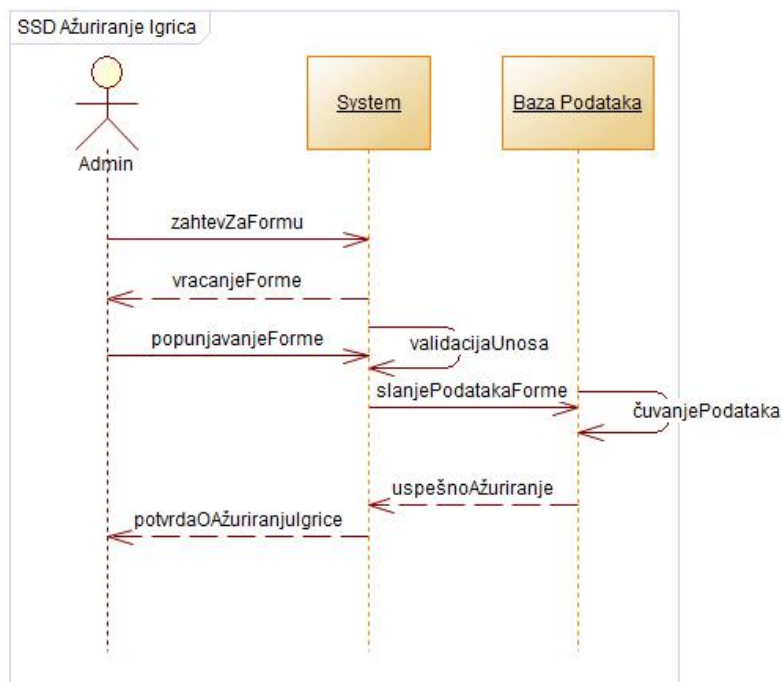
5.2.2 Dodavanje igrice



Slika 22 - SSD - Dodavanje igrice

Administrator šalje zahtev sistemu za formu za dodavanje nove igrice, sistem mu vraća stranu sa formom. Pritom administrator popunjava formu, sistem proverava validaciju unosa podataka, i pritom je šalje u SQL upit bazi podataka. Baza podataka čuva informacije, i vraća sistemu i sistem administratoru o uspešnom dodavanju.

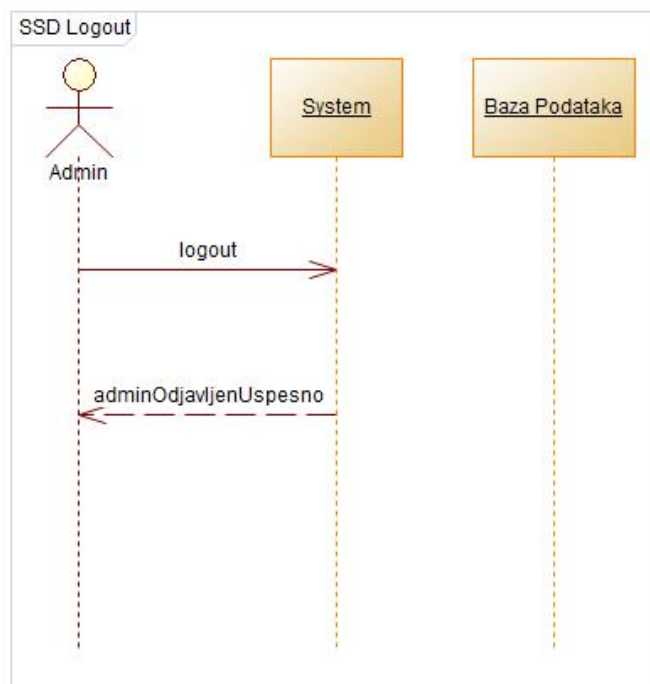
5.2.3 Ažuriranje igrice



Slika 24 - SSD - Ažuriranje igrice

Administrator ažurira igrice na isti način kao što ih dodaje. Prate se isti koraci, samo što se šalje upit za ažuriranje stanja igrice u bazi podataka, umesto unosa nove.

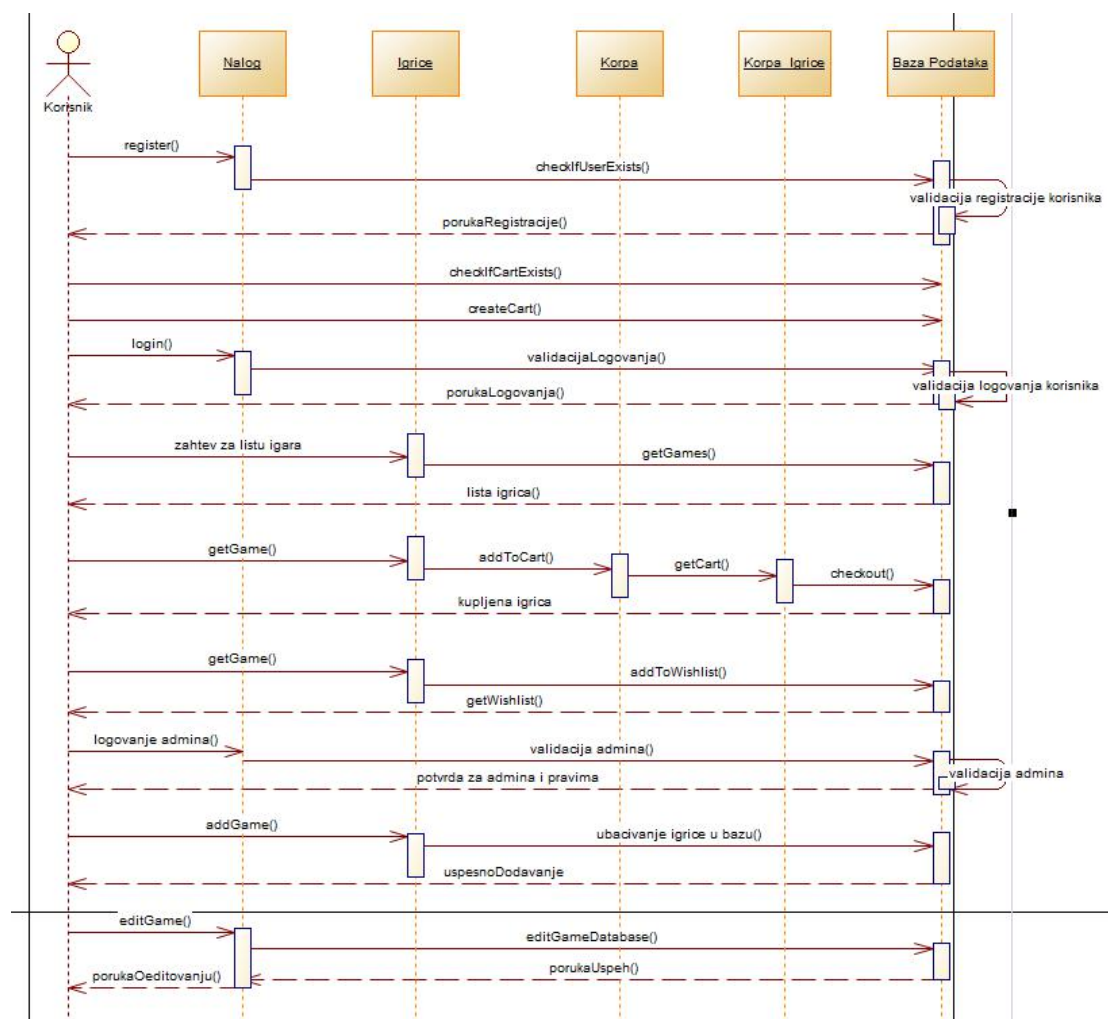
5.2.4 Odjavljivanje



Slika 24 - SSD - Odjavljivanje administratora

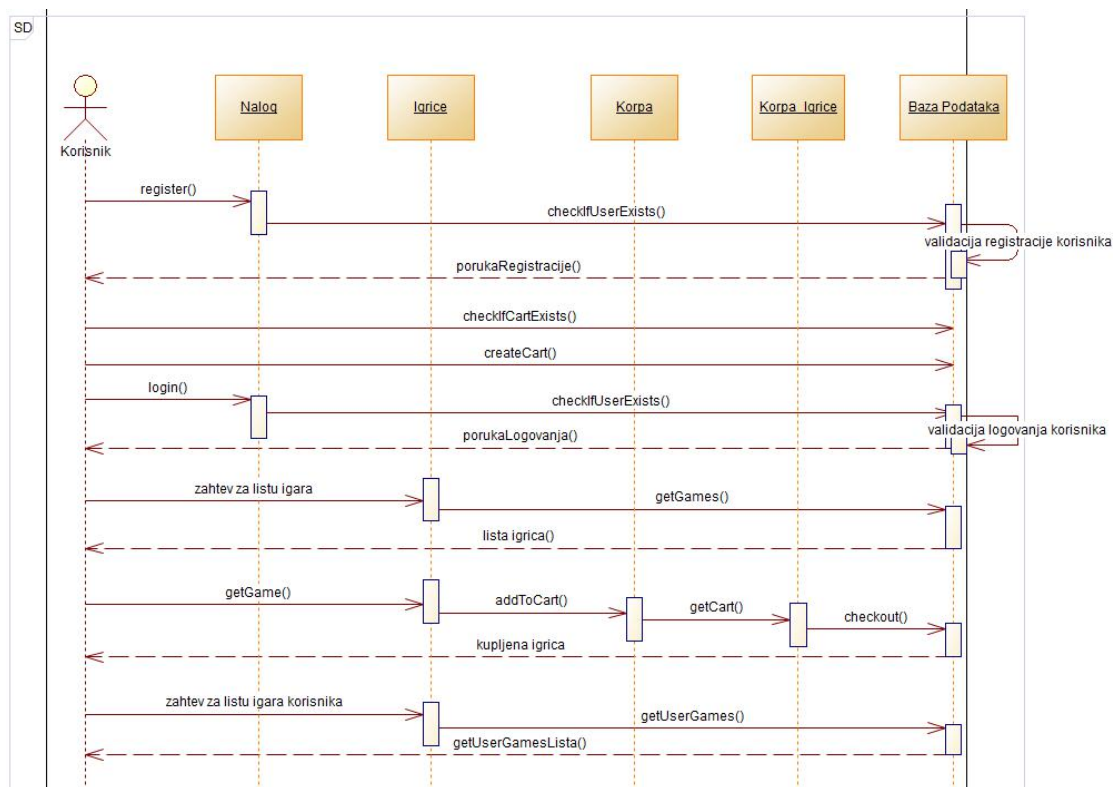
Administrator se odjavljuje sa sistema na isti način kao i korisnik, klikom na logout dugme.

5.3 Sekvencijalni dijagram(nivo upotrebljenih klasa)



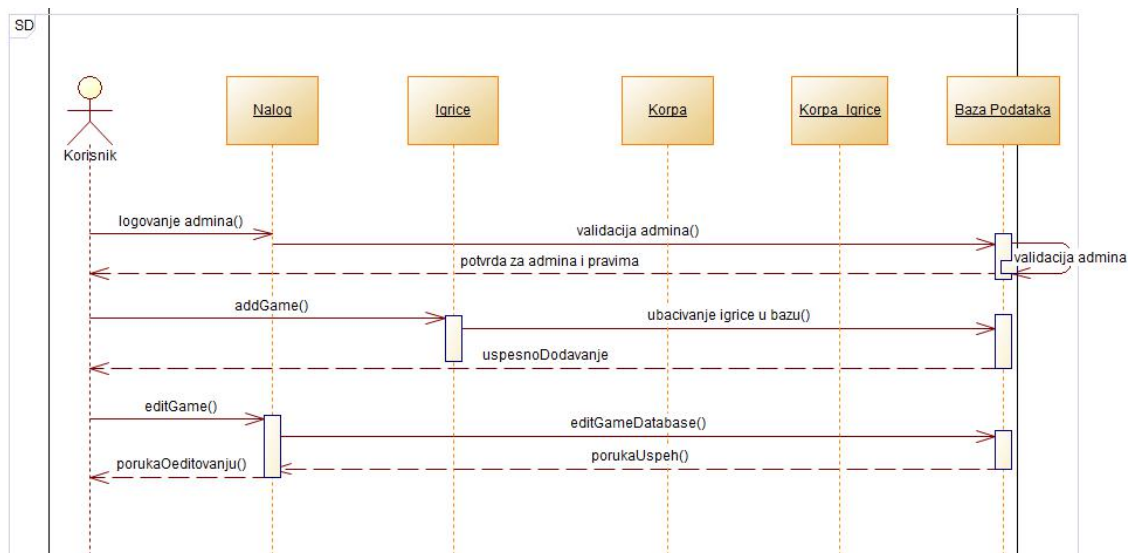
Slika 25 - Sekvencijalni dijagram

5.3.1 Korisnik



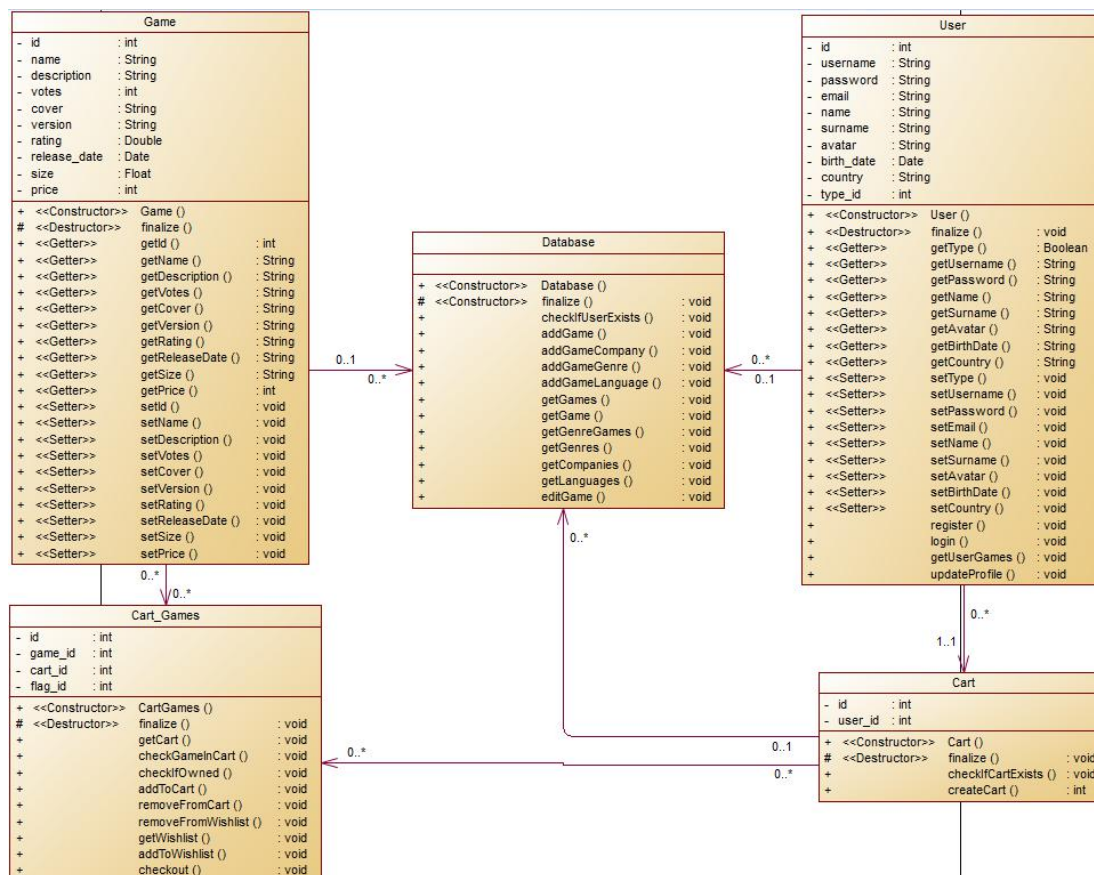
Slika 26 - SD - Korisnik

5.3.2 Administrator



Slika 27 - SD - Administrator

5.4 Klasni dijagram



Slika 28 - Klasni dijagram

Prikazan je klasni dijagram sistema, gde su prikazane klase: game, user, database, cart i cart_items.

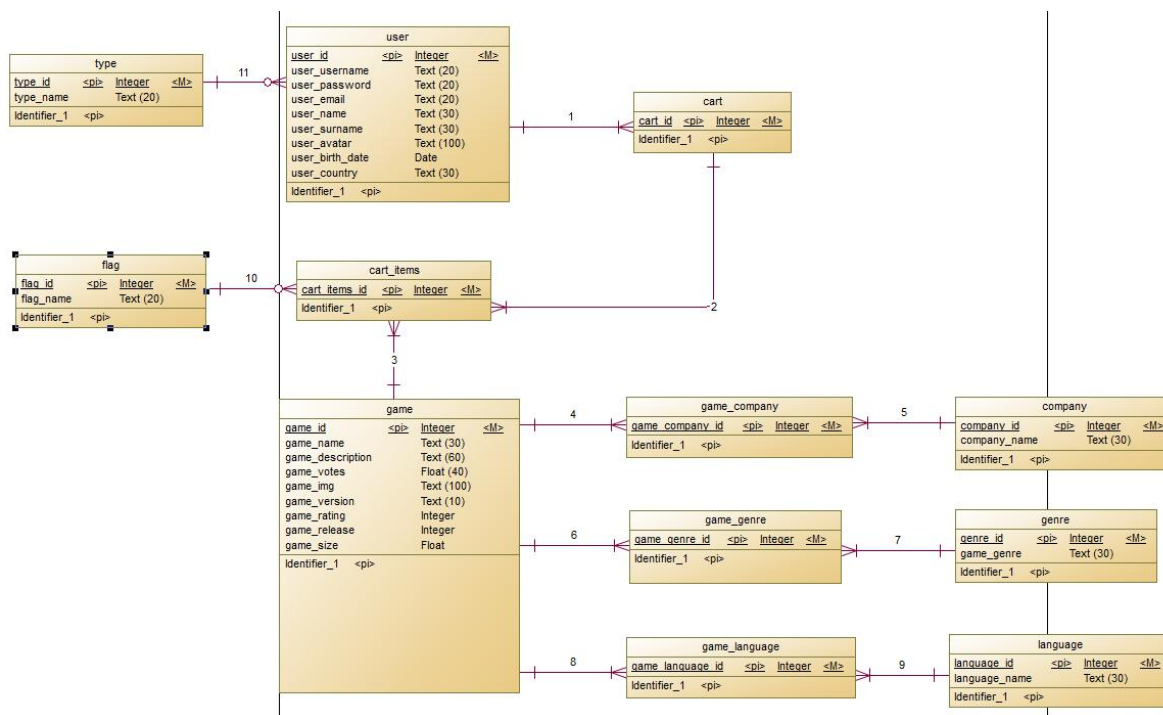
User metode: register(), login(), getUserGames(), updateProfile()

Cart metode: checkIfCartExists(), createCart()

Cart_Games metode: getCart(), checkGameInCart(), checkIfOwned(), addToCart(), removeFromCart(), removeFromWishlist(), getWishlist(), addToWishlist(), checkout()

Database metode: checkIfUserExists(), addGame(), addGameCompany(), addGameGenre(), addGameLanguage(), getGames(), getGame(), getGenreGames(), getGenres(), getCompanies(), getLanguages(), editGame().

5.5 Model baze podataka



Slika 29 - Konceptualni model baze podataka

Ovako izgleda konceptualni model baze podataka sistema. Konceptualni model baze podataka daje pogled na skup podataka(entitete i njihove atribute) i na veze koje ih karakterišu.

Ovde vidimo dakle tabele(entitete) iz baze, sa svim njenim atributima i relacijama. Prvobitno, imamo entitet "user", koja sadrži attribute kao što su ID, primaran ključ koji jedinstveno označava korisnika u bazi, koji se inkrementuje kao i ostali ID-jevi. Potom imamo username, password i email, kao "not null" attribute, koji će biti kreirani u togu registracije i uneti u bazu, heširana šifra, i neponovljiv username, kao i email. Tabela type, nam daje uvid u to da li je korisnik admin ili običan.

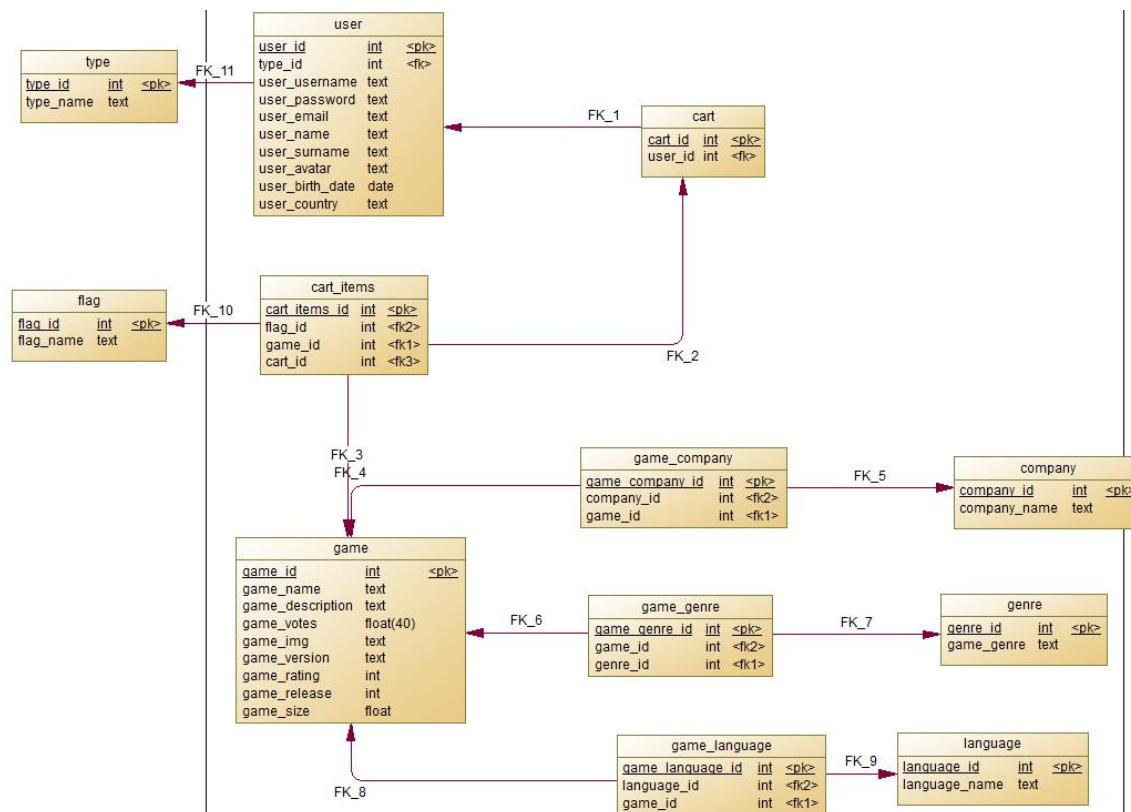
Svi ostali atributi entiteta user, kao što su name, surname, itd, su opcioni atributi korisnika koje može dodati prilikom kreiranja naloga, ovi atributi mogu biti null.

Kao najveći entitet, "Game", ima svoje attribute kao što su naziv, opis, verzija, itd, kao i ostale koje su odvojene radi kardinalnosti i veze više-na-više. Ti atributi su zapravo zbog te veze više-na-više napravljene, odnosno izdvojene kao posebne tabele sa tabelom između. Ti atributi su: "company"(developer), "genre", kao i "languages", jer igrice može biti izdata u više jezika.

Korpa je napravljena tako, što postoji cart i cart_items. U okviru cart entiteta, id korisnika ima svoj id korpe. Cart_items predstavlja porudžbine, odnosno sve igre koje korisnik bude stavio u korpu.se sastoji iz id-ja igrice, id-ja korpe u kojoj će se nalaziti igrice, kao i flag.

Flag u korpi predstavlja tri stanja korpe.

- 1) Prvo stanje predstavlja stanje bivanja u korpi(inCart).
- 2) Drugo stanje predstavlja stanje kada korisnik poseduje, tj je kupio igricu(owned).
- 3) Treće i finalno stanje predstavlja da je igrice u wishlisti.



Slika 30 - Fizički model baze podataka

U fizičkom modelu možemo videti zapravo kako izgleda naša relacionalna baza podataka, sa svim primarnim i sekundarnim(*foreign key* - *FK*) ključevima.

6. Arhitektura sistema

Arhitektura softvera ukazuje nam kako bi sistem trebalo da bude organizovan i prikazuje ukupnu strukturu sistema. Projektovanje arhitekture je prva faza procesa projektovanja softvera. To je kritična veza između inženjerstva zahteva i projektovanja softvera, jer utvrđuje glavne strukturne komponente u sistemu i veze između njih.

Ovde ćemo definisati korišćen šablon arhitekture sistema, kao i definisati aplikacijski dijagram i infrastrukturne dijagrame.

6.1 Šablon arhitekture sistema - MVC

Dizajn šablon za kreiranje arhitekture sistema je **Model-View-Controller(MVC)** arhitektura, koja odvaja prikaz informacija od interakcije korisnika sa tim informacijama.

MVC odvaja elemente sistema dozvoljavajući im da se menjaju nezavisno jedni od drugih. Dodavanje novog pogleda ili menjanje već postojećeg se može izvršiti bez bilo kakvih promena osnovnih podataka u modelu.

Korisnimo MVC arhitekturu jer i sam Angular implementira MVC arhitekturu radi razdvajanja komponenti za prezentaciju, podatke i logičkih komponenti.

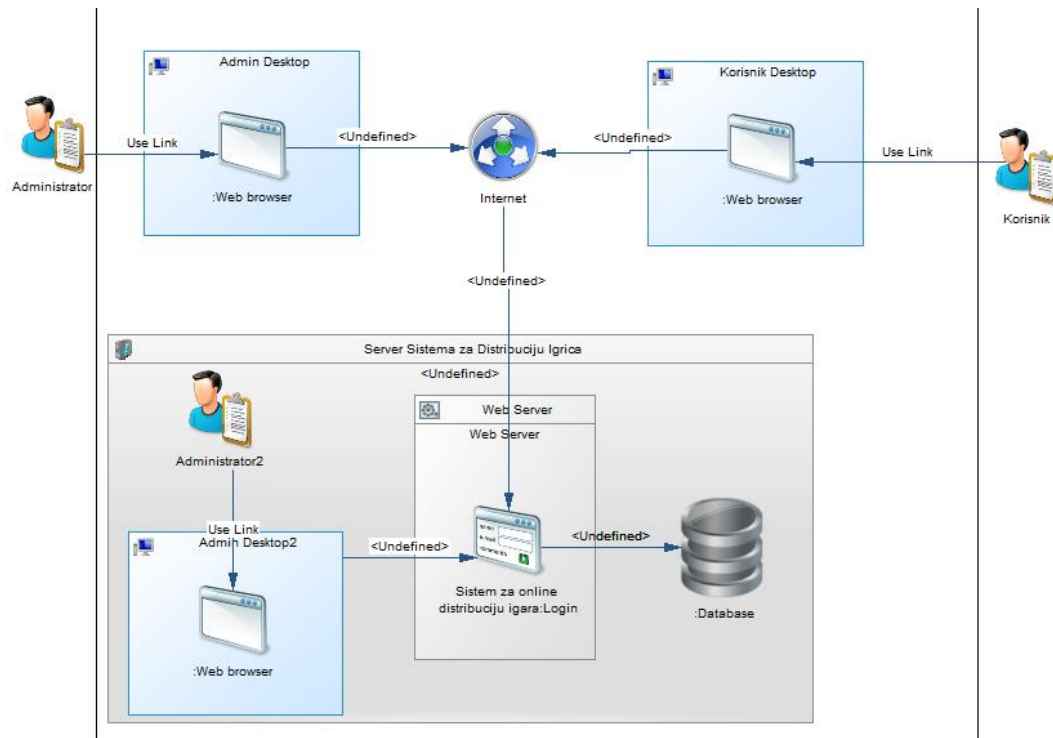
MVC deli softverski sistem u 3 dela:

- 1) View - obežbeđuje pogled, interfejs korisniku, pomoću kojeg korisnik obavlja i izvršava funkcije koje treba da se izvrše nad modelom.
- 2) Controller - osluškuje i prihvata zahtev od klijenta za izvršenje operacije. Nakon toga poziva operaciju koja je definisana u modelu, i ukoliko model promeni stanje, obaveštava view o promeni stanja.
- 3) Model - predstavlja stanje sistema koje mogu promeniti operacije modela.

Razlog za korišćenje ove arhitekture, je pre svega jer Angular implementira ovu arhitekturu, view je definisan kroz HTML, dok je model i kontroler kroz Javascript odnosno Typescript.

Prednosti ove arhitekture jesu što nam omogućava brži rad u kodu, mogućnost da napravimo više pogleda za jedan model, promena koda ne utiče na ceo model, ali MVC zato jeste jako kompleksan dizajn patern.

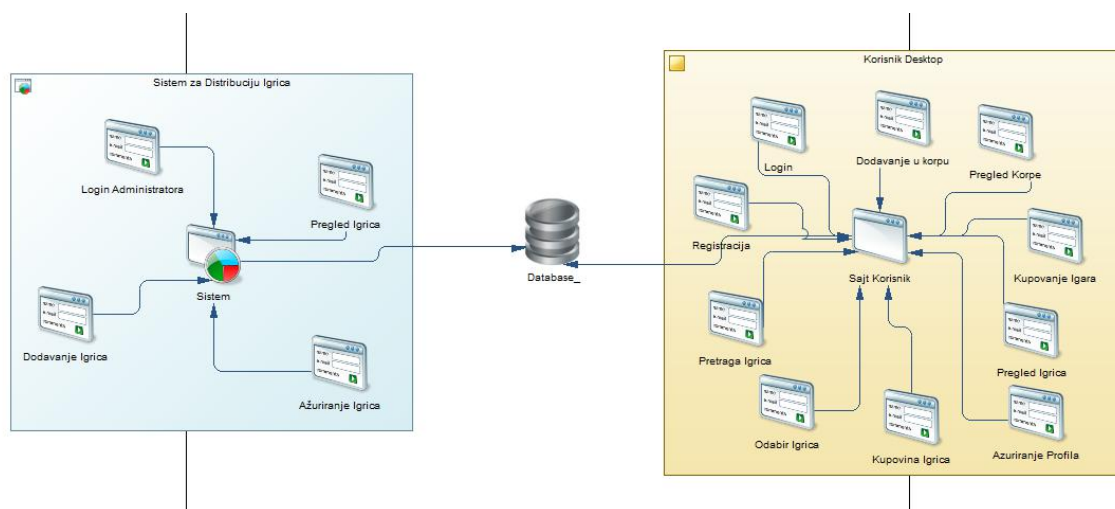
6.2 Infrastrukturni dijagram



Slika 31 - Infrastrukturni dijagram

Na slici 31. možemo videti infrastrukturni dijagram sistema. Postoji korisnik i primer 2 administratora, koji pristupaju preko desktop računara. Administrator može pristupiti sistemu direktnim pristupom, kao i bazi podataka, tako i preko internet pretraživača. Korisnik može pristupiti sistemu preko internet pretraživača.

6.3 Aplikacijski dijagram



Slika 32 - Aplikacijski dijagram

U aplikacijskom dijagramu, smo podelili korisnike, odnosno aktere u korisnika i administratora, odnosno sistem. U okviru ova dva područja vidimo funkcije koje su vezane za jednog i drugog aktera posebno, koja oba imaju pristup bazi podataka.

Komponente u dijagramu su već bile opisane, kao što su login ,registracija, dodavanje igrica u korpu, ili kod administratora u sistem, itd.

7. Testiranje i implementacija

Testiranje služi da se utvrdi da li program radi ono bi trebalo da radi i da se otkriju greške u programu pre nego što se pusti u upotrebu. Testiranje se vrši puštanjem da softver radi, pri čemu se koriste test podaci. Onda se proveravaju rezultati testiranja da bi se videlo da li oni ukazuju na neku grešku, nepravilnost i da bi se došlo do informacija o nefunkcionalnim svojstvima softvera.

Testiranje ima cilj da pokaže inženjeru kao i kupcu proizvoda, da dokaže da softver zaista ostvaruje svoje ciljeve, i radi kako treba.

Još jedan cilj jeste otkrivanje situacije u kojem ponašanje softvera nije ispravno, ili se ne reaguje kako je zamišljeno, otkrivanje bagova softvera.

Mi ćemo sada sprovesti testove za proveru ispravnosti softvera za online distribuciju igrica, koristeći *Selenium*.

7.1 Selenium

Selenium je besplatan framework za testiranje veb aplikacija, koji podržava sve veb pretraživače. On se koristi tako što ga pre svega preuzmemo kao google plugin, i pritom pokrenemo ga.

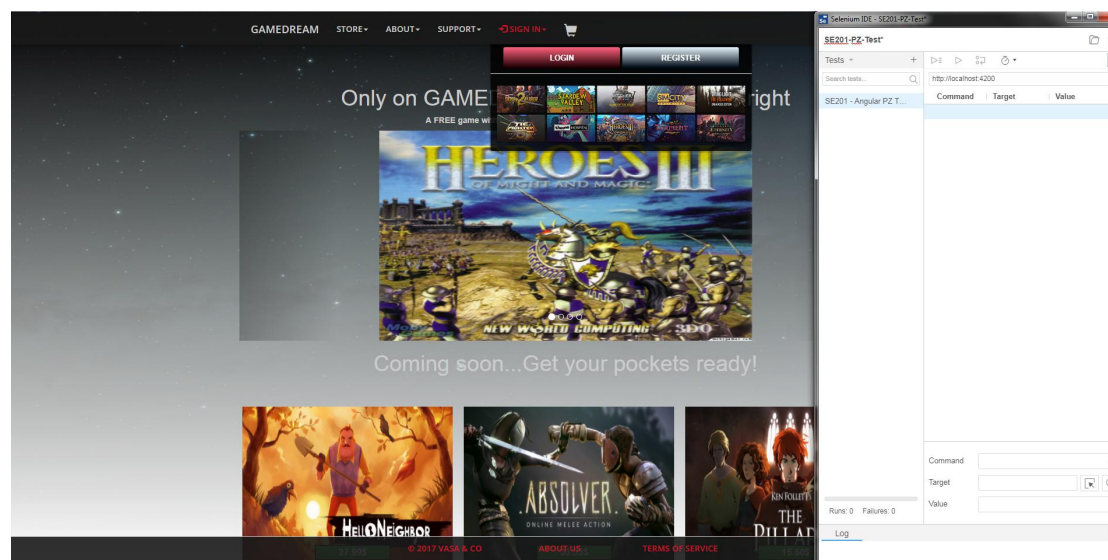
Kada se pokrene, kliknemo na *start recording*, gde selenium softver počinje da snima sve akcije koje se dešavaju na veb pretraživaču. Svaki klik, odabir html taga i elementa, kao i samu vrednost odabranog elementa, on uči.

Kada sprovedemo sve korake i klikove, selenium ih je sve pohvatao, onda možemo ići na *Run current test* gde će Selenium softver pritom sam obaviti sve korake koje je on snimio. Kroz svaki korak, registraciju, login će proći.

Selenium je veoma dobari kvalitetan softver za testiranje veb aplikacija.

7.2 Testiranje i prikaz softvera

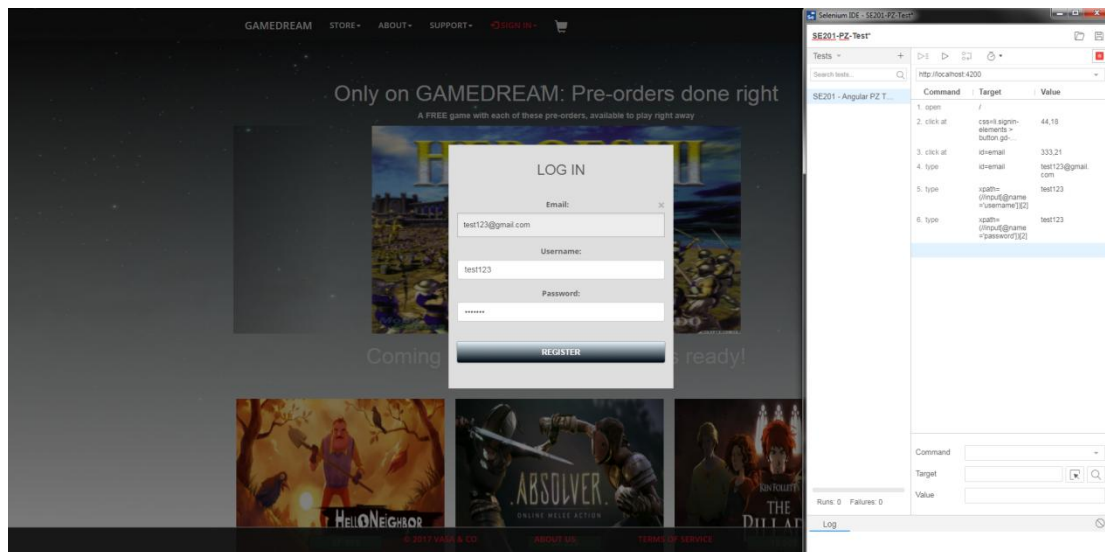
Sada ćemo obaviti sve funkcije sistema, i snimićemo ih, prikazati sam sistem kroz slike, i dokazati uspešan test.



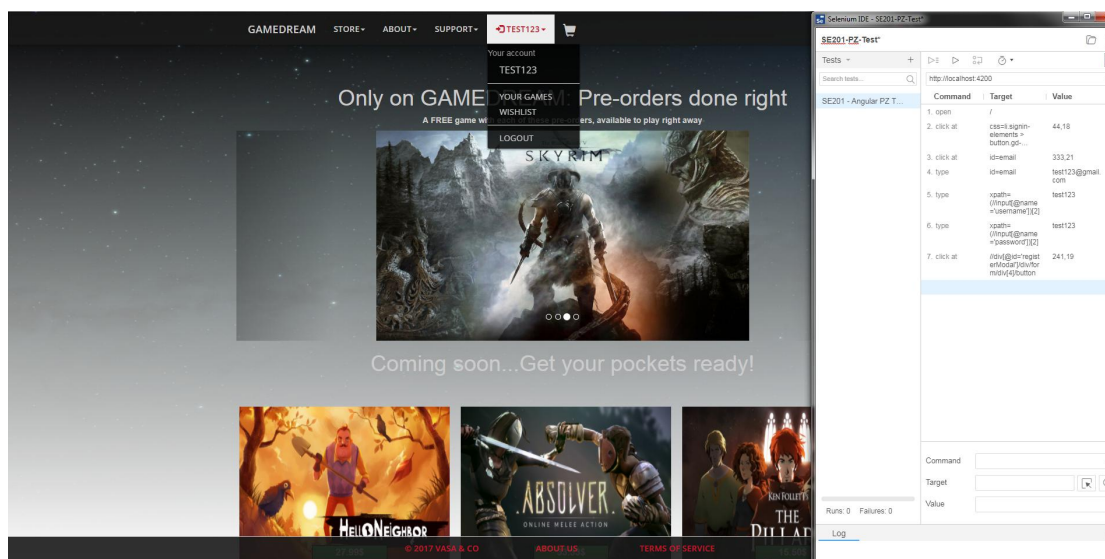
Slika 33 - Selenium testiranje - početna strana

Na slici 33 možemo videti početku stranicu kao i Selenium IDE sa desne strane ekrana. Počinjemo live snimanje klikom na crven krug, koji je sada kockica crvena, što znači da već snima podatke.

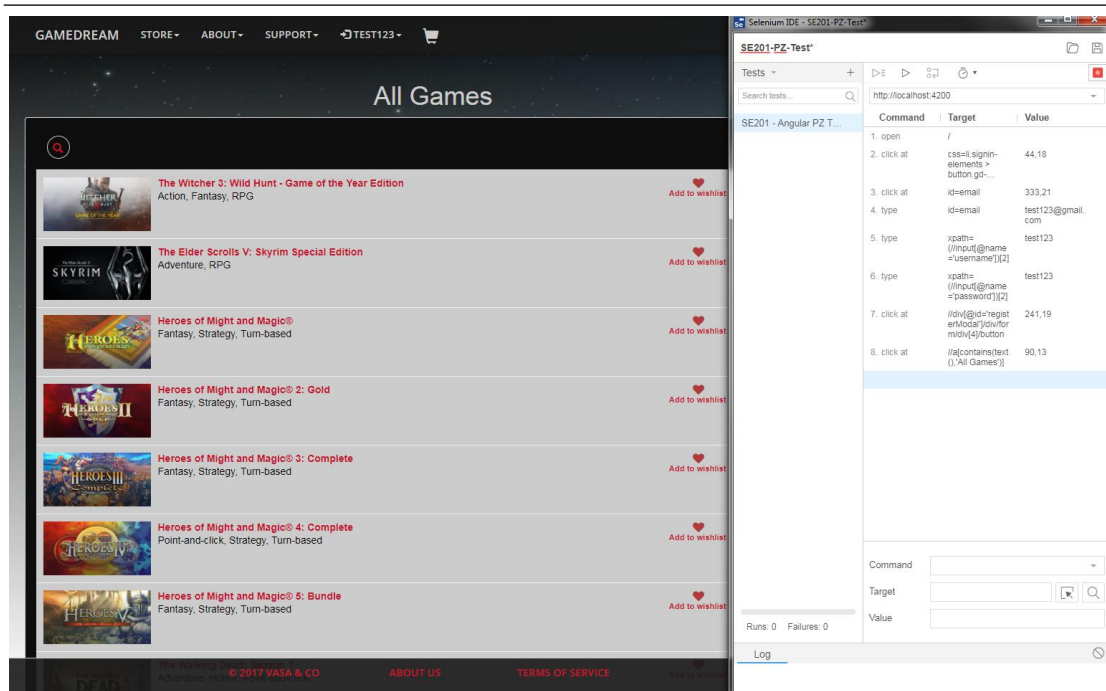
Na svakoj slici ćemo sa desne strane moći videti kako selenium softver hvata sve podatke.



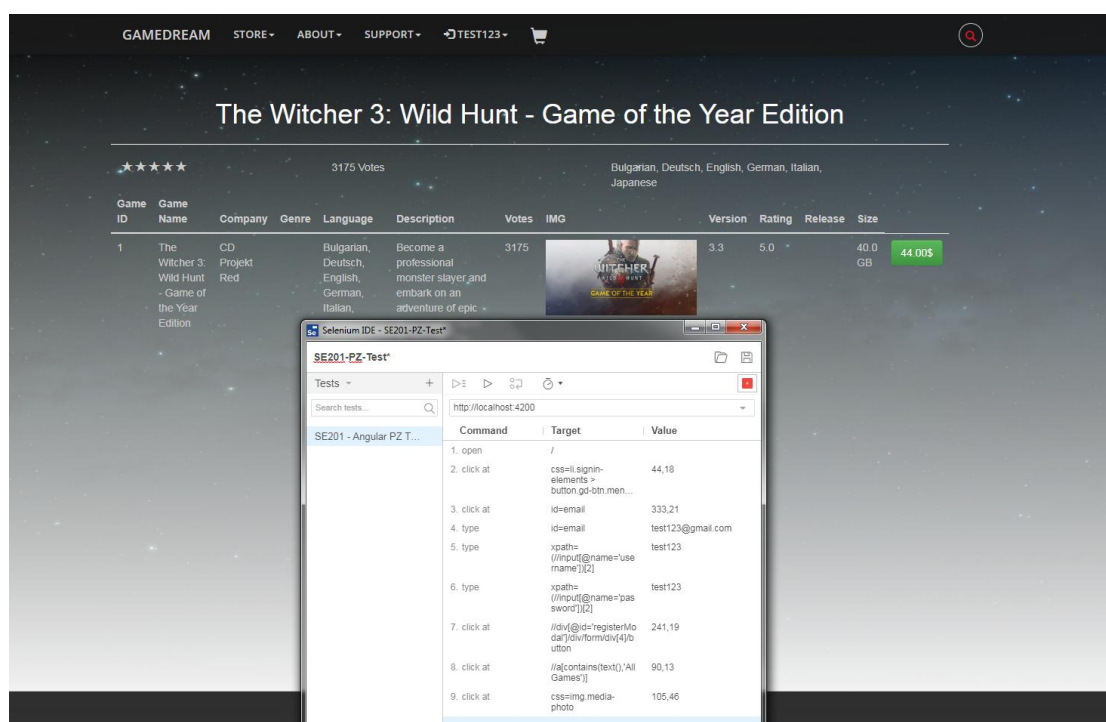
Slika 34 - Selenium testiranje - Registracija



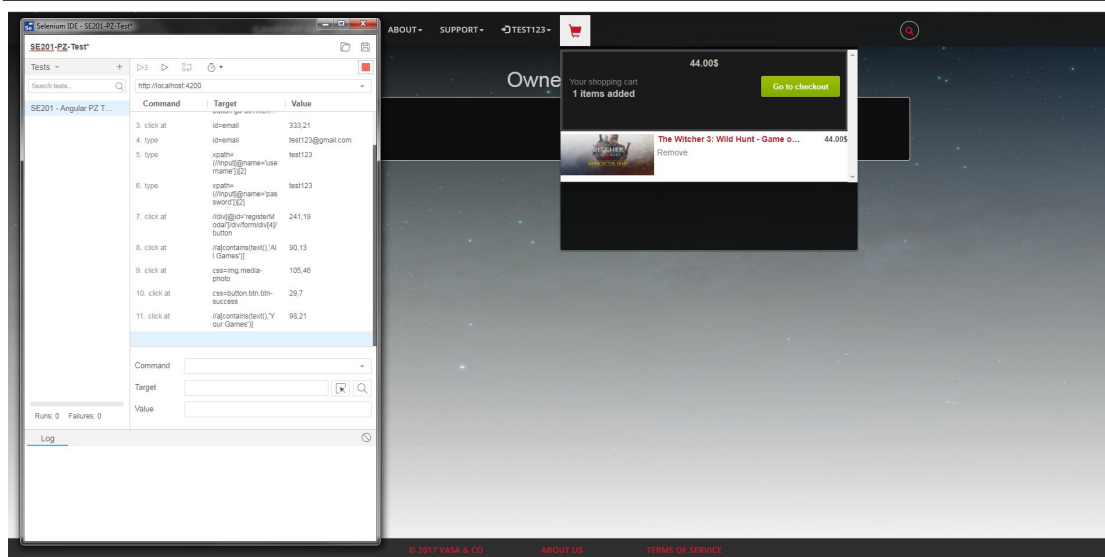
Slika 35 - Selenium testiranje - Uspešna registracija



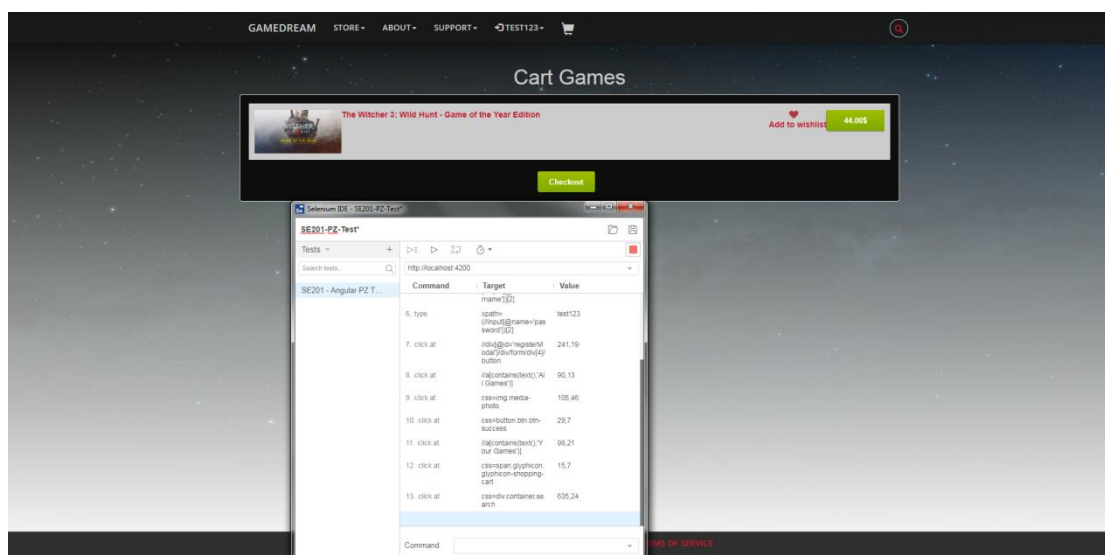
Slika 35 - Selenium testiranje - Prikaz svih igrica



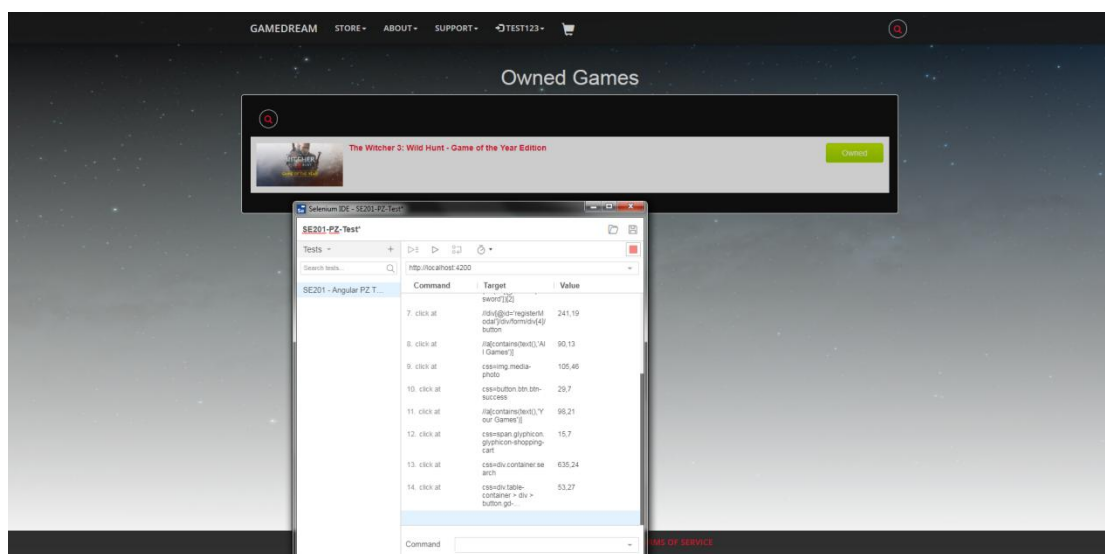
Slika 36 - Selenium testiranje - Prikaz profila igrice



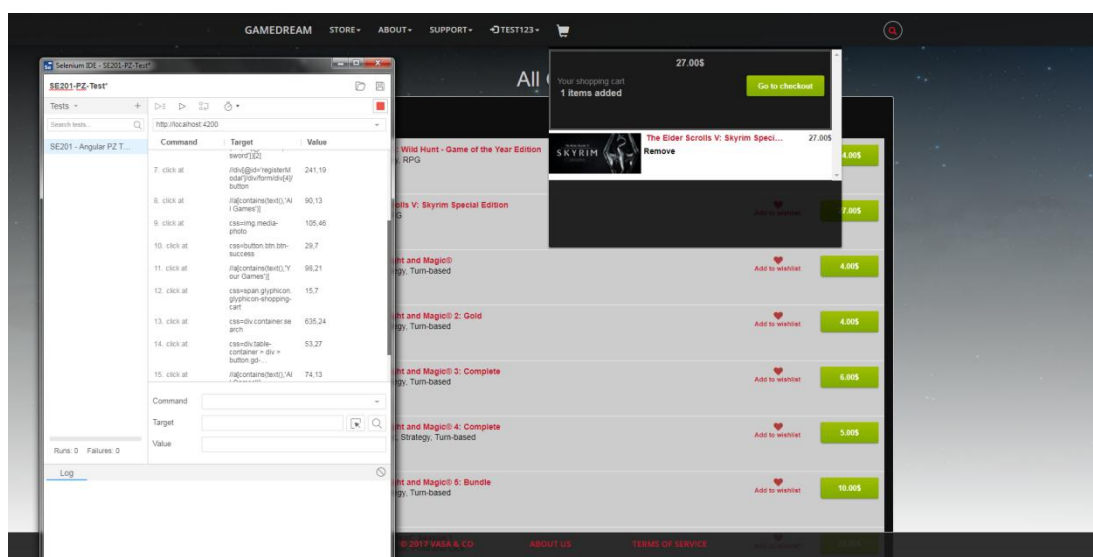
Slika 37 - Selenium testiranje - Dodavanje igrice u korpu



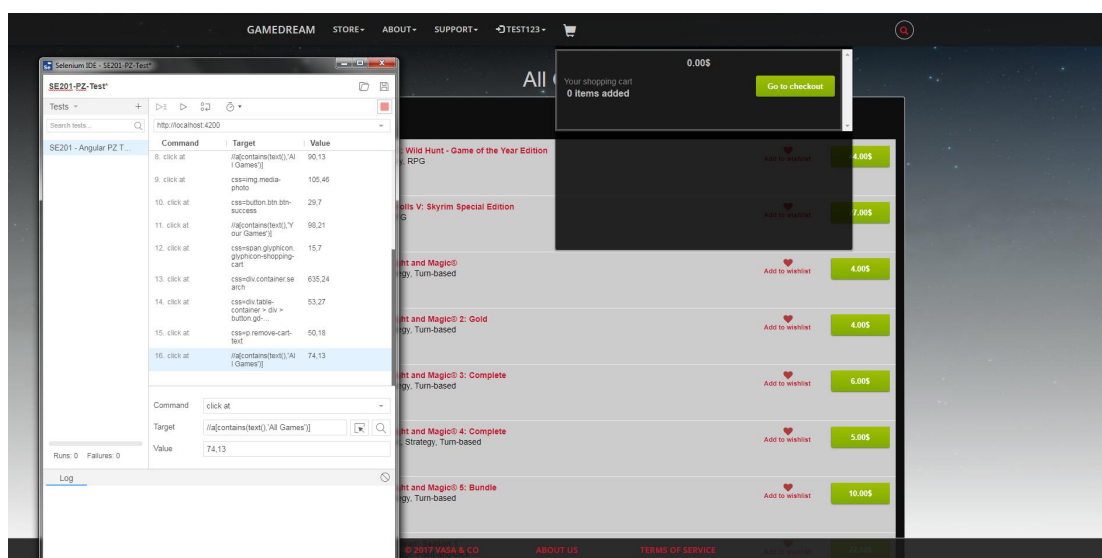
Slika 38 - Selenium testiranje - Prikaz igrice iz korpe



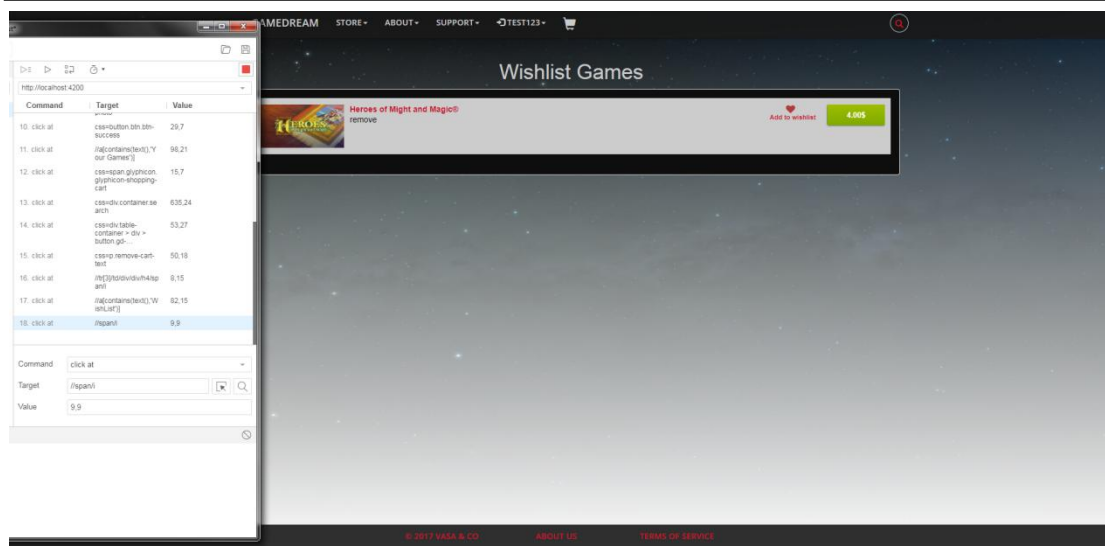
Slika 39 - Selenium testiranje - Kupljena igrice, prikaz kupljenih igrice



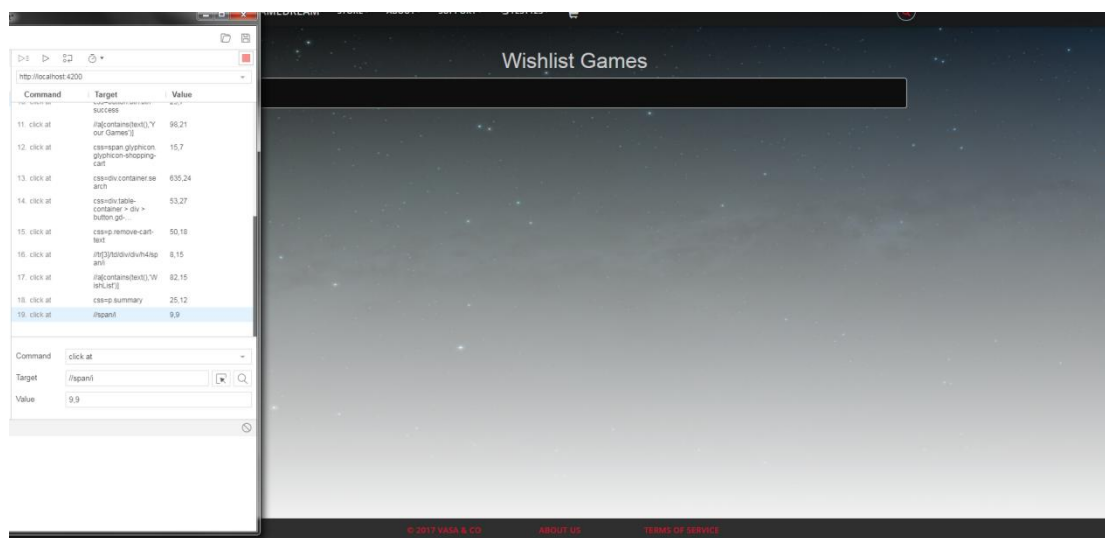
Slika 40 - Selenium testiranje - Uklanjanje igrice iz korpe



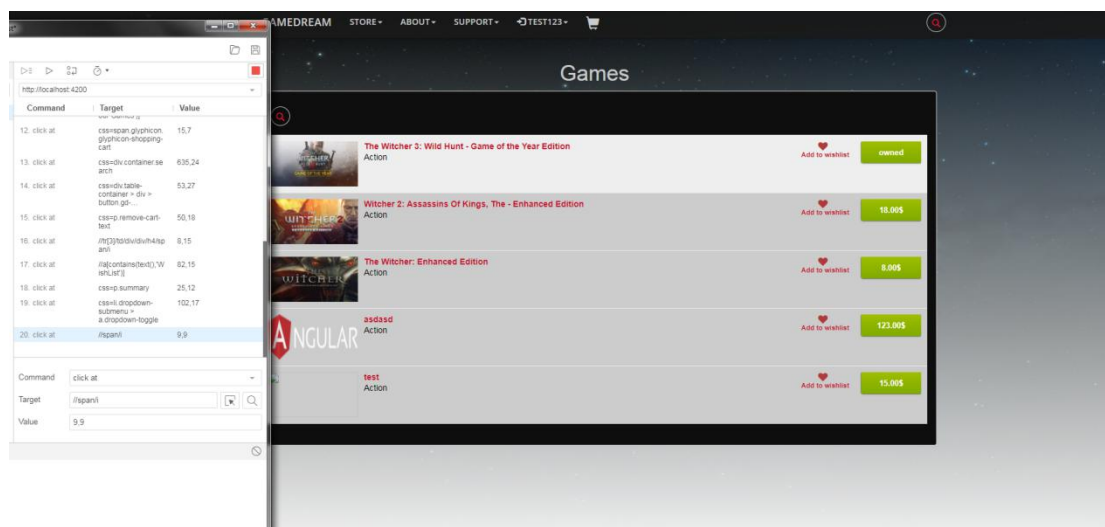
Slika 41 - Selenium testiranje - Uklanjanje igrice iz korpe uspešno



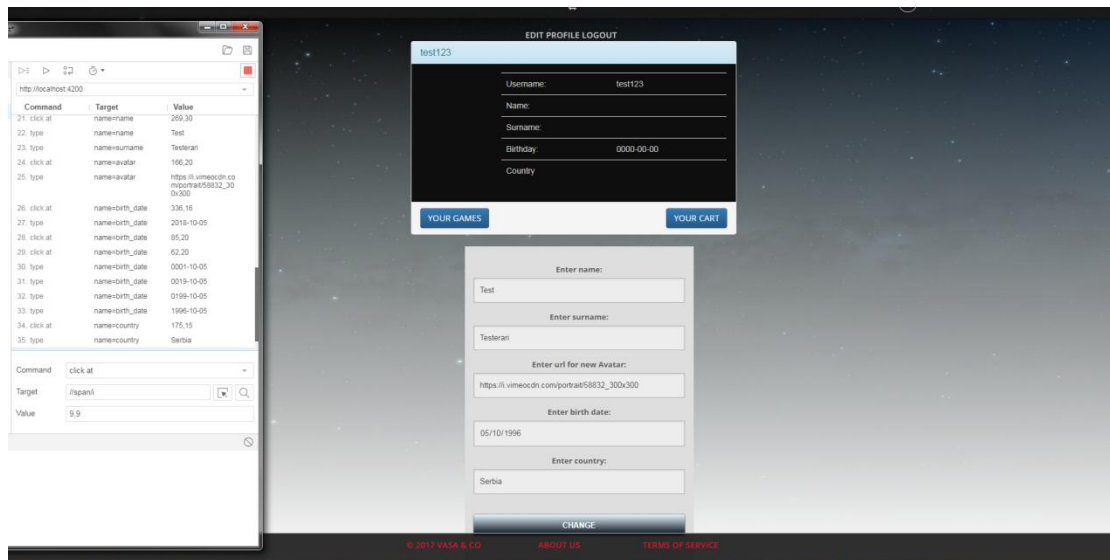
Slika 42 - Selenium testiranje - Dodavanje igrice u wishlistu, prikaz wish liste



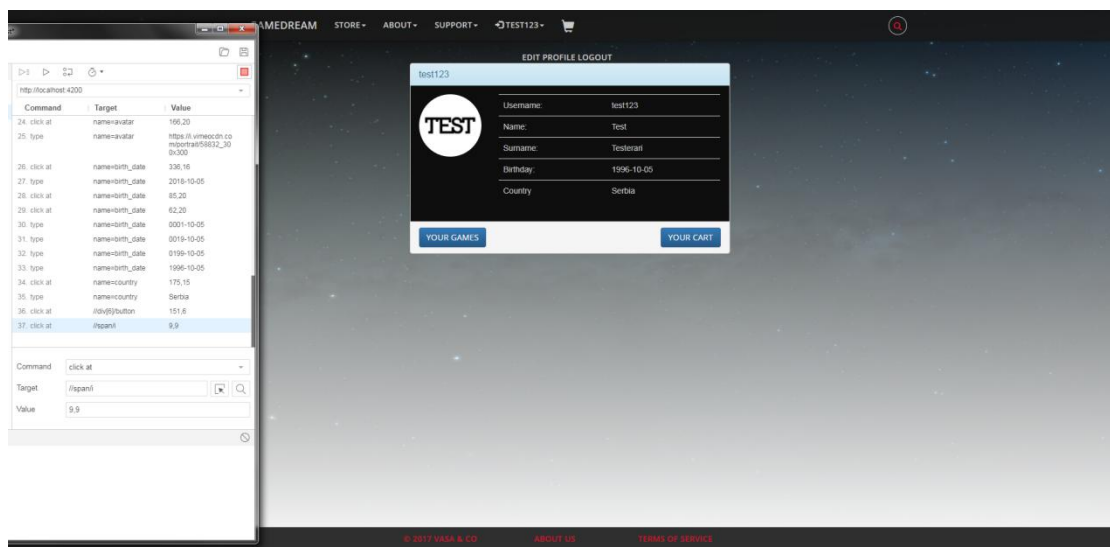
Slika 43 - Selenium testiranje - Uspešno uklonjena igrice iz wishliste



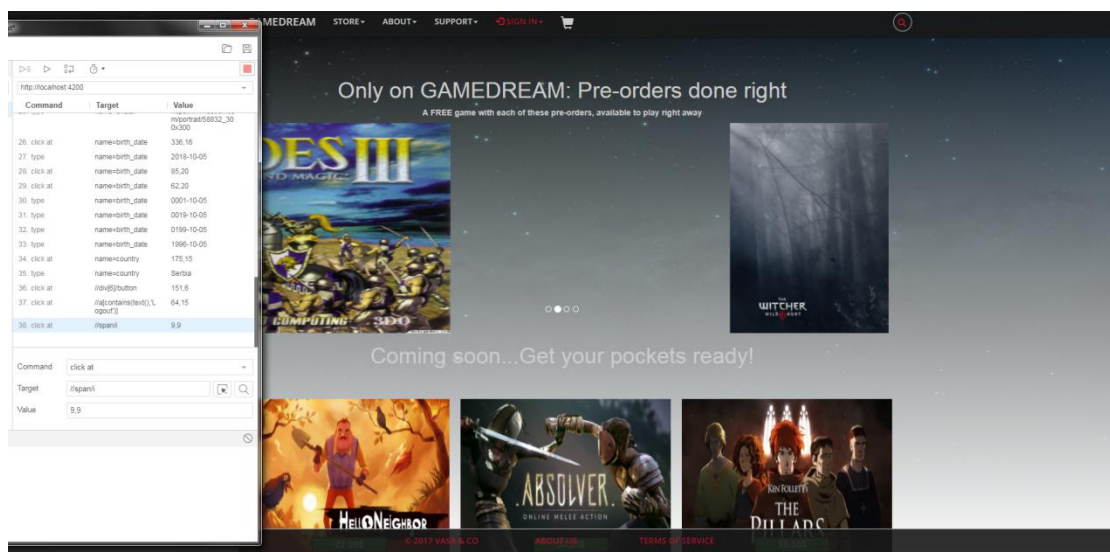
Slika 44 - Selenium testiranje - Prikaz igrice po jednom žanru



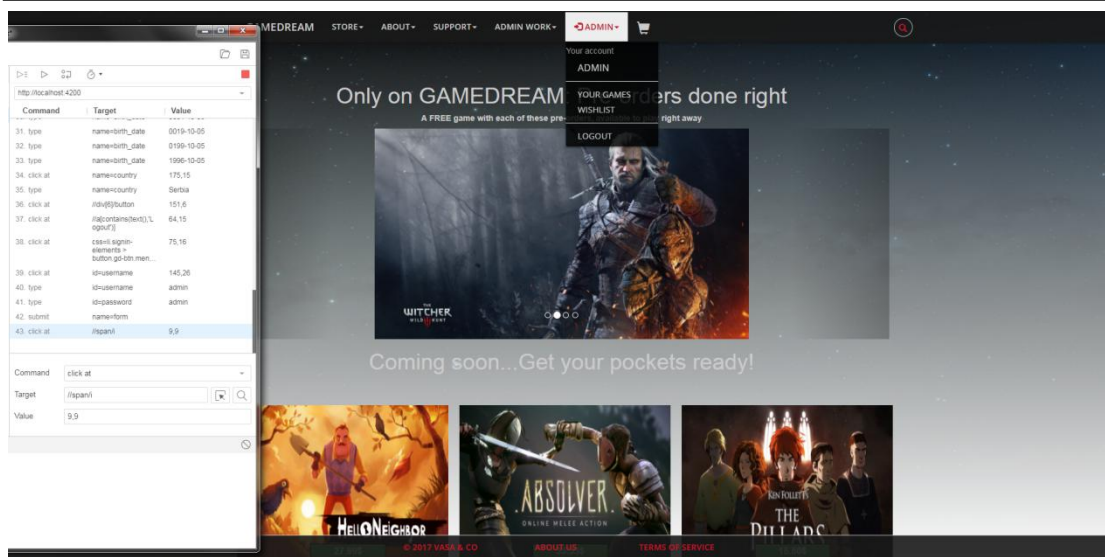
Slika 45 - Selenium testiranje - Prikaz profila, editovanje i unos podataka



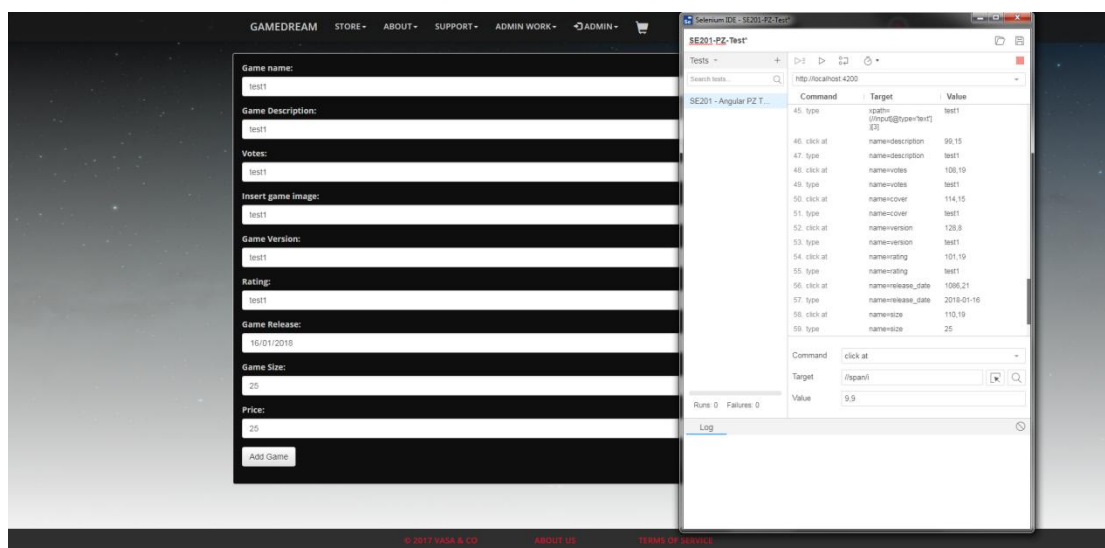
Slika 46 - Selenium testiranje - uspešno ažuriranje profila



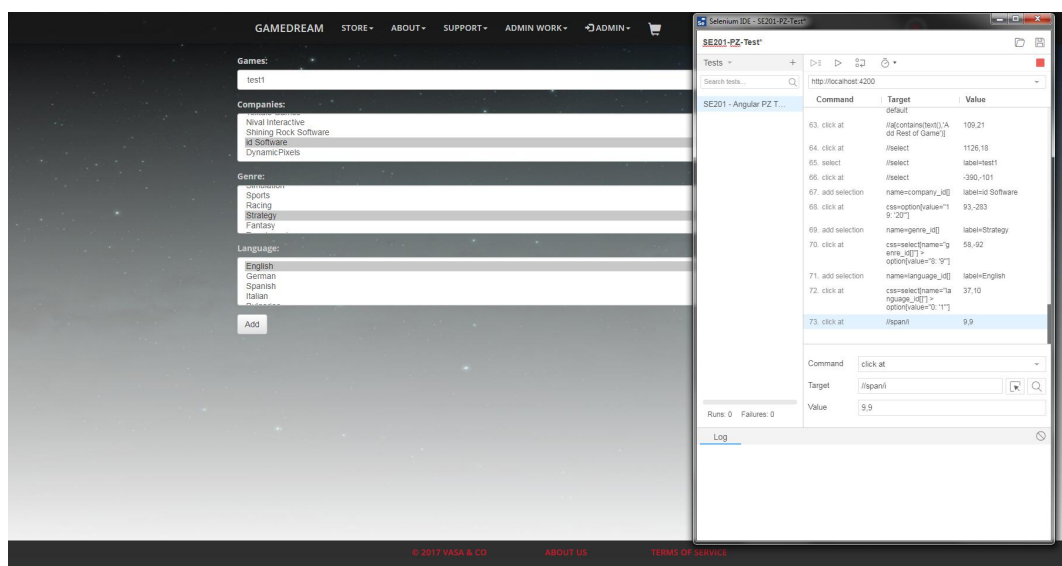
Slika 47 - Selenium testiranje - Odjavljivanje



Slika 48 - Selenium testiranje - Administrator prijavljen



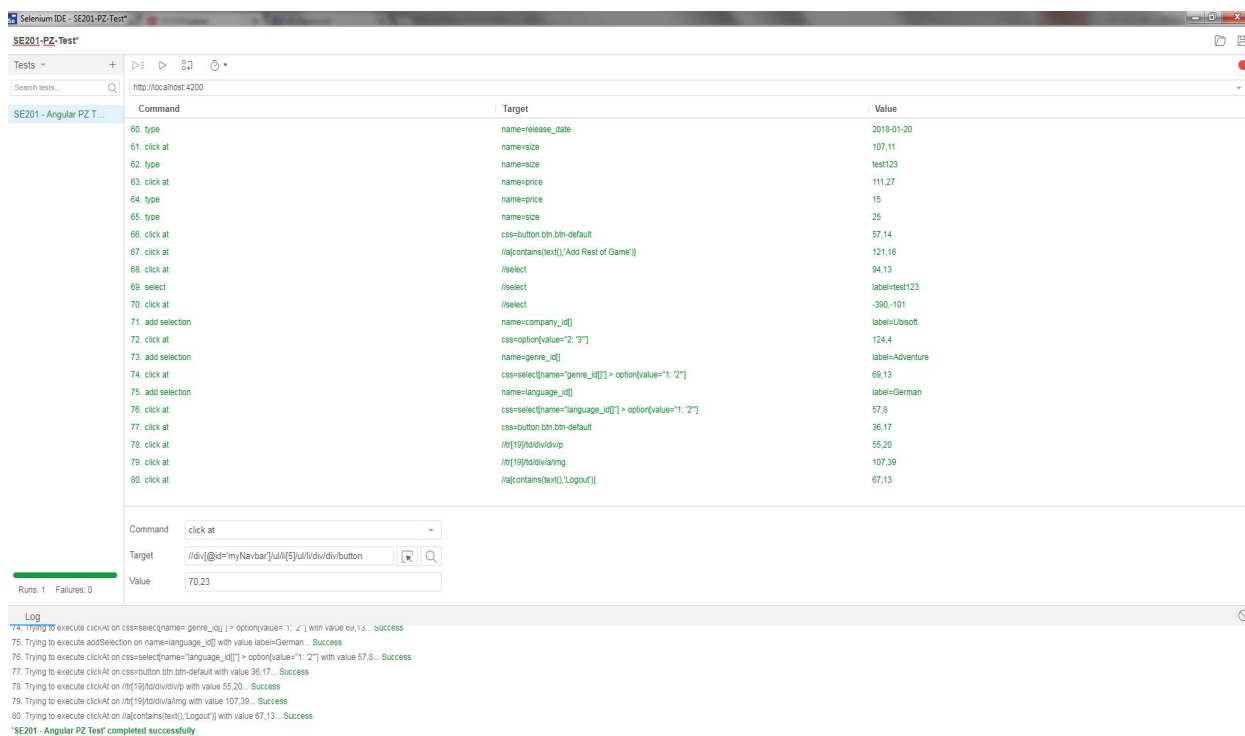
Slika 49 - Selenium testiranje - dodavanje igrice i parametara



Slika 50 - Selenium testiranje - ažuriranje igrice



Slika 51 - Selenium testiranje - uspešno dodavanje i ažuriranje igrice, prikaz iste



Slika 52 - Selenium testiranje - Run current test - Uspešan u svakom koraku

Na kraju pokrećemo test, odnosno dajemo Selenium softveru da sam uradi sve korake koje smo i mi obavili. Možemo videti da su svi koraci uspešno prošli.

Napomena: Svi testovi su prošli bez greške, samo što, prilikom registrovanja ili sličnih akcija, selenium nije mogao nastaviti, razlog za tim je što recimo kod registracije, dati korisnik sa datim uhvaćenim podacima od strane Seleniuma su već uneti u bazi i postojeći, a postoji uslov da se ne može registrovati isti korisnik u bazi, ili dodati u korpu već kupljena igrice ili slično. U toku tih nemogućih koraka, ja sam samo malo izmenio unos u podatke, kao prilikom registrovanja, i Selenium je nastavljao sve do kraja i uspešno obavio.

8. Korisničko uputstvo

Za pokretanje aplikacije, potrebno je smestiti više stvari.

Prvo, potrebno je instalirati XAMPP ili WAMP lokalni server, preko kojeg ćemo pristupiti phpmyadmin-u, i tamo importovati 'gamedream.sql', tada imamo bazu, na bazu navigiramo se u pretraživaču na <http://localhost/phpmyadmin/>

PHP folder 'gamedream' smeštamo u "xampp/htdocs" uz prefiks "C:/" ukoliko nije premešten folder.

Potom sledi Angular framework. Za koji je potrebno imati Node JS packate manager (NPM). Potrebno je pokrenuti 'cmd' i u njemu otići na putanju gde se nalazi folder sa angularom. Kada se navigira tamo, potrebno je ukucati komandu 'npm install' pre svega za instaliranje modula, nakon čega se može pokrenuti aplikacija koristeći 'ng serve' ili 'npm start'. Pritom se navigiramo na <http://localhost:4200>

Na veb sajtu se možete ili registrovati, ili se ulogovati sa sledećim parametrima:

Za admina -> username: admin, password: admin

Za korisnika -> username: Torrun, password: jackzila

9. Zaključak

Ova dokumentacija i projektni zadatak je predstavljao sistem za online distribuciju igrica. Ceo sistem je opisan bio, kako od strukture sistema, baze podataka, do dijagrama za predstavljanje svih slučajeva korišćenja i funkcionalnosti, do kako se razvio i testirao na kraju. Ova dokumentacija je od velikog značaja, kako za predaju nekome ko želi rekonstruisati možda sličan ili isti sistem, ili da služi kao podsetnik, ili knjiga, za samog inženjera da ima u uvid za čitav proces kako se razvio sistem.