



**Prolećni semestar, 2017/18**

**PREDMET: CS230 - Distribuirani Sistemi**

# **GameDream**

(Projektni zadatak)

**Profesor & Asistent:**

**Vladimir Milićević**

**Autor:**

**Sava Jeremić 2733**

*Beograd, 2018.*

## Sadržaj

Sadržaj.....	1
Uvod:.....	2
1. Funkcionalnosti sistema i alati.....	3
2. Nefunkcionalni zahtevi projektnog zadatka:.....	4
2.1 Adaptivni dizajn.....	4
2.2 Performanse.....	4
2.3 Bezbednost.....	4
4. Baza podataka.....	5
4.1 Konceptualni model baze podataka.....	5
4.2 Fizički model baze podataka.....	6
4.3 Konačni model baze podataka(PHPMyAdmin Designer).....	7
4.4 Povezivanje na bazu podataka.....	7
5. Implementacija i prikaz sistema.....	12
6. Zaključak.....	20

## Uvod:

# GameDream

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

Samo što projekat 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.

GameDream sistem će pružiti korisnicima mogućnost da kupe PC Igre, koje će moći videti sve u listi, u obliku imena igre, slike za igricu, žanra, jezika, kompanije i slično.

Korisnik će moći preko svog naloga da vidi svoje kupljene igre.

## 1. Funkcionalnosti sistema i alati

Front-end veb tehnologije koje će se koristiti za kreiranje sistema, će biti odrađene u alatima kao što su HTML5, CSS3, JavaScript, jQuery okvir, kao i najnovija verzija frameworka Bootstrap 4.7.

Za back-end će se koristiti Java EE EJB zrna sesije, kontroleri za svaki entitet, 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.

Za izradu projekta korišćene su sledeći alati i tehnologije:

- NetBeans 8.2,
- Java EE 7,
- MySQL,
- HTML 5,
- Bootstrap 4.7,
- GlasFish 4,
- PowerDesigner (Za izradu modela baze podataka).

Veb aplikacija će imati dva tipa korisnika: korisnika i administratora. Zbog ovakve podele, funkcionalnosti sistema možemo podeliti na sledeći način:

### Globalni nivo permisija

- Login
- Registraciju
- Pretraga igrica

### Administratorski nivo:

- CRUD korisnika
- CRUD igrica
- CRUD žanrova
- CRUD kompanija
- CRUD jezika

### Korisnički nivo:

- Dodavanje igrica u korpu
- Brisanje igre iz korpe
- Pregled korpe

Globalni nivo permisija uključuje sve korisnike, kao i ne-registrovane ili ulogovane korisnike sistema.

### Ključne reči

Java EE, Web Client, Baza podataka, CRUD (Create Read Update Delete)

## 2. Nefunkcionalni zahtevi projektnog zadatka:

### 2.1 Adaptivni dizajn

Adaptivan dizajn samog sistema, aplikacije, je veoma bitan aspekt prilikom kreiranja bilo kakvih sistema. Kada kažemo adaptivna, to podrazumeva da prilikom dizajniranja sistema, da ga moramo prilagoditi tako da može biti upotrebljiv na uređajima različitih veličina i performansi, 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.

### 2.2 Performanse

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

Treba prilagoditi performanse tako da korisnik ne mora da čeka duže od 1-2 sekunde da se učita sajt, tačnije i najpoželjnije 1019ms. 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

Veb strane trebaju biti vidljive korisnicima, da ne budu blanko, i sajt mora podržati minimum 100 korisnika u istom trenutku.

### 2.3 Bezbednost

Bezbednost, 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.

Ovo sprečavamo time što kreirani entiteti klasa iz baze podataka nam kreiraju @NamedQuery-je, uz pomoću koji pravimo upite za sve CRUD funkcionalnosti sa bazom podataka, i gde u okviru upita kada uzimamo vrednosti na osnovu nekog parametra(u okviru WHERE klauzule ili slično), postavljamo vrednost bindovanjem parametra.

To je primaran i glavni način za sprečavanje i rešavanje SQL injection napada, kojim maliciozni korisnici se mogu opslužiti kako bih naudili sistemu.

```
@NamedQuery(name = "Game.findAll", query = "SELECT g FROM Game g")
, @NamedQuery(name = "Game.findById", query = "SELECT g FROM Game g WHERE g.gameId = :gameId")
, @NamedQuery(name = "Game.findByName", query = "SELECT g FROM Game g WHERE g.name = :name")
, @NamedQuery(name = "Game.findByDescription", query = "SELECT g FROM Game g WHERE g.description = :description")
, @NamedQuery(name = "Game.findByCover", query = "SELECT g FROM Game g WHERE g.cover = :cover")
, @NamedQuery(name = "Game.findByReleaseDate", query = "SELECT g FROM Game g WHERE g.releaseDate = :releaseDate")
, @NamedQuery(name = "Game.findBySize", query = "SELECT g FROM Game g WHERE g.size = :size")
, @NamedQuery(name = "Game.findByPrice", query = "SELECT g FROM Game g WHERE g.price = :price"))
```

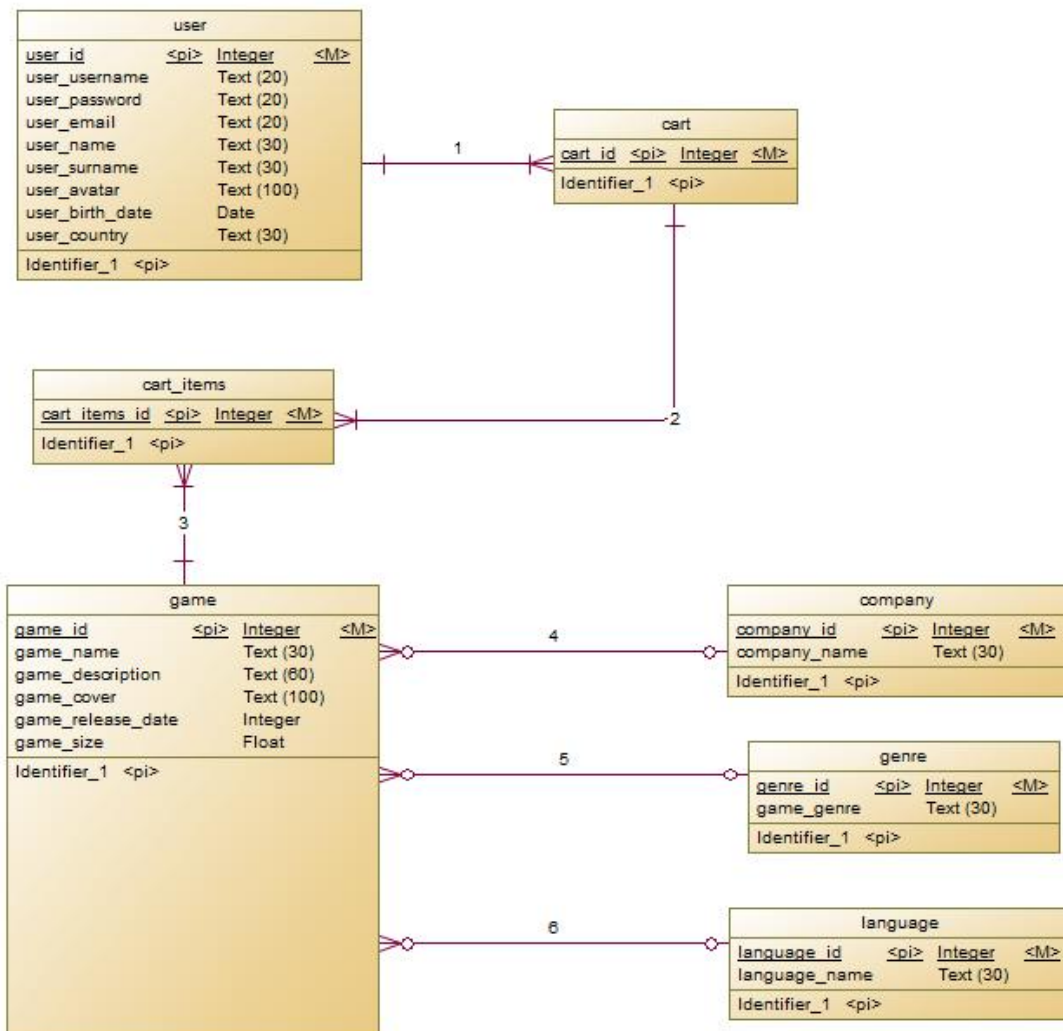
Slika 1 - Ispravan način pisanja upita

Sa slike 1 možemo videti primer pravilnog pisanja upita gde koristimo dvotačku i parametar(npr. gameId = :gameId).

Loša praksa bi bila pisanjem upita i uslova na sledeći način: gameId = 1.

## 4. Baza podataka

### 4.1 Konceptualni model baze podataka



Slika 2 - Konceptualni model baze podataka

Ovako bi izgledao konceptualni model baze podataka sistema *GameDream*, koji je kreiran uz pomoć alata *PowerDesigner*.

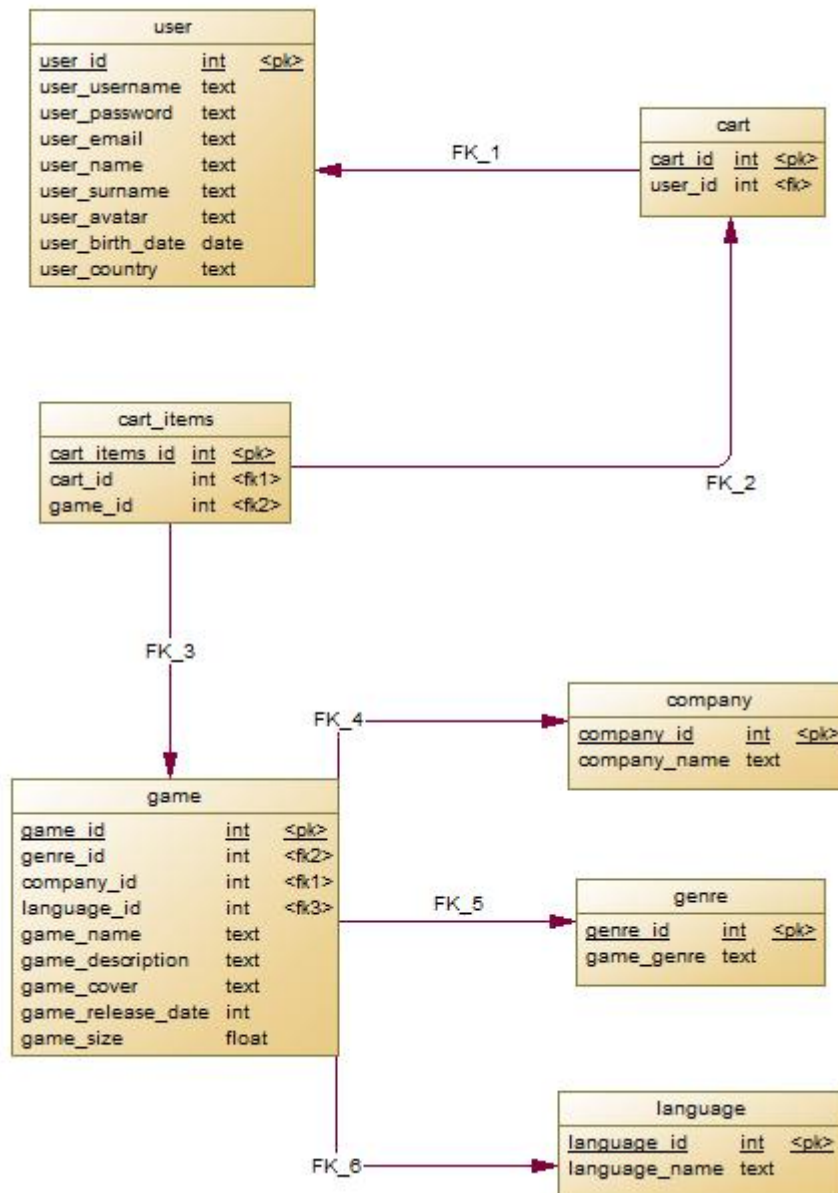
Ovde vidimo dakle tabele u bazi, sa svim njenim relacijama. Prvobitno, imamo entitet "user", koja sadrži attribute kao što su ID, primaran ključ koji jedinstveno označava korisnika u bazi, koji se inkrementuje kao i ostali ID-jevi. Potom imamo username, password i email, kao "not null" attribute, koji će biti kreirani u toku registracije i uneti u bazu, neponovljiv username, kao i email. Ostali atributi entiteta *user*, kao što su *name*, *surname* atributi korisnik koje može dodati prilikom kreiranja naloga.

"*Cart\_Items*" tabela, služi kao korpa, odnosno za smeštanje kupljenih igara korisnika, gde vidimo da je veza više-na-više između *cart*-a i *games*-a odvojena istom (jedan korisnik/cart može kupiti više igara, i igre mogu kupiti više korisnika), gde svaki korisnik ima svoj cart prilikom registrovanja.

Kao najveći entitet, "*game*", ima svoje attribute kao što su naziv, opis, veličina, itd, kao i ostale koje su odvojene radi kardinalnosti i veze više-na-jedan (*Many-To-One*).

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*”.

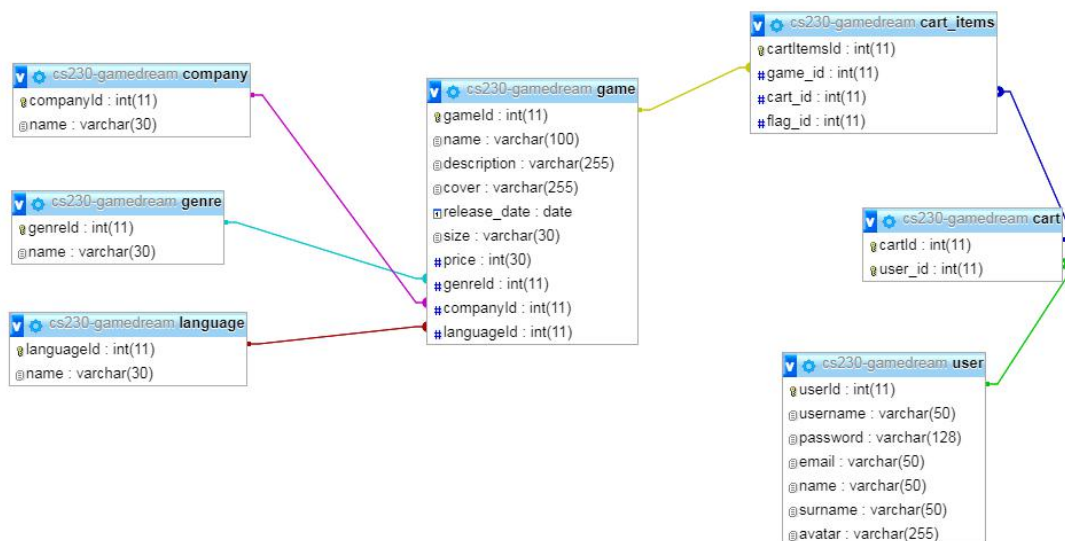
## 4.2 Fizički model baze podataka



Slika 3 - Fizički model baze podataka

Fizički model zapravo predstavlja kako će model izgledati u bazi podataka (phpMyAdmin). Vidimo kako su svi primarni i sekundarni ključevi sada povezani, i u kojoj tabeli. U “*cart\_items*” recimo, sada možemo videti kako ima i ID kupovine, koje igrice korisnik kupuje, koji korisnik.

### 4.3 Konačni model baze podataka(PHPMyAdmin Designer)



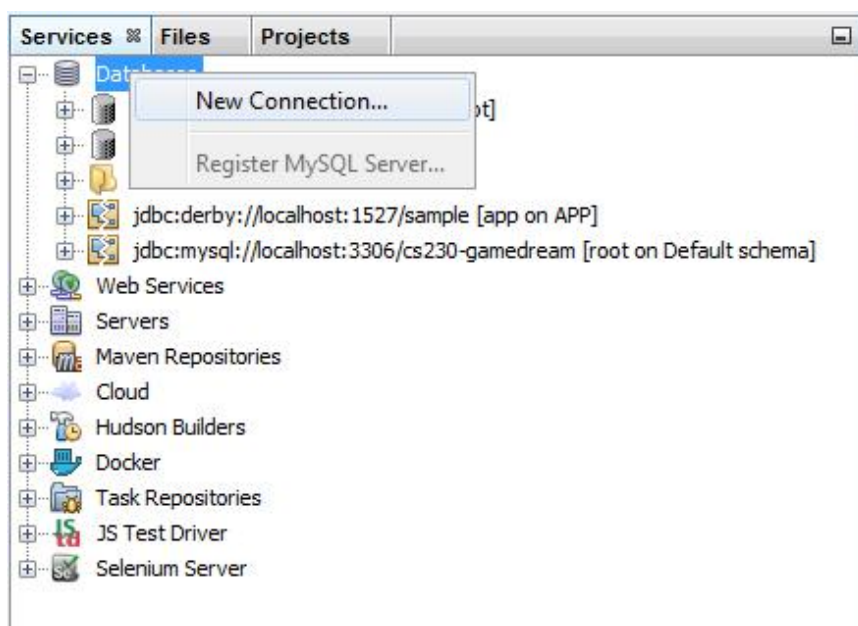
Slika 4 - Konačni model baze podataka

Konačni model baze podataka predstavlja pregled modela baze podataka u okviru PHPMyAdmin-a gde je već importovana baza.

Tabela *cart* sadrži svoj jedinstveni id kao i id korisnika kome ta korpa pripada. *Cart* tabela je povezana sa *user* tabelom, kao i *cartItems*. *CartItems* tabela predstavlja tabelu koja će sadržati sve porudžbine, odnosno sve igre koje korisnik bude stavio u korpu. Tabela se sastoji iz id-ja igrice, id-ja korpe u kojoj će se nalaziti igrice.

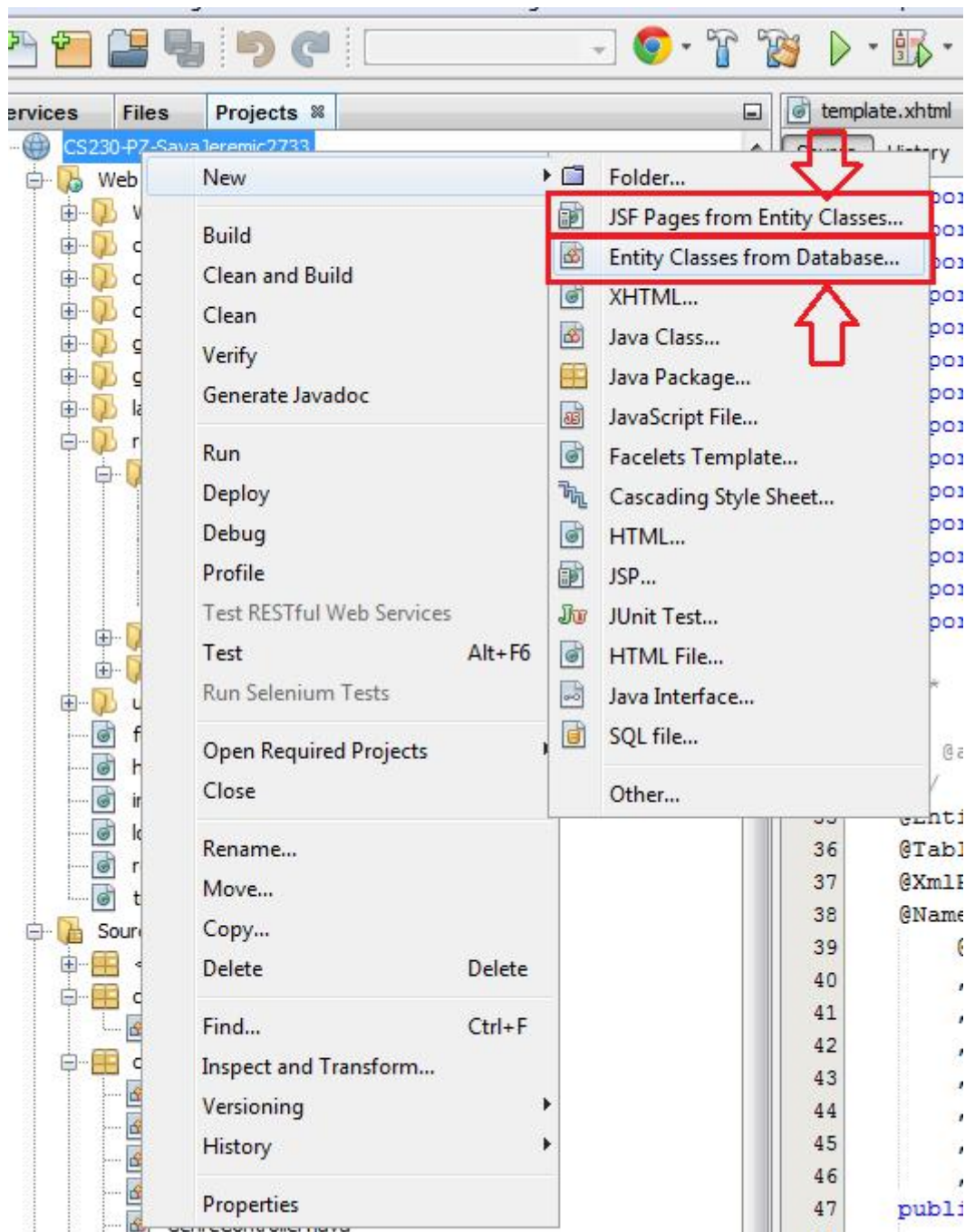
### 4.4 Povezivanje na bazu podataka

Za konektovanje projekta na bazu podataka se koristi Connection Wizard za pravljenje nove konekcije sa bazom podataka, koju ćemo definisati imenom baze podataka, i kao i obezbediti povezivanje konekcije sa veb projektom.



Slika 5 - Pravljenje nove konekcije na bazu podataka

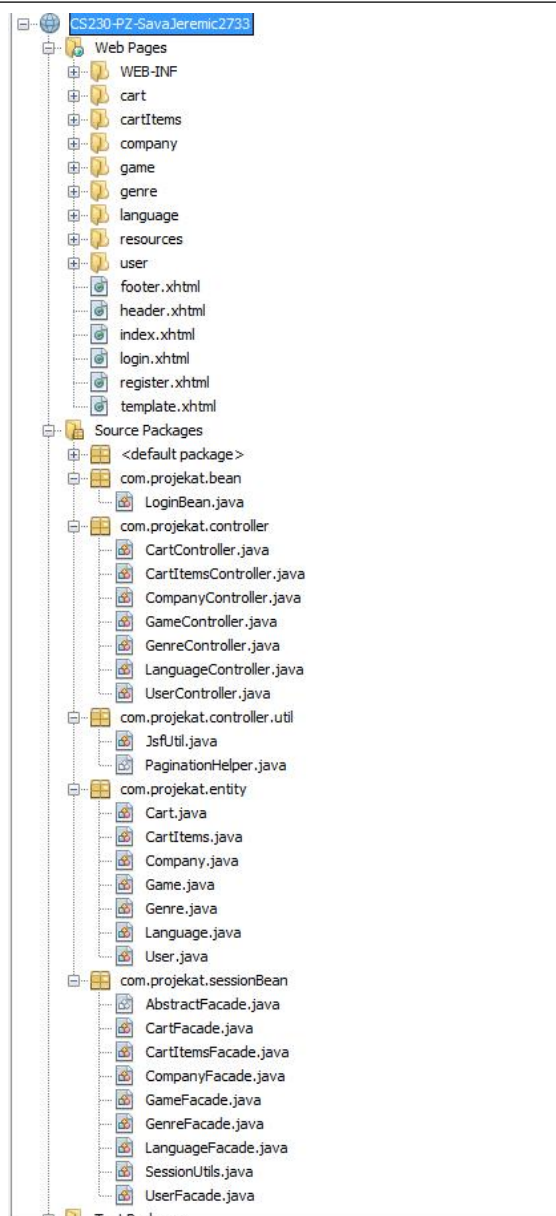




Slika 6 - Generisanje entiteta iz baze podataka - JSF strana iz entiteta

Pre svega kada napravimo nov projekat, generišemo entitete na osnovu svih tabela iz baze podataka.

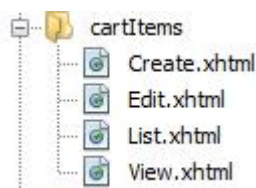
Nakon toga generišemo JSF stranice, kao i zrna sesije i kontrolere za svaki entitet.



Slika 7 - Prikaz hijerarhije generisanih fajlova

Prilikom generisanja fajlova, za kontrolere, zrna, jsf strane, za svaki posebno smo obezbedili konkretno definisan folder.

U *com.projekat.controller* paketu se nalaze kontroleri entiteta, u *com.projekat.entity* paketu se nalaze entiteti klasa, dok u *com.projekat.sessionBean* paketu nalaze fasade.



Slika 8 - Prikaz generisanih JSF strana - CRUD

Ovako izgleda generisan folder, odnosno stranice od datog entiteta. Postoji *Create* za dodavanje, *Edit* za ažuriranje, *List* za prikaz svih itema, kao i *View* za konkretan item sa prosleđenim konkretnim ključem.

```

102 public String login() throws IOException {
103     User user = null;
104     boolean validator;
105
106     try {
107         user = (User) em.createNamedQuery("User.findByUsername")
108             .setParameter("username", username)
109             .getSingleResult();
110         validator = validateUser(user);
111         getAllOrders();
112     } catch (NoResultException e) {
113         validator = false;
114     }
115
116     if (validator) {
117         HttpSession session = SessionUtils.getSession();
118         session.setAttribute("data", user);
119         int id = Integer.parseInt(session.getAttribute("data").toString());
120         System.out.println("SESSION: " + id);
121         loggedIn = true;
122         if (username.equalsIgnoreCase("admin")) {
123             admin = true;
124         }
125         Cart cart = new Cart(userId);
126         return "/index.xhtml";
127     } else {
128         FacesMessage fm = new FacesMessage(FacesMessage.SEVERITY_ERROR, "Input values are incorrect.", "Please try again!");
129         FacesContext.getCurrentInstance().addMessage(null, fm);
130         return "index";
131     }
132 }
133
134 public List<CartItem> getAllOrders() {
135     ArrayList<CartItem> list;
136     list = new ArrayList<>(em.createNamedQuery("CartItem.findByCartId", CartItem.class).setParameter("cartId", ajDi()).getResultList());
137     System.out.println("list " + list);
138     return list;
139 }
140
141 private boolean validateUser(User user) {
142     if (username.equalsIgnoreCase(user.getUsername()) && password.equals(user.getPassword())) {
143         return true;
144     }
145     return false;
146 }
147
148 public String logout() {
149     HttpSession session = SessionUtils.getSession();

```

Slika 9 - Prikaz login zrna

LoginBean predstavlja ustvari zrno, koji koristi EntityManager, za menadžment i pokretanje upita, uz pomoću kojeg obezbeđujemo funkcije kao što su

login() - za logovanje korisnika,

getAllOrders() - vraća sve korpa iteme iz korpe za određenog korisnika,

validateUser() - validacija unosa,

logout() - logout korisnika

```
@Entity
@Table(name = "game")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Game.findAll", query = "SELECT g FROM Game g")
    , @NamedQuery(name = "Game.findById", query = "SELECT g FROM Game g WHERE g.gameId = :gameId")
    , @NamedQuery(name = "Game.findByName", query = "SELECT g FROM Game g WHERE g.name = :name")
    , @NamedQuery(name = "Game.findByDescription", query = "SELECT g FROM Game g WHERE g.description = :description")
    , @NamedQuery(name = "Game.findByCover", query = "SELECT g FROM Game g WHERE g.cover = :cover")
    , @NamedQuery(name = "Game.findByReleaseDate", query = "SELECT g FROM Game g WHERE g.releaseDate = :releaseDate")
    , @NamedQuery(name = "Game.findSize", query = "SELECT g FROM Game g WHERE g.size = :size")
    , @NamedQuery(name = "Game.findPrice", query = "SELECT g FROM Game g WHERE g.price = :price"))
public class Game implements Serializable {

    @OneToMany(cascade = CascadeType.ALL, mappedBy = "gameId")
    private Collection<CartItems> cartItemsCollection;

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "gameId")
    private Integer gameId;
    @Size(max = 100)
    @Column(name = "name")
    private String name;
    @Size(max = 255)
    @Column(name = "description")
    private String description;
    @Size(max = 255)
    @Column(name = "cover")
    private String cover;
    @Column(name = "release_date")
    @Temporal(TemporalType.DATE)
    private Date releaseDate;
    @Size(max = 30)
    @Column(name = "size")
    private String size;
    @Basic(optional = false)
    @NotNull
    @Column(name = "price")
    private int price;
    @JoinColumn(name = "genreId", referencedColumnName = "genreId")
    @ManyToOne(optional = false)
    private Genre genreId;
    @JoinColumn(name = "companyId", referencedColumnName = "companyId")
    @ManyToOne(optional = false)
    private Company companyId;
    @JoinColumn(name = "releaseDate", referencedColumnName = "releaseDate")
    private Date releaseDate;
```

Slika 10 - Prikaz klase entitet - Game

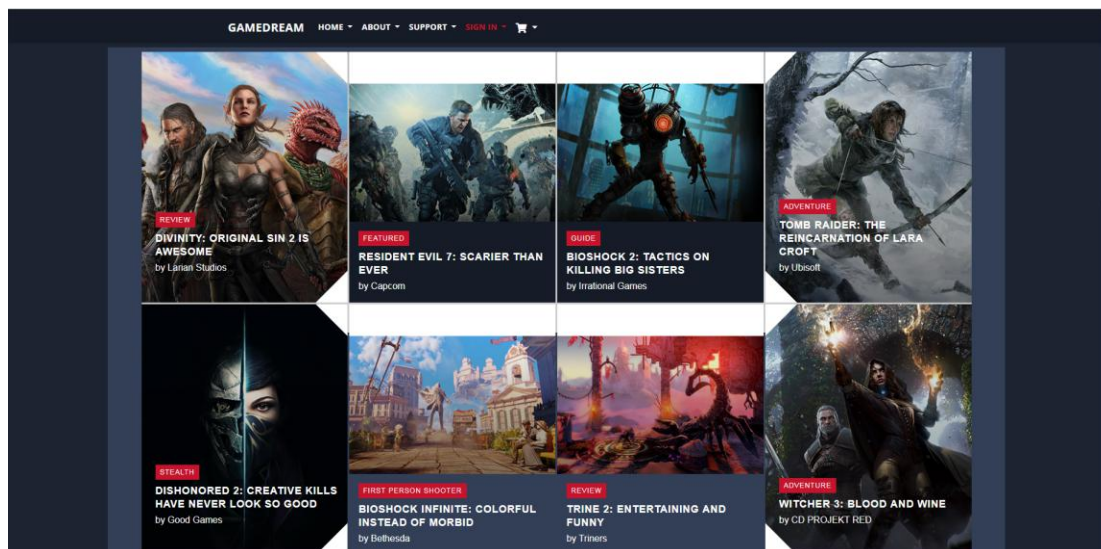
Na slici 10 možemo videti Game entitet, koji je obeležen anotacijom `@Entity`, definisan koju tabelu predstavlja anotacijom `@Table`, u sadrži više anotacija `@NamedQuery`, koji predstavljaju upite koje će druga crna moći pozivati, kao što smo uradili na slici 9 sa menadžerom entiteta.

Osnovne parametre, odnosno attribute entiteta definišemo anotacijama `@Column`, obeležavamo da li predstavlja ID anotacijom `@Id`, dok tip tabele postavljamo u zavisnosti od zavisnosti atributa i njihovih veza sa drugim tabelama, kao što imamo običan atribut name, on je obeležen anotacijom `@Basic`, dok je genreId obeležen anotacijom `@JoinColumn`, kojom spajamo tabele, odnosno FK(Foreign Key - strani ključ) sa primarnim ključem tabele, i postavljamo kompleksnost veze anotacijom `@ManyToOne`.



## 5. Implementacija i prikaz sistema

U ovom delu će biti prikazane stranice veb sajta i opisan način implementacije.



Slika 11 - Početna strana

Ovako izgleda početna strana aplikacije. Na home strani su postavljene neke igre stilizovane Bootstrapom. U okviru menija postoji Home strana, Dugme za Sign in, Korpa. Ovaj meni, odnosno header, se nalazi na svim stranama bez obzira gde se korisnik navigira, što je obezbeđeno uz pomoću JSF templejta.

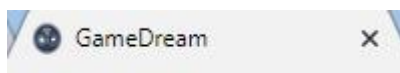
Zato što se ovaj header svuda prikazuje, moramo ograničiti i definisati šta će koji korisnik videti u zavisnosti koji korisnik je logovan, ili ne ulogovan.

Ukoliko korisnik nije ulogovan, korpa će prikazati korisniku dva dugmeta za login i registraciju, ukoliko jeste ulogovan prikazaće samu korpu.

Na mesto *Sign In* opadajućeg menija, ne-ulogovanom korisniku se prikazuju dugmad za login i registraciju, dok ukoliko je ulogovan korisnik mu se prikazuje profil, u okviru kojeg može kupovati igre i izlistavati kupljene igre.

Administrator će imati ekskluzivan opadajući meni za sebe, u okviru kojeg će moći izvršavati sve svoje administratorske CRUD funkcije nad svim tabelama.

Sve što je navedeno iznad, biće prikazano u sledećim slikama.



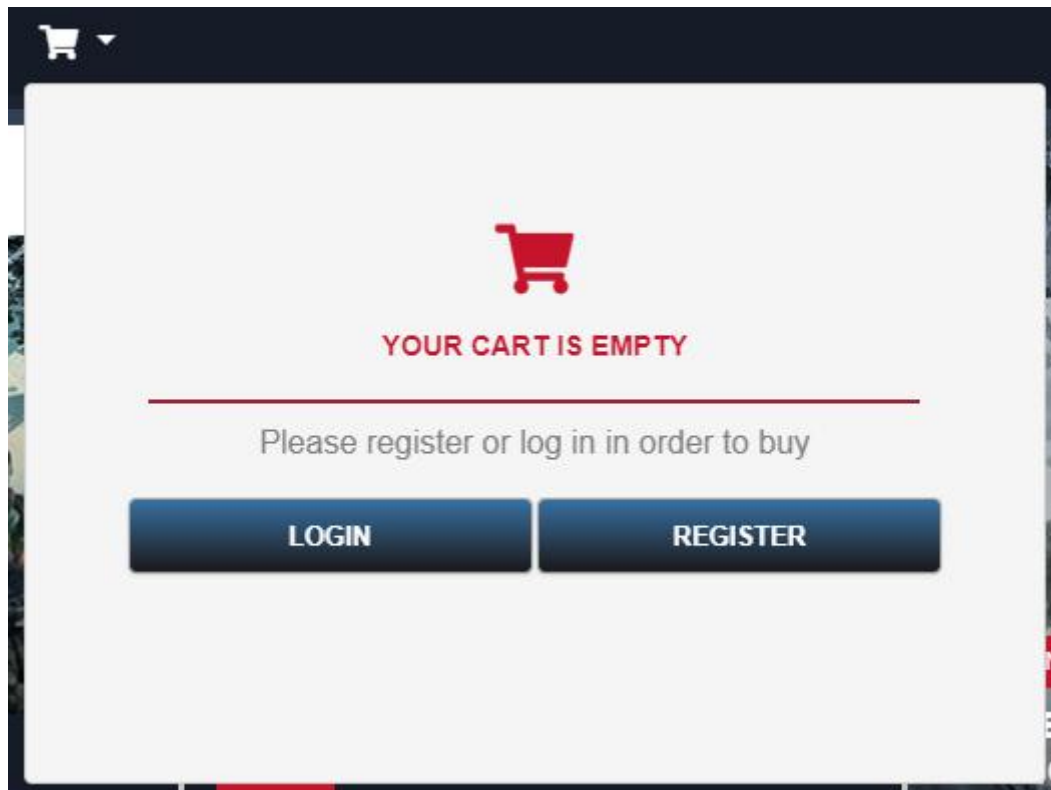
Slika 12 - Ikona veb sajta

Ovako izgleda ikona aplikacije.



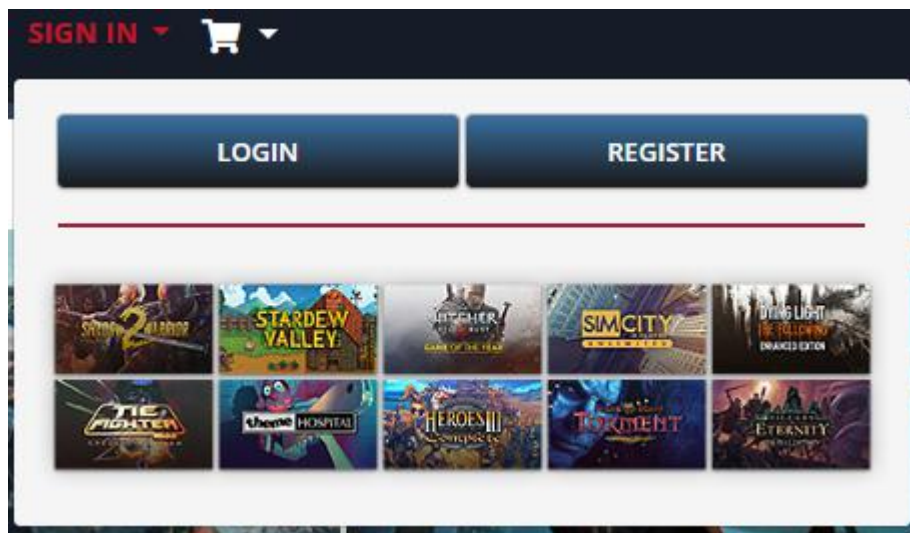
Slika 13 - Home dropdown meni

Kada kliknemo na Home dropdown padajući meni, prikaže se tekst za prikaz svih igara na klik *SHOW ALL GAME ITEMS*, dok se ispod linije izlistavaju svi žanrovi iz baze podataka.



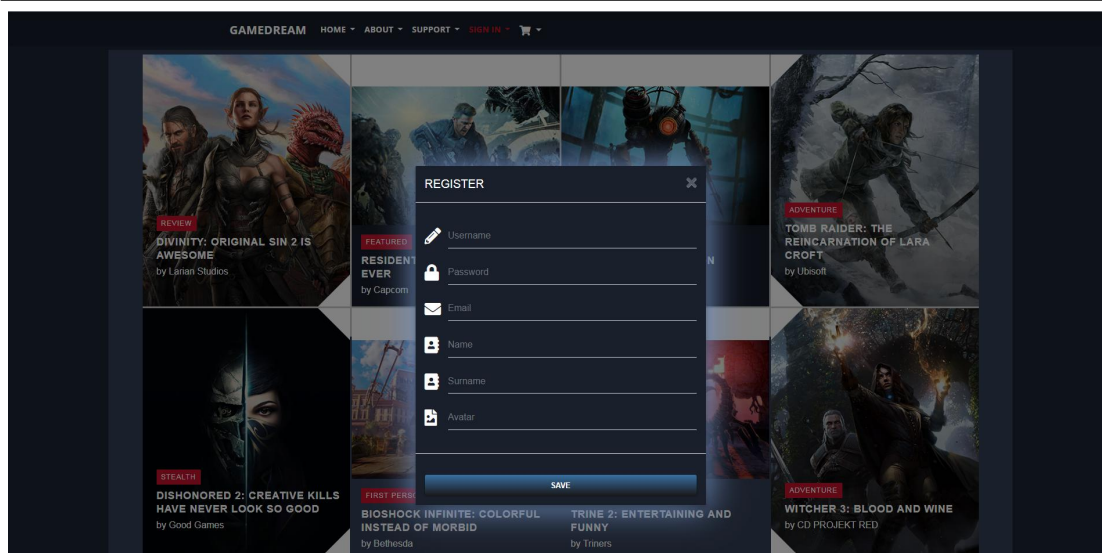
Slika 14 - Prikaz opadajućeg menija - Cart - Neulogovan korisnik

Korisnik koji nije ulogovan vidi korpu opadajući meni kao što je prikazano na slici 14. Nudi mu se login dugme, koje izbacuje login modal, ukoliko ima već nalog, ukoliko nema nalog, može odabrati dugme za registraciju.



Slika 15 - Prikaz menija - Sign In - Neulogovan korisnik

Prikaz Sign In opadajućeg menija za korisnike koji nisu ulogovani. Kao kod korpe, nude mu se dugmad za login i registraciju iz iste namere.

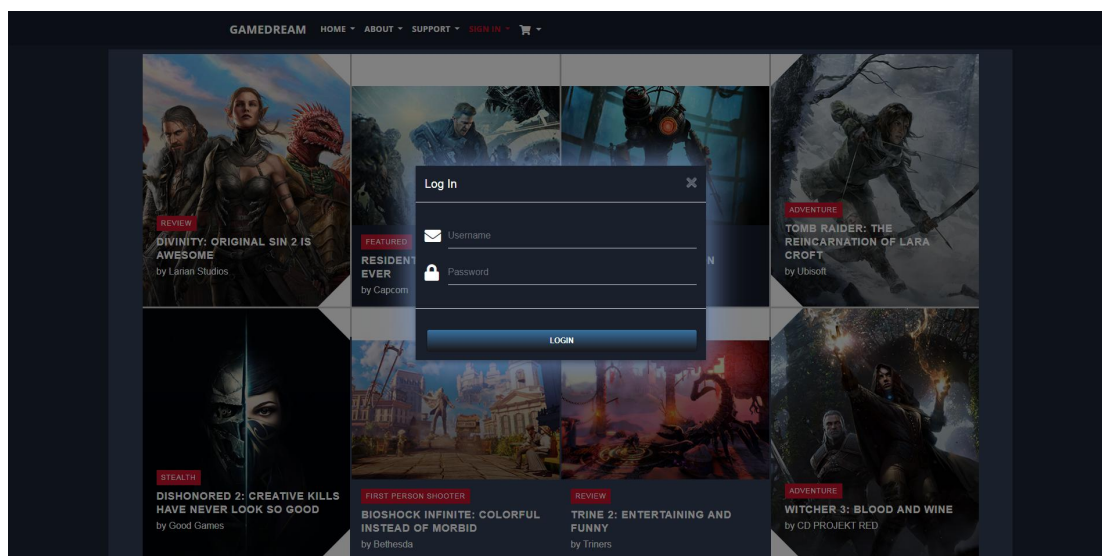


Slika 16 - Prikaz register modala

Kada korisnik pritisne na dugme *Register*, prikazaće mu se forma za registraciju u obliku modala, gde unosi parametre *Username*, *Password*, *Email*, *Name*, *Surname*, *Avatar*, odnosno korisničko ime, šifru, elektronsku poštu, ime i prezime, kao i sliku.

Korisnik mora popuniti dobro podatke, kao što je na primer email u obliku kombinacije karaktera i simbola kakav standardni format za email treba da bude.

Ukoliko korisnik ne unese podatke kako treba, prikazaće mu se greška da ispravi pogrešno unete podatke, ukoliko unese kako treba isto tako mu će se prikazati poruka da se uspešno korisnik registrovao na sistem.

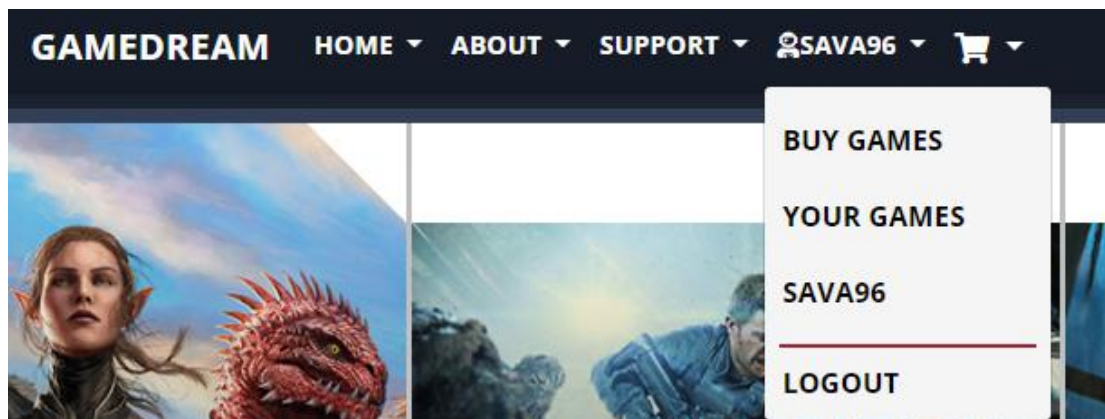


Slika 17 - Prikaz login modala

Kada korisnik pritisne na dugme *Login*, prikazaće mu se forma za login u obliku modala, gde unosi parametre *Username* i *Password*, odnosno korisničko ime i šifru.

Kao i kod forme za registraciju, login forma će izbaciti poruke u zavisnosti od unetih parametara.





Slika 18 - Prikaz opadajućeg menija - Korisnik - Ulogovan korisnik

Kao što možemo videti na slici 18, sada kada se korisnik sava96 uspešno ulogovao na sistem, promeniće se meni sistema.

Sada, namesto *Sign In* dugmeta sa slike 15, se zamenjuje opadajućim menijem korisnika.

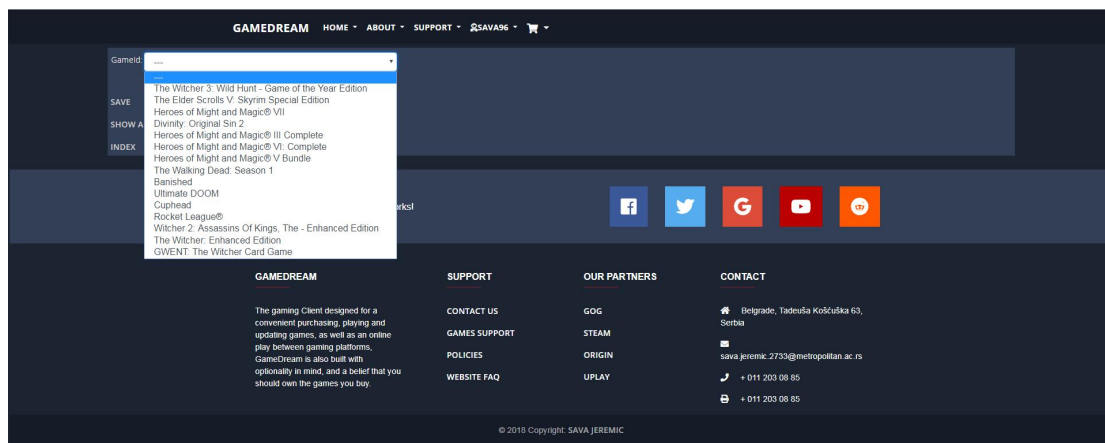
U njemu korisnik može otići na stranu gde će kupiti igrice, prelistati svoje igrice, i dugme za logout.

GAMEDREAM HOME ABOUT SUPPORT SAVA96									
1 1015 NEXT 10									
Name	Description	Cover	ReleaseDate	Size	Price	Genre	Company	Language	
The Witcher 3: Wild Hunt - Game of the Year Edition	Become a professional monster slayer and embark on an adventure of epic proportions! Upon its release, The Witcher 3: Wild Hunt		08/29/2017	40.0 GB	44	RPG	CD Projekt Red	English	<a href="#">VIEW</a>
The Elder Scrolls V: Skyrim Special Edition	Winner of more than 200 Game of the Year Awards, Skyrim Special Edition brings the epic fantasy to life in stunning detail. The		10/27/2016	12.0 GB	27	RPG	Bethesda Game Studios	English	<a href="#">VIEW</a>
Heroes of Might and Magic VII	The realms of Might and Magic are expanding. New lands have been discovered and you must rise to the challenge of conquering the		05/05/1996	353.6 MB	4	Strategy	Ubisoft	English	<a href="#">VIEW</a>
Divinity: Original Sin 2	The eagerly anticipated sequel to the award-winning RPG. Gather your party. Master deep		09/16/2017	356 MB	4	Strategy	Ubisoft	English	<a href="#">VIEW</a>

Slika 19 - Prikaz igrica

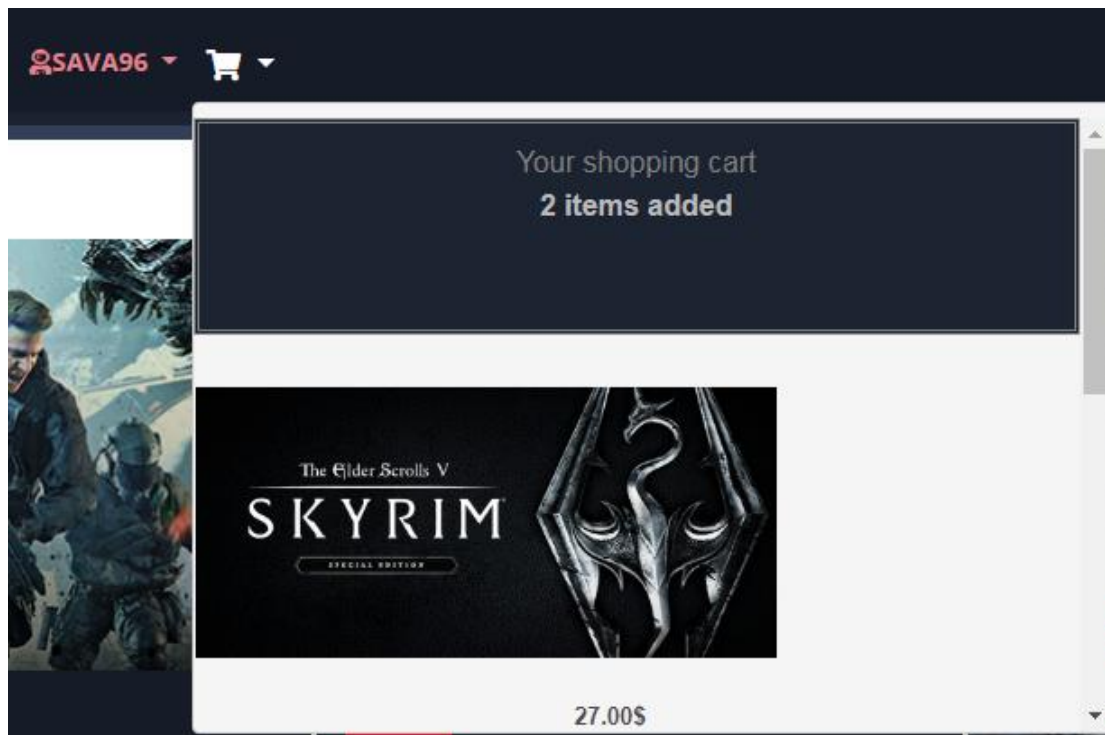
Možemo videti izlistavanje svih igrica kada je korisnik ulogovan, što je isto kao i kad korisnik nije ulogovan, dostupno mu je samo dugme *View* koje ga navodi na stranu gde se samo odabrana igrice prikazuju.

Svi parametri entiteta *Game* se prikazuju u ovom prikazu, ne uključujući ID naravno.



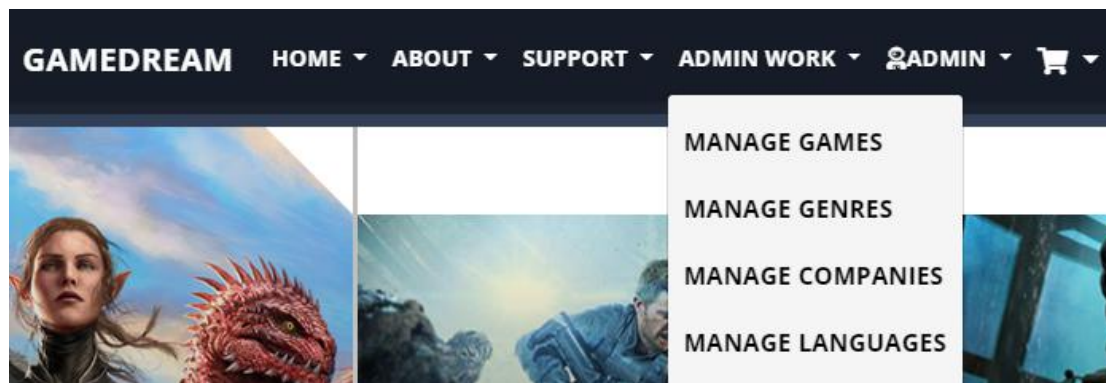
Slika 20 - Strana za kupovinu igrice

Korisnik može na ovoj strani se navigirati u okviru svog profila klikom na dugme *Buy Games*. Ovde korisnik bira ime igrice koju hoće da kupi i klikne na *save*.



Slika 21 - Prikaz opadajućeg menija - Cart - Ulogovan korisnik

Sada kada je korisnik popunio svoju korpu, može je videti u svakom trenutku u okviru opadajućeg menija za korpu. Vidi se broj igrice koji se nalazi u korpi, kao i sve igrice koje može izlistati scroll barom.



Slika 22 - Početna strana - meni administratora

Sada ćemo prikazati stranice kako ih administrator sistema vidi.

Možemo videti posebno namenjen opadajući meni za administratora, u okviru kojeg može upravljati svim funkcionalnostima CRUDA nad bazom podataka. Može korisiti CRUD za igrice, žanrove, kompanije i jezike.

Name	Description	Cover	ReleaseDate	Size	Price	Genre	Company	Language	
The Witcher 3: Wild Hunt - Game of the Year Edition	Become a professional monster slayer and embark on an adventure of epic proportions! Upon its release, The Witcher 3: Wild Hunt		08/29/2017	40.0 GB	44	RPG	CD Projekt Red	English	VIEW EDIT DESTROY
The Elder Scrolls V: Skyrim Special Edition	Winner of more than 200 Game of the Year Awards, Skyrim Special Edition brings the epic fantasy to life in stunning detail. The		10/27/2016	12.0 GB	27	RPG	Bethesda Game Studios	English	VIEW EDIT DESTROY
Heroes of Might and Magic® VII	The realms of Might and Magic are expanding. New lands have been discovered and you must rise to the challenge of conquering the		05/05/1996	353.6 MB	4	Strategy	Ubisoft	English	VIEW EDIT DESTROY
Dumky Original Sin	The eagerly anticipated sequel to the award-winning D&D. Gather your party		09/16/2017	356 MB	4	Strategy	Ubisoft	English	VIEW

Slika 23 - Prikaz igrica administratoru

U prikazu igrica administratoru se postavlja dva dodatna dugmeta, *Edit* i *Destroy*, gde je *Edit* za ažuriranje igrice, recimo cene igrice ili slično, i gde je *Destroy* za brisanje igrice.

Userld	Username	Password	Email	Name	Surname	Avatar	Cart	
1	sava96	sava96	sava@gmail.com	Sava	Jeremic	avatar	1	VIEW EDIT DESTROY
2	admin	admin	admin@admin.com	admin	admin	admin	2	VIEW EDIT DESTROY
3	test	test	test@test.com	test	test	test		VIEW EDIT DESTROY

CREATE NEW USER

INDEX

Slika 24 - Prikaz korisnika administratoru

CRUD nad korisnicima.

---

```
<c:if test="#{loginBean.loggedIn eq true}">  
  <c:if test="#{loginBean.admin eq false}">
```

*Slika 25 - if za ograničavanje prikaza*

Na slici 25 možemo videti način na koji se menjaju privilegije, odnosno prikazi elemenata sistema. Ovaj `<c:if>` postavljamo između elemenata koje želimo da ograničimo. Iz `loginBean` klase uzimamo boolean parametre, od kojih je jedan za korisnika, dok je drugi za administratora i postavljamo uslov ukoliko jeste korisnik - prikaži jedno, ukoliko je administrator - prikaži drugo.

## 6. Zaključak

Kroz ovu dokumentaciju se prošlo većina segmenata veb aplikacije. Mnogo funkcija je bilo izostavljeno za prikaz, što je zbog obimnosti za dokumentaciju samu, ali su prikazane najbitnije, kao i one koje se ponavljaju uz malo modifikacije. Prikazana je aplikacija, njene funkcionalnosti, konceptualni, fizički kao i finalni model baze podataka.

U toku izrade ovog projekta sam mnogo više naučio o Java EE, JSF i Enterprise Bean tehnologijama, proširio svoje znanje Java programskog jezika, da proširim još više svoje znanje pridodato sa predavanja i vežbi. Rukovanje sesijom i pamćenje parametara između strana kao i njihovo prosleđivanje su bili izazov, kao i Join tabela, dodavanje parametra koji sadrže strane ključeve, kao i prikaz parametra iz druge tabele na osnovu jedne na osnovu njegovog stranog ključa. Snalaženje oko upita, kao i pravljenje funkcija za login, baze i zrna, njihova sinhronizacija, su bili prevaziđen izazov.

Ovaj projekat je još jedan ispunjen cilj, a to je proširivanje svog znanja korišćenju web tehnologija i Jave, kao i stilizovanju elemenata uz pomoć Bootstrap okvira, korišćenjem i praćenjem najnovijih verzija.