



Univerzitet u Sarajevu
Elektrotehnički fakultet u Sarajevu
Odsjek za računarstvo i informatiku



Objektno-orjentirana analiza i dizajn

Projekat: GameHub

Refactoring izvještaj

Grupa 4 – Oštri Momci (#momci):

Elvedin Smajić

Vedad Grbo

Tarik Beganović

Promjene funkcionalnosti tokom implementacije

Pošto nismo imali puno vremena za implementaciju projekta zbog drugih obaveza i manjka vremena, nismo sve planirane funkcionalnosti sistema implementirali. Implementirali smo osnovnu funkcionalnost aplikacije, a to je: kreiranje accounta, mogućnost pregleda igrica, ocjena i komentara, kao i unos ocjene i komentara.

Kreiranje početnog projekta

U Visual Studiu smo kreirali ASP.NET aplikaciju sa MVC dizajnom. Osnovni izgled aplikacije je kreiran automatski od strane okruženja. Nakon toga smo ubacili temu u **bootstrap**, te smo u View HTML fajlovima promijenili boje elemenata aplikacije poput headera i footera. Pošto nismo mogli promijeniti boju direktno u bootstrap.css fajlu, u HTML fajlu smo promijenili klasu izgleda iz bootstrapa:

Prije:

```
<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-primary border-bottom box-shadow mb-3">
```

Poslije:

```
<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-danger border-bottom box-shadow mb-3">
```

Na ovaj način smo mijenjali stil ostalih elemenata aplikacije. Pored toga dodali smo logo u navbar i promijenili početni izgled homepagea.

Dodan logo u class navbar:

```
<a asp-area="" asp-controller="Home" asp-action="Index"><img src='https://i.postimg.cc/xXxcP0ZF/LogoV1.png' border='0' width=50px height=50px alt='LogoV1' /></a>
```

Promjena homepage-a (Home/Index.cshtml):

```
<div class="text-center">
  <h1 class="display-4">Dobrodošli na GameHub!</h1>
  <h3>Pravo mjesto za najveće gaming entuzijaste</h3>
  <form width="200">
    <p>
      <a asp-controller="Igrica" asp-action="Top10"><input type="button"
value="Top 10 lista" class = "guestButton"/></a>
    </p>
  </form>
  <a><img src='https://i.postimg.cc/FFYjN7L5/background.jpg' border='0'
alt='background' /></a>
</div>
```

Dodali smo u privacy page pravila korištenja aplikacije:

```
@{
  ViewData["Title"] = "Pravila korištenja";
}
<h1>@ViewData["Title"]</h1>

<p><b>Svi korisnici aplikacije GameHub se moraju pridržavati sljedećih
pravila:</b></p>
<p>1. Korisničko ime (username) mora biti prikladno.</p>
<p>2. Objave i komentari korisnika ne smiju imati uvredljiv sadržaj na bilo kojoj
osnovi.</p>
<p>3. Budite pristojni prema drugim korisnicima aplikacije. Više neprikladnih
poruka, kao i spam će biti sankcionisani.</p>
<p>4. Slikovne objave se moraju ticati video igrice. To mogu biti screenshotovi iz
igrice, kao i slike iz stvarnog
života koje se tiču igrice (npr. slike gaming mercha i slično).
</p>
<p>5. Ako nikada niste čuli za <a
href="https://www.youtube.com/watch?v=S008DnW2GeM">pravilo lopate</a>, molimo vas
da napustite aplikaciju.</p>
<p><br /></p>
<p><br /></p>
<p><br /></p>
<p><b><i>Vaš Admin team.</i></b></p>
```

Ubacivanje klasa modela, scaffolding i migracija

Dodali smo model klase:

- Igrica
- KomentarIgrica
- KomentarObjava
- Objava
- Zanr (enum).

Objava i KomentarObjava nismo stigli iskoristiti jer nismo stigli implementirati sistem objavljivanja na profil.

Ukinuli smo nasljeđivanje iz apstraktne klase Komentar u KomentarIgrica i KomentarObjava jer nam je tako bilo lakše.

Izvršili smo prvi scaffold i migraciju, te su se navedene klase dodale u bazu. Nakon toga je bilo izmjena u modelu Igrica gdje smo dodali Sliku kao atribut string koji čuva path od slike covera igrice. Isto tako, dodali smo u KomentarIgrica ID korisnika kao atribut kasnije, nakon što smo riješili na koji način ćemo implementirati sistem korisnika. Nakon promjena su izvršene promjene u bazi i kontrolerima.

Promjena izgleda pogleda putem HTML i CSS

Defaultni izgled pogleda je promijenjen putem HTML i CSS fajlova. Kod prikaza liste igrice su uklonjeni nepotrebni atributi, kao i oni koji se ne smiju vidjeti (npr. id). Dodane su slike za prikaz covera igrice, povećani su fontovi naziva i srednje ocjene, te je dodana ikonica zvjezdice pored broja ocjene.

```
@foreach (var item in Model) {  
    <tr>  
        <td>  
              
        </td>  
        <td>  
            <font size="5">@Html.DisplayFor(modelItem => item.Naziv)</font>  
        </td>  
        <td>  
              
            <font size="5">@Html.DisplayFor(modelItem =>  
item.SrednjaOcjena)</font>  
        </td>  
        <td>  
            @Html.DisplayFor(modelItem => item.Autor)  
        </td>  
        <td>  
            @Html.DisplayFor(modelItem => item.Zanr)  
        </td>  
        <td>  
            @Html.DisplayFor(modelItem => item.RRated)  
        </td>  
        <td>  
            @if(SignInManager.IsSignedIn(User)) {  
                <a asp-controller="KomentarIgrica" asp-action="Create" asp-route-  
id="@item.Id">Komentiraj</a>  
            }  
            <a asp-action="Details" asp-route-id="@item.Id">Details</a>  
        </td>  
    </tr>  
}
```

Pogledi Details su isto promjenjeni (odnosi se na prikaz profila igrice i korisnika). Prije svega u HTML-u su dodane dvije div-klase: detail-container i detail-child:

```
<div class="detail-container">
  <div class="detail-child">
    <dl class="cover">
      <dt>
        
      </dt>
    </dl>
  </div>
  <div class="detail-child">
    <dl class="row">
      <dt class="col-sm-2">
        <font size="4">@Html.DisplayNameFor(model =>
model.Naziv):</font>
        ... # ostali detalji igrice
      </dt>
    </dl>
  </div>
</div>
```

Pozicija detail-childova je podešena putem CSS-a (u site.css) tako da stoje jedan pored drugog. Prvi child čini slika (nalazi se lijevo), a drugi čine podaci (nalazi se desno).

```
.detail-container {
  padding: 10px;
}

.detail-child {
  width: 40%;
  float: left;
  padding: 10px;
}
```

Kontroleri

Većina kontrolerskih funkcija je automatski bilo generisano. Za određene kontrolere smo dodali vlastite funkcije koje su nam bile potrebne:

Top 10 lista sa compare funkcijom (IgricaController):

```
public async Task<IActionResult> Top10()
{
    var igrice = await _context.Igrica.ToListAsync();
    igrice.Sort(compareIgrice);

    igrice = igrice.GetRange(0, 10);

    return View(igrice);
}

public static int compareIgrice(Igrica ig1, Igrica ig2)
{
    return ig2.SrednjaOcjena.CompareTo(ig1.SrednjaOcjena);
}
```

Komentiranje (IgricaController):

```
// GET: Igrica/Komentiraj/5
public IActionResult Komentiraj(int? id)
{
    var model = new KomentarIgrica();
    model.IgricaId = (int) id;
    return View(model);
}

[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Komentiraj([Bind("Id,Tekst,Ocjena,IgricaId")]
KomentarIgrica komentarIgrica)
{
    if (ModelState.IsValid)
    {
        _context.KomentarIgrica.Add(komentarIgrica);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(komentarIgrica);
}
```

Pogledi komentara igrice (KomentarIgricaController):

```
public async Task<IActionResult> KomentarVecPostoji()
{
    return View(await _context.KomentarIgrica.ToListAsync());
}

public async Task<IActionResult> KomentariKorisnika(string id)
{
    var igrica = await _context.KomentarIgrica.ToListAsync();
    igrica.RemoveAll(igrice => igrice.KorisnikId!=id);

    return View(igrica);
}
```

Otvaranje profila (HomeController):

```
public IActionResult MojProfil()
{
    return View();
}
```


Problemi sa uploadom slike

Za upload igrice smo omogućili upload covera u obliku slike. Međutim, skripta koja pretvara sliku u BASE64 i posprema u bazu radi samo na posebno dugme „SPREMI SLIKU“ koje se mora kliknut prije „SUBMIT“. To bi se dalo ispraviti nekim refactoringom, ali mi nismo znali kako to da uradimo, a da upload radi.

Ipak, nismo jedini koji imaju bespotrebnih viška komandi ;) #windows11.

