

Relatório DIAM

Games HUB

Descrição do tema do site:

É um site onde se pode consultar e comentar sobre jogos. A comunidade pode ver as páginas dos jogos e comentar sobre as mesmas, atribuindo uma classificação ao jogo de uma a cinco estrelas. Todos os utilizadores podem ver os comentários que a comunidade fez sobre um determinado jogo, e ver as respetivas classificações.

Principais funcionalidades:

Temos dois tipos de utilizadores. Membros da comunidade que podem ver, comentar, gostar, e adicionar aos favoritos os jogos; Os administradores (superuser) podem adicionar jogos ao site, remover um jogo do site, eliminar comentários, e fazer tudo o que os membros da comunidade fazem; Quem não está registado no site pode visualizar a página de detalhe de cada jogo mas não pode comentar nem gostar ou adicionar aos favoritos.

Gerais

Qualquer utilizador (que não esteja registado) pode visualizar o conteúdo da página de detalhe, mas não pode comentar ou gostar ou adicionar os jogos aos favoritos. Sendo que se tentar alguma destas ações é reencaminhado para a página de login.

Membros da comunidade (registados no site)

Os Membros da comunidade podem visualizar, gostar, adicionar aos favoritos, e comentar sobre o jogo selecionado atribuindo-lhe um rating de 1-5 estrelas. Para manter a integridade do site o comentário tem de ter obrigatoriamente associado o comentário e o rating atribuído.

Administradores do site

Têm todos os privilégios dos utilizadores referidos acima, podem adicionar jogos ao site, remover os mesmos e têm o poder de eliminar comentários.

Modelos de dados desenvolvidos e suas relações

```
class Image(models.Model):
    image = models.ImageField(null=True)
    name = models.CharField(null=True, max_length=200)
    description = models.CharField(max_length=20000)
    likes = models.PositiveIntegerField(default=0)
    views = models.PositiveIntegerField(default=0)
    bookmark = models.BooleanField(default=False)
    pub_date = models.DateTimeField('Pub_Date', null=True)

class Rats(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    perfil_image = models.ImageField(null=True, blank=True)

class Comment(models.Model):
    image = models.ForeignKey(Image, on_delete=models.CASCADE, null=True)
    user = models.ForeignKey(Rats, on_delete=models.CASCADE, null=True)
    comment_text = models.CharField(max_length=20000)
    comment_data = models.DateTimeField('Comment Date')
    rating = models.PositiveIntegerField(default=0)

class LikedGames(models.Model):
    class Meta:
        unique_together = (('user', 'image'),)
    user = models.ForeignKey(Rats, on_delete=models.CASCADE, null=True)
    image = models.ForeignKey(Image, on_delete=models.CASCADE, null=True)

class BookmarkImage(models.Model):
    class Meta:
        unique_together = (('user', 'image'),)
    user = models.ForeignKey(Rats, on_delete=models.CASCADE, null=True)
    image = models.ForeignKey(Image, on_delete=models.CASCADE, null=True)
```

Figure 1 Class Models

A classe **Image** vai ter associado uma imagem, o nome do jogo, uma ligeira descrição, o número de likes e de views associados aquele jogo, um booleano que indica se o jogo foi adicionado aos favoritos pelo utilizador que está a navegar na página e uma data de publicação que indica quando o jogo foi adicionado ao site. Este último atributo e o do número de likes e views servem para organizar a lista dos jogos consoante esses atributos.

A classe **Rats** corresponde ao utilizador registado no site, e tem uma relação de um com um com a classe **User**, o que indica que todos os **Rats** são **User**. Para além disso tem associada uma imagem de perfil, a qual pode ser alterada em qualquer altura.

A classe **Comment** tem associada a si o texto do comentário, a data em que foi feito o mesmo, e a classificação que foi atribuída. Esta classe tem também atribuídos duas chaves estrangeiras (a imagem e o utilizador). Isto acontece porque na página dos comentários mostramos quem foi o utilizador que efetuou o comentário e a foto do mesmo associada.

As classes **LikedGames** e **BookmarkImage** foram criadas para que fosse possível que o utilizador só conseguisse dar like e adicionar aos favoritos uma vez cada jogo. Após ter adicionado à sua lista de likes ou de favoritos, quando tentar realizar a mesma ação, este reverte-a, removendo dos jogos gostados ou dos favoritos. Para fazer esta validação temos associado à classe uma chave composta com o id do utilizador e o id da imagem.

Flow do utilizador normal

Um user novo não registado no site tem acesso às páginas podendo apenas visualizar os conteúdos nela contidos, não tendo acesso a algumas funcionalidades. Para isso este user irá criar uma conta, associando um username, password e um mail. Mais tarde poderá adicionar também uma imagem de perfil.

Após a criação da conta, este poderá adicionar os jogos aos favoritos, dar like a um jogo, comentar e remover o seu próprio comentário.

Pontos fortes e inovadores do projeto

- Desenvolvemos um carousel de imagens através de uma biblioteca de javascript (owl-carousel.js)

```
<div class="owl-carousel-wrapper">
  <h1 class="owl-title"> Top 10 most viewed games </h1>
  <br>
  <div class="owl-carousel">
    {% for image in latest_image_list %}
    <div> <a href="{% url 'games_hub_app:image_detail' image.id %}"> <img class="games_cover"
    {% endfor %}
  </div>
</div>
```

Figure 2 Carrousel de imagens

- Design do site: Quando passamos com o rato por cima da imagem de perfil esta faz um efeito em que rodopia; quando passamos com o rato por cima das imagens na página principal estas ampliam ligeiramente; a amostragem dos comentários; o site encontra-se todo ele em dark mode, e mantemos a consistência dos templates entre todas as páginas; os botões têm efeito néon; os gradientes no background vão de azul para roxo

- O utilizador só consegue adicionar os jogos aos favoritos, e dar like ao jogo uma vez. Isto é feito através da chave composta explicada acima.

- Validações feitas ao longo do site para manter integridade do mesmo evitando que o utilizador não registado/sem permissões consiga aceder a páginas que não devia. Por exemplo caso um utilizador tente através do url chegar à página de adicionar jogos, este vai ser reencaminhado para a página de login.