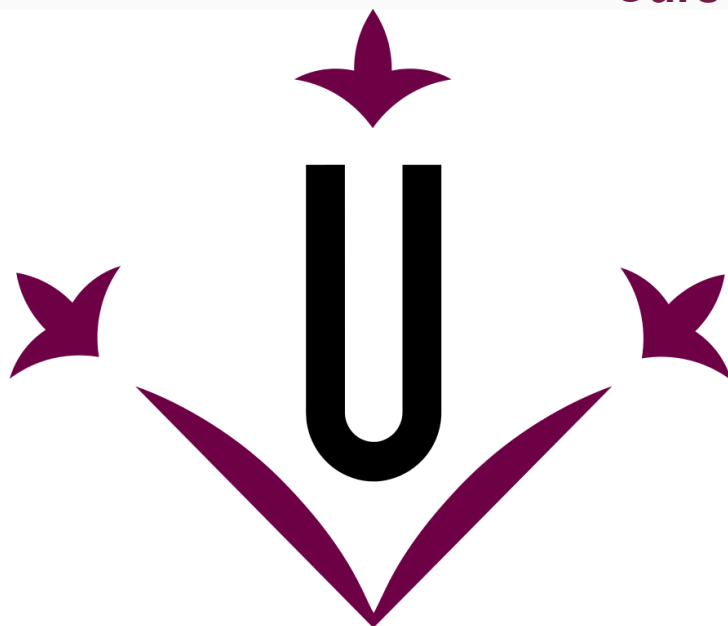


ESCOLA
POLITÈCNICA SUPERIOR
UNIVERSITAT DE LLEIDA

Curs 2023-2024



Universitat de Lleida

PROJECTE WEB **Deliverable 1**

Giovanni Maerean Rubiano

DNI: 48251274B

Albert Martín López

DNI: 49381774S

Pau Barahona Setó

DNI: 49380060A

Index

Functionality implemented	3
Sign up	3
User Authentication (Log in & Log out)	3
Components	3
Models	4
SteamUser	4
Product	4
Achievement	4
Developer	4
Publisher	4
Models diagram	5

Functionality implemented

Sign up

The user registration functionality is implemented using the SignUpView class.

When a user accesses the registration page (/accounts/signup/), they are presented with a form generated by Django's UserCreationForm.

Upon successful registration, users are redirected to the login page (/accounts/login/).

User Authentication (Log in & Log out)

Django's built-in authentication system is used for user authentication.

The login and logout functionalities are provided by Django's default authentication views, which are included via the include("django.contrib.auth.urls") in webProject/urls.py.

Users can access the login page (/accounts/login/) to authenticate themselves and access restricted areas of the application.

Components

As it was mentioned, the project uses some Django base components like user authentication views. The project applies Django's built-in authentication views and forms for user login and logout, and also Django's user model.

Some of the custom components that can be seen in the project are user Registration View (SignUpView), which is a custom view for user registration using Django's CreateView class. There are also defined some custom URL Patterns, additional URL patterns specific to user authentication, such as the registration page URL, and other custom url patterns as the home page.

Templates are also an important part of the project that should be pointed out, as user registration template, a custom HTML template for the user registration form, and also other templates as base, which will be extended by other templates, home for the home page and for the registration we have login and signup templates.

Models

Within models.py, as we mentioned in the project proposal, we have considered creating these models to have and store the data we need from the API, such as users, products, achievements, developers, and publishers.

SteamUser

This model represents the users of the Steam platform. Each instance of SteamUser has attributes such as steamID, which is a unique identifier for each user, realName that stores the user's real name, personaName that holds the name the user displays on Steam, profileUrl which is the URL of their profile on Steam, countryCode indicating the user's country, and timeCreated recording the date and time the user's Steam account was created. Additionally, it has relationships with other users (friends) and games (products) through many-to-many relationships.

Product

The Product model represents the games available on the Steam platform. Each game has attributes such as appId, which is a unique identifier for the game, name which is the game's name, type indicating the game type, age representing the game's age rating, free which is a boolean indicator to check if the game is free, among others. It also has a total_achievements property that calculates the total number of achievements available in the game.

Achievement

The Achievement model represents the achievements available in Steam games. Each achievement has a name and is associated with a specific game (product). Additionally, it can be unlocked by multiple users (users), so it has a many-to-many relationship with the SteamUser model.

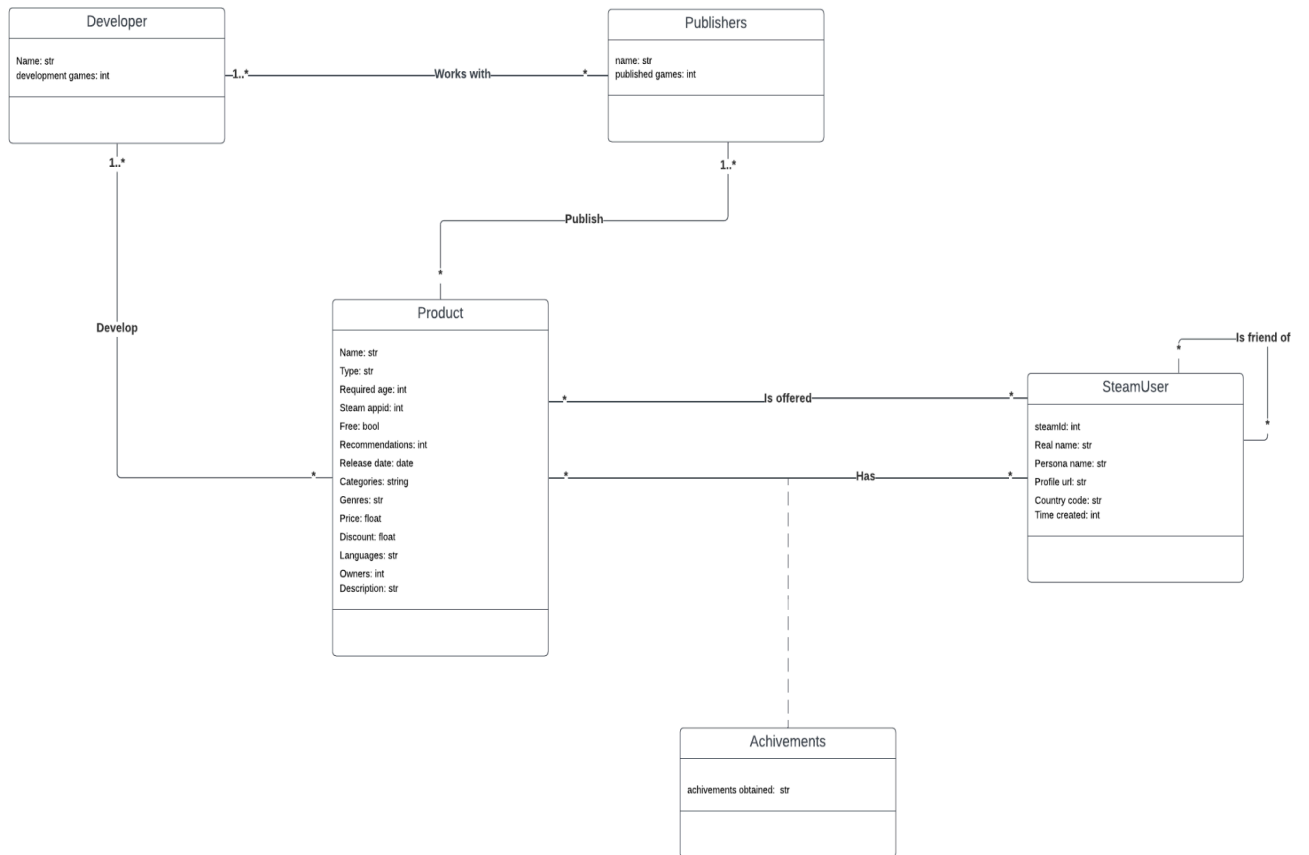
Developer

This model represents game developers. Each developer has a name and may have developed several games (products). Additionally, the total number of products developed by the developer (developedProducts) is recorded.

Publisher

The Publisher model represents game publishers. Each publisher has a name and may have published several games (products). Additionally, the total number of products published by the publisher (publishedProducts) is recorded.

Models diagram



Repository link: <https://github.com/GiovanniMaerean/webProject>