CS480 – Course project

Summer 2021

Database: epic_price_tracking

Ajay, Sanjay and Sampath

Description:

The store has games which have a unique name, publisher, genre, and other details associated with the game. Each publisher had published multiple games over the years. Each game might have addons like the DLC but require the base game to function. Our web scraping tool will scrape the Epic Game Store for prices weekly and record the prices for all the games. Users are also given a tool to filter the entire database of games by choosing a specific genre and publisher. When a user finds a game they are interested in they can track the game and add it to their list. Users will also have the option to remove these games from their list whenever they choose. Additionally, every time we run our web scraping tool, we allow users to compare the current price to the previous month's price, and notify all users who subscribed to be notified for that particular game. Our GUI tool, a web application for users to select games by publisher, genre, and view price history of the game.

Part 2 – CRUD (Create, read, update,and delete)

List of strong entities:

1. game
2. user
3. genre

List of weak entities:

1. user_interested_game
2. game_genre
3. price_record

We will implement a web scraping script capable of fetching the required information about the games from the Epic Games Store www.epicgames.com.

This Script will be exposed via an API that will be available to the admin to trigger a fetch from the website or can also be scheduled to perform a web scraping operation on a timely basis.

We will implement the following functionality using Java and SQL with necessary GUIinterfaces.

1. Insert/delete/update/read a game (all attributes except the game_id). The game_id should be generated by the system automatically using MySQL autoincrement.

2. Insert/delete/update/read a user (all attributes except the user_id). The user_id should be generated by the system automatically using MySQL autoincrement.

3. Insert/delete/update/read a user_tracks_game table which contains the games that users subscribe to receive discount information.

4. Insert/delete/update/read a genre table (all attributes except the genre_id). The genre_id should be generated by the system automatically using MySQL autoincrement.

Part 3 – Queries

Based on the Demo, we will implement the following functionality using Java and SQL with necessary GUI interfaces.

Trivial Queries:

1. List all genres
2. List all Publishers
3. Get the current logged in user information along with username and email etc.
4. Show games tracked by a user
5. List all games in alphabetical order

Non-trivial Queries:

1. Show all the games along with their publisher names
2. Filter games by a particular publisher
3. Filter games by a particular genre
4. Filter games by both a particular genre and publisher
5. Lists all users tracking a particular game
6. List games which have been discounted (on a month to month basis) and are tracked by a particular user. This would help send notifications to the user regarding price discounts for the games they are interested in.