Middlesex University Mauritius

CST 3130 Advanced Web Development with Big Data

Coursework 1: Price Comparison Website

2021 Oct Intake

Muddathir Muhammad Ibney Noorani Joomun

(M00830556)

Contents

Table of Figures

1. Description:

In the development of this game price comparison website, we will harness a combination of advanced technologies and programming methodologies to ensure robust, efficient, and scalable application performance. Here's a breakdown of the technologies and methods we plan to employ:

- Multi-Threading: To enhance the efficiency of real-time data scraping from multiple websites simultaneously, we will implement multi-threading in our Java application. This approach allows the concurrent execution of two or more parts of a program for maximum utilization of CPU. Each thread will pull data from different websites, ensuring a swift aggregation of data without bottlenecking the application's performance.
- Maven: As our build tool, Maven will manage the project's build, reporting, and documentation from a central piece of information. It will simplify the build process, manage project dependencies, and allow for easy project setup based on various templates (archetypes). While other tools like Ant or Gradle could be considered, Maven's widespread usage and strong community support make it a favorable choice.
- Spring Framework: We will leverage the Spring Framework for comprehensive infrastructure support for developing Java applications. Primarily, we'll utilize it for dependency injection to manage our objects, making our codebase more cohesive yet less coupled. Spring's robust ecosystem, including security and data modules, can also be beneficial as the project grows.
- Hibernate: As our ORM (Object Relational Mapping) tool, Hibernate will facilitate the storage and retrieval of Java domain objects in our SQL database. It's a stable framework that simplifies database operations, efficiently manages transactions, and helps to avoid common issues with database portability and compatibility.
- Web Scraping: To aggregate data from third-party websites, we will use JSoup or a similar Java library for web scraping. These tools will parse the HTML from these sites, allowing us to extract and manipulate data and, subsequently, populate our database with up-to-date information. Our approach will adhere to ethical scraping practices, including compliance with each site's terms of service and robots.txt rules.
- Database Manageent: We will use a SQL-based database, emphasizing complex table structures, efficient data storage, and adherence to SQL naming conventions and best practices. The complexity of our database design will correlate with the project's requirements, ensuring data integrity, normalization, and optimal performance.
- Testing: Rigorous testing is vital to our project's success. We will avoid trivial tests and focus on meaningful cases that assess our application's functionality, reliability, and performance. All tests must pass to ensure the application is market-ready. Source code will be submitted along with screenshots of the test results in the final report
- REST API: We'll implement a RESTful API to enable our application to communicate with external systems, or for future scalability purposes, allowing other clients like mobile apps or desktop apps to retrieve data from our system.

- Node.js: Though our primary backend technology will be Java, we might also consider using Node.js for certain parts of our system, especially if we need non-blocking, event-driven servers due to a high volume of data scraping

This tech stack was chosen for its robustness, maintainability, and community support, ensuring that our game price comparison website is reliable, fast, and user-friendly**.**

2. Wireframes

2.1 Home Page

The home page shall display the search for you to access the games and same shall show the latest top 5 updated games.

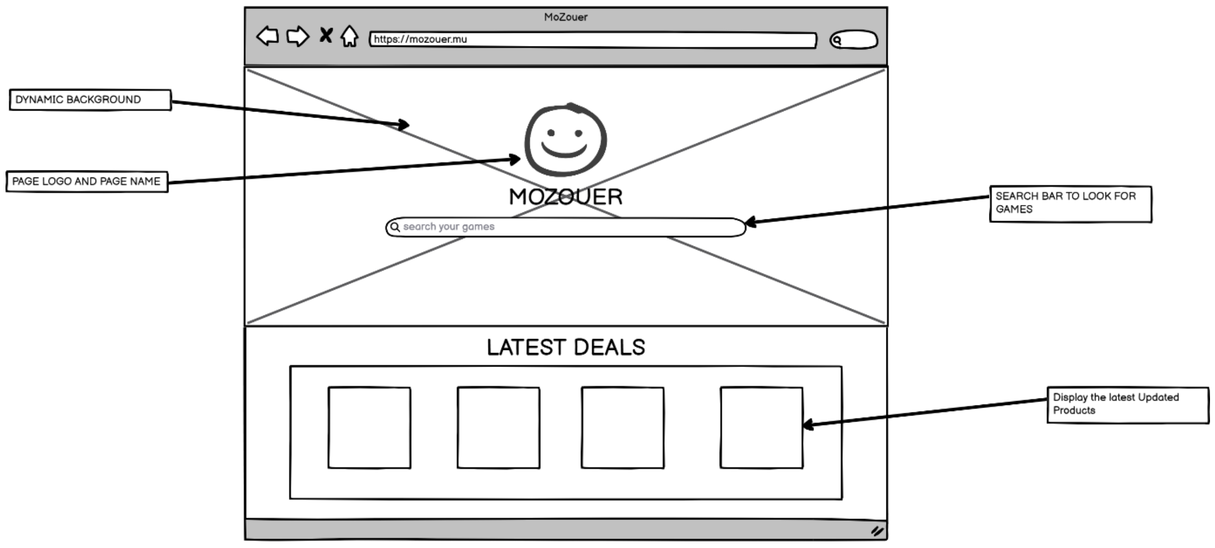

Figure 1: Home Page Wireframe

2.2 Product List

The product List Page shall display the product with different Prices from Different Vendors.

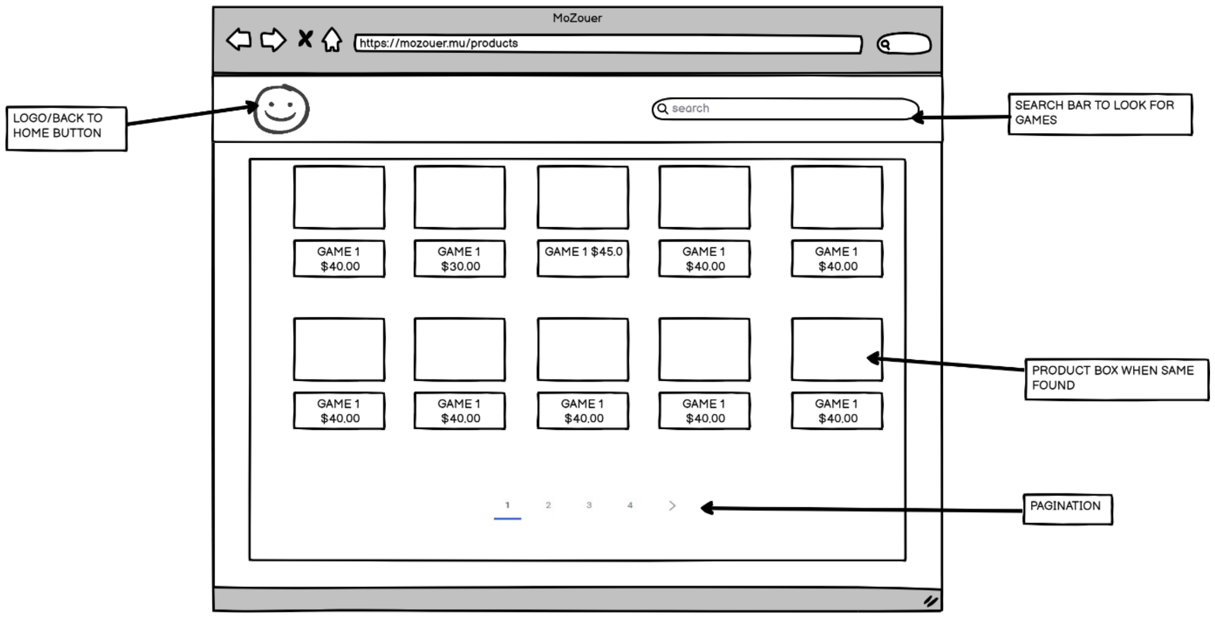

Figure 2: Product List Wireframe

2.3 Product Page

The product Page shall display the selected product with the description and the link to redirect same to the main vendors page

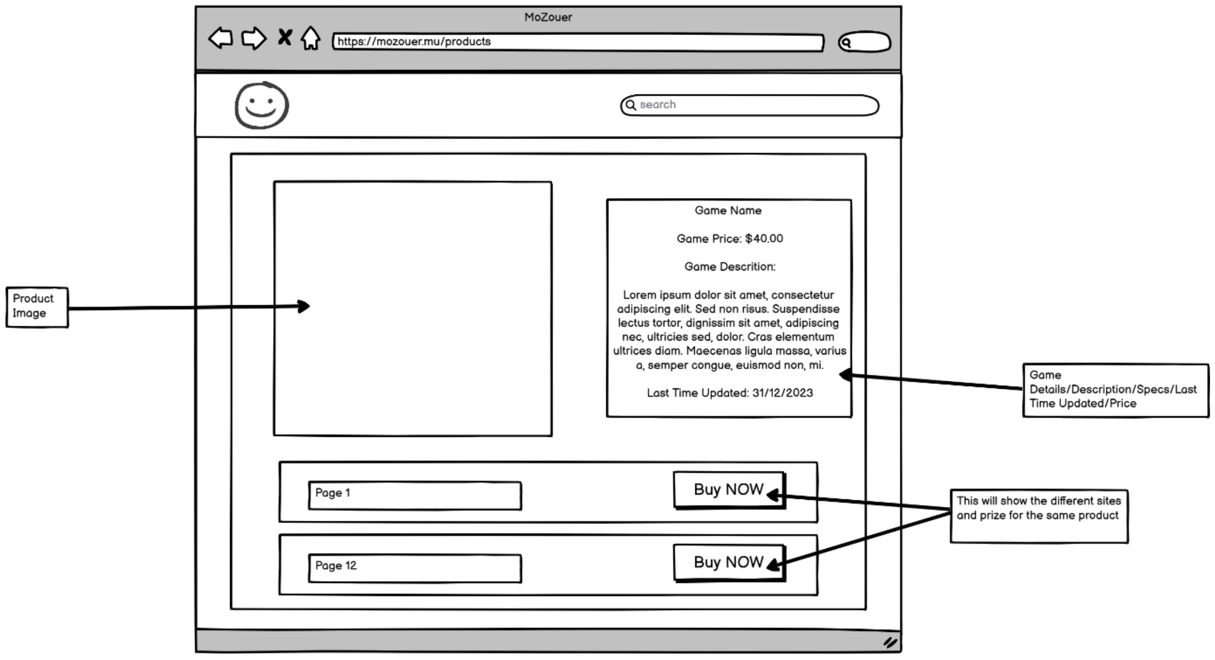
MoZouer
https://mozouer.mu/products
search
Game Name
Game Price: $40.00
Game Descrition:
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi.
Last Time Updated: 31/12/2023
Product Image
Game Details/Description/Specs/Last Time Updated/Price
Page 1
Buy NOW
Page 12
Buy NOW
This will show the different sites and prize for the same product

Figure 3: Product Page Wireframe

3.Data Sources:

Our project relies on scraping pricing and game-related data from several online gaming stores. It is essential to maintain ethical and responsible web scraping practices. We intend to respect the guidelines set forth by the websites by referring to their robots.txt files. Here's a breakdown of the data sources and how we plan to interact with them:

1. **Steam** - https://store.steampowered.com/
2. **GOG (Good Old Games)** - https://www.gog.com/
3. **Gamivo** - https://www.gamivo.com/ :
4. **K4g** - https://k4g.com/
5. **Amazon**- https://amazon.com/

4. Database Design:

Our database design is designed to efficiently manage and organize data for our game price comparison website. It includes the following key elements:

- Games is the core entity that represents different games with their title, SystemRequirements,and descriptions.

- Comparison stores details of various Price, Game ID and URLs.
- Supported Platform will store the Game_ID together with the Platform

4.1 Relationships & Indexing

Games is the core entity representing different games, storing their titles, system requirements, and descriptions.

Comparison stores details of various game prices, including GameID, URLs, and price information.

SupportedPlatform associates each game with the platforms on which it's supported.

Indexing:

Indexes should be created on foreign keys (GameID) for efficient data retrieval.Additional indexes can be established on frequently queried fields to optimize search performance.

4.5 Normalization

The database design is normalized to eliminate redundancy and maintain data consistency. Data is organized into separate tables to prevent data duplication and ensure data integrity.

Below is the design of the database

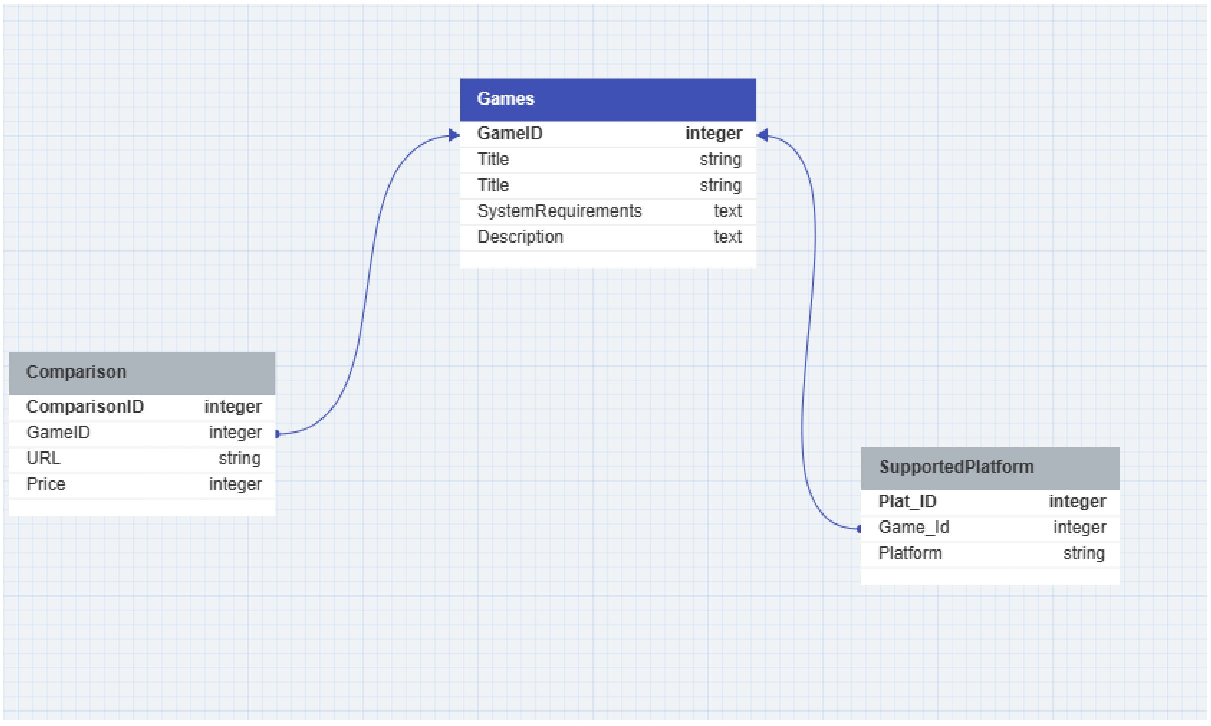

Figure 4: Database Design