



DATABASE AND WEB DEVELOPMENT – BTIT058

Gary McNeill – 10187871

Joe Lloyd - 10177148



TOPIC AND OBJECTIVES

Our chosen business for this web application is an online PC game store. This store will allow customers to create accounts, view product details and purchase games for digital download. The target audience for this site is people who are passionate about PC gaming and will most likely have invested in high speed broadband and building their own machines. This site will offer a fast and convenient way to buy games as well as see news.

HIERARCHICAL STRUCTURE

The structure of this site is very simple. All forms but one are on the root of the server. The exception to this is the game details page which is behind the game list page. We are using two folders for images in this project. 'GameImg' is used for holding images which are uploaded from the admin page when adding a new game. The 'Img' folder contains images for the slideshow on the main page.

VISUAL DESIGN

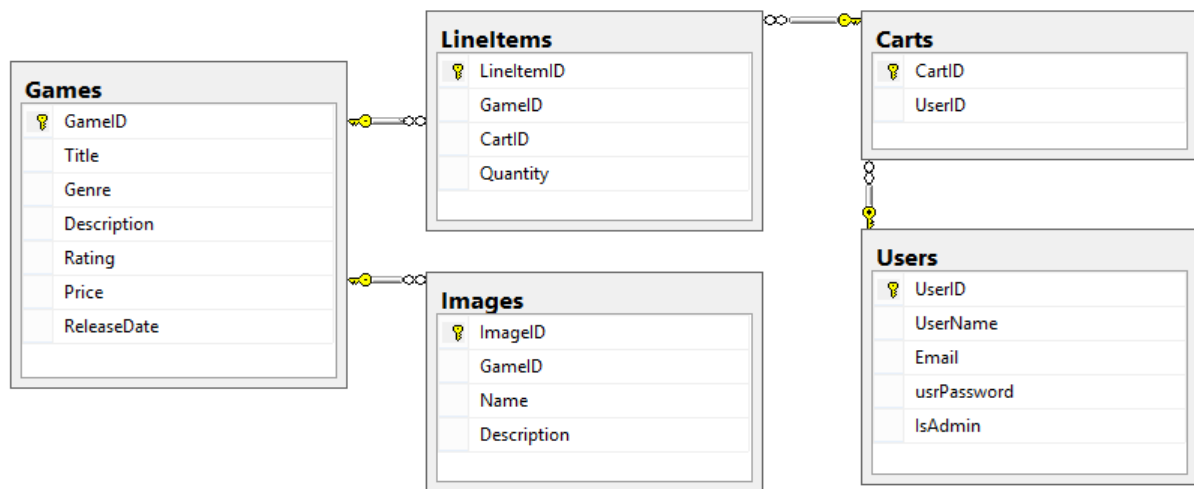
The theme for this website was chosen based on similar game and game store websites. The clean and dark aesthetic makes colours from the game images stand out and draw the eye. This means that users are more likely to stay on the website longer which will hopefully result in more purchases.

This site is primarily based on a single column layout using bootstrap. There are pages however, with both single, multiple column rows. On the game list page for example, the games are laid out in 4 columns across which are formatted using `asp:ListView` and `ItemTemplate`. Bootstrap has saved us a lot of time in the design phase and has allowed us to move more quickly on to the code behind it.

This site maintains its layout on every screen size we have tested on from 720p-1080p. This is due to the built-in scaling of bootstrap. This means the site fits into mobile with no additional styling required, this was tested at the smallest browser size possible to replicate a phone screen.

CSS classes were all taken from bootstrap and provided the jQuery necessary to animate buttons and outline focused textboxes with a slight glow. There were no special fonts used as the default sans-serif font suited the minimalist aesthetic quite well.

The database for this project contained five tables, seen below.



On the admin page there are four options. Add, update, delete and serialize a game. When the admin wants to add a game they must fill in the required information and provide an image to be displayed with the game. When the 'Add Game' button is clicked the games table is populated with the information provided and a new name is generated for the image. This new name is stored in the Images table. The image itself is stored in the 'GameImages' folder.

In the game list and game details page when a game is called the image name is taken from the database and appended to the 'src' field in the list view item template.

Update is nearly identical to this bar the image upload and delete uses a game id to find a record and remove it from the database.

Serialize to XML uses a stored procedure to pull a single record from the database using a game id which is input by the user. It then binds this record to a data table. The 'WriteToXML' function is then called and the XML file is saved to the desktop of the admin.

A shopping cart is created when a user is registered and each user will only have one shopping cart. When a user wishes to buy an item they click the 'buy now' button. A 'LinelItem' entry into the database is then generated and acts like a reference telling us that what games a specific cart contains.

TECHNOLOGIES & TECHNIQUES USED

Naturally, the primary technologies used for this project were ASP and C#. ASP providing the controls and validation and C# allowing us to add functionality behind these controls and communicate with the database.

We made use of both ADO and Entity Framework during development. Entity Framework was used for all create, retrieve, update and delete tasks and ADO for binding information datasets and our password encryption when a user registers. The reason we used ADO for these two functions was that we felt we could accomplish the tasks more quickly and not because we believed it was a better option.

We have added two session states in our site. One for administrators and another for normal users. The navigation button for admin controls is hidden for users using some basic if else logic. If a normal user attempts to gain access to the admin page they will be automatically redirected to the default page. Similarly if an admin logs in they will be redirected to the admin controls page.

If neither of these session states are active then the site will redirect to the login/registration page and all other pages will be inaccessible.

We made use of the asp Ajax control on the shopping cart item in the navigation bar of the site. When a user logs into the site this control shows the number of games the user has in their basket. This is displayed using a label in an asp Ajax update panel. This label is updated each time the user adds a new game to their cart.

MAINTENANCE

Given the simplicity of the admin page we believe this site will be quite easy to maintain up to a point. Once the game list page reaches a certain amount of products it will become unfeasible to have them all on one page. At this point we would then separate them into separate pages based on genre, rating and price. A search feature would also make navigating the site significantly easier as the product base grows.

In terms of the database we do not see any reason to extend it currently. However, given more time additional tables could be added for the news items. This would allow for a

similar layout to the game list page meaning users could choose an article they wish to read and then be redirected to a page for that article.

Given more time the entire site could be dynamically updated in the same way games are updated, added and deleted. We could design a back end that any non-programmer could edit things like the carousel images or the about us text i.e. a fully comprehensive content management system.

SEARCH ENGINE OPTIMISATION

Much of SEO was taken care of in the early stages of development. At the top of each page we used the <meta> description tag. This will put a description below the website name when it comes up as a search result and contributes to the keywords people may use when searching for it. Having appropriate page names for the HTML files is also took a high priority. In addition to we named all image files and pages with useful keywords. For example naming an image you use to link to the detail page for a game 'Warcraft' and not something nondescript such as 'wpage' or 'image1'.

In the case of images the 'alt' attribute will show text if the image fails to load. This is very useful for people with disabilities who may be using screen readers. In our image table we have a description column, this was intended for use on alt tags for when images were rendered in the view. However due to time constraints we were unable to implement this

STRENGTHS AND WEAKNESSES

From an ease of access perspective, the main strength of this site is the clear navigation. There is very little if no ambiguity and because of this anyone with basic website knowledge could easily purchase a product.

As mentioned previously, page redirects have been put in place to ensure non-logged in users cannot access any page except login/register.

The front end admin controls for interacting with the database are easy to understand which means that even non-programmers can manage site content.

Validation of the login, register and admin forms are also quite robust. Users cannot re-register an email address, login without correct credentials or leave null values in forms. Regular expression and required field asp validators were used to accomplish this.

We failed to include a 'short description' column in our database on the games to use in the head of the 'GameDetails.html' pages, this would allow us to have a unique description in the header tags on each games details page.

We also came across a bug with our Ajax content panel, when a customer buys a game the number does not update to the correct number but instead always remains one behind where it should be. The conclusion was that the number was being updated before the database 'lineItem' was created. Upon navigating to the cart page the correct number of games is shown at the top.

CONCLUSION

This project was a great opportunity to learn all sorts of technologies and methodologies. During the initial few weeks of the project we took a 'pair programming' approach to design. This allowed us to get used to the entity framework and bootstrap integration in an environment where both of us could help address any problems we encountered individually.

After a few weeks we both felt comfortable enough to work on separate parts of the site. At this point we created a repository on git hub and used git bash for version control. Git was an invaluable tool. Even with both of us working side by side for the majority of the project we didn't exchange a single item via data stick, drop box etc. Git made merging our code easy and quick and also left a virtual paper trail that we were able to use to easily catch bugs and problems quickly, it also kept us accountable for what we pushed to the repository.

Use of the entity framework greatly increased our production speed. The time consuming process of creating stored procedures was nearly totally eliminated. We were able to get 'C.R.U.D' (Create Retrieve Update Delete) up and running relatively fast with entity framework. Using the linq queries we were able to pull all the necessary information from the database quickly which made the process far more efficient.

As stated above we used bootstrap to style our site, the use of this frame work left us with more time to focus on the more difficult coding process and freed up many hours for us. We both agree that bootstrap was an invaluable tool for any site even if it was just used for its column layout.

Overall we felt by doing this project we learned a lot about database driven web development. We found ASP to be a very powerful and useful tool and look forward to further developing our skill sets in the future.