Use Cases and Logical Architecture

* **XID:** X00191019
* **Name:** Jason O’Connor
* **Project Title:** GameVault – Video Game Search Website

Section 1: For Each Use Case:

|  |  |
| --- | --- |
| Title (goal) | What is the name of this function? |
| Primary Actor | Who is the user? |
| Story | Describe in detail what happens here |

## Section 2: Logical Architecture

In this section use [www.draw.io](http://www.draw.io) to show how your architecture is distributed. Your architecture should show:

|  |  |  |
| --- | --- | --- |
| Software Components | App Engines | Databases |
| API’s & Languages | Deployment | Security |

A diagram of a computer

Description automatically generated

## Section 3: Logical Architecture Discussion

In this section, discuss each component of your architecture.  
Add numbers to the arrows and discuss the flow of data.

Section 1: For Each Use Case:

|  |  |
| --- | --- |
| **Title (goal)** | Search for Video Game |
| **Primary Actor** | Unregistered, Registered, Premium Users |
| **Story** | The website loads to the home page, where a search bar is present in the middle of the screen. User clicks the search bar and types in the name of the video game they wish to find, and the search returns related results to the input. |

|  |  |
| --- | --- |
| **Title (goal)** | Create Account |
| **Primary Actor** | Unregistered Users |
| **Story** | In the Navigation Bar, a Sign-Up button is clicked, taking the user to the account page. Here the user enters their email address and password into the designated fields and clicks Sign Up. The page will refresh and direct them to the Sign In page upon successful sign up if requirements are met. |

|  |  |
| --- | --- |
| **Title (goal)** | Sign In |
| **Primary Actor** | Registered, Premium Users |
| **Story** | In the Navigation Bar, a Sign In button is clicked, taking the user to the account page. Here the user enters their email address and password in the designated fields and clicks Sign In. The page will refresh, taking the user to the home page now signed in to their account. |

|  |  |
| --- | --- |
| **Title (goal)** | Leave a Review on Video Game |
| **Primary Actor** | Registered, Premium Users |
| **Story** | On the video game information page, the user chooses a star rating between 1 – 5, including half stars, and then clicks into the description box where they can enter their review. Once input, the user clicks post review, and the review is added to the video game information page in the review section. |

|  |  |
| --- | --- |
| **Title (goal)** | Add Video Game to Wishlist |
| **Primary Actor** | Registered, Premium Users |
| **Story** | On the video game information page, there is a heart icon button near the cover art of the video game. When clicked, the user confirms they wish to add the game to their wish list. Once confirmed, the game is added to the wish list after the page refreshes and reloads to the game page. |

|  |  |
| --- | --- |
| **Title (goal)** | Add Premium to Cart |
| **Primary Actor** | Registered Users |
| **Story** | Registered user clicks icon in the Navigation Bar named Get Premium, redirects to page informing the user of premium benefits. They click purchase and it adds premium to the cart, redirecting them to the cart page. |

## Section 2: Logical Architecture

A diagram of a computer

Description automatically generated

## Section 3: Logical Architecture Discussion

For the architecture of my website, which will be coded in Django Python, the above graph demonstrates a user accessing the website on a PC, but it can also be accessed on a mobile device such as a phone or tablet using the web browser, shown by arrow 1.

When the computer accesses the website, shown by arrow 2, the website will be hosted using a service such as Amazon AWS to ensure maximum availability and minimum downtime or issues to the hosting of this website.

Shown by arrow 3, when the website is accessed, it is populated by data stored in Amazon AWS services such as S3 Bucket for media and RDS MySQL for user profiles. Since my website is centred around video games and information relating to them, I will use an API to populate my database with the required information. This API is shown by arrow 4 and it will fill my database will hundreds of records of video games to ensure every game a user could want to find is available, and by using an API and storing on AWS I will ensure this information is available at all times and not stored locally on a PC which would be inefficient and not suited for a website such as this.

Arrow 5 demonstrates that when a user wants to carry out statistical analysis of their video game habits, they can create graphs and view their input information in easy-to-understand graph format. These graphs are then saved to the user’s profile for viewing at any time whenever they are needed.

Arrow 6 demonstrates that when these graphs are created, they are saved to the website and available with the rest of the website once it is deployed. Lastly, the website is viewable and ready to use on the PC when the user visits the website.