

Phase 2: Technical Report

Website URL: <http://gamedb.us-east-1.elasticbeanstalk.com/>

Team Information:

Canvas Group: morning-1

Project Name: Game-DB

GitHub Repo Link: <https://github.com/numan201/game-db>

Phase Lead: John Nguyen

Members:

Albert Garza

- **GitHub ID:** possumrapture
- **Email:** albertsgarza@utexas.edu

Numan Habib

- **GitHub ID:** numan201
- **Email:** numanhabib@gmail.com

John Nguyen

- **GitHub ID:** johnnguyen3196
- **Email:** johnnguyen3196@gmail.com

Alejandro Rodriguez

- **GitHub ID:** JustAlejandro
- **Email:** rodriguezalejandro@utexas.edu

David Wolf

- **GitHub ID:** rambisco
- **Email:** david.wolf@utexas.edu

Motivation:

We want to make a useful video game database for gamers. By incorporating data from various APIs, other websites, and communities, we can have a variety of useful information all in one place for our users.

Users:

All types of gamers, people interested in learning about different games, people shopping for games, and people who like to consume gaming related entertainment.

Phase 1 User stories:

1. As a grandmother with a grandchild who likes Fortnite, I want to know what Fortnite is so that I know my boy is not playing inappropriate games.
 - a. Estimated: 5 hours
 - b. Actual: 8 hours
 - c. Assumptions: user has no gaming experience and does not normally keep up with the current games
2. As a fan of Rockstar Games, I want to know some of the games that they made so that I can find other cool sandbox games that I can spend all day on.
 - a. Estimated: 5 hours
 - b. Actual: 6 hours
3. As a potential game buyer, I want to preview screenshots and videos of the game so I have an idea of the game type and how fun it can be.
 - a. Estimated: 3 hours
 - b. Actual: 3 hours
4. As a software engineering professor, I want to know who is making commits to a GitHub repo so I know how much each person is contributing.
 - a. Estimated: 3 hours
 - b. Actual: 12 hours
5. As a person who likes to watch Twitch streams, I want to watch a stream for a game I find in the database so that I can see its gameplay live.
 - a. Estimated: 6 hours
 - b. Actual: 8 hours

6. As a frugal video game purchaser, I want to know how many 5/4/3/2/1 star reviews there are for a game so that I can make an educated decision in purchasing a game and not waste time on a bad game.
 - a. Estimated: 3 hours
 - b. Actual: 1 hour
 - c. Assumptions: user does not have much free time because they are normally busy with work
7. As an owner of a PlayStation 3, I want to know what platforms each game is on so that I know if I can play it on my console
 - a. Estimated: 1 hour
 - b. Actual: 2 hours
8. As a video game historian, I want to know the release date of my favorite games, so that I can be factually correct in my discussions with peers.
 - a. Estimated: 1 hour
 - b. Actual: 1 hour
9. As a software engineer interested in game development, I want to know which studio made my favorite games, so that I can send in an application.
 - a. Estimated: 2 hours
 - b. Actual: 4 hours
10. As a software engineer interested in web development I want to know the tools used to make a website, so that I have an idea of the capabilities of tools and which tools I should invest time in learning.
 - a. Estimated: 0.1 hours
 - b. Actual: 0.5 hours

Phase 2 User stories:

1. As a user, I want to see the latest and top games and gaming news so I don't have to spend a long time searching through the website to find these games and news.
 - a. Estimated: 8 hours
 - b. Actual: 8 hours
2. As a gaming enthusiast, I want to check the news for specific games so I can make conversation with my friends and know what games have interesting gameplay.
 - a. Estimated: 5 hours
 - b. Actual: 7 hours
3. As a user, I would like to be able to easily view all games, developers, and publishers in the database page by page (pagination) so the website is more pleasant to view.
 - a. Estimated: 3 hours
 - b. Actual: 3 hours
 - c. Assumptions: page by page means pagination
4. As a gamer looking to purchase a video game, I would like to see how many people are currently playing the game on Steam so I can know if I will be able to find an online lobby quickly.
 - a. Estimated: 2 hours
 - b. Actual: 1 hours
5. As a Game-DB user, I would like to keep track of games I would like to purchase so I can get them in the future when I have enough money.
 - a. Estimated: 4 hours
 - b. Actual: 6 hours

Design:

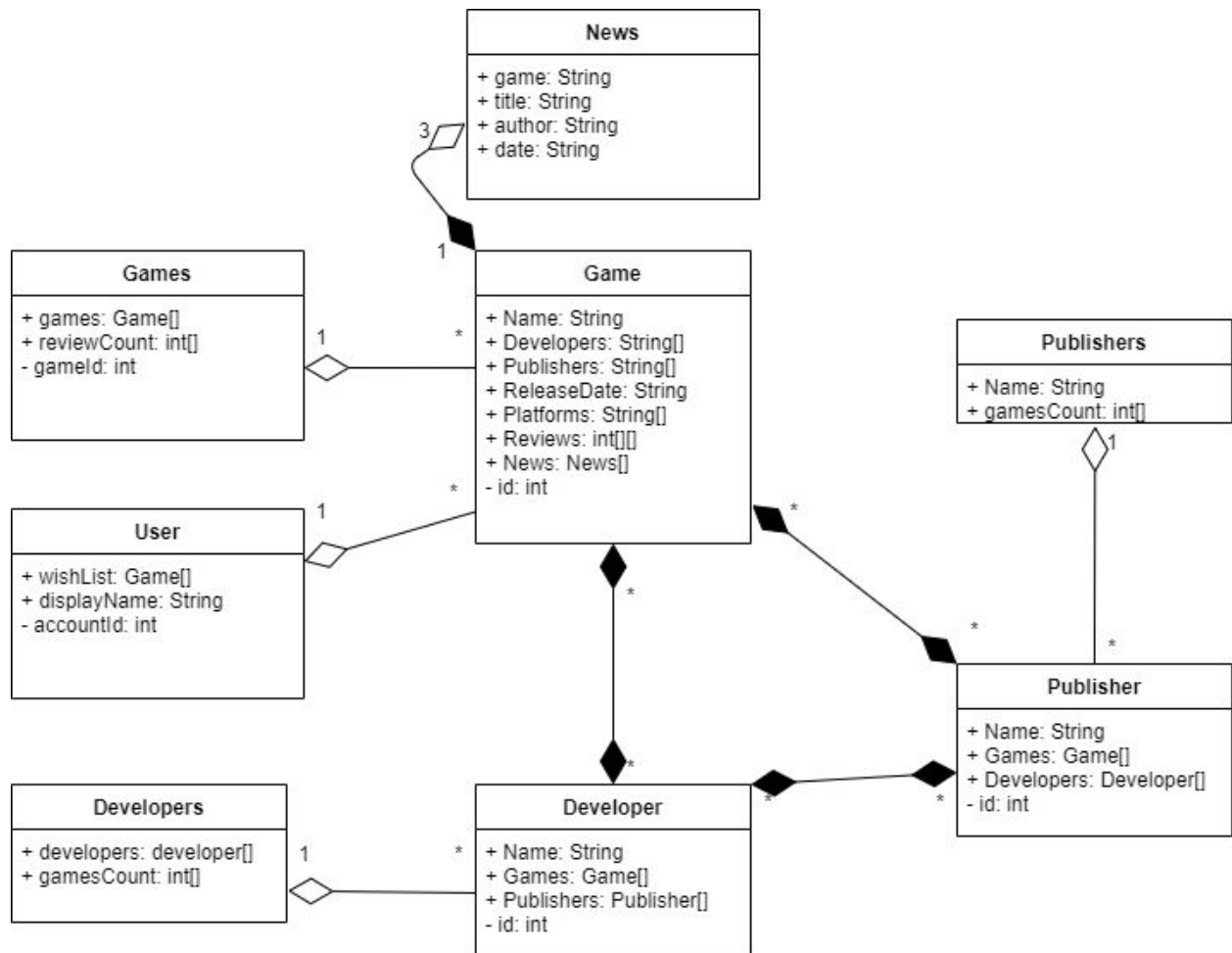
Phase 1:

For our design we used modern tools to expose us to technologies commonly used in the industry right now. Our backend is Node.js with MongoDB, and our website is hosted through AWS. For our framework, we chose Express since it is a common framework used to make web applications with Node. It makes web development with Node easier and more structured. When a user visits a URL on our domain, the framework routes it to the appropriate router file which will handle the request. The router will query and process data from MongoDB/APIs necessary to render the page. After this, it will provide this data in an object to a template file (.ejs) specific to this page. The .ejs file created for this page will render the provided data in HTML to serve the browser request\generate web page. Our .ejs files use various CSS classes and Bootstrap 4 to make our pages aesthetically pleasing and keep design simple.

Phase 2 Update:

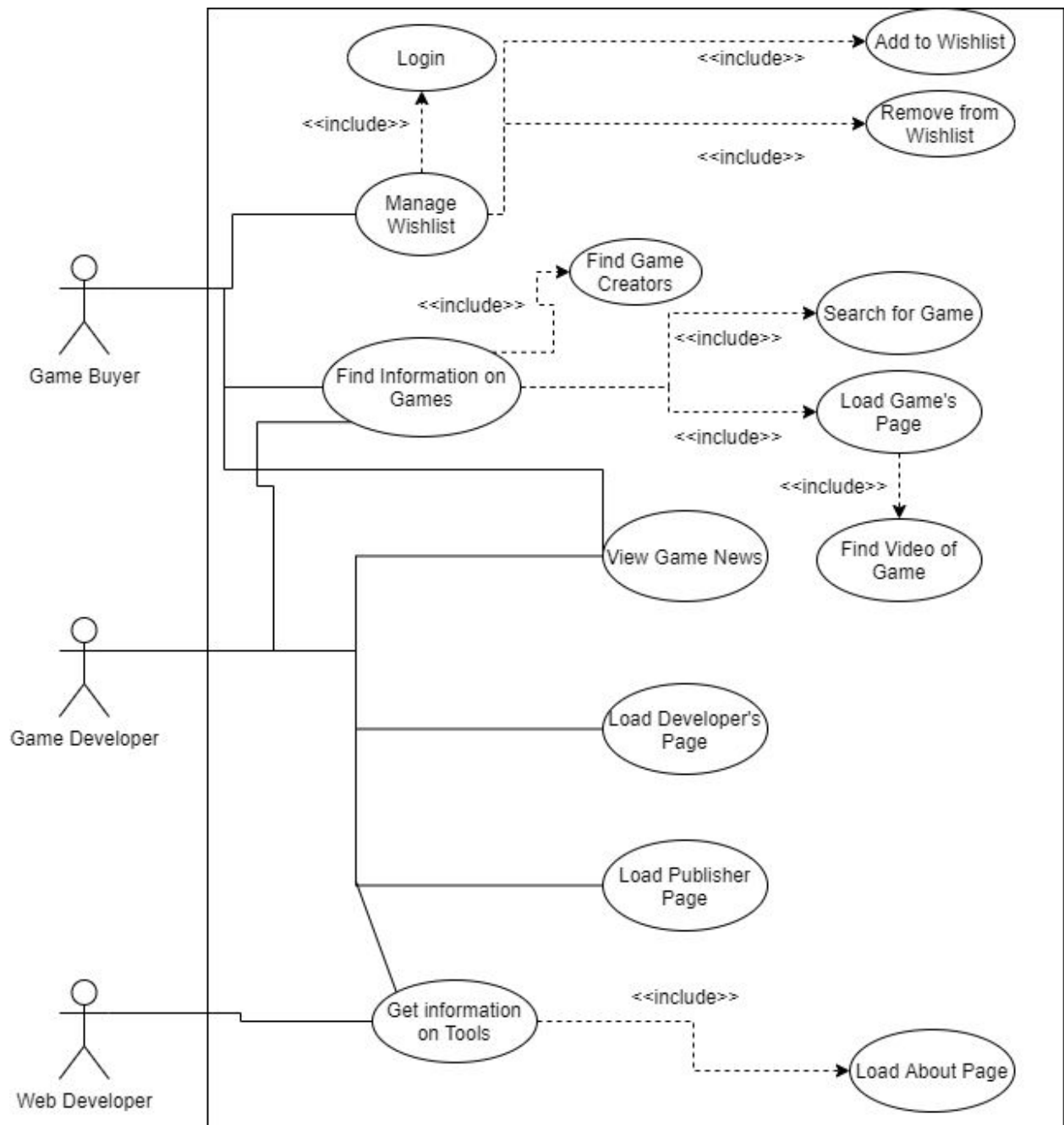
To create a user system for wishlists, we used Passport.js to help us create our user authentication. Passport.js assisted us in creating user sessions and implementing OAuth login through Google Accounts. It also allowed us to restrict certain functionality\web pages on the website only to logged-in users.

UML:



Class Diagram, Figure 1

Class diagram showing the relationships between different models, and the functionality that can exist between each.



Use-Case Diagram, Figure 2

Diagram showcasing the different actors for the website, and the processes they can go through to interact with the system.

Testing:

Manually tested publicly viewable pages and all of their features.

We did GUI testing by using Selenium to automatically test different parts of our webpage. All unique types of connections were tested such as: Developers -> Games, the GitHub Repo Link, pagination, and the tab to go to the Games model. Since we have a login feature, there are tests exclusive to being logged in and logged out. There are 5 test suites: Logged In, Logged Out, Doesn't Matter If Logged In/Out, Smoke Test, and All Tests (which is used to keep track of which tests have not been assigned to a suite).

We tested JavaScript code by ensuring that all pages and JavaScript routes return a successful code. Each URL is routed to a JavaScript file which is then rendered with a JavaScript template. We tested URLs that did not require parameters by ensuring a successful load, and tested URLs that did have parameters by getting sample objects from the database and ensuring that loads still completed successfully.

Models (required 3 are underlined):

1. Games:

a. Attributes:

- i. Using RAWG API
 - 1. Name
 - 2. Release Date
 - 3. Developers
 - 4. Publishers
 - 5. Platforms
 - 6. Reviews
 - 7. Related Screenshots
- ii. Using YouTube API
 - 1. Related Videos
- iii. Using Twitch API
 - 1. Related streams
- iv. Using Steam API
 - 1. Steam player count
 - 2. News

2. Developers:

a. Attributes:

- i. Using RAWG API
 - 1. Name
 - 2. # of Games made
 - 3. List of games made
 - 4. List of publishers worked with

3. Publishers:

a. Attributes

- i. Using RAWG API
 - 1. Name
 - 2. Developers worked with
 - 3. Games published

4. News

a. Attributes

- i. Using Steam API
 - 1. Title
 - 2. Author
 - 3. Date

Tools:

1. Express JS
 - a. Framework to create websites with Node more easily.
2. Axios
 - a. Library to make HTTP requests easily.
3. Bootstrap:
 - a. Used to make our website look better and more modern.
4. Passport.js
 - a. Authentication library to make user authentication system.
5. Postman
 - a. Test get requests for APIs
6. MongoDB
 - a. Database for storing site data. Also helps to reduce API calls.
7. Amazon Web Services (AWS)
 - a. Host website
8. WebStorm
 - a. IDE used to run website and debug
9. Visual Studio
 - a. IDE used to run website and debug
10. Visual Studio Code
 - a. Text editor with syntax highlighting
11. Google Chrome (Chrome Developer Tools)
 - a. Used to find bugs on website with Inspect Element tool
12. RAWG API
 - a. Used to pull information about games, developers, and publishers
13. YouTube API
 - a. Used to get relevant Youtube videos of a game
14. Twitch API
 - a. Add videos of people livestreaming games
15. Github API
 - a. Dynamically get commit and issue numbers from our repository
16. Google News API
 - a. Used to get overall gaming news on frontpage
17. Steam Web API
 - a. Used to fetch player numbers
 - b. Serves relevant news for each game
18. draw.io
 - a. Make UML diagrams to describe our website design

19. Selenium

- a. GUI testing

20. Mocha

- a. Unit Testing for JavaScript

Reflection:

Phase 1:

We started working on this project early and finished 95% of Phase 1 half a week before it was due. We were also very communicative by having daily SCRUM meetings, properly distributing work, and reallocating workloads when necessary. The frameworks and APIs that we used were also chosen early in the planning process. Some improvements that we can make is planning a full schedule for meetings, instead of figuring out what we want to do for the meeting when we arrive at the meeting place. We can also plan weekly meetings at a specific time that works for everyone, instead of playing it by ear. We all learned more about web development tools. We also learned that we work effectively on our individual assignments when being in a group because we can easily use each other as resources instead of having to always find help online.

Phase 2:

During Phase 1, we made our website dynamic and stored a lot of data on MongoDB, so for Phase 2 we were able to focus more on additional features beyond the requirements such as adding a wishlist for games. There was significantly less communication between group members during this phase, compared to the previous phase, because we could not meet in person, due to Spring Break and the ongoing pandemic. However, we extensively communicated through Slack and helped each other debug our issues. We learned that in-person meetings make it easier to engage everyone and communicate more effectively because it is easier to understand visual cues and share what is on our screens.