

GameEye: The Gamer's Eye Into The Gaming Industry

Brenden Lewis

CS410

Team Yellow

Professor Janet Brunelle

Old Dominion University

Table of Contents

Table of Contents	2
Introduction	4
Societal Problem	4
Solution Description	5
GameEye Product Description.....	6
Key Product Features and Capabilities	6
<i>General Features & Notifications</i>	6
<i>Authentication & Account Management</i>	9
<i>Game Tracking & Searching</i>	10
<i>Web Scraping & Machine Learning</i>	12
Major Components.....	14
<i>Hardware</i>	14
<i>Software</i>	15
Frontend. GameEye is a PWA that utilizes the WebStorm IDE, Angular Framework, ..	15
Backend. The backend of GameEye uses the IntelliJ IDEA IDE, Spring Framework, .	15
Testing. Testing for GameEye and all its functions will be	15
Databases. MongoDB, MongoDB Compass and the.....	15
Third-Party Software. For third-party software, GameEye uses the Auth0	15
Machine Learning. The software used to handle the machine learning algorithms	15

Natural Language Processing. For natural language processing, GameEye will	15
Identification of Case Study.....	16
Glossary	17
References	19

List of Figures

Figure 1. Current Process Flow	4
Figure 2: Solution Process Flow	6
Figure 3: Information Categories Mock-Up	7
Figure 4: News Articles Mock-Up.....	8
Figure 5: Settings Mock-Up.....	9
Figure 6: User Watch List Mock-Up	10
Figure 7: Searching Mock-Up	11
Figure 8: Web Scraping Flow Diagram	12
Figure 9: Web Scraping Service Algorithm.....	13
Figure 10: Importance Score Scale	13
Figure 11: Major Functional Component Diagram	14

Introduction

Societal Problem

In the current online landscape, it can be difficult for avid gamers to follow their favourite games. Thousands of games are released annually (Gough, 2019a; Gough, 2019b), with some of those games spending upwards of 7-9 years in development (Dietz, 2011). Information about these games are decentralized and often widespread across many different news sites, developer blogs, streaming platforms, and social media. This spreading of information can make it both difficult and time-consuming for gamers to find new information for upcoming games in development or current games that are receiving new updates or content.

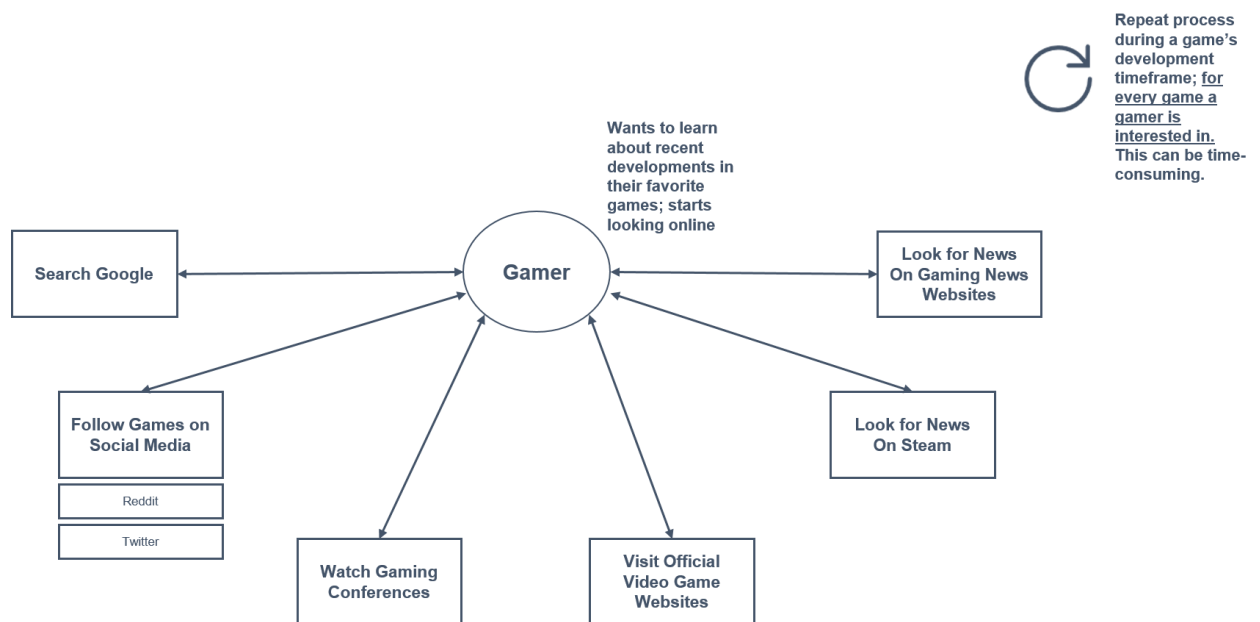


Figure 1. Current Process Flow

Figure 1 shows the current process flow of a gamer looking for information about a game that is coming out soon that they want to play. To stay up-to-date with the game's development prior to release, the gamer may have to actively and regularly check each of these different sources of information in case any new information is released; this is an active process dealing with a lot of noise both before and after a game is released, and is often applied to multiple

games at a time. This kind of routine can be both time-consuming and tiring, which can lead to gamers easing up on news-searching and risk missing out on new information. As a result, gamers can often forget about the games they, at one point, expressed interest in. This poses a huge challenge for small, independent (indie) developers to keep attention on their games, both released and upcoming, as they become overshadowed by larger, more popular AAA franchises.

Solution Description

In order to alleviate these issues, gamers need a singular location to find all the news for their favourite games; they need a platform that is accessible, customizable, and capable of notifying them when new news is released while also filtering out any unwanted noise. This will keep gamers up-to-date with the games they follow and help prevent loss in retention for smaller games.

This platform is GameEye, a progressive web application (PWA) for computers and mobile devices that allows user to craft a personal watch-list of their favourite games and be notified when any news or updates for those games are released. GameEye utilizes web scraping to find news for games online and links the user to those sources so they don't have to bother searching themselves. Machine learning is also utilized to classify these new sources using a multi-factor scoring system to determine the importance of news updates for use in filtering unimportant content and notification customization. Users will be able to easily search for and add games, both current and unreleased, to their personal watch-list which can be customized however the user desires.

GameEye Product Description

Key Product Features and Capabilities

GameEye is able to scour the internet to find news about specified games and collect them for convenient viewing.

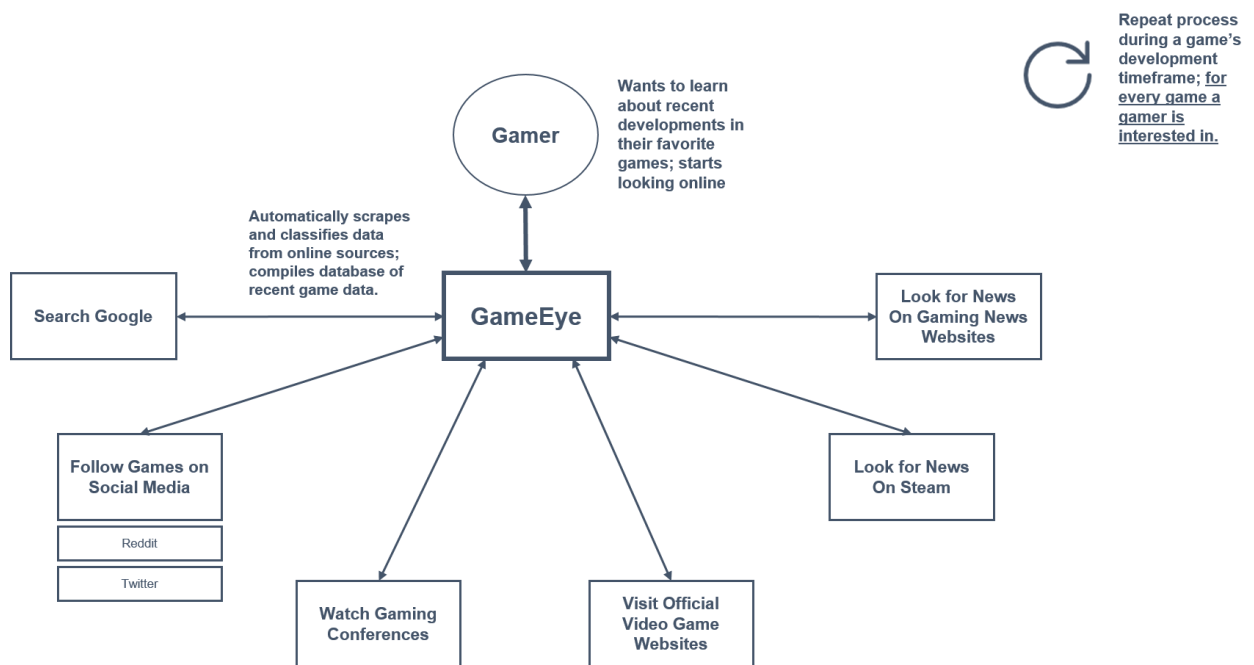


Figure 2: Solution Process Flow

Figure 2 represents the updated process flow with the use of GameEye; instead of the gamer having to constantly go back and forth between news sources to find information, GameEye handles the same workload while the gamer now only has to check a single source to find new information. This cuts out both the time and effort needed to stay up-to-date with a game's development as it currently stands.

General Features & Notifications

The user is able to build their own watch-list of games; these can be games that are either already released or still in development. Each game in a user's watch-list contains categorized information as represented in Figure 3. When a new piece of information is found, it is labelled

appropriately depending on information source and a notification is sent to the user when it's added into the database.



Figure 3: Information Categories Mock-Up

These categories include important updates such as changes in release dates or new game updates, news articles pertaining to the game, social media posts from development teams, official or community artwork, and videos on streaming sites such as trailers or ViDocs. When new content is discovered, notifications are sent to user through either push-notifications on mobile or email if the user specifies. On the application itself, the number of new notifications since last visited are shown next to each category.

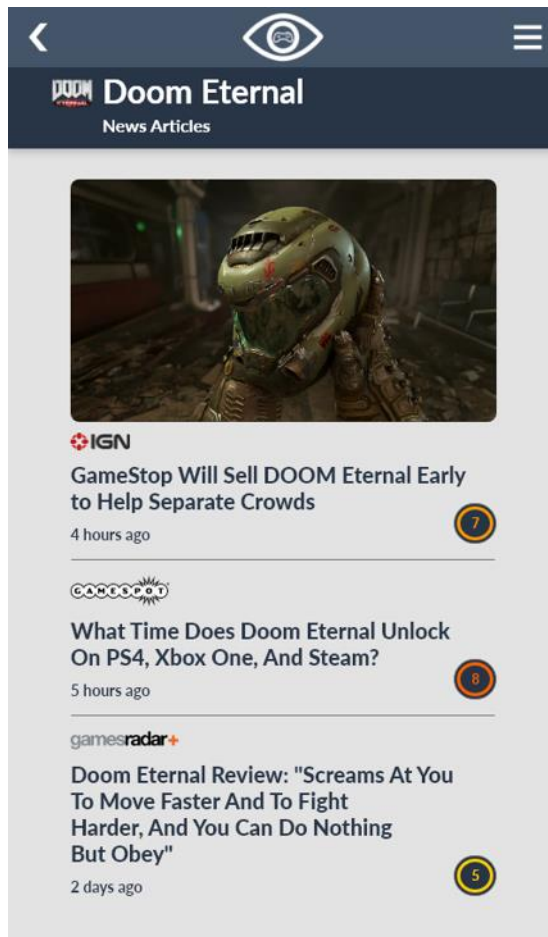


Figure 4: News Articles Mock-Up

Figure 4 shows how the articles are represented when visiting the ‘news articles’ category. If available, an image from the most recent article is provided for viewing above the articles listed, and each article is accompanied by the name of the news outlet. Each article is given a score based on its importance generated by machine learning algorithms, and directs the user to the original news source when interacted with. Any content in these categories that are over ninety days old will be removed from the listing and archived for later viewing if, for instance, the user is interested in older updates for a game they just bought or are revisiting.

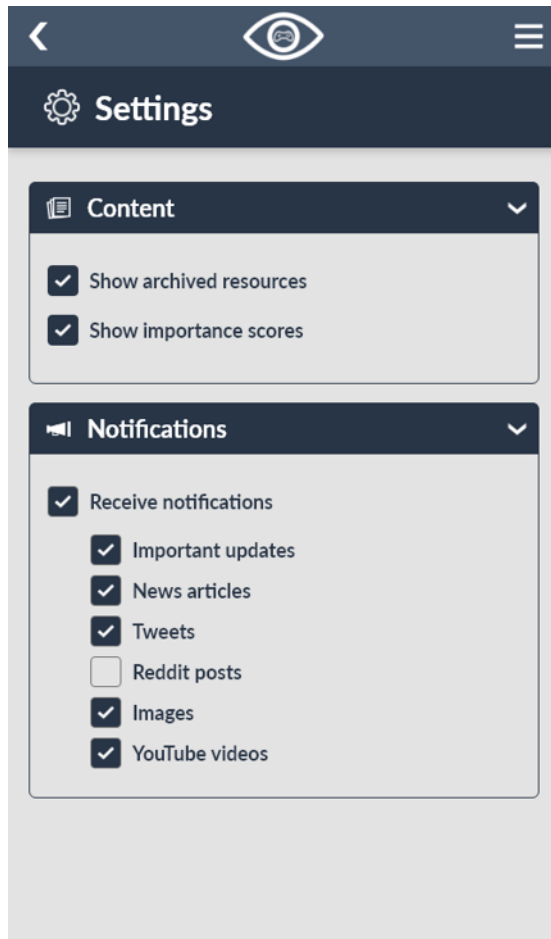


Figure 5: Settings Mock-Up

Shown in Figure 5, the user is able to modify notification settings to change what kind of notifications they get and the frequency in which they are notified. Users have the option to customize the notifications they receive from the different categories; and, if desired, users have the capability to enable or disable the scoring for incoming news and the ability to view archived information. Users can also customize which games from their watch-list they wish to receive notifications from alongside their frequency.

Authentication & Account Management

GameEye requires registering an account with an email and password in order to access its features. GameEye users take on one of two main roles: guest and registered user. A guest is

anyone that visits the application who is not registered; a guest is only able to register for an account and learn more about the GameEye application. Registered users are those who have registered an account and can begin curating their watch-list and receiving notifications.

Game Tracking & Searching

Games are tracked through the user's watch-list as represented in Figure 6. Through the watch-list, a user can see all the games they follow and if any new content is available for viewing; interacting with a title in the watch-list takes the user to the categorized information related to that game. The watch-list can also be customized to fit the user preferences such as sorting order and which games can send alerts.

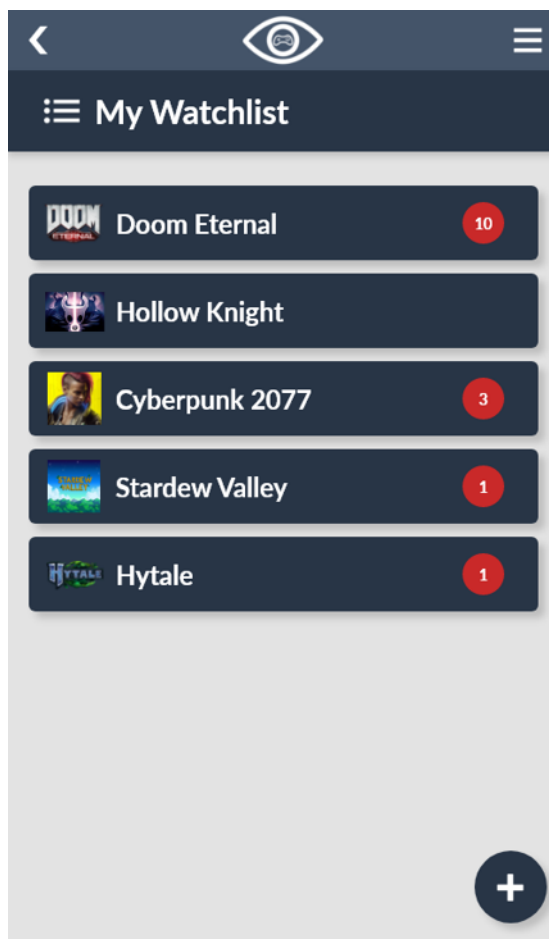


Figure 6: User Watch List Mock-Up

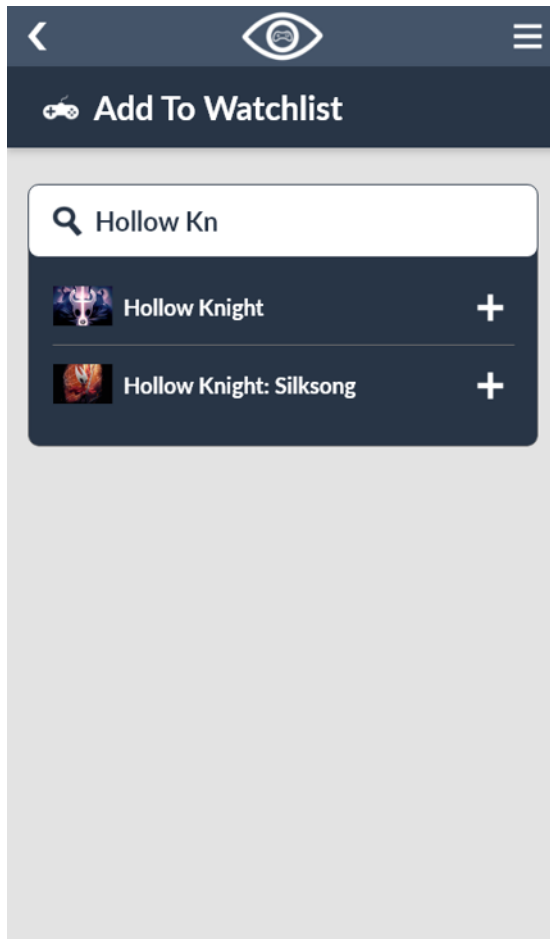


Figure 7: Searching Mock-Up

When it comes to searching for games to add to the watch-list, GameEye utilizes a search bar with auto-completion, as demonstrated in Figure 7, to find games and directly add them to the watch-list. When searching for a game, game titles and their respective information is retrieved from an existing video game database, such as the IGDB database or potentially a curated database of games. GameEye also provides a “hit list” of current popular games based on user activity to serve as suggestions for new users or users looking for something new to follow.

Web Scraping & Machine Learning

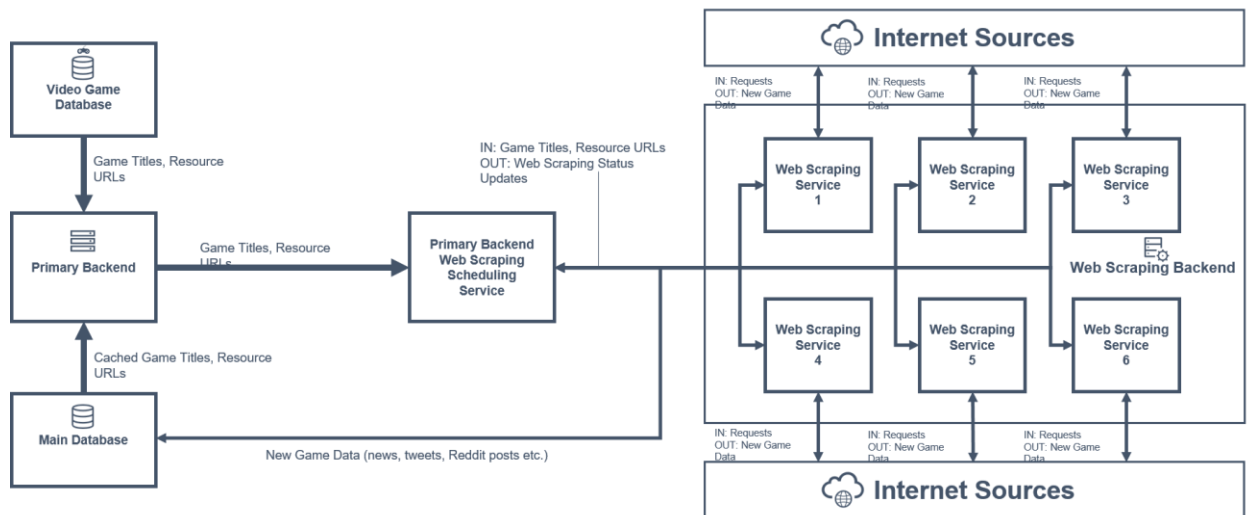


Figure 8: Web Scraping Flow Diagram

Figure 8 shows the flow diagram for how the web scraping works on GameEye's backend. Web scraping services are developed to monitor different sources of information across the internet such as news aggregators, social media, online store pages like Steam, and streaming services. When new information is found, the scraper is able to provide collectively the title of resource found, the text tied to the resource, the resource URL, a timestamp of when the resource was found, and any potential image tied to the resource back to the GameEye database to be used for processing; this process is represented in Figure 9. From there, the resource type will be determined and stored in the appropriate category under the related game.

[This space intentionally left blank]

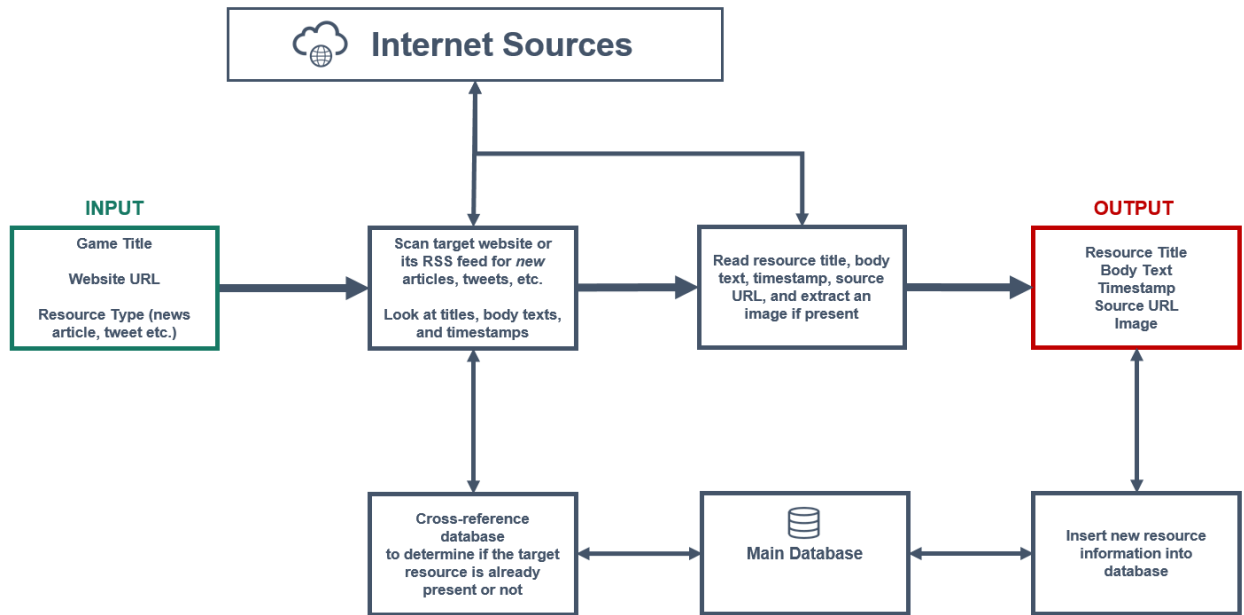


Figure 9: Web Scraping Service Algorithm

With the information provided by the web scrapers, GameEye will utilize machine learning algorithms to observe the resources and determine “importance” scores to help users decide which information they need to pay more attention to. These scores range from one to ten with one being of least importance and ten being of most importance. This scale is shown in Figure 10.



Figure 10: Importance Score Scale

Major Components

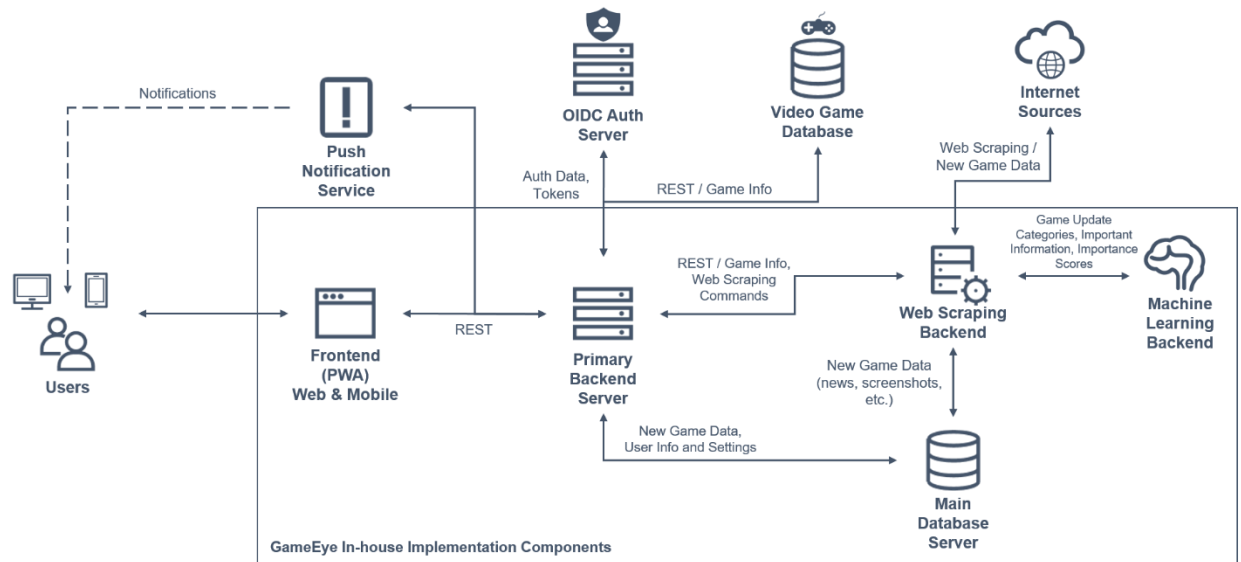


Figure 11: Major Functional Component Diagram

Figure 11 shows the Major Functional Component Diagram (MFCD) for GameEye. The main components between the frontend, backend, and main servers are the video game database to retrieve video game titles, the web scraping backend, and the associated machine learning backend.

Hardware

The hardware used by GameEye are the frontend server, main backend server, web scraping server, main database server, and machine learning backend server. The frontend server is an external-facing web server responsible for handling the frontend of the web application. The main backend is an external-facing web server exposing a REST API for communication between the frontend, web scraping backend, databases, and other third-party services. The web scraping server is also external-facing, but doesn't incoming traffic; this server also exposes a REST API for launching web scraping operations. The machine learning backend is an Internal GPU-powered "TensorFlow Serving" server exposing a REST API for machine learning

inference; it communicates directly with the web scraping server to extract information and classify data. The main database server is an internal MongoDB server that contains all data related to the GameEye application.

Software

Frontend. GameEye is a PWA that utilizes the WebStorm IDE, Angular Framework, and Google Workbox to handle the frontend work. The main programming languages used for the frontend are HTML, SASS/CSS, and TypeScript

Backend. The backend of GameEye uses the IntelliJ IDEA IDE, Spring Framework, jsoup library, and the MongoDB Java Driver. The main programming language used for the backend is Java.

Testing. Testing for GameEye and all its functions will be done in the JUnit Java Framework and the Jest JavaScript Framework.

Databases. MongoDB, MongoDB Compass and the IGDB REST API will be used to build and maintain the GameEye databases.

Third-Party Software. For third-party software, GameEye uses the Auth0 Single Page App SDK for authentication and Firebase Cloud Messaging to handle notifications on mobile devices.

Machine Learning. The software used to handle the machine learning algorithms are the Keras library, scikit-learn library, and a TensorFlow Serving model server. The primary programming language that GameEye uses for all forms of machine learning is Python.

Natural Language Processing. For natural language processing, GameEye will utilize the spaCy library which is used with Python.

Identification of Case Study

GameEye is intended for those who play video games either as serious hobby or casually. These gamers will want to stay informed about their favourite games throughout their development cycles; they will be updated and notified about new information as it's released. Small indie developers can be considered an indirect customer as they benefit from their games gaining more exposure and player retention over a longer period of time as they release new games or updates to existing titles.

Glossary

AAA: Classification of video games produced by a mid-sized or major game publisher that typically have large development and marketing budgets; analogous to “blockbuster” in film.

Angular Framework: Platform for building mobile and desktop applications.

API: Application Programming Interface; a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other services.

AWS: Amazon® subsidiary that provides on-demand cloud computing platforms and APIs

CSS: Cascading Style Sheets; used to stylize webpages.

Guest: Initial role for users who have not created an account on GameEye.

Hitlist: List of highly watched video games by users.

HTML: Hypertext Markup Language; used as markup for documents meant to be displayed in a web browser.

IGDB: Database of known video games, accessed by REST API to populate GameEye's database

Indie Games: Games developed by individuals or smaller teams of people without the financial support of larger game publishers.

IntelliJ Idea: IDE developed by JetBrains to write Java applications and will be used in the back-end development of GameEye.

JavaScript: Object-oriented language used to create dynamic, interactive effects on webpages.

Jest JavaScript Framework: Testing framework maintained by Facebook Inc.

JSoup Library: Java library for working with real-world HTML.

JUnit Java Framework: A testing framework for Java.

Keras (Python Deep Learning Library): Open-source neural-network library written in Python.

MongoDB: A cross-platform document-oriented database program

Noise Filtering: Information/news articles shown that caters to an individual's content preferences.

OIDC Authentication: Authentication protocol based on the OAuth2.0 family of specifications.

PWA: Progressive Web Application; a type of application software delivered through the web which is built using common web technologies including HTML, CSS, and JavaScript.

Python: Interpreted, high-level, general-purpose programming language.

REST: Software architectural style used in creating web services.

RSS Feed: Web feed that allows users and applications to access updates to websites in a standardized, computer-readable format.

Scikit-learn Library: Software machine learning library for the Python programming language.

SpaCy Library: Open-source software library for advanced natural language processing.

Spring Framework: Application framework and inversion of control container for the Java platform.

TensorFlow: Open-source software library for differential programming and dataflow; used in machine learning applications.

Tester: GameEye beta testers; users of the application in its prototype phase who will provide feedback on their experience.

Web Scraping: Data scraping for extracting data from websites.

WebStorm: IDE developed by JetBrains to write JavaScript code.

References

- Anderton, K. (2019, June 26). The Business Of Video Games: Market Share For Gaming Platforms in 2019. <https://www.forbes.com/sites/kevinanderton/2019/06/26/the-business-of-video-games-market-share-for-gaming-platforms-in-2019-infographic/#c1793427b254>
- Dietz, J. (2011, June 23). 30 Games That Emerged from Development Hell. <https://www.metacritic.com/feature/games-that-shed-vaporware-status>
- Gough, C. (2019, August 9). Number of games released on Steam 2018. <https://www.statista.com/statistics/552623/number-games-released-steam/>
- Gough, C. (2019, August 9). Number of gamers worldwide 2021. <https://www.statista.com/statistics/748044/number-video-gamers-world/>
- Gough, C. (2019, October 9). Google Play: Number of available games by quarter 2019. <https://www.statista.com/statistics/780229/number-of-available-gaming-apps-in-the-google-play-store-quarter/>
- Humphries, M. (2019, September 18). Twitch Acquires Gaming Database Website IGDB. <https://www.pcmag.com/news/twitch-acquires-gaming-database-website-igdb>
- Rose, M. (2014, May 15). How the surge of Steam releases will affect game developers. https://www.gamasutra.com/view/news/217583/How_the_surge_of_Steam_releases_will_affect_game_developers.php
- WePC. (2020, January 1). 2020 Video Game Industry Statistics, Trends & Data. Retrieved May 4, 2020, from <https://www.wepc.com/news/video-game-statistics/#video-game-monetization>