I. Introduction

A. Societal Problem

1. Thousands of new video games released every year
2. Video games often have long development timeframes, making it tiresome to stay updated for so long
3. Current methods for staying up to date about video games are time-consuming
4. Decentralized information and news about video games makes it more difficult to find the desired information
5. Independent game developers often have difficulty in maintaining public attention

B. Solution Description

1. A platform where users can search for and follow/watch games
2. By following a game, *GameEye* will keep track of it and notify users when new content is released online
3. Users will have a personal watchlist
4. Using machine learning, news articles and tweets be classified depending on what they are about
5. Using machine learning, news articles and tweets will be analyzed, and a multi-factor score will be computed and assigned based on the impact the news have on the reference game and players.

II. GameEye Product Description

A. Key Product Features and Capabilities

1. General

    a) Cross-platform support (desktop and mobile devices) based on progressive web application technologies

    b) Offline support, local caching, and connectivity interruption resiliency features

2. Authentication

    a) Secure user login and registration

    b) External provider user login and registration (e.g. with social media accounts)

    c) Persistent sessions so that users do not have to login every time

    d) Two-factor authentication for increased security

    e) Password recovery mechanisms in case users forget their passwords

3. Account Management

    a) The ability for users to change their passwords

    b) The ability for users to modify their profile information

    c) The ability for users to delete their accounts

4. Game Tracking

    a) Personal Watchlists

        (1) Each user has a personal watchlist of video games they want to stay updated about

        (2) Game thumbnails displayed next to game titles

    b) New Game Updates

(1) Organized by important updates, news articles, tweets, Reddit posts, images, and videos

(2) Thumbnails included for news articles and tweets

c) Most-Watched Games List

(1) Shows the most-watched games on the platform

5. Searching

a) Ability to search for video games based on their titles

b) Support for autocompletion

6. Web Scraping

(1) News articles from video game news websites

(2) Tweets from official game Twitter feeds

(3) Reddit posts from official game subreddits.

(4) Videos from official game YouTube channels

(5) Images

7. Notifications

a) Cross-platform push-notifications for new game updates

b) UI count of notifications for each game and each resource category

c) Suggested video game notifications based on the most-watched games list

8. Settings

a) Users choose if they want to see archived resources

b) Users choose if they want to see impact scores for news articles and tweets

c) Users choose if they want to receive notifications and for which resource categories to be notified about

d) Users can submit feedback

9. Machine Learning

a) Game update classification (release date announcement, delay, major game update, minor game update, etc.)

b) Impact scoring for news articles and tweets

c) Important information extraction for news articles and tweets

B. Major Components

1. Hardware

a) Frontend server

b) Main backend server

c) Web scraping backend server

d) Machine learning backend server

e) Main database server

2. Software

a) Frontend

(1) WebStorm IDE

(2) Angular Framework

(3) Google Workbox (PWA libraries)

(4) Languages

(a) HTML

(b) SASS/CSS

(c) TypeScript

b) Backend

(1) IntelliJ IDEA IDE

(2) Spring Framework

(3) jsoup library (web scraping)

(4) MongoDB Java Driver

(5) ElasticSearch Java Driver

c) Testing

(1) JUnit Java Framework

(2) Jest JavaScript Framework

d) Machine Learning

(1) Python

(2) Keras (deep learning library)

(3) scikit-learn library

(4) TensorFlow Serving (model server)

e) Natural Language Processing

(1) Python

(2) spaCy library

f) Databases

(1) MongoDB

(2) MongoDB Compass (MongoDB GUI)

(3) ElasticSearch

(4) IGDB REST API

g) 3rd Party Software

(1) Firebase Authentication

(2) Firebase Cloud Messaging (using Firebase Admin Java SDK)

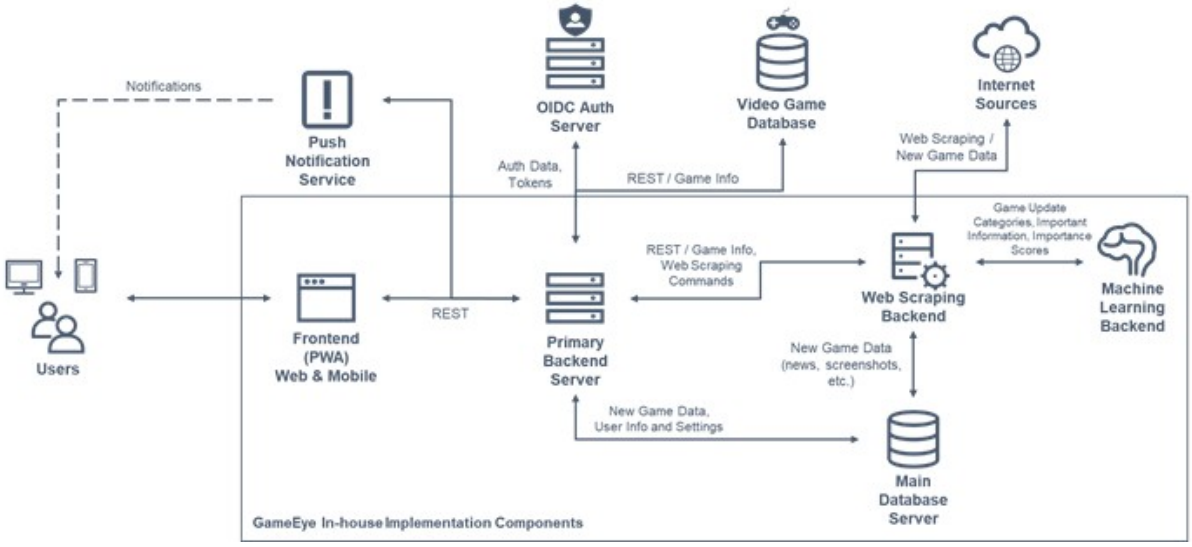

Figure 1: Major Functional Component Diagram

III. Identification of Case Study

A. Who is the product for?

1. Gamers

B. What will the product be used for?

1. Game update notifications
2. Game tracking

C. Who else might use it in the future?

1. Future gamers

IV. GameEye Product Prototype Description

A. Prototype Features and Capabilities

1. The prototype will provide:

a) Robust user authentication and external provider authentication support

b) Account management features for users to change their passwords and modify their profile information

c) Game searching with autocompletion

d) Personal game watchlists

e) Web scraping for news articles

f) Redirection to sources upon clicking scraped resources

g) Most-watched games list

h) Settings for toggling the visibility of archived resources, toggling the visibility of impact scores, and for notifications

i) Impact scoring of news articles using machine learning

j) Cross-platform push-notifications for new game updates and UI count of notifications

2. Provided features will demonstrate that GameEye *successfully*:

a) Scrapes data from multiple online sources

b) Offers robust authentication mechanisms for users

c) Offers robust game searching and fast autocompletion

d) Offers each user a personal watchlist

e) Offers a list where users can see the most-watched games on the platform

f) Offers customization capabilities and a variety of settings regarding content and notifications

g) Uses machine learning to score news articles based on the impact of the news for a game and gamers

h) Offers robust cross-platform support and push-notifications

3. Risk Mitigation

a) Scrape RSS feeds to protect against source website structure changes which would cause web scraping to fail

b) Store video game database content on our database for redundancy in case IGDB goes down

c) Make platform scalable, use load balancers and multiple instances of servers and databases to protect against high load

d) Cache content of user devices so that the application does not appear blank in case the main database fails; have more instances of the main database for redundancy

e) Use database encryption and 3rd party authentication provider to safeguard user personally identifiable information

f) Provide manual to help users understand the application

g) Use proper coding practices to prevent NoSQL injection, XSS, data exposure and broken authentication and access controls

B. Prototype Architecture

1. The prototype will utilize the same architecture as the real-world product except for the replicated servers for load balancing that would be present in the real-world product.

Category	Feature	RWP	Prototype
General			
	Cross-Platform Support (Desktop, Mobile)	Full Functionality	Full Functionality
	Offline Support	Full Functionality	Partial Functionality
	Local Caching	Full Functionality	Partial Functionality
	Connectivity Interruption Resiliency	Full Functionality	Partial Functionality
Authentication			
	User Login & Registration	Full Functionality	Full Functionality
	External Provider Login & Registration	Full Functionality	Full Functionality
	Persistent Sessions	Full Functionality	Full Functionality
	Two-Factor Authentication (2FA)	Full Functionality	No Functionality
	Password Recovery	Full Functionality	Full Functionality
Account Management			
	Change Password	Full Functionality	Full Functionality
	Modify Profile Information	Full Functionality	Full Functionality
	Delete Account	Full Functionality	No Functionality
Searching			
	Search For Games	Full Functionality	Partial Functionality
	Search Autocompletion	Full Functionality	Full Functionality
Game Tracking			
	Add Games To Watchlist	Full Functionality	Full Functionality

Figure 2: RWP vs Prototype Features (1)

Category	Feature	RWP	Prototype
	News Articles (Web Scraping)	Full Functionality	Partial Functionality
	Tweets (Web Scraping)	Full Functionality	Partial Functionality
	Reddit Posts (Web Scraping)	Full Functionality	No Functionality
	Images (Web Scraping)	Full Functionality	No Functionality
	Videos (Web Scraping)	Full Functionality	No Functionality
	Resource Thumbnails (Includes Website Logos)	Full Functionality	Partial Functionality
	Game Thumbnails	Full Functionality	Full Functionality
	Source Website Redirection	Full Functionality	Full Functionality
	Resource Organization	Full Functionality	Full Functionality
	Show Archived Resources	Full Functionality	Full Functionality
	Most Watched Games List	Full Functionality	Full Functionality
Settings			
	Show Archived Resources Option	Full Functionality	Full Functionality
	Show Importance Scores Option	Full Functionality	Full Functionality
	Receive Notifications	Full Functionality	Full Functionality
	Receive Notifications Per Category	Full Functionality	Full Functionality
	Submit Feedback	Full Functionality	No Functionality
Machine Learning			
	Importance Scoring	Full Functionality	Partial Functionality
	Resource Classification	Full Functionality	Partial Functionality
	Important Information Extraction	Full Functionality	No Functionality

Figure 3: RWP vs Prototype Features (2)

Category	Feature	RWP	Prototype
Notifications			
	Push-Notifications For New Game Updates	Full Functionality	Full Functionality
	UI Count of Notifications For Each Game	Full Functionality	Full Functionality
	UI Count of Notifications For Each Resource Category	Full Functionality	Full Functionality
	Cross-Platform Notifications	Full Functionality	Full Functionality
	Suggested Video Game Notifications	Full Functionality	No Functionality

Figure 4: RWP vs Prototype Features (3)

C. Prototype Development Challenges

1. Learning new technology and frameworks required for implementation
2. Collecting and labeling data for machine learning models
3. Properly securing communication between the multiple backends
4. Implementing robust caching of frontend content for connectivity interruption resiliency and performance
5. Implementing robust web scraping with future-proof mechanisms in case source websites change

V. Glossary

List of terms and abbreviations:

***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

Impact Score: A score from 1 to 3 on the impact some news has on a game and its players. It is computed using machine learning.

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.

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.

VI. References

Anderton, K. (2019, June 26). The Business Of Video Games: Market Share For Gaming Platforms in 2019 [Infographic]. 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