Lab 2 - GameEye Product Specification Outline

Jacob L. Cook

Old Dominion University

CS411W

Professor Janet Brunelle

November 22, 2020

Version 2

Table of Contents

List of Figures

List of Tables

1 Introduction

Video games are one of the most popular forms of entertainment worldwide. According to Gough (2019a), the rate at which video games are released worldwide on Steam, a prominent video game platform, is increasing exponentially. Gough (2019a) provides data indicating that, worldwide, the number of new video games released on Steam grew from 4,207 new games in 2016 to 9,050 new games in 2018. In addition to the increasing video game release rate, the community of active gamers is expanding significantly. Gough (2019b) predicts that, by 2021, the number of video gamers worldwide will surpass 2.7 billion. For new and experienced gamers alike, the expanding library of video games is overwhelming.

The large number of video games and long development timeframes increase the difficulty for gamers to find new desirable games. Gamers struggle to maintain awareness throughout the development of new video games. There is great variation in the number of years from the date a game is announced to the date a game is released. Gamers are negatively impacted by long development timeframes for video games. Dietz (2011) illustrates that a prime example of a game with a long development timeline is Team Fortress 2, which was announced in 1998 and eventually released in 2007. While under development, games have distinct statuses, such as “announced,” “delayed,” and “released.” For games with extensive development timeframes, gamers face the difficulty of staying apprised of new information and status updates. GameEye is a software-based solution to solve the problems facing many video gamers.

1.1 Purpose

GameEye is a platform for consumers to access game news and updates from a central location. Users can search for their favorite games and curate a personal watchlist. Each user can add a game they wish to follow to their watchlist. Once a game is added to the user’s watchlist, they will receive notifications when GameEye finds new related content. Rather than

bombarding users with all information related to a game, machine learning classifies news articles and Tweets into specific content categories. In addition to content categorization, each news resource is prioritized by a multifactor score computed by machine learning. GameEye informs users of the most popular games currently added to other watchlists. The target community consists of gamers who struggle to track video game news. By centralizing game news, providing users with personal watchlists, and ranking information based on impact, GameEye is the gamer's eye into the video game industry.

1.2 Scope

GameEye's most important goals are to provide personalized user watchlists and notify users about new video game content, such as news articles. Machine learning is used to extract important information from news articles. Personal watchlists are populated with video game titles that users find within the GameEye search feature. Users receive notifications when new content is found for a game in their watchlist. In addition, video game content is ranked by impact to eliminate irrelevant information. GameEye's design leverages progressive web application (PWA) technologies – enabling usage on desktop and mobile devices. PWA architecture uses common web technologies to deliver the same user experience on any device with a modern web browser.

The GameEye prototype will demonstrate proof of concept by implementing the core product features. Users will have the ability to authenticate to the GameEye frontend via third-party services including Google and Microsoft. Video game data will be largely derived from actual entries in IGDB and populated into the MongoDB database. To complement raw video game data from IGDB, video game news articles will be retrieved via web scraping. News sources will include prominent video news websites and a mock news website curated for this prototype demonstration. Users will be able to search for video game titles with the auto-

completion feature supported by Elasticsearch. Users will be able to perform account management tasks, such as changing passwords and updating profile information.

In the prototype, each user will have a personal video game watchlist. User watchlists enable tracking of released and unreleased video games titles. Users will be able to configure their account settings such as visibility of archived resources, visibility of impact scores, and notification preferences. The prototype will provide web scraping capabilities in the backend for obtaining news article data. Many of the challenges from the real-world product version of GameEye are mitigated by a reduced number of product features. The architecture, including the MFCD, of the GameEye prototype is still functionally equivalent to the real-world product.

1.3 Definitions, Acronyms, 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 Web Services): Amazon® subsidiary that provides on-demand cloud computing platforms and APIs

CSS: Cascading Style Sheets; used to stylize webpages.

Elasticsearch: Search and analytics engine that integrates with schema-less (e.g., MongoDB) database systems. Based on the Lucene library and provides an HTTP web interface.

Firebase Authentication: Backend authentication services and software development kits (SDKs) for integrating authentication into applications.

Google Workbox: A set of JavaScript libraries for offline support in PWA.

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.

IDE (Integrated Developer Environment): Application providing a set of tools and resources for programmers. Includes source code editors, automation tools, compilers, and debuggers.

IGDB (Internet Games Database): Database of known video games: accessed by REST API to populate GameEye's database.

Impact Score: A score from 1-3 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: The act of removing data that is determined to be "noise" (i.e., subjectively irrelevant data) to suit 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 (Representational State Transfer): Software architectural style used in creating web services.

RSS Feed (Really simple syndication): web feed that allows users and applications to access updates to websites in a standardized, computer-readable format.

SASS/CSS (Syntactically Awesome Style Sheets/Cascading Style Sheets): an extension of CSS used for more robust development of style sheets for web design.

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.

TypeScript: An open-source language which augments the capabilities of JavaScript. The language adds static type definitions and is compiled into JavaScript via the TypeScript compiler or Babel compiler.

Web Scraping: Data scraping for extracting data from websites.

WebStorm: An IDE developed by JetBrains to write JavaScript code.

1.4 References

Anderton, K. (2019, June 26). Video game market share and growth. [Infographic]. *Forbes.* https://www.forbes.com/sites/kevinanderton/2019/06/26/the-business-of-video-games-market-share-for-gaming-platforms-in-2019-infographic/#600d6dd97b25

Angelopoulos, A., Cook, J., Diasanta, C., Epps, J., Hund, B., Lewis, B., Wilson, A. (2020). *Lab 1 – GameEye Product Description.* https://www.cs.odu.edu/~411yello/pages/labs.html

Dietz, J. (2011, June 23). *30 Games that emerged from development hell.* Metacritic. https://www.metacritic.com/feature/games-that-shed-vaporware-status

Gough, C. (2019, August 9). *Number of games released on Steam 2018.* Statista. https://www.statista.com/statistics/552623/number-games-released-steam/

Gough, C. (2019, August 9). *Number of gamers worldwide 2021.* Statista. https://www.statista.com/statistics/748044/number-video-gamers-world/

Gough, C. (2019, October 9). *Google Play: Number of available games by quarter 2019.* Statista. https://www.statista.com/statistics/780229/number-of-available-gaming-apps-in-thegoogle-play-store-quarter/

Humphries, M. (2019, September 18). Twitch acquires gaming database website IGDB. *PCMag.* 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.* Gamasutra. https://www.gamasutra.com/view/news/217583/How_the_surge_of_Steam_releases_will_affect_game_developers.php

Web, K (2019, Oct 1). *The $120 billion gaming industry is going through more change than it ever has before, and everyone is trying to cash in*. Business Insider. https://www.businessinsider.com/video-game-industry-120-billion-future-innovation-2019-9

1.5 Overview

The remaining sections of this product specification outline provide the prototype architecture, prototype functionalities, software interfaces, and requirements of the GameEye prototype. This specification outline describes the purposes and relationships of the components found in the MFCD. Frontend and backend components and functionalities are described in sections 2.1 and 2.2.

2 General Description

The GameEye prototype is comprised of the primary components found in the real-world product. Primary components of the prototype include *Primary Backend Server*, *Main Database Server*, and *Frontend (PWA) Web & Mobile*. The frontend components provide functionalities to the GameEye users. Backend components support the frontend with functionalities such as web scraping and database interfacing. The primary components provide core functionalities including video game data collection and storage, machine learning impact score computation, and watchlist interfacing. GameEye uses several software interfaces such as MongoDB for storing news articles and video game data. IGDB is an external software interface for collecting game data. Firebase Authentication is used for user account access and Firebase Cloud Messaging for notification services.

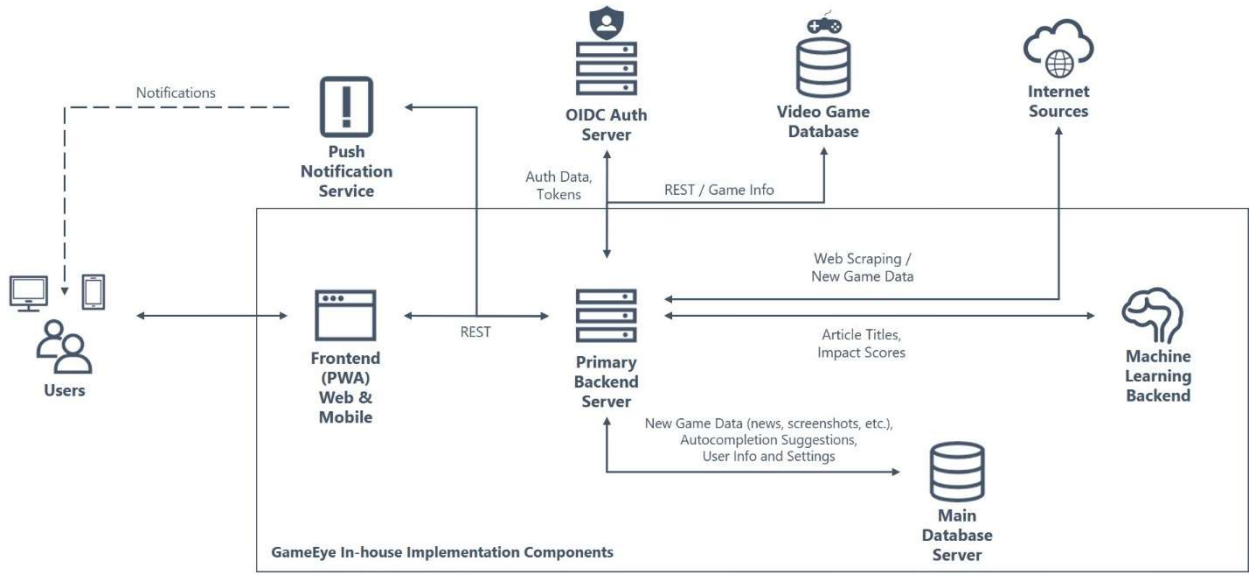
2.1 Prototype Architecture Description

GameEye provides a frontend user interface consisting of a single-page cross-platform web application. As shown in Figure 1, the *Frontend (PWA) Web & Mobile* component connects to the *Primary Backend Server* component via a REST API. User authentication is handled with auth data tokens by the *OIDC Auth Server* component, which integrates with the Primary Backend Server. OIDC Auth Server uses Firebase authentication from Google for account

management. Users receive notifications via the *Push Notification Service* component, which uses Firebase Cloud Messaging. The frontend components are supported by Angular.

Figure 1

Prototype Major Functional Component Diagram (MFCD)



Backend implementation includes components that support the frontend interactions with video game data. The *Main Database Server* component depicted in Figure 1 is responsible for storing GameEye video game data. MongoDB will be the primary database technology for the Main Database Server component, and the database will be populated with data from IGDB. The frontend retrieves new video game data to serve to users from the Main Database Server. GameEye provides the frontend with search autocompletion via the Elasticsearch database, which is contained in the Main Database Server. Impact scores are computed by the *Machine Learning Backend* and stored in the Main Database Server. Backend components are implemented with Spring Boot.

2.2 Prototype Functional Description

Backend components will provide functionalities including data collection and storage, video game title searching, web scraping, machine learning, user notification generation, and supporting the frontend components. The backend will retrieve and process video game data from IGDB. Relevant data will be stored in the Main Database Server using MongoDB. News article data will be collected from Internet sources via web scraping services contained within the Primary Backend Server. For the purposes of the prototype, the mock news website is scraped. The mock news website is pre-populated with realistic video game news articles. Scraped news articles will be classified based on relative impact for the referenced game by the Machine Learning Backend.

Table 1

Prototype Features Table 1 of 2

Feature	Description	Implementation
	General	
Cross-Platform Support (Desktop, Mobile)	Ability to use GameEye on desktop and mobile devices.	Full Functionality
	Authentication	
User Login	Access an existing account on GameEye.	Full Functionality
User Registration	Create an account on GameEye.	Full Functionality
External Provider Login & Registration	Login with a Google or Microsoft account. If logging in for the first time on GameEye, an account is automatically created.	Full Functionality
Persistent Sessions	Access account without having to log in again after closing GameEye without logging out and reopening it.	Full Functionality
Password Recovery	Send a link to the user's email address that allows them to reset their password.	Full Functionality
	Account Management	
Change Profile Information	Allow users to change their name and email address.	Full Functionality
Change Password	Allow users to change their password while logged in.	Full Functionality
	Searching	
Search for Games	Allow users to search for video games to add to their watchlist.	Partial Functionality: searching mechanisms will be fully functional but not all games will be available
Search Autocompletion	Search results appear based on characters typed in the watchlist search bar (e.g. Hollow Knight and Hollow Knight: Silksong appear when "Hollow Kn" is typed in the search bar).	Partial Functionality: autocompletion mechanisms will be fully functional but not all games will be available

Table 2

Prototype Features Table 2 of 2

Feature	Description	Implementation
	Game Tracking	
Add Games to Watchlist	Allow users to add games to their watchlist.	Partial Functionality: Not all games will be available
Remove Games from Watchlist	Allow users to remove games from their watchlist.	Full Functionality
News Articles (Web Scraping)	Web scraping used to obtain news articles about video games by scraping popular gaming news websites.	Partial Functionality: not all intended news websites will be scraped
Resource Thumbnails (Includes Website Logos)	Display thumbnails for various resources such as news articles. These are scraped images. News website logos are also collected and displayed next to articles.	Partial Functionality: not all resources will be implemented, only news articles. Not all news websites will be used.
Game Thumbnails	Display images that represent a game. Shown in the watchlist, title bar, and search results.	Partial Functionality: not all games will be available
Source Website Redirection	Redirect a user to the official news article page when clicking on a news article inside GameEye.	Partial Functionality: not all news websites will be available
Resource Organization	The various resources will be organized by they type (e.g. news articles, tweets, etc.).	Partial Functionality: only news websites will be available
Most-Watched Games List	A list of the most-watched games by GameEye users.	Full Functionality
	Settings	
Show Impact Scores Option	Users can choose whether or not they want to see the impact scores of a resource.	Partial Functionality: only for news websites
Impact Score Levels Option	Users can choose for which impact scores to receive notifications for.	Partial Functionality: only for news websites
Receive Notifications Option	Users can choose whether or not they want to receive notifications.	Full Functionality
	Machine Learning	
Impact Scoring	Machine learning is used to give a score to a resource based on how impactful it is.	Partial Functionality: only for news articles, not enough training data to get the desired accuracy
	Notifications	
Push-Notifications for New Resources	Users will receive push-notifications when new resources have been scraped.	Partial Functionality: only for news articles
UI Count of Notifications for Each Game	An indicator showing how many unseen notifications there are for each game in a user's watchlist.	Full Functionality
UI Count of Notifications for Each Resource Category	An indicator showing how many unseen notifications there are for each resource category.	Partial Functionality: only for news articles
Cross-Platform Notifications	Notifications will be received in both the desktop and mobile versions of GameEye.	Full Functionality

The backend facilitates user searches with autocompletion assistance for video game titles. When users add a game to their watchlist, the frontend will trigger the backend to add game titles to the user's database entry. The notification service will send users notifications when new articles are found. All communication between the backend and frontend is provided by a REST API. For the purposes of the prototype, utilities for developers and testers will be implemented.

Frontend components will provide end-user functionalities for account creation, account management, login, watchlists, notifications, profile settings, and content preferences. The login and account creation page will allow users to register for the GameEye web application. The frontend will facilitate authentication via email and password, Google accounts, and Microsoft accounts. Separate pages will be generated for adding new games to a watchlist, viewing watched games, and accessing resources for a watched game. A page will be provided for users to change their profile settings and notification preferences.

2.3 Software Challenges

GameEye is ambitious and presents many development challenges. For instance, the external video game data retrieved from IGDB is challenging. Developers must use an API to retrieve the IGDB data, however, that API was recently updated. Future API updates may impact the development of the GameEye prototype. In addition, IGDB data will be updated in MongoDB frequently and updating thousands of game entries introduces the risk of data corruption if not properly tested. Elasticsearch, which is used for search autocompetition poses another challenge since it will be storing a subset of the data in MongoDB and must be kept up to date. In total, GameEye requires that the IGDB, MongoDB, and Elasticsearch databases all function properly together. Integration of the frontend and backend components also presents another challenge. Video game data from IGDB and web scraping is stored in the backend database, however, this

data must be delivered to users in an aesthetic way. While proper testing can mitigate many issues, GameEye presents many software challenges.