Lab 2 – Product Specification Outline

Brenden Lewis

Old Dominion University

CS411W

Professor Janet Brunelle

18 October 2020

Version 1

Table of Contents

List of Figures

1 Introduction

Given the scale of the web and how far information can be spread, it can be difficult for avid gamers to follow news for their favorite games. Thousands of games are released annually (Gough, 2019a; Gough, 2019b), with some of those games spending up to nine years in development (Dietz, 2011). Information regarding game development is often decentralized and widespread across many different news sites, developer blogs, streaming platforms, and social media. This spread of information can make it both difficult and time-consuming for gamers to find new information for games in development. A solution for organizing and discovering video game news is needed.

1.1 Purpose

GameEye is a progressive web application (PWA) for computers and mobile devices that allows users to craft a personal watch-list of their favorite games and receive notifications when any news or updates for those games is released. GameEye is able to scrape news sources over the internet to find news about specified games and consolidate it all in one location for convenient viewing. This is handled through the use of web scrapers to collect links to game-related news from various sources of game-centric news aggregators. Machine learning is utilized to classify these news sources using a multi-factor scoring system to determine the importance of news updates for use in filtering unimportant content and notification customization. Users are able to easily search for and add games, both current and unreleased, to their personal, customizable watch-list.

The purpose of GameEye is to provide both active and casual gamers easier means to keep up to date with their favorite games by notifying them of relevant news. GameEye is intended to compliment other news sources and game media by aggregating their news, not replace them. Each article preview shown by GameEye redirects users back to the news source it originated as to provide deserved web traffic to the source material. News information is collectively scraped twice every day from multiple sources and users are notified about any relevant information.

1.2 Scope

GameEye is a news aggregator that uses machine learning to notify users of video game news that may be relevant to them. Machine learning is utilized to classify news articles and assign “impact” scores to scraped news to signify how important the news is to the user. Users can create personalized watch-lists with the games they want to follow and receive notifications when new news content is available for viewing. The goal of GameEye is to be a single location gamers can access to get regularly updated on video game news information.

The GameEye prototype will demonstrate the major functions of the real-world product. These functions include watch-list construction, scraping for game news articles, impact scoring for scraped articles, account creation and management, and database construction. The biggest difference in functionality between the prototype and the real-world prototype is the scope of resources available to users. The prototype will only feature news information from a few online news sources for a limited number of games, while the real-world product will also provide Twitter posts, YouTube videos, and Reddit posts as sources and a much larger collection of games from the Internet Games Database (IGDB).

1.3 Definitions, Acronyms, and Abbreviations

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 Web Services; 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.

Hit List: 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: Internet Games Database; 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: OpenID Connect; 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.

[This space intentionally left blank]

1.4 References

Anderton, K. (2019, June 26). The Business Of Video Games: Market Share For Gaming Platforms in 2019. *Forbes.* 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*. Metacritic. https://www.metacritic.com/feature/games-that-shed-vaporware-status

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

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

Gough, C. (2019, October 9*). Google Play: Number of available games by quarter*. Statista. 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. *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

WePC. (2020, January 1). *2020 Video Game Industry Statistics, Trends & Data*. WePC. Retrieved May 4, 2020, from https://www.wepc.com/news/video-game-statistics/#video-game-monetization

1.5 Overview

This product specification details the software components, hardware components, external interfaces, features capabilities of the GameEye prototype. The remaining sections of the document include detail descriptions of the key software and hardware components utilized in implementing the GameEye prototype; the key features that the prototype will demonstrate including any of its limitations; and the interfaces that will be provided to and from other software by GameEye.

[This space intentionally left blank]

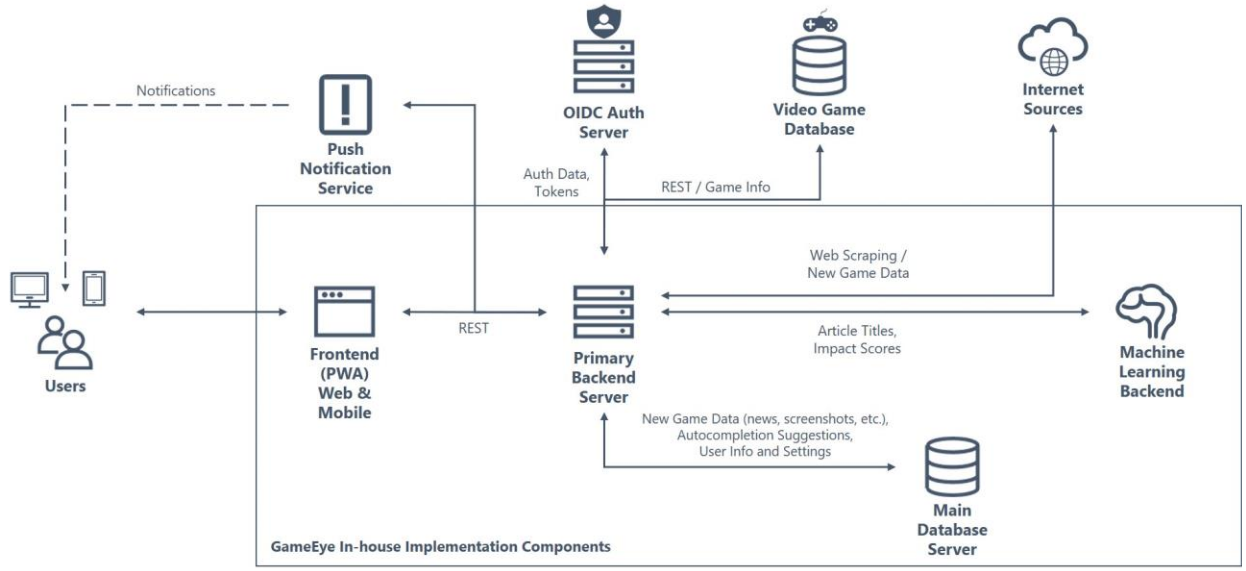
2 General Description

2.1 Prototype Architecture

The GameEye prototype is comprised of the following major components which are highlighted in Figure 1:

Figure 1:

Prototype Major Functional Component Diagram


Notifications
Push Notification Service
OIDC Auth Server
Video Game Database
Internet Sources
Auth Data, Tokens
REST / Game Info
Web Scraping / New Game Data
Users
Frontend (PWA) Web & Mobile
REST
Primary Backend Server
Article Titles, Impact Scores
Machine Learning Backend
New Game Data (news, screenshots, etc.), Autocompletion Suggestions, User Info and Settings
Main Database Server
GameEye In-house Implementation Components

- Frontend: The GameEye frontend is a cross-platform web application with a user interface. The frontend is able to communicate with the primary backend using its REST API. The frontend component also connects to all the Firebase services required by the prototype and is built using Angular.
- Primary Backend: The GameEye primary backend provides a REST API for the frontend and is used to orchestrate all web scraping for the prototype. The primary backend communicates with the main database, machine learning backend, ElasticSearch backend, IGDB, and Firebase services and is built using Spring Boot to handle web applications.

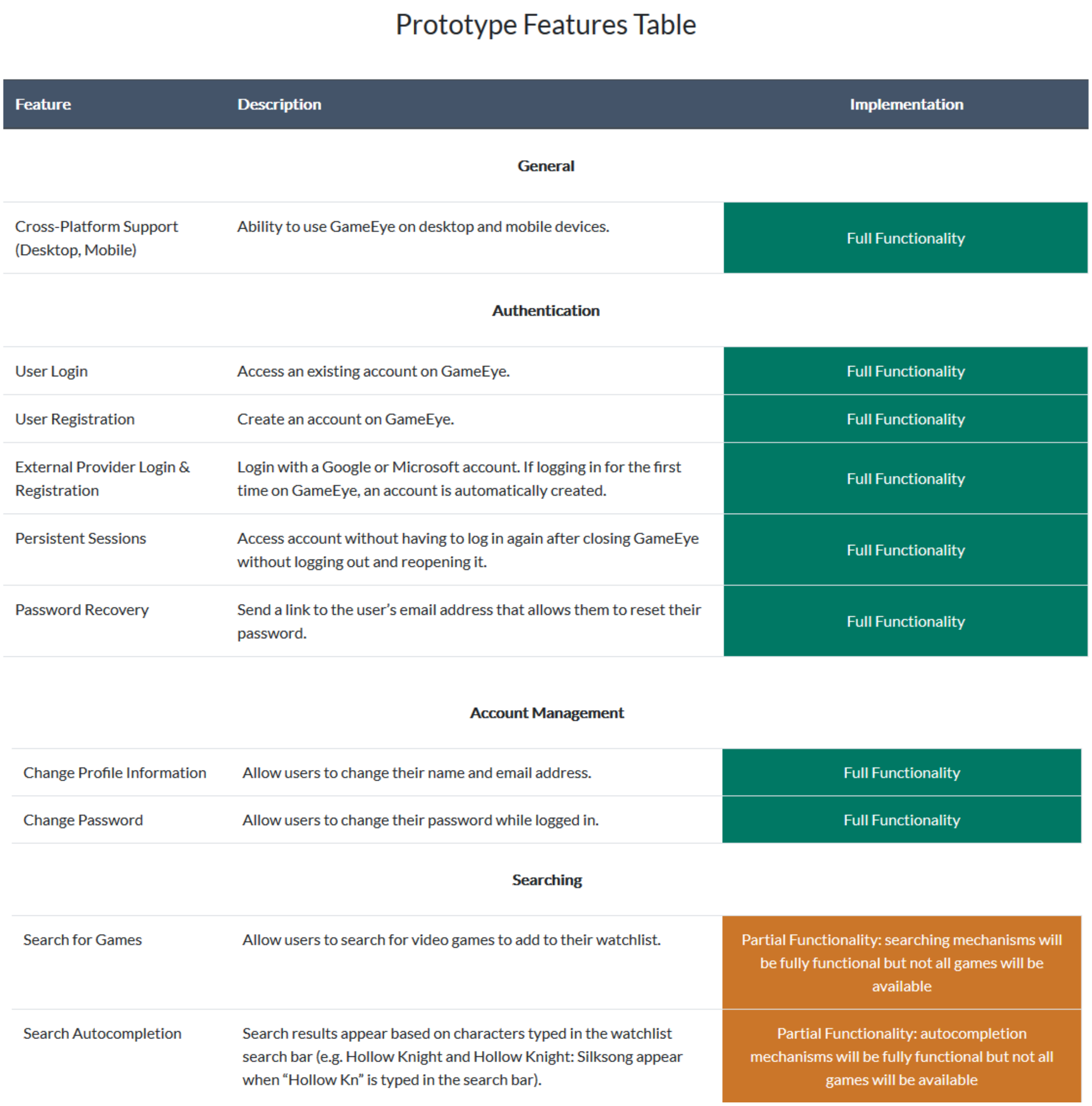
- Machine Learning Backend: The GameEye machine learning backend is a server that is responsible for the prototype's impact scoring functionality for scraped news articles. A REST API is also provided for the backend to interact with and is built using TensorFlow Serving, which is serving system used for designing machine learning models.
- Main Database: The prototype's main database is constructed and managed in MongoDB. The main database contains all the data related to the GameEye prototype; this includes all data used to direct scrapers and store their contents, user data, and game data.
- ElasticSearch Database: The GameEye prototype utilizes an ElasticSearch database to store game titles and allows for efficient searching of stored titles. This database is the basis for GameEye's auto-completion search functionality.
- Firebase Authentication: The GameEye prototype uses Firebase authentication to authenticate users when signing into the application.
- Firebase Cloud Messaging: The GameEye prototype utilizes Firebase cloud messaging to allow for push-notifications within the application.

2.2 Prototype Functional Description

The major functions of the GameEye prototype can be broken down into the frontend and backend components. The backend components handle all forms of collection, processing, and storage of data from web scrapers in the primary backend, machine learning backend, and main database. The frontend components work to provide users with the physical pages they can interact with and allows for storing user information in the main database. Figure 2 below shows the major features present in the GameEye prototype.

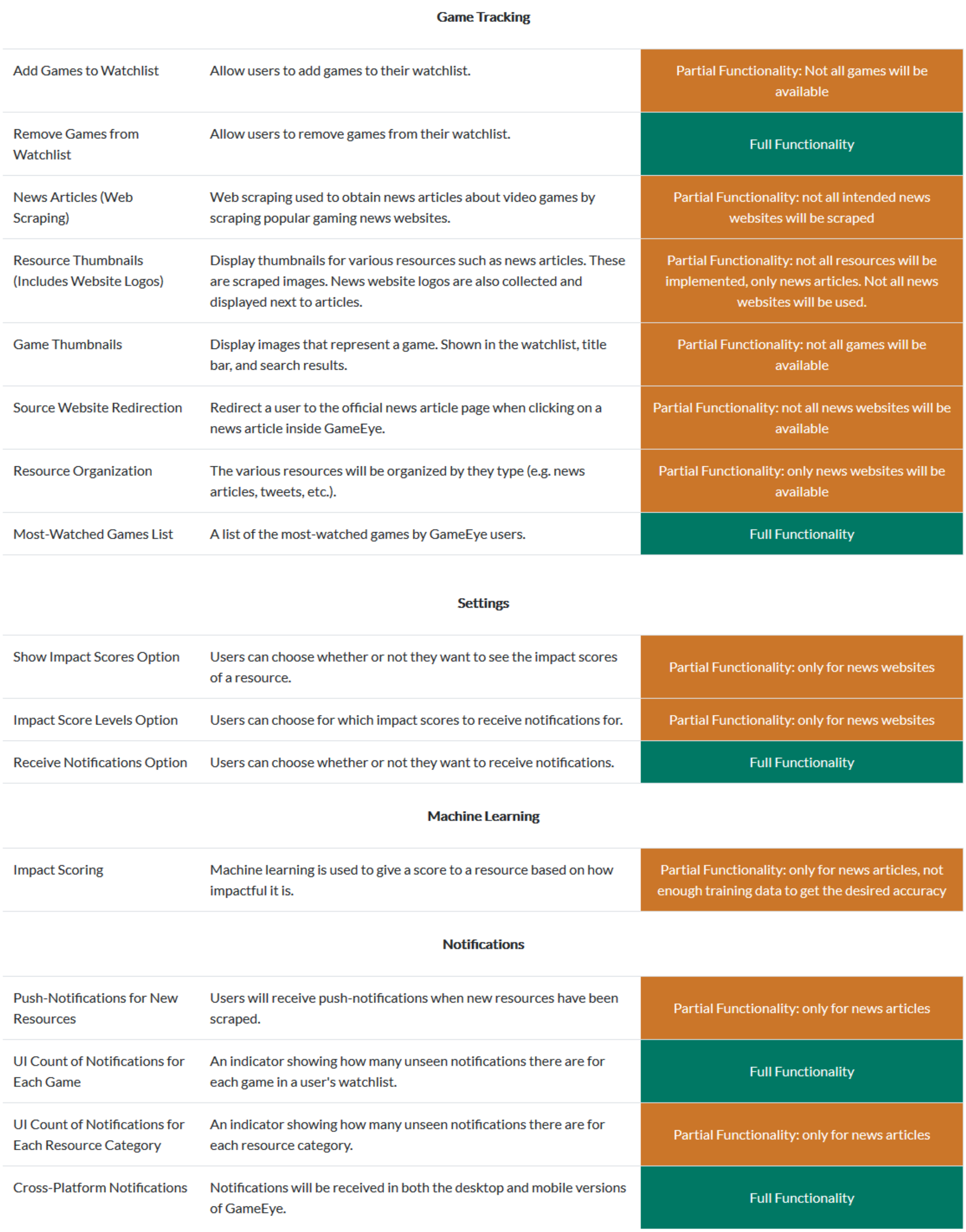
Figure 2:

Prototype Features Table



Prototype Features Table

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



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 prototype's backend components are responsible for collecting metadata for video games such as release dates, game descriptions, available platforms, and media in the IGDB to be stored in the main database. There is also support for searching for games within the database using ElasticSearch. These components are utilized by the web scrapers to regularly scrape news websites and properly store news articles within the main database; this entails correctly identifying the game an article refers to, ensuring the scraped article is not already present in the database to prevent duplicate storage, and ensuring that the article explicitly mentions a game within the title to prevent storage of non-game-related news. The machine learning backend is then called by the scrapers to compute impact scores for their collected articles using machine learning. A REST API is provided from the backend to the frontend to allow for communication and data transfer between the two. In order to prevent unauthorized users from accessing the backend, role-based endpoint authorization is enforced.

The frontend components collaborate to form the UI the users see when using the application. The prototype will provide a number of pages users can interact with to utilize the features of the prototype. There is a login page where users can create an account and sign in using an existing Google or Microsoft account. There is a watch-list page where users can view their watch-list and search for new games to add to it. There is a page that allows users to view different kinds of resources and new articles for a game. There is a settings page where users can customize account and notifications settings; based on these notification preferences, push notifications will appear when new articles are added for a game in their watch-list.

2.3 Software Interfaces

The GameEye prototype utilizes several software interfaces in its implementation. MongoDB is a database program that houses the main GameEye database which is used for storing user information and game information collected from the primary backend and IGDB. ElasticSearch is a search engine used by GameEye to search for games stored in the database for auto-completion search in the frontend and web scraper usage for data storage in the backend. The IGDB API is used to collect metadata on existing video games from IGDB to be stored in the main database by the primary backend; IGDB contains information for hundreds of thousands of games that can potentially be available for GameEye aggregation. Firebase Authentication and Cloud Messaging are Google services that allow for user authentication and push notifications within the GameEye application.