# Project1 Requirements & Design

## Project Requirements

* Project 1 will be a Scala console application that is retrieving data using Hive or MapReduce. Your job is to build a real-time news analyzer. This application should allow users to view the trending topics (e.g., all trending topics for news related to "politics", "tv shows", "movies", "video games", or "sports" only [choose one topic for project]).
* You must present a project proposal to trainer and be approved before proceeding with project.

### MVP

* ALL user interaction must come purely from the console application.
* Hive/MapReduce must:
  + Scrape data from datasets from an API based on your topic of choice.
* Your console application must:
  + Query data to answer at least 6 analysis questions of your choice.
  + Have a login system for all users with passwords.
    - 2 types of users: BASIC and ADMIN.
    - Users should also be able to update username and password.
* Implement all CRUD operations.
* Implement bucketing and partitioning.
* Can use hive with screenshots but make as program in IntelliJ (or VSCode) too with appropriate dependencies.

### Stretch Goals

* Passwords must be encrypted.
* Export all results into a JSON file/can optional because (changes done).
* Find a trend.

### Presentations

* You will be asked to run an analysis using the console application on the day of the presentation, so be prepared to do so.
* We'll have 5-10 minutes a piece, so make sure your presentation can be covered in that time, focusing on the parts of your analysis you find most interesting.

### Technologies

* Hadoop MapReduce
* YARN (by default)
* HDFS
* Scala 2.11 (or 2.12)
* Hive
* Git + GitHub

### Due Date

* Presentations will take place on Monday, 1/3.

## Proposal

* I will be fetching video game news data using the API through the site https://www.gamespot.com/api/, focusing on the following endpoints:
  + **http://www.gamespot.com/api/articles/?api\_key=[YOUR API KEY]**
  + **http://www.gamespot.com/api/games/?api\_key=[YOUR API KEY]**
  + **http://www.gamespot.com/api/reviews/?api\_key=[YOUR API KEY]**
* The six analytical questions I want to answer are as follows:
  1. What are the most-mentioned games in the last 10 years per year per month?
  2. What are the least-mentioned games in the last 10 years per year?
  3. Which games’ news involves the word “cheat”?
  4. How many games, articles, and reviews would we remove from a given period?
  5. How does a game’s reviews look across the 1-10 spectrum 1, 5, 15, and 30 days out after a news article is released?
  6. What new games have been released, and what games have received new reviews and articles, and how many new reviews and articles, from a given date?

## Functional Requirements

* Connect via Hive.
  + Perform all CRUD operations.
* Connect to Gamespot API.
  + Handle all status codes appropriately.
  + Handle timing.
* Provide command-line interface that allows the following functionality:
  + Log into application service.
    - Register a new account.
      * Prompt for new username.
      * Prompt for new password.
      * Confirm password.
        + Store hashed version of password.
  + Exit application.
  + If BASIC:
    - Home screen displays any new games that have been released, as well as what games have received new reviews and articles and how many, every API update, only if there is something to show.
      * Press enter to go to main menu.
        + Options to:

Return to Home Screen.

Execute Saved Query.

Rename username provided current password.

Change password.

Store hashed version of password.

Logout.

* + If ADMIN:
    - Same Home Screen as BASIC.
    - Main Menu options:
      * Return to Home Screen.
      * Manage Queries.
        + Issue Query.

Option to save query (as view if supported in Hive) after creating it.

* + - * + View Saved Query.
        + Execute Saved Query.
        + Delete Saved Query.
        + Return to Main Menu.
      * Rename username provided current password.
      * Change password.
        + Store hashed version of password.
      * Manage Users.
        + View Usernames.
        + Delete BASIC User.
        + Elevate BASIC User to ADMIN.
        + Return to Main Menu.

## Non-Functional Requirements

# UML Class Diagram



# Database Diagram

