Розробка системи для збору та обробки даних з використанням Apache Spark

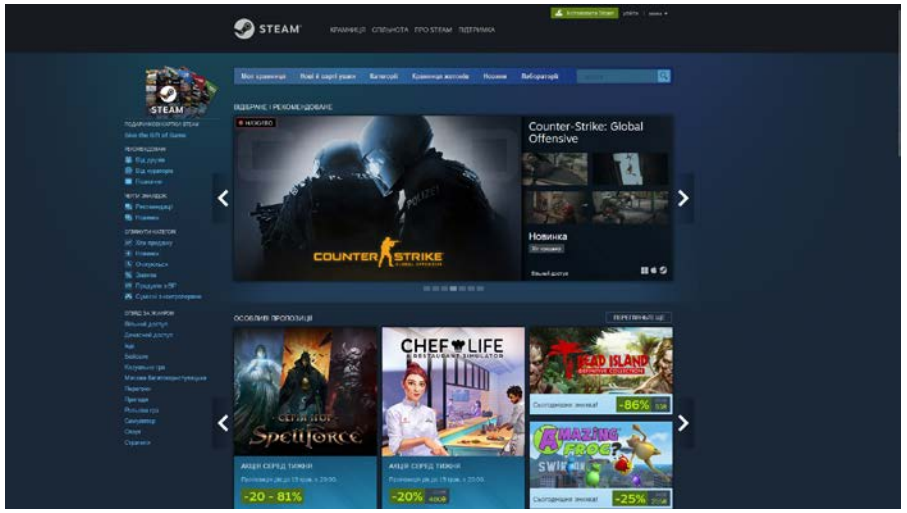
Виконав: Пінкевич В. М.

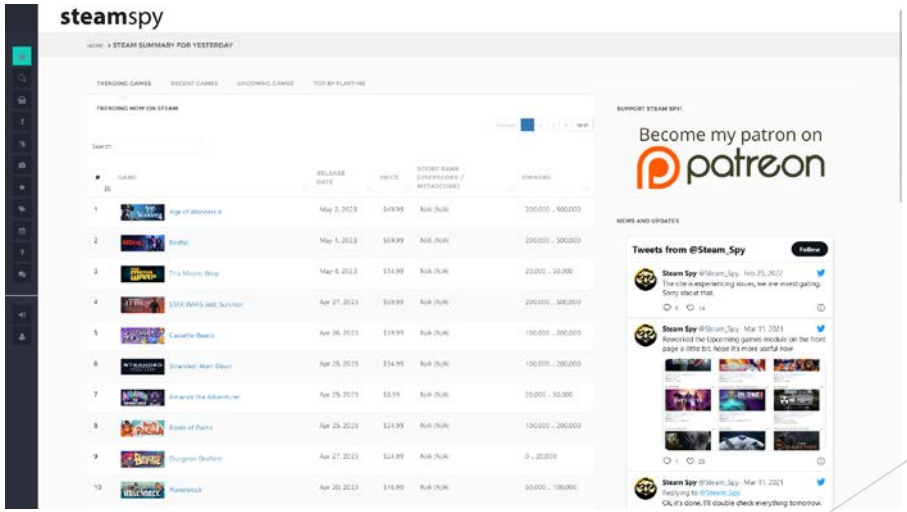
Науковий керівник: Борозенний С. О.

Мета роботи

- Розробка системи для збору даних зі сторонніх публічних веб API, їх обробки та збереження;
- Дослідження особливостей роботи зі сторонніми публічними веб API;
- Дослідження особливостей роботи з великою кількістю даних із використанням фреймворку Apache Spark.

Джерела даних





Архітектура системи

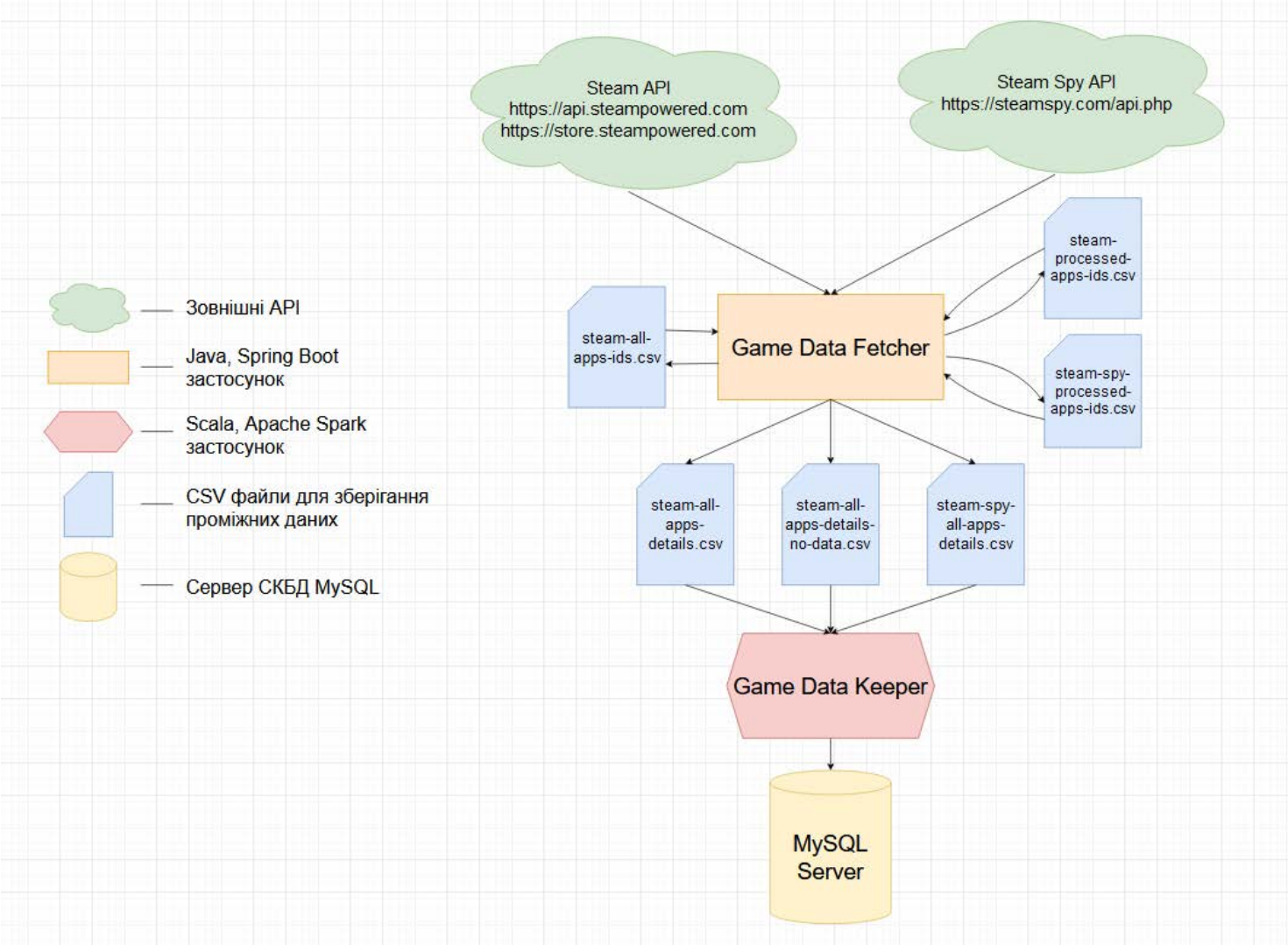


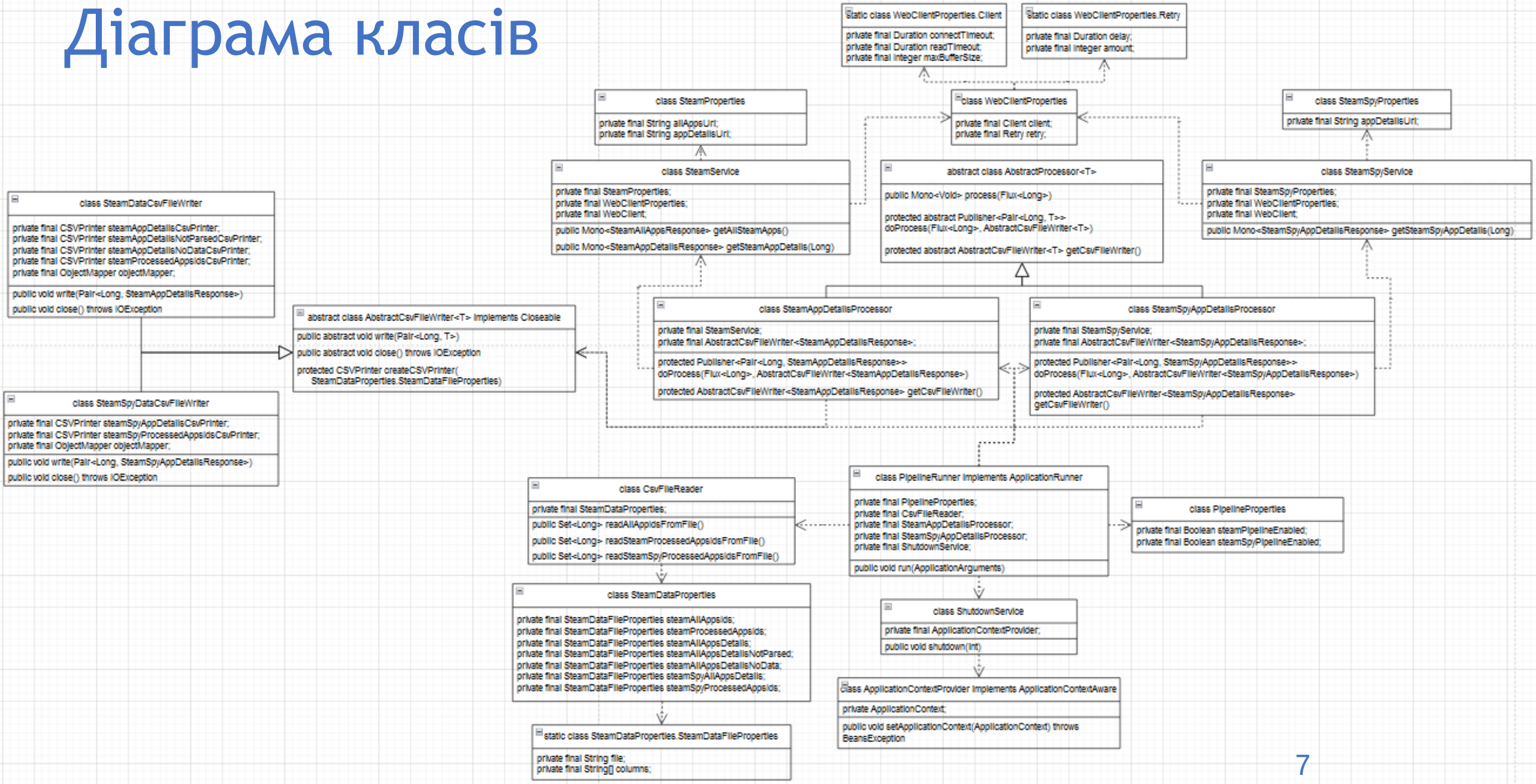
Схема бази даних

Розробка застосунку для збору даних

Spring Boot

Spring Webflux

Діаграма класів



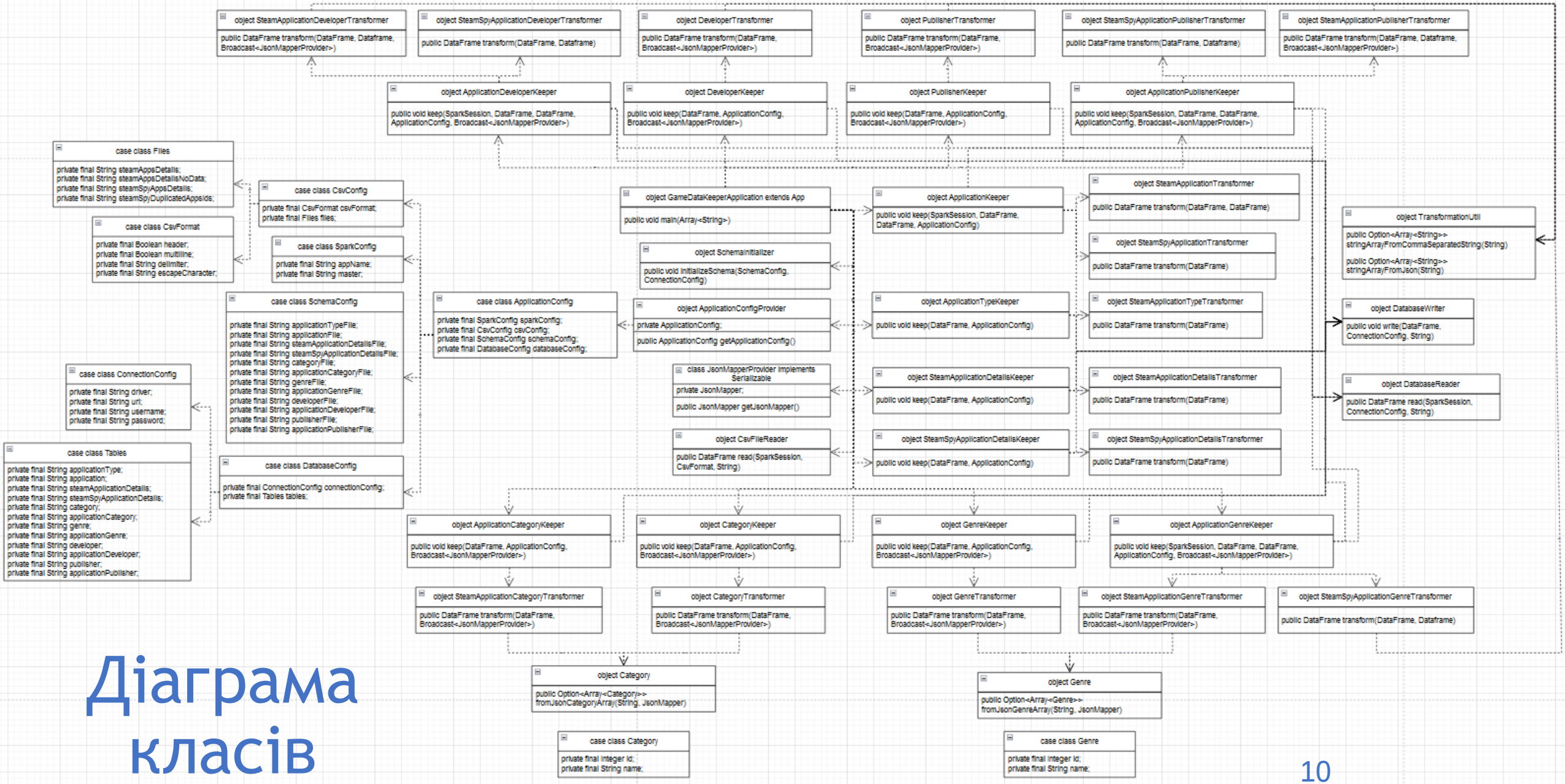
Проблеми взаємодії зі сторонніми публічними веб API

Можливість відмови API у будь-який момент (технічні роботи, перевантаження сервера, тощо).

Наявність обмежень на кількість запитів до API за певний проміжок часу від одного клієнта

Відсутність стандартизованого формату даних у відповідях від API.

Розробка застосунку для обробки даних



Діаграма класів

Можливості покращення системи

- Використання проксі-серверів, які дозволять робити декілька запитів до сторонніх веб API паралельно та збільшать швидкість отримання даних;
- Застосування брокерів повідомлень для передачі даних частинами системи в реальному часі;
- Розгортання повноцінного Spark-кластеру – це дозволить задіяти більше ресурсів для обробки даних.

Результати роботи

- Розроблено систему, яка отримує дані від сторонніх веб API, виконує їх обробку та зберігає уже оброблені дані;
- Описано можливості для розширення та покращення розробленої системи, які дозволять системі отримувати та працювати із даними у реальному часі;
- Було отримано дані про увесь публічно доступний контент у сервісах Steam та Steam Spy та збережено їх у реляційній базі даних. Збережені дані можуть бути використані для подальшого аналізу стану відеоігрового ринку на момент отримання даних.

Висновки (1)

- При використанні сторонніх публічних веб API, над якими розробник не має контролю, необхідно враховувати можливі проблеми, які можуть виникнути та не залежать від розробника, а також розглядати непередбачені сценарії, які можуть відбутися;

- Розподіл системи на різні самостійні частини за їх функціоналом дозволяє забезпечити більшу надійність, кращу відмовостійкість, а також більш ефективне використання ресурсів;

Висновки (2)

- Використання фреймворку Apache Spark показало високу ефективність при роботі із великою кількістю даних. Він надає зручні інструменти для роботи із даними, підтримує велику кількість джерел даних, має можливості для розширення та додавання власної логіки, а також підтримує як горизонтальне, так і вертикальне масштабування.

Дякую за увагу!