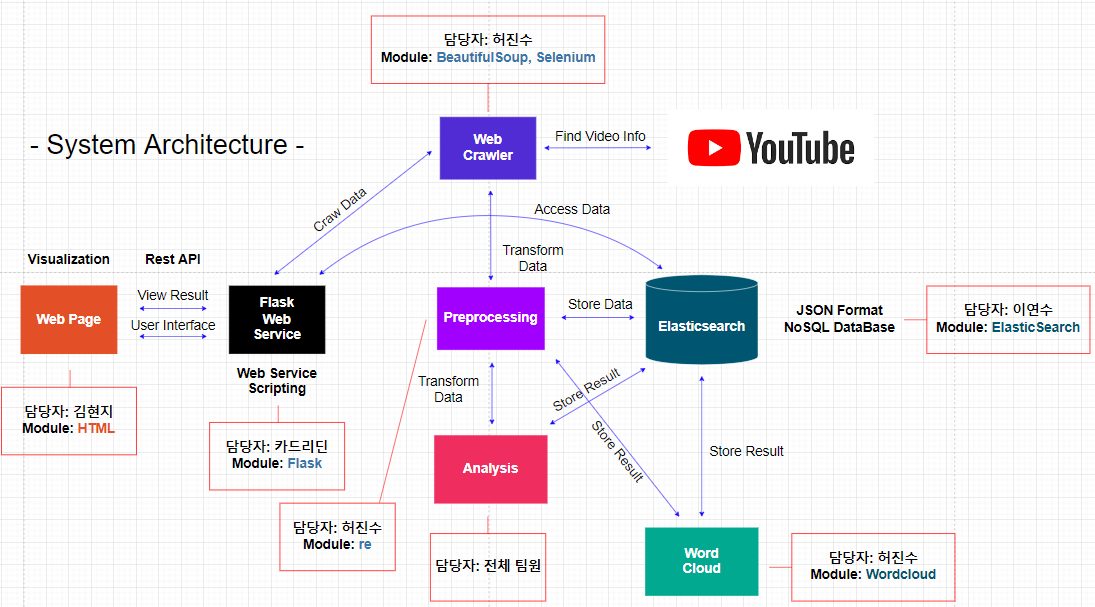
**- 시스템 구조**



※ 설명

**사용한 언어**: HTML, CSS, JavaScript, Python

**사용한 Tool**: Flask, Elasticsearch, Docker

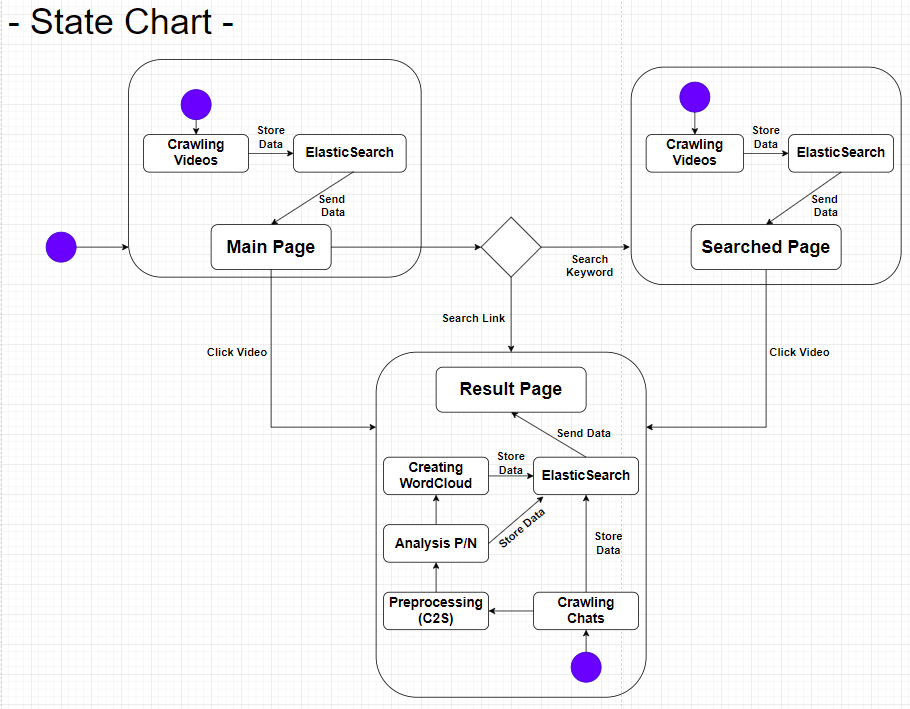
**Version 관리**: Github

**프로젝트 일정이나 회의 관리**: Notion

**개발 환경**: Visual Studio Code, PyCharm

시스템을 개발할 때 실행환경으로 VMware를 설치하여 정상적으로 작동하는지 확인하였고, Docker는 사용할 사람만 Container를 생성하여 시스템이 작동되는지 확인했다.

전반적인 Component 간의 상호작용은 위 그림을 참고하면 된다.



※ 설명

- 처음 시스템을 실행할 때 유튜브의 메인 페이지에 있는 영상을 10개 크롤링한다. 이때 영상의 이미지, 제목, 주소, 조회수, 좋아요 수 데이터를 가져오는데 만약 크롤링이 안되면 None 값으로 둔다. 크롤링이 완료되면 Elasticsearch에 데이터를 저장하고, 저장한 데이터를 Elasticsearch에서 가져와서 Main 페이지로 보낸다.

- Main 페이지에서는 사용자가 원하는 키워드(단어) 혹은 동영상 주소(링크)를 검색할 수 있고, 크롤링으로 가져온 10개의 동영상에 대한 데이터를 카드 형태로 나열하여 보여준다. 동영상에 대한 각 카드를 사용자가 클릭하여 영상이 긍정적인지, 부정적인지 확인할 수 있는 결과 페이지로 넘어갈 수 있다.

- 사용자가 키워드(단어)로 검색을 하는 경우, 크롤러를 실행하여 입력된 키워드에 대한 10개의 동영상 정보(이미지, 제목, 주소, 조회수, 좋아요 수)를 가져오는데 만약 해당 정보가 없으면 None으로 둔다. 크롤링이 완료되면 Elasticsearch에 데이터를 저장하고, 저장한 데이터를 Elasticsearch에서 가져와서 검색 결과 페이지로 보낸다.

- 검색 결과 페이지에서는 사용자에게 크롤링으로 가져온 10개의 동영상에 대한 데이터를 카드 형태로 나열하여 보여주고, 사용자는 각 카드를 클릭하여 영상이 긍정적인지, 부정적인지 확인할 수 있는 결과 페이지로 넘어갈 수 있다.

- 사용자가 키워드가 아닌 동영상의 링크로 검색을 하는 경우, 크롤러를 실행하여 해당 동영상의 댓글을 가져오고, 댓글 데이터를 문장 단위로 나누어서 데이터를 가공한다. 그 후에, 해당 영상이 긍정적인지, 부정적인지 분석하고 긍정/부정 비율이 더 높은 쪽의 문장 데이터를 가져와서 Word Cloud를 만든다. 그리고 분석한 결과와 생성한 Word Cloud 이미지를 Elasticsearch에 저장한다. 모든 작업이 끝나면 해당 동영상에 대한 데이터를 Elasticsearch에서 불러와서 결과 페이지로 보낸다.

- 사용자가 영상 카드를 클릭한 경우, 사용자가 동영상 링크를 검색했을 때 시스템에서 처리하는 작업과 동일하다.

- 결과 페이지에서는 해당 영상에 대한 긍정/부정 분석 결과 데이터와 Word Cloud 데이터를 영상과 함께 사용자에게 보여준다. 구체적으로는 가장 많이 나온 긍정적인 단어 Top 3와 부정적인 단어 Top 3, 해당 영상과 영상에 대한 Word Cloud 이미지, 긍정/부정 비율을 보여준다.

**- 느낀점**

여러 사람이 하나의 팀 프로젝트에 참여하여 서로 다른 작업 환경에서 코드를 작성하고 Git에 업데이트하니까 version 관리가 정말 힘들었다. 또, Docker를 사용하여 시스템을 실행했을 때와 VMware에서 시스템을 실행했을 때가 너무 달라서 환경 설정하기가 무척 까다로웠다. 그리고 코드를 작성할 때마다 정상적으로 실행되는지 확인하는 시간이 오래 걸려서 개발 기간이 예상보다 더 오래 걸린 것 같다.