Assignment-2 Problem 2:

CODE A: To create a Java program to extract data from NewsAPI.

PsuedoCode:

```
String[] keywords = {"Canada", "University", "Dalhousie", "Halifax", "Canada Education",
"Moncton", "hockey", "Fredericton", "celebration"};

// Join all the keywords with "OR" and encode the resulting query string

String query = joinWithOr(keywords);

String encodedQuery = urlEncode(query);


// Construct the URL for fetching the news from the API

String apiUrl = "https://newsapi.org/v2/everything?q=" + encodedQuery +
"&apiKey=ab3a6355793f4756bcda3e9f15b8b792";

URL url = new URL(apiUrl);


// Open a connection to the URL and fetch the news data

HttpURLConnection connection = (HttpURLConnection) url.openConnection();

connection.setRequestMethod("GET");

BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));

StringBuilder response = new StringBuilder();

String inputLine;

while ((inputLine = bufferedReader.readLine()) != null) {

   response.append(inputLine);

}

bufferedReader.close();


// Pass the news data to the DataProcessingEngine for processing

DataProcessingEngine dataProcessingEngine = new DataProcessingEngine();

dataProcessingEngine.process(response.toString());
```

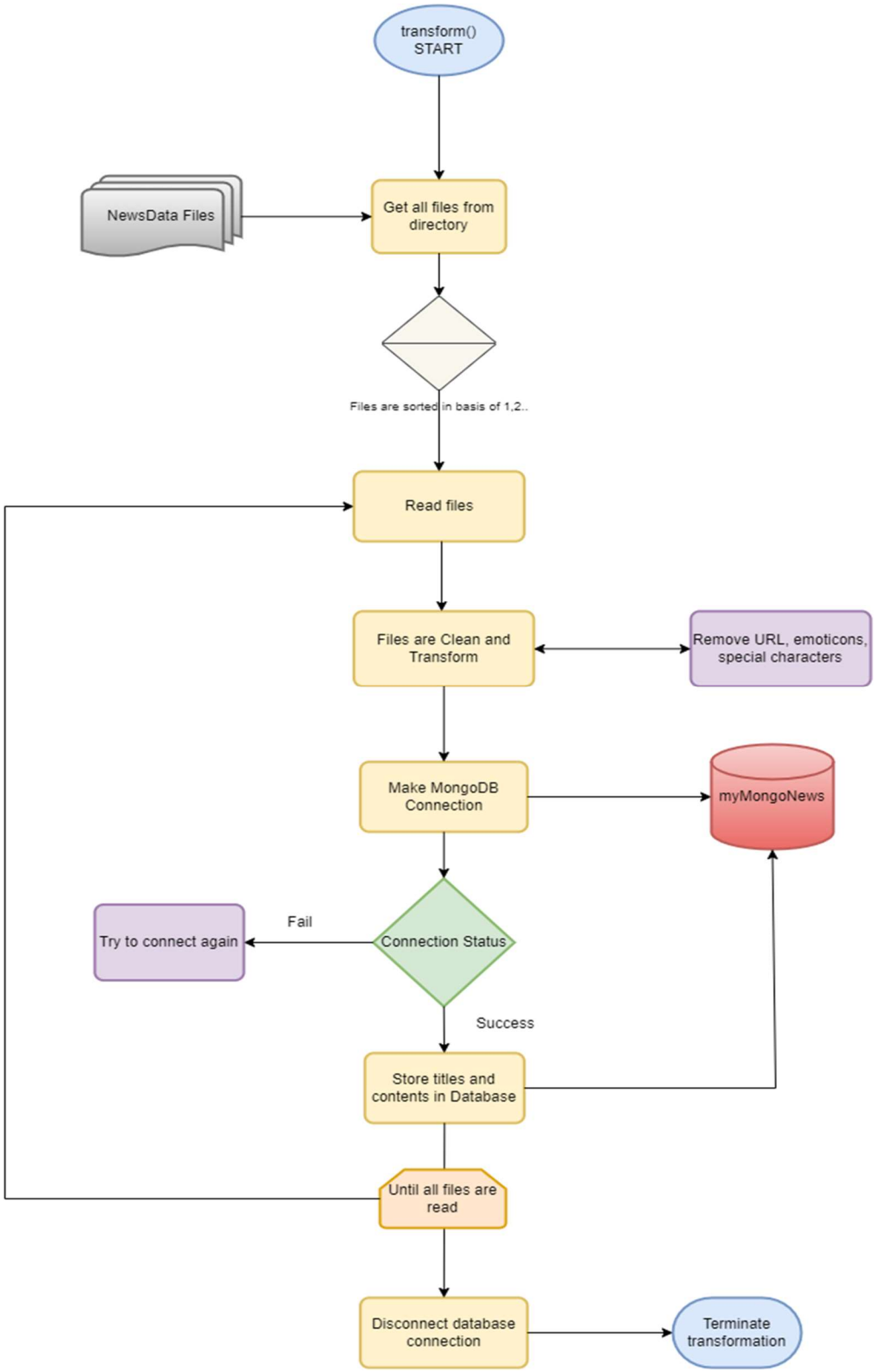
PsuedoCode with sentence explanation:

1. Create an array of keywords to search for in the news API
2. Join the array of keywords with "OR" and encode the resulting query string using a function called "joinWithOr" and "urlEncode" respectively
3. Construct the URL for fetching the news from the API using the encoded query string and API key
4. Open a connection to the URL using HttpURLConnection and set the request method to "GET"
5. Create a BufferedReader to read the response from the API
6. Read the response line by line and append each line to a StringBuilder called "response" until there is no more data to read
7. Close the BufferedReader
8. Create a DataProcessingEngine object
9. Pass the news data, stored in "response" as a string, to the DataProcessingEngine object for processing using the "process" method.

CODE B: Write titles and content in raw files, each files having 5 or less articles.

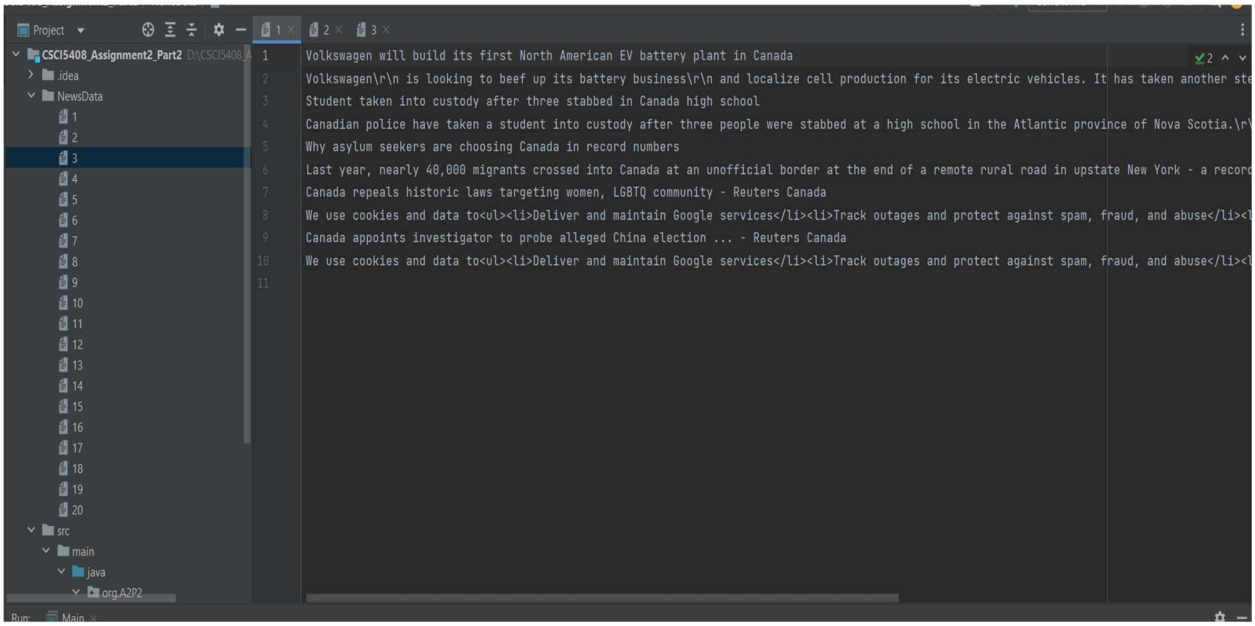
1. Start of the DataProcessingEngine class
2. Define the process method that takes a StringBuilder object called "response" as input parameter
3. Compile two regex patterns, one for the title and one for the content
4. Initialize variables including fileName, fiveNewsCounter, and path
5. Create a new FileWriter object to write data to a file
6. Loop through all titles and contents and write them to the file
   a. If there are five articles in a file, create a new file
   b. Extract the title and content from the response
   c. Write the title and content to the file
   d. Increment fiveNewsCounter
7. Close the writer
8. Create a new TransformationEngine object
9. Call the transform method to store data in MongoDB

CODE C: Flowchart for transformation Engine:

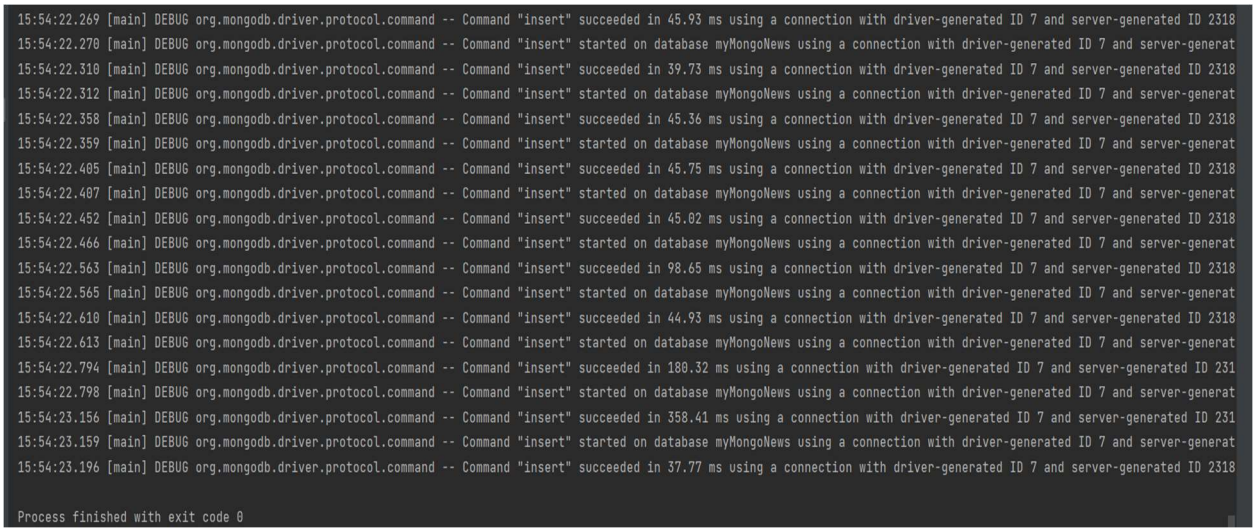


Test cases and Screenshots

1. Title and contents store in files.



2. Documents inserted in MongoDB:



3. Documents (titles and contents) visible in Database with removed URL, emoticons and special characters.