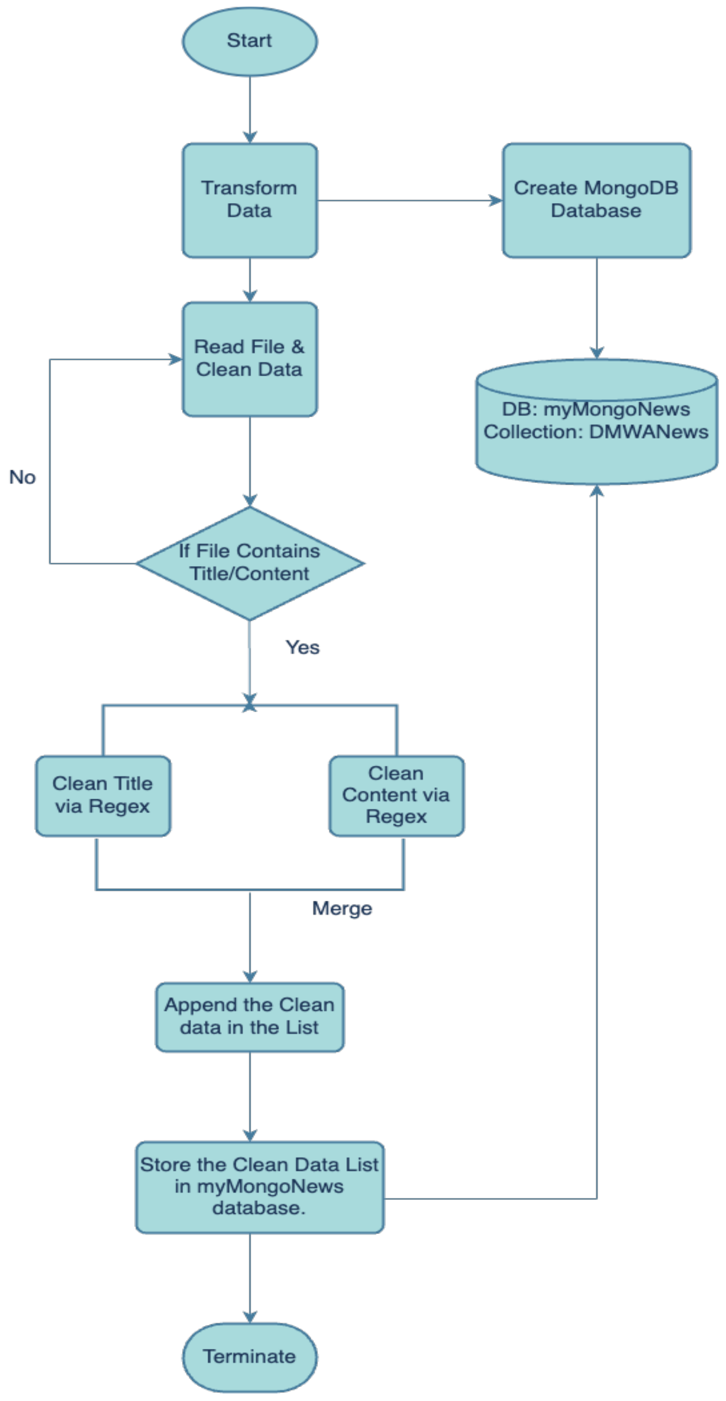
Assignment – 2 (Problem 2)

Pseudocode for Data Extraction (Code A)

- Define a package named "org.example".
- Import the required classes.
- Define a class named "CodeA".
- Define a public method named "fetchAPI" that throws an exception.
- Declare three String variables: "keywords", "apiKey", and "httpsUrl".
- Assign appropriate values to the "keywords", "apiKey", and "httpsUrl" variables.
- Create a new URL object named "myurl" using the "httpsUrl" variable.
- Create a new BufferedReader object named "in" by calling "openStream ()" method on "myurl" and wrapping it in an InputStreamReader.
- Declare a String variable named "inputLine".
- Create a new StringBuffer object named "response".
- While there is input available from the BufferedReader, read each line of input into the "inputLine" variable and append it to the "response" StringBuffer.
- Close the BufferedReader.
- Create a new instance of CodeB class named "codeB".
- Call the "dataProcessing" method on the "codeB" instance, passing the "response" StringBuffer as a parameter.
- Create a new instance of CodeC class named "codeC".
- Call the "transformData" method on the "codeC" instance.

Flowchart of Transformation Engine


Start
Transform Data
Create MongoDB Database
Read File & Clean Data
DB: myMongoNews Collection: DMWANews
No
If File Contains Title/Content
Yes
Clean Title via Regex
Clean Content via Regex
Merge
Append the Clean data in the List
Store the Clean Data List in myMongoNews database.
Terminate

Explanation of Flowchart

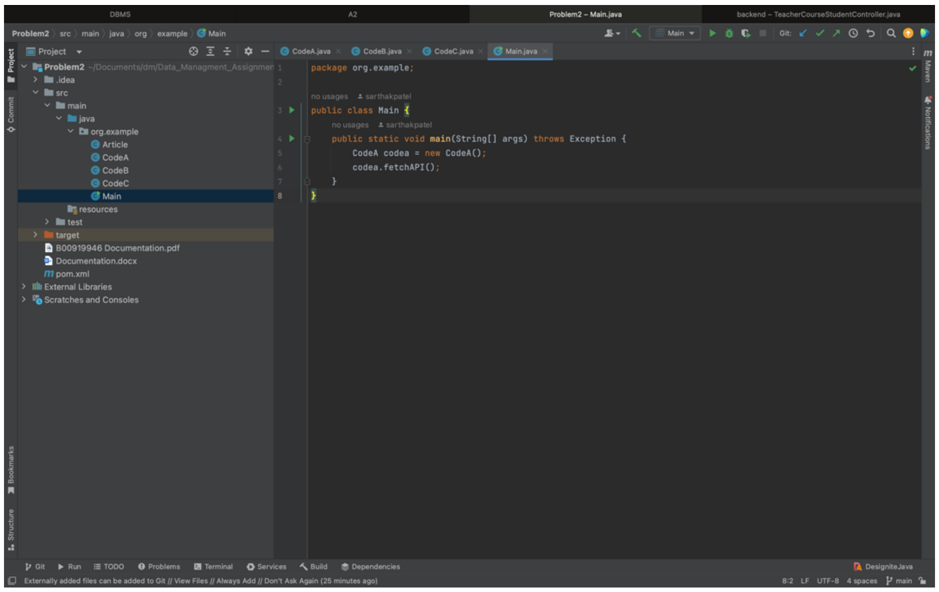
Before the function transformation data is called we would already have fetched the data from the API and processed it to store in the files with utmost 5 article in each file. After that we would read and clean all the articles from the files by removing URLS and emoticons from the title and content part of the files with the help of regex. Each article consisting of title with its respective content will be stored in the list of articles. After that we will push the data as a document in the MongoDB database, myMongoNews we made.

Solid Principles Implemented

I have tried to implement as much SOLID principles as I can and below are the principles which my code follows:

- **Single Responsibility Principle (SRP):** The CodeA, CodeB, and CodeC classes each have a clear and singular responsibility. CodeA is responsible for fetching data from the News API, CodeB is responsible for processing and writing the data to files, and CodeC is responsible for transforming and storing the data in a MongoDB database.
- **Open/Closed Principle (OCP):** The code does follow the Open/Closed principle. The code can be extended without modifying the existing code.
- **Liskov Substitution Principle (LSP):** The provided code does not seem to have any inheritance hierarchy, so there is no application of LSP.
- **Interface Segregation Principle (ISP):** There are no explicit interfaces defined in the provided code, so ISP hasn't been followed.
- **Dependency Inversion Principle (DIP):** The classes are loosely coupled.

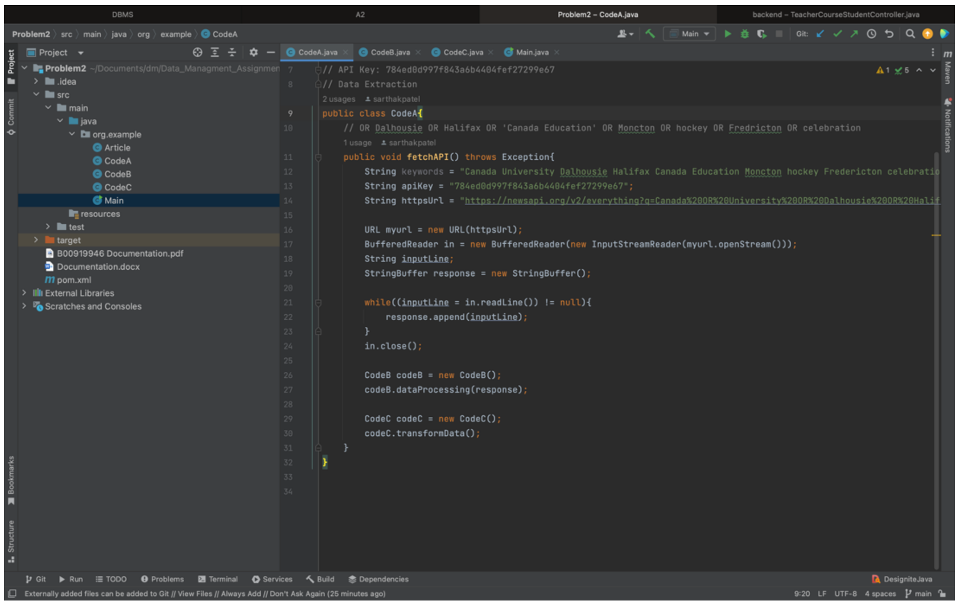
Screenshots of Implemented Test Cases



```java
package org.example;

public class Main {
    public static void main(String[] args) throws Exception {
        CodeA codea = new CodeA();
        codea.fetchAPI();
    }
}
```

Fig1: Main.java class of Problem 2.

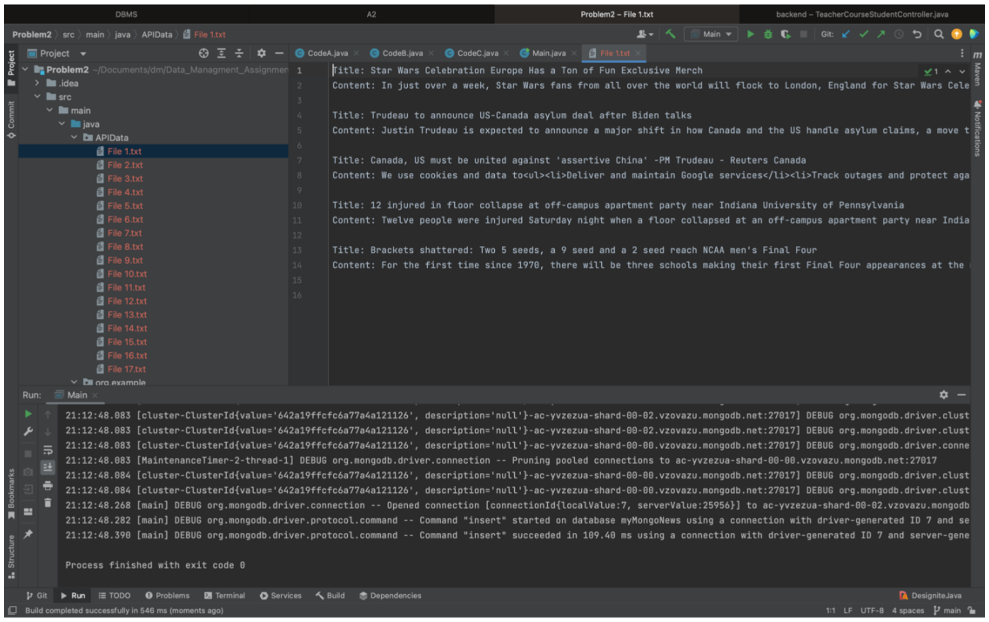
The Main.java class only take object of Code A that is the object for extracting data from the news API.



```java
// API Key: 784ed0d997f843a6b4404fef27299e67
// Data Extraction
public class CodeA{
    // OR Dalhousie OR Halifax OR 'Canada Education' OR Moncton OR hockey OR Fredricton OR celebration
    public void fetchAPI() throws Exception{
        String keywords = "Canada University Dalhousie Halifax Canada Education Moncton hockey Fredericton celebratio
        String apiKey = "784ed0d997f843a6b4404fef27299e67";
        String httpsUrl = "https://newsapi.org/v2/everything?q=Canada%20OR%20University%20OR%20Dalhousie%20OR%20Halif

        URL myurl = new URL(httpsUrl);
        BufferedReader in = new BufferedReader(new InputStreamReader(myurl.openStream()));
        String inputLine;
        StringBuffer response = new StringBuffer();

        while((inputLine = in.readLine()) != null){
            response.append(inputLine);
        }
        in.close();

        CodeB codeB = new CodeB();
        codeB.dataProcessing(response);

        CodeC codeC = new CodeC();
        codeC.transformData();
    }
}
```

Fig2: CodeA.java class of Problem 2.

The Code A class takes objects of Code B and Code C responsible for data processing and transformation of data. Hence it fulfills the condition of problem, that code A -> code B -> code C. As all classes are depended on each other to execute their part.



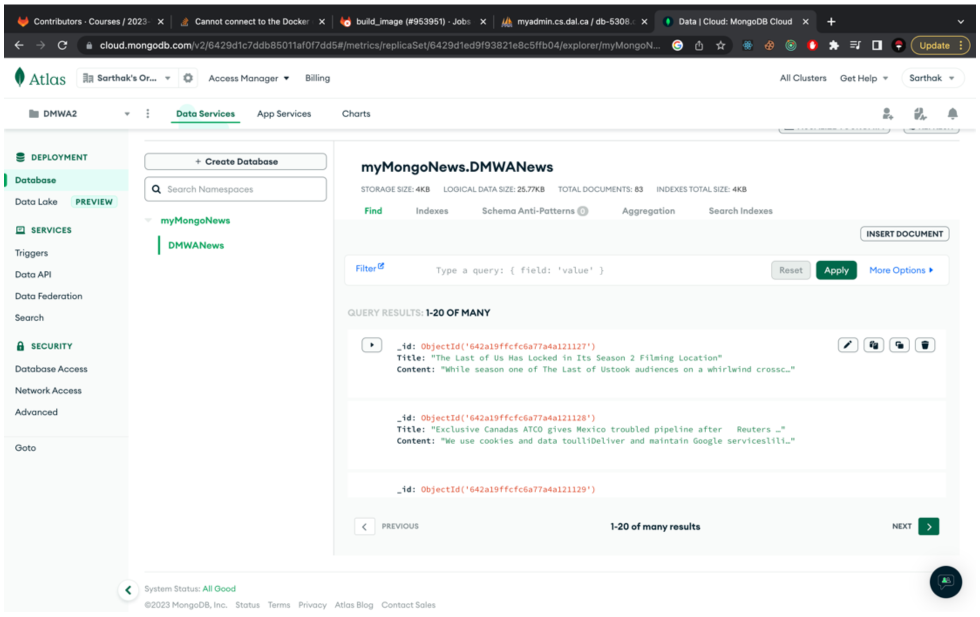
```
Title: Star Wars Celebration Europe Has a Ton of Fun Exclusive Merch
Content: In just over a week, Star Wars fans from all over the world will flock to London, England for Star Wars Cele

Title: Trudeau to announce US-Canada asylum deal after Biden talks
Content: Justin Trudeau is expected to announce a major shift in how Canada and the US handle asylum claims, a move t

Title: Canada, US must be united against 'assertive China' -PM Trudeau - Reuters Canada
Content: We use cookies and data to<ul><li>Deliver and maintain Google services</li><li>Track outages and protect aga

Title: 12 injured in floor collapse at off-campus apartment party near Indiana University of Pennsylvania
Content: Twelve people were injured Saturday night when a floor collapsed at an off-campus apartment party near India

Title: Brackets shattered: Two 5 seeds, a 9 seed and a 2 seed reach NCAA men's Final Four
Content: For the first time since 1970, there will be three schools making their first Final Four appearances at the
```

```
21:12:48.083 [cluster-ClusterId{value='642a19ffcfc6a77a4a121126', description='null'}-ac-yvzezua-shard-00-02.vzovazu.mongodb.net:27017] DEBUG org.mongodb.driver.clust
21:12:48.083 [cluster-ClusterId{value='642a19ffcfc6a77a4a121126', description='null'}-ac-yvzezua-shard-00-02.vzovazu.mongodb.net:27017] DEBUG org.mongodb.driver.clust
21:12:48.083 [cluster-ClusterId{value='642a19ffcfc6a77a4a121126', description='null'}-ac-yvzezua-shard-00-00.vzovazu.mongodb.net:27017] DEBUG org.mongodb.driver.conne
21:12:48.083 [MaintenanceTimer-2-thread-1] DEBUG org.mongodb.driver.connection -- Pruning pooled connections to ac-yvzezua-shard-00-00.vzovazu.mongodb.net:27017
21:12:48.084 [cluster-ClusterId{value='642a19ffcfc6a77a4a121126', description='null'}-ac-yvzezua-shard-00-00.vzovazu.mongodb.net:27017] DEBUG org.mongodb.driver.clust
21:12:48.084 [cluster-ClusterId{value='642a19ffcfc6a77a4a121126', description='null'}-ac-yvzezua-shard-00-00.vzovazu.mongodb.net:27017] DEBUG org.mongodb.driver.clust
21:12:48.268 [main] DEBUG org.mongodb.driver.connection -- Opened connection [connectionId{localValue:7, serverValue:25956}] to ac-yvzezua-shard-00-02.vzovazu.mongodb
21:12:48.282 [main] DEBUG org.mongodb.driver.protocol.command -- Command "insert" started on database myMongoNews using a connection with driver-generated ID 7 and se
21:12:48.390 [main] DEBUG org.mongodb.driver.protocol.command -- Command "insert" succeeded in 109.40 ms using a connection with driver-generated ID 7 and server-gene

Process finished with exit code 0
```

Build completed successfully in 546 ms (moments ago)

Fig3: Output generated by executing the code.

When the code executes, the data from the News API will be stored in the files which would have at most of 5 new articles with their titles and content.



myMongoNews.DMWANews

STORAGE SIZE: 4KB LOGICAL DATA SIZE: 25.77KB TOTAL DOCUMENTS: 83 INDEXES TOTAL SIZE: 4KB

QUERY RESULTS: 1-20 OF MANY

```
_id: ObjectId('642a19ffcfc6a77a4a121127')
Title: "The Last of Us Has Locked in Its Season 2 Filming Location"
Content: "While season one of The Last of Ustook audiences on a whirlwind crossc…"

_id: ObjectId('642a19ffcfc6a77a4a121128')
Title: "Exclusive Canadas ATCO gives Mexico troubled pipeline after   Reuters …"
Content: "We use cookies and data toulliDeliver and maintain Google serviceslili…"

_id: ObjectId('642a19ffcfc6a77a4a121129')
```

1-20 of many results

Fig4: MongoDB database output

Code C was responsible was for transforming and cleaning data as well as to store it in the MongoDB database.

Note

Once the program has been executed it will successfully execute Code A, Code B, Code C without the manual creation of files and cleaned and transformed data would be stored in MongoDB database. But when you run the program again it would again fetch the data from the API and store it in the existing file, hence duplicity of data in file. So, if someone wants to run the program again correctly, delete the APIData folder and run it again. I could have designed the program to delete all existing files of data in APIData folder, but one wouldn't be able to see the data again even after the data is successfully stored in MongoDB database.