IT5100B Industry Readiness Stream Processing Project Report

Name: Wong Shang Yi
Matriculation Number: A0276553W

1. Features of the application

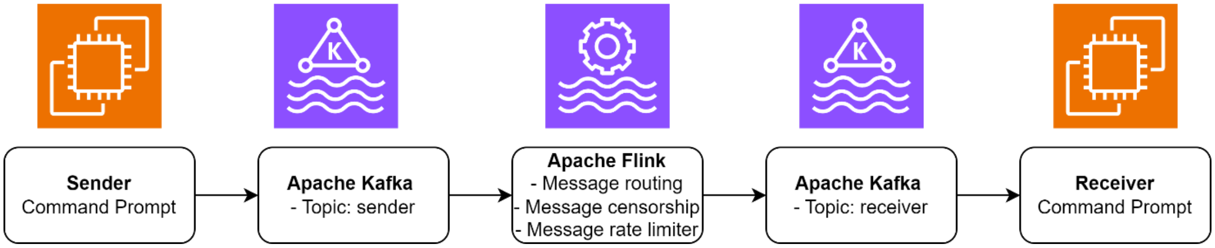
1. Users can send messages to other users.

2. Users can receive messages from other users.

3. Profanities in messages are censored.

4. Message rate limiter (User cannot send messages if the duration between messages is less than 500ms and will receive a warning message in the receiver)

2. Configuration



1. As the application is still in the prototyping and testing stage, the resources are kept to the bare minimum for rapid development.
   a. No. of consumer: 2
      i. 2 users with their respective sender application
   b. No. of producer: 2
      i. 2 users with their respective receiver application
   c. Topics: 2
      i. Sender
      ii. Receiver
   d. Apache Flink task managers
      i. 1
2. However, the application can be scaled horizontally and vertically when the number of applications increase, by:
   a. Apache Kafka/Apache Flink
      i. Increasing the size of the clusters
      ii. Increasing the number of topics for each geographic region (regions)
      iii. Increasing the number of partitions for each region
      iv. Increasing the number of replications distributed across different availability zones in the same region

3. Running the application

a. Package my_console_application in IntelliJ by running:

  i. Menu -> Build -> Build Artifacts -> Build/Rebuild

b. Package my_flink_job in IntelliJ by running:

  i. Maven plug in -> execute -> mvn package

c. To start the docker image, run the following in command prompt;

  i. docker-compose up -d

d. To view the Apache flink task manager in browser, enter:

  i. localhost:8081

e. Submit 'my_flink_job-1.0.jar' and run with parallelism of 1

f. Create new topics by running in the kafka0 shell:

  i. kafka-topics.sh --create --replication-factor 2 --partitions 3 --bootstrap-server kafka0:9094 --topic sender

  ii. kafka-topics.sh --create --replication-factor 2 --partitions 3 --bootstrap-server kafka0:9094 --topic receiver

g. Open 4 command prompt windows, and run:

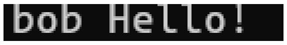
  i. cd .\code\my_console_application\out\artifacts\my_console_application_jar

  ii. java -jar my_console_application.jar

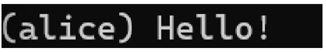
h. Follow the prompt on the screen to start the application

4. Expected outcomes

1. Messaging (Alice to Bob)

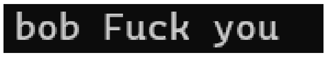
a. Input: 

```
bob Hello!
```

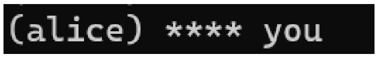
b. Output: 

```
(alice) Hello!
```

2. Message censorship

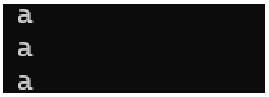
a. Input: 

```
bob Fuck you
```

b. Output: 

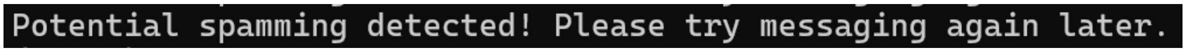
```
(alice) **** you
```

3. Message rate limiter

a. Input: 

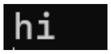
```
a
a
a
```

b. Output:



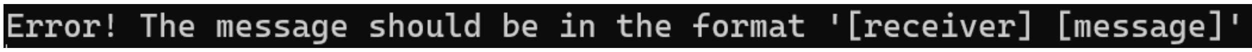
```
Potential spamming detected! Please try messaging again later.
```

4. Message format checker

a. Input: 

```
hi
```

b. Output:



```
Error! The message should be in the format '[receiver] [message]'
```