**CS6310 Spring 2021 A6**

**Group 4**

Rui Tan

Zhaoran Yang

Qi Li

Chenyu You

Lanxin Zhang

1. **Functioning Application**

For this project, a web application is built with Spring Boot.

To run this web application, use the source code in ***source\_code.zip*** and import the source code “spring” folder as project in IDE intelliJ, with jdk13, run the following three commands to build the project into docker(backend/frontend separated), then composed the docker docker container:

*docker build -t gatech/backend -f ./images/Dockerfile.backend ./backend*

*docker build -t gatech/frontend -f ./images/Dockerfile.frontend ./frontend*

*docker-compose -p gatech -f docker-compose.yml up -d*

(PS: in order to see the configuration of the load balancer (Scalability), comment the api instance configuration, and uncomment the api0, api1, api2 configuration in the docker-compose.yml file, then stop the running docker instances and remove the container and images, re-build the backend/frontend docker, then run:

*docker-compose -p gatech -f docker-compose.yml up --scale api=3 -d* )

While the docker container is up and running, navigate to <http://localhost:3001> to interact with our user interface. (Configurability)

At the frontend interface you may input data manually, create each group (demographic group, studio, etc) the data will persist to database. When click display, you need to input with the identified (such as shortName for studio, shortName for streaming service) to display the full information:

If you wish to view input data at backend, navigate to the following endpoints listed as below:

Demographic Group:

http://localhost:8080/crud/demographicgroup

StreamingService:

http://localhost:8080/crud/streamingservice

Studio:

http://localhost:8080/crud/studio

Event:

http://localhost:8080/crud/event

Offer:

http://localhost:8080/service/offer

WatchRecord:

http://localhost:8080/service/watch

We use a simulation to forward the time, open a terminal, when you are ready to forward to the next month, input:

curl -i -X GET "localhost:8080/service/nextmonth/next"

if you want to see the current time:

curl -i -X GET "localhost:8080/service/nextmonth/display"

if you would like to check the database, use:

$: docker ps

to see the container ID for the database postgre

$: docker exec -it <containerID> bash

$:psql -U postgres

$:\dt

then you can see the database schema

From here you may use SQL command to check the entry for each table. (reference to the Archivability part of the demo video)

Sample test case file (Attachment\_test\_scenario\_passed\_30.txt) used in the demo video is attached here.