

Team A Iteration 2

Good job overall. Keep the design documentation up to date as you continue the project.

90 / 100

High-level design

Components, package, classes

The high-level design has good diagrams, but could use more discussion of the responsibilities of each class.

Good discussion of spark and docker.

Specification of internal interfaces (APIs, WebAPIs, sequence diagrams)

Very good collection of sequence diagrams. The more important use cases are addressed.

WebAPI seems more complete.

Database Schemata

I'm not sure that all your classes that need to be persisted are represented in the schema. Comments, test reports, grades, for example.

Implementation

Good start. Keep it up.

Other comments

It's good to see that you tackled the high risk element of isolation early. I'm hoping for some use cases involving DB to be complete for iteration 3.

Notes from Zizui

1. Please attach exported SQL file with your project, e.g. db.sql. This script should drop existing tables and import at least initial data.
2. It is recommended to allow setting database connection string with environment variables. For example:

```
var host = process.env.MYSQL_HOST ;
```

By using this approach, it simplified both testing and development work, because the DB connection string can be changed on-the-fly without modifying your program :-) To set the DB connection, simply executing:

```
export MYSQL_HOST=localhost
export MYSQL_USERNAME=root
export MYSQL_DBNAME=codedrop
export MYSQL_PASSWORD=TOPS3CT3T!!
```

1. Please move your code to root directory, then organize all documents/submissions to doc/.
2. You required a big docker images which crashed by docker :-) Maybe it is better to put a note on your program to suggest users switch to overlay instead of aufs to avoid the bug I suffered (<https://github.com/docker/docker/issues/6325>).
3. Why its keep reporting "read from process!:"? Are you pooling the output and waiting for user input? If so, then I think it is better to be replaced by other approaches. Even namedpipe is better than this.
4. Another suggestion is to replace your command line (docker run ...) with docker API and a Dockerfile (maybe even kubernetes) Your ant configuration is incomplete. It is better to follow the standard approach (e.g. <https://ant.apache.org/manual/tutorial-HelloWorldWithAnt.html>) to define "compile", "jar" and "run" targets. Thus, it enable the testers (me) to build a jar file and do automatic deployment.