My Contributions and Application for Project Create a CLI tool for managing InterMine instances

Application:

Past experience with this free software community and project:

- I started with reading and following the tutorial to setup the local version of intermine.
- After that, I cloned intermine_boot repo and read about the requirements
- I have read through all the code in intermine_boot and docker-intermine-gradle repository
- I have worked on two issues. I have created PRs for the same. I intent to get PRs merged soon and work on more issues.

Past experience with other free software communities:

- I have taken part in hacktoberfest last year and completed that.
- My experience with open source communities is a bit limited otherwise.
- But I use git, python, bash and other tools very frequently in my day to day research work so I am well versed in them.

Relevant personal or class projects that relate to this project:

- some of my research projects implemented in python are linked here
https://github.com/22PoojaGaur/Improving-Diversity-Performance-of-Association-Rule-Based-Recommender-Systems
https://github.com/22PoojaGaur/Extracting-Diverse-Pattern-With-UCH

- I am a beginner in docker but I have written docker files for creating docker images to get my feet wet.

- I had written a web scraper using python (mechanize, beautiful soup and selenium) as part of an internship.

- I have studied and worked in bash in the early years of my college. I am fairly confident to read and write bash scripts for a task.

Verified the time commitments listed on initial application are correct.

Applicant is not applying to Google Summer of Code.

Timeline of tasks for the internship period:

During April

I have read the code already present. I would resolve more issues and get a better knowledge of things. I would also learn more about the CI tools that might be useful (I had worked a bit on project that used jenkins for CI/CD and aws for hosting but I haven't written jenkins code myself)

May 04 - May 16

Talk with mentor to finalize the timeline and goals (pr deadlines) for the project.
- store the config file and make sure that build takes info from config file to run.

May 18 - May 30 May

Create single command setup for CI use case
- set up travis CI in github
- add .travis.yml to git
- in before install check dependency, load configs, download required docker image and run tests.

Jun 1 - Jun 6

Implement versioned data uploading
- The archives stored locally can be versioned based on config.

- If config changes, an archive with new version is stored and since version changed it is uploaded to the cloud with the version tagged.

Jun 8 - Jun 13

Buffer time to get above features merged

Jun 15 - Jun 20

add wizard and configurator instances to local setup to get config files from users

Jun 22 - Jun 27

modify code to ask and use configurations from the file and orchestrate docker image

Jun 29 - Jul 4

Buffer time to get above features merged

Jun 6 - Jul 18

Look into way to upload data files

Jul 20 - Jul 25

Buffer time to get the above merged

Jul 27 - Aug 15

Remaining time for additional work or writing tests

Contributions:

Create a CLI tool for managing InterMine instances

Contribution #1:

started April 3, 2020. merged.

https://github.com/intermine/docker-intermine-gradle/pull/9

https://github.com/intermine/intermine_boot/pull/21

```
Currently, intermine_builder docker container builds a
intermine instance from scratch even if the build already
exists on the system.

This PR resolves this issue by ->
- Adding a condition to check whether build directories exist
- running only script to start tomcat webapp if the build
directory exists.
```

Contribution #2:

started March 26, 2020. merged.

https://github.com/intermine/intermine_boot/pull/17

```
Problem ->
To add support to upload archives of the intermine build to
cloud servers. As a start, make it for `amazon s3 bucket` in
an extendable manner.

Solution ->
- Created an aws educate account.
- Looked at steps involved in managing aws buckets and
uploading files to them
- Looked at which part of code creates the archive
- built a function to upload the same archives to the s3
buckets
- for making it extendable:
    - Make the credentials as a dictionary which maps
credential for each method to the credentials required.
    - Make a function which is called for all upload related
methods and which calls individual methods based on the
argument sent.
```