**Assignment-Guidelines**

**Must-have(s):**

* Should be individual work
* First things first - have a clear Problem Statement on what all your Project is going to accomplish and NOT going to DO.
* Clear JAVA coding following best practices. Use JAVA >= 1.8
* Follow OOP Principles in clearly identifying Interfaces, Classes,
  + Also, ensure to handle Exception Handling, Logging, etc.
* Unit Tests
  + Positive Scenarios
  + Negative Scenarios
  + Edge Cases/Boundary Conditions
  + NOTE: Write as many UTs as possible but meaningful ones. Also, Unit Tests should be independent, repeatable (idempotent).
* Check-in your code in Git (local Git is fine for now).
* Build a CI/CD pipeline. It should:
  + C/O code from Git
  + Run maven build (clean install)
  + Run all your UTs.
  + Run through your SONAR QUBE Static Code Analysis
  + Deploy the binaries to your local Tomcat (>8.5 version)
  + Write Integration Tests. This can be again using Java’s JUNIT but it need to assume Tomcat is running and hit the end-points.
  + Post deployment, run as part of the pipeline the Integration Tests in an automated way.
  + SEND EMAIL NOTIFICATION TO ME AND YOURSELF AFTER THESE STEPS (IN BOTH CASES - EITHER PASS/FAIL). Please keep this to the last otherwise there’d be flurry of notification emails during your development.
* Documentation:
  + README in your project. Should cover overview of the project in couple of lines; Steps to build & install if not done as part of the pipeline. Plus, any other information that would be meaningful
  + DESIGN Document:
    - Should contain class diagrams, sequence diagrams, use-case diagram
  + Developers Guide:
    - Steps, with clear examples showing how would you like a Developer to start using your Services.
  + NOTE: You can look at our current projects (only the Documentation 😊) for your reference.
* All your development should be on Linux (any flavor is fine) and using Java and RESTful Webservices. Spring Framework would be preferred.
* If you would like other languages, please contact me.

**Nice-to have:**

* Bonus: Build a Docker Image of your work and keep it in your local registry. README file on how to get the image, run your container. It should provide a shell interface to login as well