**Project requirements**

**This is an engineering proof of concept. It is not a planned marketable project. The goal is to exercise the Spring Enterprise Framework according to “company” guidelines to validate its use in future projects.**

**The presentation should emphasize demonstrating the Enterprise technical capabilities of Spring. It should NOT be primarily a demonstration of a user interface.**

**The man hours spent on the project is expected to be ~100 hours for a 4 person team. This should be tracked in the daily README file and in a spreadsheet.**

* **Use the technologies we learned** 
  + Annotation Driven
  + Hibernate Persistence [JPA]
  + Performance
  + EMail
  + AOP
  + Spring Integration [EAI]
  + REST - security
  + Security ; Login, Authorization
  + Messaging [ JMS/AMQP]
  + MVC \*\*\* This is should NOT be an emphasis for the Project
  + Batch
  + Testing
  + Maven
  + **Documentation should include EXACTLY where technologies are used.**
  + **Documentation should include EXACTLY how to test the functionality.**
  + **Documentation should include WHO did WHAT**
  + **DO NOT use technologies that we haven’t covered…**
* Other “technical” requirements include
  + Follow “corporate” architectural decisions
    - This isn’t a one-off design/implementation
    - Pivotal Server [Tomcat] based
    - MySQL based
    - Adhere to project structure as used in class
    - Adhere to configuration as learned in class
    - GitHub Manual usage…
  + Good naming practices [[Oracle naming](http://www.oracle.com/technetwork/java/codeconventions-135099.html) ]
    - Proper package naming
    - Descriptive names [ Product product .vs. Product p .vs. Product prod ]
* Every Team member is to “own” a “unique technology” scenario
  + **Documentation should include WHO did WHAT**
* Use design pattern of Spring and/or separation of concerns
  + Put the data manipulation in service layer
  + Put your data computation in service, and domain model[ invoked through service]
  + Do the data access in service through repository
* Design documentation
  + Class & sequence diagrams, etc.[**CLASS DIAGRAM**: Not ERD & NOT reverse engineered]
  + Assumptions, risks, dependencies
* Documentation should include how to configure /Install Application
  + database name should be **team name**
  + Application should have pre-populated data [ populate.sql].

**REQUEST:**

**BUT I want to use SQL Lite … ANSWER: Follow the Guidelines**

**BUT I want to use Jetty … ANSWER: Follow the Guidelines**

**BUT I want to use Hibernate without JPA … ANSWER: Follow the Guidelines**

**BUT I want to work with my Teammates on EVERYTHING … ANSWER: Follow the Guidelines**

**BUT I want to use the Spring Boot… ANSWER: Follow the Guidelines \*\*\***

**BUT I want to use the Cloud… ANSWER: Follow the Guidelines \*\*\***

**BUT I want to use ERD diagrams… ANSWER: Follow the Guidelines \*\*\***

**BUT I want to use the Gradle… ANSWER: Follow the Guidelines**

**BUT I want to use Spring Data… ANSWER: Follow the Guidelines**

**BUT I want to work with my Teammates on EVERYTHING … ANSWER: Follow the Guidelines**