These are the user stories to follow. No change in base requirements, just some notes for clarification

## Quiz Management System (QMS)  
\* You will show your project to me on Monday, 1/6/2020  
\* No powerpoint is required - my expectation is that you have everything up and running that Monday, and simply run me through the user stories  
\* I will speak with Stanley about allowing you 11 to focus on this project in Staging

### User Stories  
\* A manager can log in  
\* A manager can create quiz questions  
\* The expectation here is *\_only\_* for Multiple Choice questions  
\* A manager can create quizzes  
\* A manager can assign quizzes to users  
\* A user can log in  
\* A user can take assigned quizzes  
\* A user can see all previously completed quiz grades

### OPTIONAL USER STORIES  
\* A user can view explanations for any missed question  
\* A user can register  
\* A user can be added to a group  
\* A manager can assign quizzes to a group  
\* A manager can create questions of other types than Multiple Choice  
\* Some that come to mind are Short Answer, True/False, Fill in the Blank, Match Terms to Definitions, etc

## Technical Requirements  
\* Java 8  
\* Tomcat 8.5 or newer  
\* Front Controller design pattern is suggested, but not required.  
\* JDBC  
\* Oracle 11g, Oracle 12c, MySQL 5.6, or MySQL 8.0  
\* Can be local, can be AWS RDS  
\* Angular 6 or newer  
\* Does not have to be as complex as how we have made Angular apps in the past;  
\* By all means lazily load modules, but that is not a requirement.  
\* Component reuse also is suggested, but not a requirement.  
\* Git  
\* My suggestion is to have a master branch with the most up to date code, and practice the creation of feature branches and merging them into master. Bonus points for pull requests.  
\* Again, my suggestion, not a requirement.  
\* Maven  
\* Logging to a file using the RollingFileAppender for log4j  
\* You might find [this article helpful](<https://www.baeldung.com/java-logging-rolling-file-appenders>)

## For David, Jean, Boris, Takumi and Chris  
\* The rest of you may also complete this if you would like. Knowledge of monitoring tools will only help you  
\* Cognizant wishes to speak with you again about joining the R2 Engineering Team  
\* [Sign up for a DynaTrace free trial](<https://www.dynatrace.com/trial/>) and install the *\_OneAgent\_* on your machine  
\* You might [find this link helpful](<https://www.dynatrace.com/support/help/technology-support/operating-systems/windows/>)  
\* It supports *\_out of the box\_* Java, JDBC, JVM/JMX, Garbage Collection, MySQL, Oracle DB, Tomcat, and [many more](<https://www.dynatrace.com/technologies/>)  
\* Using DynaTrace, please be able to show  
\* The SQL Query that takes the longest amount of time to complete  
\* The *\_action\_* which consumes the most CPU  
\* The *\_action\_* which consumes the most Memory  
\* I am defining an *\_action\_* as one leg of the critical path;  
\* for example, if the critical path is  
(a.) signing in  
(b.) starting a quiz  
(c.) completing a quiz, and  
(d.) viewing your score  
\* Then your job is to tell me which action (a, b, c, or d) consumes the most CPU and which consumes the most Memory