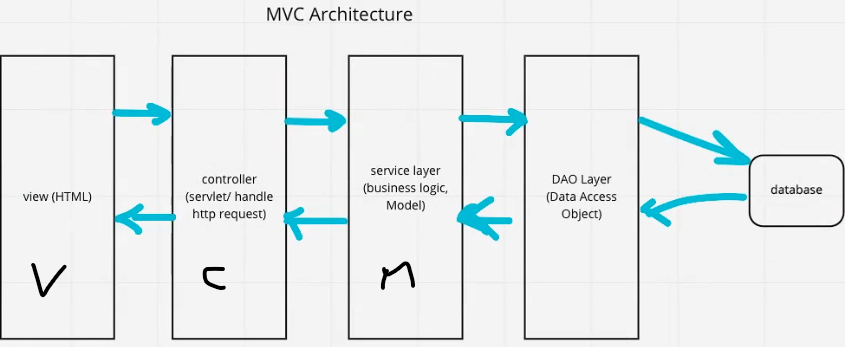
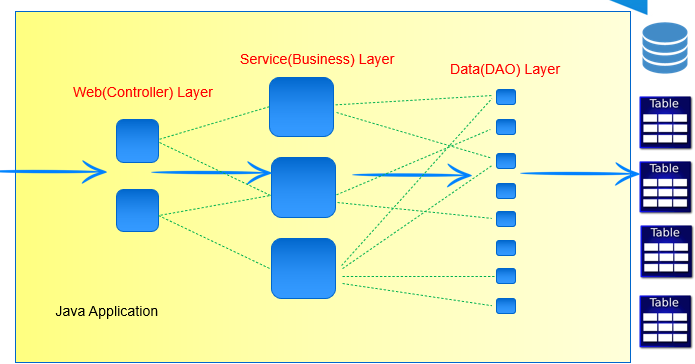
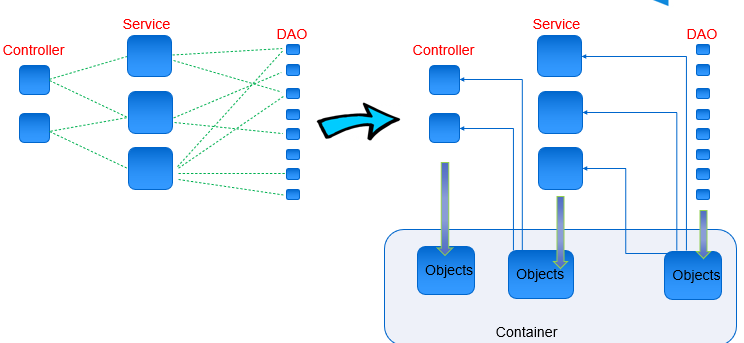
Checklist review today:

* Three Layer (tier) architecture





* IOC (Inversion of Control)



AOP (implement by dynamic proxy design pattern)

* Aspect oriented programming
* aspects enable the modularization of concerns such as transaction management, logging or security that cut across multiple types and objects (cross cutting corners)
* Aspect
  + a class that implements enterprise application concerns that cut across multiple classes, such as transaction management, logging, security
* Join Point
  + a point during the execution of a program, such as the execution of a method or the handling of an exception
* PointCut
  + a predicate that matches join points,
  + Advice is associated with a pointcut expression and runs at any join point matched by the pointcut
  + Spring framework uses the AspectJ Pointcut expression language
* Advice
  + before: advices that executes before a join point
  + after: advice to be executed regardless of the means by which a join point exits (normal or exceptional return)
  + after return: advice to be executed after a join point completes
  + after throwing: advice to be executed if a method exits by throwing an exception
  + around: advice that surrounds a join point such as a method invocation

Learned Plan tomorrow:

* + Review Spring MVC
    - dispatcherServlet
  + Review Spring Boot