~~1. Create a database in postgres or use h2 in memory database. Create 10 entity tables, where should be One-to-one, One-to-many, many-to-many relationships (join table won’t be counted as entity table). Create a DATABASE UML diagram. Upload your diagram with project as PDF file.~~ **~~FILE SHOULD BE LOCATED INSIDE YOUR PROJECT FOLDER.~~**

~~2. Upload database backup file with your project, if you use postgres database.~~ **~~Name your database – {$your\_variant\_{$your\_lastname}}.~~** ~~For example~~ **~~variant3\_Urmanov.tar~~**

**~~spring.datasource.url=jdbc:postgresql://localhost:5432/ variant3\_Urmanov~~**

~~3. Create Readme.MD file in project structure. In this file write your project's idea, functionality that you're going to implement etc. (https://github.com/tchapi/markdown-cheatsheet/blob/master/README.md~~

~~4. Use different type of beans annotations.~~

~~5. Use different type of Dependency Injections. (ONLY CONSTRUCTOR and Setter injection. NO FIELD injection)~~

**6. Write good service logic in service classes.** (If your most port of code will consist only calling repo method, -50% from your grade)

~~7. Use @PropertySource, @Lazy, @Scope, @DependsOn.~~

9. ~~Add at least 2 configuration classes.~~

~~10. Add AOP configuration. Use AspectJ annotation style.~~

11. ~~Use next annotations: @Before, @Pointcut, @After, @Aspect, @AfterReturning, @Around, @AfterThrowing.~~

~~12. Add real service/business logic in AOP code.~~

13. For your repository classes, use different and more complex methods/code for JdbcTemplate class. **DON’T USE JPA Repositories.**

14. Use batch operations.