

---

# **Sprint 3.MSD:** **Persistence Model** **Implementation**

S3 • 2020

---

# Overview

## Data Access Components Implementation:

- Persistence Model  
Implementation
- Repositories Services  
Implementations

## Course Coverage:

- Theme\_2: Course\_5
-

---

# Persistence Model Implementation

## JPA Entities

- Entities with
  - @Entity
  - @Inheritance
- Value Objects:
  - @Enumerated
  - @Convert

## JPA Aggregates

- Root Entity relationships. e.g.:
    - @OneToMany
      - mappedBy
      - fetch
      - Cascade = CascadeType.ALL
      - orphanRemoval = true
  - Component Entities relationships:
    - @OneToMany
-

---

# Repository Services' Implementation

## Spring Repositories

- Extending Spring Interface
  - JpaRepository<RootEntity, T>
- Adding custom operations, e.g.
  - findByPropertyName()
  - findByCustomFilter() with @Query to add custom filter

## (Bonus) Complete JPA Repository with

- IRepository CRUD interface
  - EntityRepositoryJPA implementation class with EntityManager injection as @PersistenceContext
-

---

# Business Rules Implementation

## Update existing Business Services and Factories

- to use JPA-based Repositories, either Spring Repositories or complete JPA repositories

## Implement computation or audit rules

- using @EntityListeners classes with @PrePersist, @PreUpdate, @PreRemove methods.
-

---

# Deliverables

persistence.xml

@Entity Classes

Repository definitions as

- Spring JpaRepository interfaces

(Bonus) Repository definitions as

- EntityRepository CRUD interfaces with
- EntityRepositoryJPA implementations

Spring App Configuration **Class** with JPA beans' factory methods for:

- EntityManagerFactory and
- JpaTransactionManager

JUnit Test **Classes**

- New tests for JPA Repository Services
- Updates Tests for Business Services