BIF5-SWEN3   # Semester-Project/Code-Review Rating Matrix

Team:
Student:
Date:

Title	Description	Max.
Functional-Requirements (20%):		
Use Cases	Understanding and Implementation of use-cases	5
REST API	Implementation of service-layers, converters, service-agents	5
Web-Frontend	UX and completeness (functionallity) of web frontend	5
additional Use Case		5
Non-Functional-Requirements:		
Queues	Implementation of async-communication between components	4
Logging	Appropriate logging implemented in all components/layers	2
Validation	Validation in all layers and components	2
Stability Patterns	Exception handling, layer-based exceptions, stability patterns implemented	2
Unit Tests	Implementation of Unit-Tests with Mocking and appropriate coverage	4
Integration Tests	End2End Tests of use-cases and critical paths	4
Clean-Code	SOLID principles, clean-code for high code quality used	2
Software Architecture:		
Packaging	Packaging and project-setup with containers (docker-compose, configs correct)	10
Loose Coupling	Implementation of interfaces, lose coupling	2
Mapper	Mapping between layers, usage of mapping framework	2
Dependency Injection	DI implemented, DI framework used	2
DAL	Implementation of persistence with ORM and Repository-Pattern (evtl. Unit-Of-	2
BL	Implementation of BL, entities, workflow, facade	2
Software Development Workflow:		
GitFlow	GitFlow based; branches used; pull-requests used	5
Issue Tracking	form of issue-tracking, kanban-board used and actual	5
CI/CD-pipelines	some sort of CI/CD automation used (e.g. GitHub Action), for test, linting,..	5
Documentation	critical aspects of the solution are documented (diagrams, README.md)	5
Code-Review (questions):		
Knowledge	Knowledge of the WHY/HOW/WHERE in the code.	
	Able to explain the background (theoretical) concepts	20

Total (in %):   **100**

> **Remarks:**
> - The score here is in percent and will be scaled to the max-points of the code-review in Moodle
>   e.g. Mid-Term Code-Review 100% → 25 points; Final Code-Review 100% → 40 points
> - You will get part score per item, if the task is not implemented complete or source quality is low

Points	Notes

0