ODSOFT 2021/2022 – Project self-assessment   TEAM ID: G4 0 1

Instructions: Please mark put an X on the non-greyed column that better reflects your project's item achievement. Fill the gaps (_____) when appropriate.
Analysis Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently 50%; 3 – Achieves partially, includes justification for main options; 4 – Achieves completely with justifications and alternatives.
Requirements Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently; 3 – Achieves with Minor Faults; 4 – Achieves completely including advanced topics.

Description	Analysis					Requirements				
	0	1	2	3	4	0	1	2	3	4
2.1 Base Pipeline – Overall Grading									●	
• The overall pipeline should be designed. This includes a high-level design for all the concerns;					●					
• The design should include a description of the process, including the Git organization (i.e., branching model) model to be adopted and how it relates to the pipeline;				●						
• The application should be completed with the specific features for the project:										
o Book concept; (or entity 1);				●						●
o Bookmark concept; (or entity 2);				●						●
o Tag concept; (or entity 3);				●						●
o Lease concept; (or entity 4);				●						●
o Book cannot be leased to different contacts at the same time; (or constraint 1);				●						●
o Contacts cannot lease more than three books at a time: (or constraint 2);				●					●	
o Tags with the same description should not be allowed; (or constraint 3);				●						●
• Unit tests are implemented, Unit Tests Report and Coverage Report are published on Jenkins;				●					●	
• Pit Mutation tests (or similar) are executed, and Mutation Coverage Report is published on Jenkins;				●					●	
• Generate the Project Report PDF file from your project's Readme.md. You may choose the technology to use. • **Used technology to generate PDF:** __MarkdownToPdfTask__								●		●
• The pipeline works both on Windows and Linux operating systems;					●					●
• Classify your final project according to the Maturity Model;					●					
• Implement the pipeline for the specifics of this concern. The base pipeline implementation should also include all the features addressed in Class Assignment – Part Two, using parallel stages when suitable. **Which steps are run in parallel?**									●	
• __Run unit tests__, __Run integration__, __Run Javadoc__, _____, _____;				●					●	
• __Run checkstyle__, __Run SpotBugs__, _____, _____, _____;				●					●	
• Publish unit tests, Publish integration tests, Publish mutation tests, Publish Javadoc, Publish jacoco, Publish Checkstyle, Publish spotbugs				●					●	
• _____, _____, _____, _____, _____;										
• _____, _____, _____, _____, _____;										

ODSOFT 2021/2022 – Project self-assessment   TEAM ID: G4 0 1

Instructions: Please mark put an X on the non-greyed column that better reflects your project's item achievement. Fill the gaps (_____) when appropriate.
Analysis Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently 50%; 3 – Achieves partially, includes justification for main options; 4 – Achieves completely with justifications and alternatives.
Requirements Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently; 3 – Achieves with Minor Faults; 4 – Achieves completely including advanced topics.

Description	Analysis					Requirements				
	0	1	2	3	4	0	1	2	3	4
2.2 Documentation and Database – Overall Grading | Student # 1210107 **Name:** Jorge Gonçalves				●					●	
• The specific stages for this concern of the pipeline should be designed;				●						
• Implement the pipeline for the specifics of this concern. Analysis results must be published and explicitly discussed in the project report;				●					●	
• The application should have a persistence layer that must use a relational database. You may choose the technology to use for this non-functional requirement; **In the CMS case, it should use an SGBD running on a different system than the CMS web application** using containerisation. • **Used SGBD:** ___H2_____			●						●	
• Generate and archive the project's Moodle zip submission file, containing the Project Report PDF, the Jenkinsfile and the generated artefacts for the build (e.g., Javadoc, reports, etc – excluding all temporary files);					●					●
o Jenkinsfile										●
o Project Report PDF										●
o Artefacts (please detail)										●
▪ source code ;										●
▪ documentation ;										●
▪ tests ;										●
▪ gradle ;										●
▪ _____ ;										
▪ _____ ;										
▪ _____ ;										
▪ _____ ;										
▪ _____ ;										
▪ _____ ;										
▪ _____ ;										
▪ _____ ;										
• The pipeline works both on Windows and Linux operating systems;										●
• **Advanced requirement:** The chosen application should be adapted to foreseen third-parties version releases. In the CMS case, explicitly document and adapt the current CMS solution for supporting Gradle's (v. 6.9.1 or superior (7.3)) latest release and latest JDK's release (v. 11 or superior).			●					●		

ODSOFT 2021/2022 – Project self-assessment · TEAM ID: G_4_0_1_

Instructions: Please mark put an X on the non-greyed column that better reflects your project's item achievement. Fill the gaps (_____) when appropriate.
Analysis Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently 50%; 3 – Achieves partially, includes justification for main options; 4 – Achieves completely with justifications and alternatives.
Requirements Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently; 3 – Achieves with Minor Faults; 4 – Achieves completely including advanced topics.

Item	A0	A1	A2	A3	A4	R0	R1	R2	R3	R4
2.3 Integration Tests and Code Quality – Overall Grading | Student #__1180813__ Name: _Miguel Sousa_			●							●
• The specific stages for this concern of the pipeline should be designed;				●		(greyed)				
• Implement the pipeline for the specifics of this concern. Analysis results must be published and explicitly discussed in the project report;			●							●
• The pipeline should include a "check" on the code quality of the project by using Checkstyle (or similar tool). Settings for code quality should be defined, including the thresholds for the build health. Analysis results must be published; **Used Thresholds:** _200-600_			●							●
• The pipeline should include a "check" on the code quality of the project by using FindBugs (or similar tool). Settings for code quality should be defined, including the thresholds for the build health. Analysis results must be published; **Used Thresholds:** _100-300_			●							●
• Integration tests should cover the specific features of the project. Integrations Tests Report and Coverage Reports should be published on Jenkins. Settings for integration test coverage should be defined, including the thresholds for the build health. The build should fail if coverage degrades more than the delta thresholds that should be configured; **Used Thresholds:** _9-9.5_			●							●
• The pipeline works both on Windows and Linux operating systems;				●						●
• **Advanced requirement:** Test the application for at least two different webserver versions. There should be a configuration file in the repository stating which versions of the webserver must be used and running the tests against more than two different versions should be supported:	●								●	
o Configuration file	(greyed)							●		
o Support more than two different webserver versions	(greyed)							●		
2.4 Staging, Smoke and Functional Testing – Overall Grading | Student #_____ Name: _____										
• The specific stages for this concern of the pipeline should be designed;						(greyed)				
• Implement the pipeline for the specifics of this concern. Analysis results must be published and explicitly discussed in the project report;										
• Acceptance tests with Cucumber and Selenium (or similar tools) should cover the specific features of the project. Acceptance Tests Report and Coverage Reports should be published on Jenkins;										
• These tests should be executed against a docker container with the application (do not forget that the application now has a database!);										
• The image for this docker container should be published in the docker hub;										
• The smoke test should be executed against a docker container with the application (do not forget that the application now has a database!);										
• The pipeline works both on Windows and Linux operating systems;										
• **Advanced requirement:** Run your chosen application acceptance tests using at least two different browsers. A configuration file, in the repository should state which browsers and versions must be used to run the acceptance tests. Use more than two different browser/versions should be supported:										
o Configuration file	(greyed)									
o Support more than two different browsers/versions in the same build	(greyed)									

ODSOFT 2021/2022 – Project self-assessment   TEAM ID: G4 0 1

Instructions: Please mark put an X on the non-greyed column that better reflects your project's item achievement. Fill the gaps (_____) when appropriate.
Analysis Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently 50%; 3 – Achieves partially, includes justification for main options; 4 – Achieves completely with justifications and alternatives.
Requirements Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently; 3 – Achieves with Minor Faults; 4 – Achieves completely including advanced topics.

**2.5 Non-Functional and Performance Testing– Overall Grading	Student #_____ Name: _____**									
• The specific stages for this concern of the pipeline should be designed;									
• Implement the pipeline for the specifics of this concern. Analysis results must be published and explicitly discussed in the project report;									
• Non-functional requirements should be tested.									
• Non-functional tests should be performed to check the system capacity using JMeter and a report should be published on Jenkins;									
• The pipeline works both on Windows and Linux operating systems;									
• **Advanced requirement**: run the capacity tests using tests written in Java (or other language) against the webserver. A Gradle configuration should state which tests should be run and write a report about the results, stating when the service will stop working as intended.									
o Use tests written in Java;									
o Configuration file;									
o Results report;									
**2.6 Continuous Deployment – Overall Grading	Student #_____ Name: _____**									
• The specific stages for this concern of the pipeline should be designed;									
• Implement the pipeline for the specifics of this concern. Analysis results must be published and explicitly discussed in the project report;									
• This should be the last stage of the pipeline. The idea is to simulate a live deployment to production;									
• This should simulate the production stage using a docker container;									
• It should be able to deploy the binaries/resources and upgrade the database if necessary;									
• It should be able to recover to the last "good" state if the upgrade fails;									
• There should be a tag in the git repository related to the release and it also should be possible to identify this version from the deployed artefacts;									
• The pipeline works both on Windows and Linux operating systems:									
• **Advanced requirement:** Perform a Blue/Green deployment of your application. For this task, the production environment should be running on at least three different machines i.e., the front-end tomcat proxy (or similar), the green production machine, and the blue production machine (do not forget that the application now has a database that should also be running on a different machine!).									

ODSOFT 2021/2022 – Project self-assessment   TEAM ID: G4 0 1

Instructions: Please mark put an X on the non-greyed column that better reflects your project's item achievement. Fill the gaps (_____) when appropriate.
Analysis Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently 50%; 3 – Achieves partially, includes justification for main options; 4 – Achieves completely with justifications and alternatives.
Requirements Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently; 3 – Achieves with Minor Faults; 4 – Achieves completely including advanced topics.

Project Report – Overall Grading					
• Evidences of continuous building of the pipeline (build history with increasing stages)					●
• Specifies the details of the design					●
• Specifies the technical details of the implementation					●
• Provides analysis of possible alternatives and the justification of the options.			●		

Teamwork Effort

- o Student #_1180813____; Accomplishment %_45____; (The sum of all percentages should equal 100%)
- o Student #_1210107____; Accomplishment %_55____; (The sum of all percentages should equal 100%)
- o Student #_____; Accomplishment %_____; (The sum of all percentages should equal 100%)
- o Student #_____; Accomplishment %_____; (The sum of all percentages should equal 100%)