ODSOFT 2021/2022: CA – Part 2 Self-Assessment TEAM ID: G_ _ _

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.

Implementation Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently; 3 – Achieves with Minor Faults; 4 – Achieves completely.

Description	**Analysis**					**Implementation**				
	0	1	2	3	4	0	1	2	3	4
Overall Team Grading										
• Each task has, at least, one issue in Bitbucket										
• Bitbucket issues are closed and refer to the commit										
• All source code is committed										
• Commit messages are clear and well formed										
• The repository readme.md is well structured, clear, and coherent with the work done										
• Issues have a clear explanation on the scope and differences to other issues										
• All pipelines must be able to run on any of the student's machines										
• An evaluation report on the performance of the CA component's, comparing the different execution times, was written (report.md)										
• Self-Assessment file is committed to the repository										
• Other issues:										
○ (____)										
○ (____)										
○ (____)										
○ (____)										
○ (____)										
○ (____)										
○ (____)										
○ (____)										
○ (____)										
Teamwork Effort										
○ Student #_____________; Accomplishment %______ ; (The sum of all percentages should equal 100%)										
○ Student #_____________; Accomplishment %______ ; (The sum of all percentages should equal 100%)										
○ Student #_____________; Accomplishment %______ ; (The sum of all percentages should equal 100%)										
○ Student #_____________; Accomplishment %______ ; (The sum of all percentages should equal 100%)										
○ Student #_____________; Accomplishment %______ ; (The sum of all percentages should equal 100%)										

ODSOFT 2021/2022: CA – Part 2 Self-Assessment TEAM ID: G_ _ _

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.

Implementation Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently; 3 – Achieves with Minor Faults; 4 – Achieves completely.

Description	Analysis					Implementation				
	0	1	2	3	4	0	1	2	3	4
Task #1 – Jenkins Sequential Build using "Promoted Build" or "Build Pipeline"										
• A sketch or schematic of the pipeline with a small description is provided										
• A Jenkins Pipeline using the **"Promoted Build"** or **"Build Pipeline"** plugin, performing a **sequential build**, is configured and described										
1. **Repository Checkout** - checkout the GIT repository										
2. **Generate and Publish deployment file** – deployment file is built and published on Jenkins										
3. **Javadoc -** the Javadoc is built and published on Jenkins										
4. **Unit Tests**										
a. The Unit Tests are executed and described										
b. The Unit Tests Report is generated and published on Jenkins										
c. The Unit Tests Coverage Report is generated and published on Jenkins										
5. **Integration Tests**										
a. The Integration Tests are executed and described										
b. The Integration Tests Report is generated and published on Jenkins										
c. The Integration Tests Coverage Report is generated and published on Jenkins										
6. **Mutation Tests**										
a. The Mutation Tests are executed and described										
b. The Mutation Coverage Report is generated and published on Jenkins										
7. **System Test**										
a. The application (e.g. .war file) is deployed to a pre-configured Server running offsite (e.g., Tomcat Server instance)										
b. An automatic smoke test is performed										
8. **UI Acceptance Manual Tests**										
a. An email notification is sent, regarding the successful execution of all the previous tests and asking to perform a manual test										
b. An UI Acceptance Manual Test is executed to cancel or proceed with the pipeline										
c. The pipeline must wait for a user manual confirmation on Jenkins										
9. **Continuous Integration Feedback** - a tag on the repository with the Jenkins build number and status is published										
• Jenkins's Job configuration files (Build Promotions and/or Build Pipelines jobs) are in the repository										
• The overall "build" fails if any of the stages fail										
• Each Gradle task is executed only once in the pipeline										

ODSOFT 2021/2022: CA – Part 2 Self-Assessment TEAM ID: G_ _ _

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.

Implementation Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently; 3 – Achieves with Minor Faults; 4 – Achieves completely.

Description	Analysis					Implementation				
	0	1	2	3	4	0	1	2	3	4
Task #2 – Jenkins Parallel Build using "Promoted Build" or "Build Pipeline"										
• A sketch or schematic of the pipeline with a small description is provided										
• A Jenkins Pipeline using the **"Promoted Build"** or **"Build Pipeline"** plugin, performing a **sequential build**, is configured, and described										
1. **Repository Checkout** - checkout the GIT repository										
2. **Generate and Publish deployment file** – deployment file is built and published on Jenkins										
3. **Javadoc -** the Javadoc is built and published on Jenkins										
4. **Unit Tests**										
a. The Unit Tests are executed and described										
b. The Unit Tests Report is generated and published on Jenkins										
c. The Unit Tests Coverage Report is generated and published on Jenkins										
5. **Integration Tests**										
a. The Integration Tests are executed and described										
b. The Integration Tests Report is generated and published on Jenkins										
c. The Integration Tests Coverage Report is generated and published on Jenkins										
6. **Mutation Tests**										
a. The Mutation Tests are executed and described										
b. The Mutation Coverage Report is generated and published on Jenkins										
7. **System Test**										
a. The application (e.g. .war file) is deployed to a pre-configured Server running offsite (e.g., Tomcat Server instance)										
b. An automatic smoke test is performed										
8. **UI Acceptance Manual Tests**										
a. An email notification is sent, regarding the successful execution of all the previous tests and asking to perform a manual test										
b. An UI Acceptance Manual Test is executed to cancel or proceed with the pipeline										
c. The pipeline must wait for a user manual confirmation on Jenkins										
9. **Continuous Integration Feedback** - a tag on the repository with the Jenkins build number and status is published										
• Jenkins's Job configuration files (Build Promotions and/or Build Pipelines jobs) are in the repository										
• The overall "build" fails if any of the stages fail										
• Each Gradle task is executed only once in the pipeline										

ODSOFT 2021/2022: CA – Part 2 Self-Assessment TEAM ID: G_ _ _

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.

Implementation Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently; 3 – Achieves with Minor Faults; 4 – Achieves completely.

Description	**Analysis**					**Implementation**				
	0	1	2	3	4	0	1	2	3	4
Task #3 – Jenkins Sequential Build using Scripted Pipeline										
• A sketch or schematic of the pipeline with a small description is provided										
• A Jenkins Pipeline using the **"Promoted Build"** or **"Build Pipeline"** plugin, performing a **sequential build**, is configured, and described										
1. **Repository Checkout** - checkout the GIT repository										
2. **Generate and Publish deployment file** – deployment file is built and published on Jenkins										
3. **Javadoc -** the Javadoc is built and published on Jenkins										
4. **Unit Tests**										
a. The Unit Tests are executed and described										
b. The Unit Tests Report is generated and published on Jenkins										
c. The Unit Tests Coverage Report is generated and published on Jenkins										
5. **Integration Tests**										
a. The Integration Tests are executed and described										
b. The Integration Tests Report is generated and published on Jenkins										
c. The Integration Tests Coverage Report is generated and published on Jenkins										
6. **Mutation Tests**										
a. The Mutation Tests are executed and described										
b. The Mutation Coverage Report is generated and published on Jenkins										
7. **System Test**										
a. The application (e.g. .war file) is deployed to a pre-configured Server running offsite (e.g., Tomcat Server instance)										
b. An automatic smoke test is performed										
8. **UI Acceptance Manual Tests**										
a. An email notification is sent, regarding the successful execution of all the previous tests and asking to perform a manual test										
b. An UI Acceptance Manual Test is executed to cancel or proceed with the pipeline										
c. The pipeline must wait for a user manual confirmation on Jenkins										
9. **Continuous Integration Feedback** - a tag on the repository with the Jenkins build number and status is published										
• Jenkins's Job configuration files are in the repository, as well as the scripted pipeline										
• The overall "build" fails if any of the stages fail										
• Each Gradle task is executed only once in the pipeline										

ODSOFT 2021/2022: CA – Part 2 Self-Assessment TEAM ID: G_ _ _

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.

Implementation Grading: 0 – No Submission; 1 – Attempt; 2 – Achieves Sufficiently; 3 – Achieves with Minor Faults; 4 – Achieves completely.

Description	Analysis					Implementation				
	0	1	2	3	4	0	1	2	3	4
Task #4 – Jenkins Parallel Build using Scripted Pipeline										
• A sketch or schematic of the pipeline with a small description is provided										
• A Jenkins Pipeline using the **"Promoted Build"** or **"Build Pipeline"** plugin, performing a **sequential build**, is configured, and described										
1. **Repository Checkout** - checkout the GIT repository										
2. **Generate and Publish deployment file** – deployment file is built and published on Jenkins										
3. **Javadoc -** the Javadoc is built and published on Jenkins										
4. **Unit Tests**										
a. The Unit Tests are executed and described										
b. The Unit Tests Report is generated and published on Jenkins										
c. The Unit Tests Coverage Report is generated and published on Jenkins										
5. **Integration Tests**										
a. The Integration Tests are executed and described										
b. The Integration Tests Report is generated and published on Jenkins										
c. The Integration Tests Coverage Report is generated and published on Jenkins										
6. **Mutation Tests**										
a. The Mutation Tests are executed and described										
b. The Mutation Coverage Report is generated and published on Jenkins										
7. **System Test**										
a. The application (e.g. .war file) is deployed to a pre-configured Server running offsite (e.g., Tomcat Server instance)										
b. An automatic smoke test is performed										
8. **UI Acceptance Manual Tests**										
a. An email notification is sent, regarding the successful execution of all the previous tests and asking to perform a manual test										
b. An UI Acceptance Manual Test is executed to cancel or proceed with the pipeline										
c. The pipeline must wait for a user manual confirmation on Jenkins										
9. **Continuous Integration Feedback** - a tag on the repository with the Jenkins build number and status is published										
• Jenkins's Job configuration files are in the repository, as well as the scripted pipeline										
• The overall "build" fails if any of the stages fail										
• Each Gradle task is executed only once in the pipeline										