D04 – Performance testing

# Identification data

\_\_4\_\_ \_\_B\_\_ \_\_https://github.com/rodddella/Acme-Planner/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Group Level GitHub repository

\_\_Seville (Spain), 07/06/2021\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Place, date

Imagen que contiene Texto

Descripción generada automáticamente\_\_Guillermo Diz Gil (Manager, developer, tester)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Student #1: Full name, roles, and signature

\_\_Francisco Rodríguez Pérez (Developer, tester)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Student #3: Full name, roles, and signature

Diagrama

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media\_\_Carmen María Muñoz Pérez (Developer, tester)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Student #2: Full name, roles, and signature

Icono

Descripción generada automáticamente\_\_George Laurentiu Bogdan (Developer, tester)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Student #4: Full name, roles, and signature

# Responsibility statements

[X] I am an author of this deliverable. I have not cheated in any way.

[X] I have collaborated with my partners on producing this deliverable; in other words, neither have I ridden their coattails nor gobbled them up.

[X] I have learnt from working on this deliverable, so that I can pass my control check.

[X] I understand that my deliverable will be considered failed if I fail to meet any of its requirements or if I fail to submit it by the deadline.

[X] I understand that I must have a contingency plan and that submitting my deliverable very close to the deadline is likely to result in disaster.

[X] I have taken into account the guidelines provided in the lectures and document “On your deliverables.pdf”, which are available at the USE’s e-learning platform.

# Deliverable and requirements

### Item 1: reports

* Workgroup report: produce a report in which you introduce your workgroup members, including their full names, recent picture, corporate email, roles played, and a global assessment of their performance.
* Work plan report: produce a report in which you list the tasks performed to produce this deliverable. For each task, you must provide a title, a short description, worker name(s), start and end dates, and total number of hours spent. Note that the report must include all of the tasks since you started working on the project; note, too, that it must report on actual tasks and times, not estimations.
* Budget report: produce a report with your budget, which must take the amortisation of your computers and the personnel costs into account. To compute the amortisation, assume that it is linearly distributed along three years. To compute the personnel costs, assume that a manager’s or an analyst’s work hour costs approximately €25.00 and the other roles’ work hour costs approximately €15.00. Note that the report must take all of the tasks into account and that it must report on actual costs, not estimations.
* Progress report: produce a report in which you describe how the work in your workgroup has gone on. Please, do not write on your subjective impressions; make a point of writing a concise and objective progress report.
* Lint report: produce a report in which you explain the bad smells found by Sonar Lint and why they are not problematic. Note that, ideally, this report must be empty because Sonar Lint should not be able to find any bad smells in your code. In the cases in which Sonar Lint finds a bad smell, you must analyse it; it may be the case that Sonar Lint is right, which means that you have to correct the problem; in the exceptional case in which Sonar Lint is not right, you must clearly justify the reason why.
* Change-log report: produce a report in which you describe the mistakes that you’ve found in your previous deliverable and how you have corrected them.
* Cloud services report: produce a file in which you provide a link to your GitHub repository and a link to your Clever Cloud application. Provide your credentials if you created an account for your deliverable; otherwise, wait for instructions. Do not share your personal credentials.
* Data model report: produce a report with a model for the data managed in project Acme Planner at the desired level. Please, make sure that the model includes all of the constraints in the requirements and the ones that result from analysing it.
* Features model report: produce a report with the features models of project Acme Planner at the desired level. Please, make sure that the model includes all of the features, actions, and constraints in the requirements and the ones that result from analysing it.
* Performance report: produce a report in which you analyse the performance of your project on two different computers. The report must include confidence intervals at the 95% confidence level for the mean of the wall time taken to serve the requests in your test cases plus a hypothesis contrast that makes it clear whether the performance on both computers can considered equal or not at the 95% confidence level. The report must include a clear description of the procedure that led to the results and detailed explanations regarding the interpretation of the results.

### Item 2: project

* Implement the requirements in the Acme Planner project at the desired level.
* Implement a suite to perform functional and performance testing. Your suite must fulfil the following requirements so that it can be accepted for evaluation:
  + The instruction coverage reported by the Coverage Runner plugin must be at least 60% in every feature, if possible. In the cases in which it is not possible, justify it clearly.
  + Every feature must have at least a positive test case per action. This is always possible.
  + Every feature must have at least a negative test case per action, if possible. In the cases in which it is not possible to implement it, justify it clearly.
  + Every test case must include a header comment in which you make it clear the feature/action that it tests (including the constraints that are violated, if any) and the expected results.