NORTHERN ARIZONA
UNIVERSITY
College of Engineering, Forestry & Natural Sciences

SCHOOL OF INFORMATICS, COMPUTING, AND CYBER SYSTEMS
CS 386 – Software Engineering
Prof. Marco Gerosa

Team Project – D.7 V&V

Deadline: Friday, April 20 11:59 pm

Grading: 3 points

This deliverable should describe the verification and validation of your system. Structure your deliverable according to the following sections. See the “Team Project Instructions” for details about formatting. Check the lecture materials and perform additional research to produce a high-quality deliverable.

1. Description

Provide 1-2 paragraphs to describe your system. This will help us to remember what your system is about.

2. Verification (tests)

Verification is about certifying that you are developing right the product. For release 2 of your system, all features should have automated tests. For this deliverable, provide examples of a unit test, an integration test, and an acceptance test.

2.1. Unit test

Provide a unit test that makes use of Mock objects. Include the GitHub links to the class being tested, the test case, and the mock objects. Also, include a print screen showing the test execution.

2.2. Integration test

Provide an integration test. Include the GitHub link to the test and a print screen showing the test execution. Explain what parts of the system your integration test is testing.

2.3. Acceptance

Provide a GitHub link to an automated acceptance test and a print screen or video that shows it execution. Some platforms make automated acceptance tests hard to implement. If it is your case, provide a justification and a script for a manual acceptance test instead.

3. Validation (user evaluation)

Validation is about certifying that you are developing the right product. The user inception and the requirement elicitation were based on empirical studies with users and stakeholders. Now it is time to check if you are on the right track conducting some user evaluation on the actual system.

In this deliverable, present the script you've used for the user evaluation and its results. The script should contain the tasks that you will give to the user, what you are going to collect, and what you are going to ask. Don't forget to add some questions about the users' general impressions. You can ask open questions (e.g. How would you describe the homepage of our app? How do you compare our system to the competitor X?) or closed questions (On a scale of 1 to 10, how would you rate the layout of our application? On the same scale, how likely you would use the system in its current state?). Take a look on the inception and requirements deliverables to help to create the script (aim to check if you are achieving your initial goals and if the features are implemented in a satisfactory way). Conduct the user evaluation with at least 3 users.

In terms of results, report the data that you collected and reflect about what you observed. Some questions that you can explore: What features worked well? What can be changed? How is the learning curve of your system? Did the user preformed the tasks as you expected? Did the user actions produce the results they expected? What the users liked the most? Is your value proposition accomplished?

4. **Group participation**
Provide this section as described in the "Team Project Instructions."

Feedback: If you have any suggestions about this document or the course, please send them via this online form: https://goo.gl/forms/peCmYZ0fXiIgzwS12