

# **AA 2017-2018 Software Engineering 2—I&T assignment goal, schedule, and rules**

***READ THIS VERY CAREFULLY  
NO EXCUSE FOR IGNORING WHAT WE WRITE HERE***

## **1 Goal and approach**

The goals of this assignment are:

- Achieve a running prototype implementation of Travlendar+.
- Test your prototype possibly using some of the automated tools that will be presented in class.
- Evaluate through acceptance testing the prototype implemented by another group.

You will deliver two artefacts:

- The Implementation and Test deliverable (ITD) that will include the code you have developed, a document describing the structure of the code, the adopted development frameworks, the installation instructions, and information on how you performed your tests.
- The Acceptance Test deliverable (ATD) in which you will describe the focuses of your validation activity and the results you have obtained.

These two deliverables will be reviewed, together with RASD and DD, during the final discussions that will take place during the winter exam sessions according to a schedule that will be proposed in the forthcoming months. The evaluation will assess the quality of the artifacts you prepare (accuracy, completeness, soundness) and the quality of your presentation (if you are able to explain your point in an appropriate way and if your presentation fits in the allowed time). Please check the introduction to the course for more information on the evaluation criteria for the mandatory project.

## **2 Project schedule**

- Registration deadline 8/11/2017
- ITD submission deadline 7/01/2018
- ATD submission deadline 14/01/2018
- Final presentation (to be scheduled)

All deadlines are assumed to expire at **23:59** of the days listed above.

## **3 Rules**

- This assignment is optional and replaces the written exam of the Software Engineering II course.
- The assignment has to be developed by the same groups of the mandatory projects. Only groups of two or three persons can participate.

- In the case a group is not able to complete the assignment, the students in the group can take the written exam to obtain a final score for the project. Should this happen, you are required to inform your reference professor before the deadline for enrolling to the written exam.
- Each group MUST register to the assignment by filling in the following form <https://goo.gl/forms/hvufh7ZihlqGeTlo2>. Do not forget to include in the form all relevant data!
- You should continue using the same github repository you used for the mandatory project. For the new items to be delivered create a specific working folder. Have, however, all released deliverables in the same DeliveryFolder directory you used for the documents of the mandatory project.
- We would like to see that you use the repository not only to upload the final versions of your deliverables, but also to commit your intermediate versions. We would like to see commits performed by all group members.
- Each group MUST provide the requested artifacts within the stated deadlines. A delay of a few days, if notified in advance to the reference professor, will be tolerated but it will also result in a penalty in the final score. It is mandatory to provide these artifacts and to present them to the reference professor in a final meeting that will be scheduled later.
- The developed code should be properly commented.
- During the development of the assignment each group will keep track of the number of hours each group member works toward the fulfillment of each deadline.
- For any question related to the project that could be interesting also for the other groups, please use the forum available on the Beep website. We will answer as promptly as possible.

## 4 What we require

Concerning the implementation, you are required to develop what could be considered the first version of Travlendar+. This software should be coherent with your RASD and DD and should include the set of functionalities that you consider to be essential for engaging your future users. Nice-to-have features could be disregarded in this implementation. Privilege stability and quality of your code rather than the implementation of a large quantity of features. Referring to the mandatory project assignment, groups composed of two students are required to provide some of the features related to flexible lunches and groups of three students are required to offer some of the mechanisms for trip arrangements.

We do not impose any requirement on the frameworks adopted for the implementation, but we expect that you will state explicitly the rationale for your choice (as part of the document accompanying the code) and the advantages and drawbacks implied by your choice.

More in detail, for what concerns ITD, we would like to see:

- The code ready for installation and execution (the code could be accompanied by a Maven file, a zip file, anything you think is useful for the purpose).
- A document that contains:
  1. The front page that identifies your team, the document, and the software (simply a link to the file/directory that includes what needs to be installed plus a link to the source code)
  2. Introduction and scope of the document

3. The requirements/functionalities that are actually implemented in the software (with motivations for including them and excluding others if applicable)
4. Adopted development frameworks (this section could either include simply references to sections in the DD where we find information about your choices and/or any new choice that you have made):
  - a. Adopted programming languages and its advantages and disadvantages
  - b. Any middleware you have adopted and its advantages and disadvantage
  - c. Any API you did not include in the DD
5. The structure of the source code
6. Information on how you performed your testing: you can refer to what you have in the DD and extend it as needed. Describe the procedure that you have used if not described in the DD. At least for system test, describe the main test cases that you have considered and their outcome
7. The installation instructions: anything that is useful for us and for the other groups to install and run your software, any prerequisite, ...

For what concerns ATD:

- The front page that identifies your team and the document
- The identification of the project you have analysed (names of authors, link to the repository, main reference documents you have considered)
- Installation setup: What you did to install the prototype as well as any problem you have faced in this phase, any incoherency you may have found when following the documentation accompanying the software
- The acceptance test cases you have applied and the corresponding outcome: you can extract test cases by analysing the RASD the other team has developed, the list of features that the team has actually developed as well as any other case that you, as a user of the application, think is appropriate (please provide motivation for your choice)
- Any additional point you want to raise on the quality of documentation and code