A.Y. 2025-2026 Software Engineering 2
I&T assignment goal, schedule, and rules

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

Table of contents

1 Goal and approach

The goals of this assignment are:

- Achieve a running prototype implementation of Best Bike Paths (BBP).
- Test your prototype.
- Evaluate through acceptance testing the prototype implemented by another group.

You will deliver two artifacts:

- 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 acceptance testing activity you will perform on the prototype of another group and the results you will obtain.

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 allotted time). Please check the introduction to the course for more information on the evaluation criteria for the I&T project.

2 Project schedule

- Registration deadline: 23/12/2025
- ITD submission deadline: 01/02/2026
- Distribution of the acceptance testing assignments: 02/02/2026
- ATD submission deadline: 08/02/2026
- Final presentation: to be scheduled

All deadlines are assumed to expire at **23:59** (Rome Time).
Note: You can submit before the deadlines, if you want/need.

3 Rules

- This assignment is optional and replaces, together with the R&DD assignment, the written exam of the Software Engineering 2 course, entirely.
- The assignment must be developed by the same groups of the R&DD project. 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 (WE1) 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://forms.office.com/e/kT9TeHqaYN. Do not forget to include in the form all relevant data!
- You should continue using the same GitHub repository you used for the R&DD 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 R&DD 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. These artifacts, as well as a demo of the system, will be presented to the reference professor in a final meeting that will be scheduled later.
- The developed code should be properly commented.
- The acceptance testing phase will start as soon as your reference professor will send you the information concerning the other prototype to be evaluated for acceptance testing.
- 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 Webeep website. We will answer as promptly as possible.

4 Code of Conduct for the use of Generative AI tools

You are welcome to use any tools that support the development of high-quality software and documentation. However, please keep in mind the following key principles:

1. **Responsibility for content**
   You are solely responsible for the entire content of your software and documents, including every line of code, all text, UML diagrams, figures, and references. While you may use any tool to assist in your work, you must ensure that all material is correct, original, and fully understood by you. During the project discussion, you should be able to explain, justify, and defend every aspect of your work in detail.

2. **Use of Generative AI**
   Generative AI tools—including, but not limited to, Large Language Models (LLMs)—can be useful for certain supporting tasks, such as improving writing style, generating summaries, suggesting code snippets, or drafting diagrams. However, they cannot replace your own reasoning or development work.

If you use Generative AI tools in your work, you must include a dedicated section in your ITD and ATD documents describing:

- Which tool(s) you used (e.g., specific chatbots, such as ChatGPT, Gemini);
- The inputs you provided (e.g., prompts, datasets, source materials, parameters, or constraints);
- The outputs you obtained;
- How you verified, refined, and integrated those outputs into your project.

Important Notes:

- Some tools—including free or online ones—may store input data for future model training. Use them cautiously and ensure that privacy and confidentiality are maintained.
- Generative AI tools may produce incorrect or misleading ("hallucinated") outputs, including errors in code, diagrams, or figures. It is your responsibility to verify the accuracy and integrity of all generated material.

All tools must be used ethically and responsibly. Compliance with this code of conduct will be evaluated, and violations may affect your final grade, up to and including failure of the project in severe cases.

5 What we require

The implementation project is a follow-up of the R&DD BBP project.

Groups composed of **two students** will implement the portion of the BBP system offering the features assigned in the R&DD part to single students.

Groups of **three students** can choose to implement the portion of the BBP system offering the features assigned in the R&DD part to groups composed of two students, with the exclusion of the part concerning the interaction with the accelerometer and gyroscope sensors, or could skip entirely the automated mode data collection and focus on the aggregation and selection of data coming from different users (last part of the R&DD assignment for groups of three).

The software should be coherent with your RASD and DD and should include the set of functions that you consider to be essential for engaging your future users. In other terms, we are not expecting a market-ready product, but a viable prototype. 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.

We do not impose any requirements 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, or a zip file, or 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/functions 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 their advantages and disadvantages.
    b. Any middleware you have adopted and its advantages and disadvantages.
    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, ...
8. Effort Spent: In this section you will include information about the number of hours each group member has worked for this document.
9. References.

For what concerns **ATD**, we expect a document with the following structure:

1. The front page that identifies your team and the document.
2. The identification of the project you have analyzed (names of authors, link to the repository, main reference documents you have considered).
3. Installation setup: What you did to install the prototype as well as any problem you have faced in this phase, any incoherence you may have found when following the documentation accompanying the software.
4. The acceptance test cases you have applied and the corresponding outcome: you can extract test cases by analyzing 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).
5. Any additional point you want to raise on the quality of documentation and code.
6. Effort Spent: In this section you will include information about the number of hours each group member has worked for this document.
7. References.