A.Y. 2022-2023 Software Engineering 2
Requirement Engineering and Design Project: goal, schedule, and rules

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

Table of contents

1 Goal and approach

The objective of this project is to apply in practice what you learn during lectures with the purpose of becoming familiar with software engineering practices and able to address new software engineering issues in a rigorous way. The project includes two assignments:

1. The preparation of a Requirement Analysis and Specification Document (RASD) for a problem we provide you.
2. The definition of the Design Document (DD) for the system considered in point 1 above.

The two assignments will be reviewed 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 (accurateness, 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 R&DD project. The two assignments are described in the rest of this document.

2 Project schedule

- Group registration deadline: 23/10/2022
- RASD submission deadline: 23/12/2022
- DD submission deadline: 08/01/2023
- 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 part of the written exam of the Software Engineering 2 course.
- The project is developed in groups of two or three persons. Groups composed of a single student are allowed even if strongly discouraged. The assignments and the corresponding expectations of the professor will be calibrated based on the size of the group.
- Each group interested in taking the project must register itself following the steps listed in Section 4. "Mixed" groups involving students of the three sections are allowed. When registering, these groups must indicate a single "reference" professor. This will be the one holding the discussion at the end of the course and deciding the grade. The choice is up to the students, but if we will realize that there is an unbalance among the groups under the responsibility of each of us, we may change your reference professor. In this case, we will inform you a few days after the group registration deadline.
- 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 will be presented to the reference professor in a final meeting that will be scheduled later.
- Each group MUST release artifacts by committing them into a specific folder of the git repository created for the project (see the following section).
- Each group MUST use the repository not only to upload the final versions of deliverables, but also to commit intermediate versions. We want to see commits performed by all group members. In the case a group wants to use collaborative writing tools (e.g., Google Docs) that keep track of individual contributions, the group will include in the readme file associated with the git repository the link to the online document, making sure that through that link the reference professor will be able to inspect the contributions to the document.
- The material included in your artifacts is not fixed in stone. You can (and are encouraged to) provide updates at any point before the final submission deadline (08/01/2023), if you think these are needed.
- During the development of the project 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 Group registration and organization of your repository

You should form your group and register it by going through the following steps:

1. Create a private repository for your project on GitHub (https://github.com). Please note that, as students, you have the possibility to create a private GitHub repository for free. Your repository should be named by combining the names of all group members. For instance, BianchiRossiVerdi will be the name of the repository of the group composed of the students Tommaso Bianchi, Maria Rossi e Veronica Verdi. Make sure that all group members have a Github account and have access to the repository (write permission). Moreover, invite your reference professor (GitHub accounts *matteocamilli* for Prof. Camilli, *dinitto* for Prof. Di Nitto, and *matteo-g-rossi* for Prof. Rossi) to access your repository (read permission is sufficient). Note: in past years we

noticed that GitHub has set some restrictions concerning the creation of repositories from accounts of people from specific countries. If you are experiencing this problem, you can use Bitbucket instead (https://bitbucket.org/product/). We do not suggest this one for all groups because it has other limitations concerning the maximum number of private repositories you can have.

2. Register your group by filling in the following form https://forms.office.com/r/ciHrwbTtsd. Do not forget to include in the form all relevant data!
3. Create a directory for each of the documents you will be working on.
4. Moreover, create a directory called *DeliveryFolder* where, by the due deadlines, you will commit and push the pdf version of your documents (name it RASD1.pdf or DD1.pdf, depending on the document you are releasing) plus any additional file you may want to include (e.g., the Alloy model and/or any UML model).
5. After the submission deadline, should you need to update your document, you can commit in the same folder another pdf file with an increased version number, e.g., RASD2.pdf. The new file should include a section that describes the performed changes.

5 The problem: eMall – e-Mobility for All

Electric mobility (e-Mobility) is a way to limit the carbon footprint caused by our urban and sub-urban mobility needs. When using an electric vehicle, knowing where to charge the vehicle and carefully planning the charging process in such a way that it introduces minimal interference and constraints on our daily schedule is of paramount importance.
e-Mobility Service Providers (eMSPs) offer to end users the possibility to:

1) know about the charging stations nearby, their cost, any special offer they have;
2) book a charge in a specific charging station for a certain timeframe;
3) start the charging process at a certain station;
4) notify the user when the charging process is finished;
5) pay for the obtained service.

Charging stations are owned and managed by Charging Point Operators (CPOs). Each CPO has its own IT infrastructure administrated through the so-called Charge Point Management System (CPMS). The CPMS handles the acquisition of energy from external (3rd party) Distribution System Operators (DSOs) and distribute it to the connected vehicles, making decisions, such as the amount of energy to be used for each connected vehicle. CPOs can dynamically decide from which DSO to acquire energy, for instance, depending on the current price and/or on the used mix of energy sources and, based on these, can dynamically decide the cost of a charging and set special offers. If batteries are available at the charging station, a CPO can also decide whether to store or not energy and whether to use the energy available in the batteries instead of acquiring it from DSOs. These decisions can be handled either manually by human operators or automatically by CPMSs.
CPOs offer the following main functions through their CPMSs:

a) know the location and "external" status of a charging station (number of charging sockets available, their type such as slow/fast/rapid, their cost, and, if all sockets of a certain type are occupied, the estimated amount of time until the first socket of that type is freed);
b) start charging a vehicle according to the amount of power supplied by the socket, and monitor the charging process to infer when the battery is full;

c) know the "internal" status of a charging station (amount of energy available in its batteries, if any, number of vehicles being charged and, for each charging vehicle, amount of power absorbed and time left to the end of the charge);

d) acquire by the DSOs information about the current price of energy;

e) decide from which DSO to acquire energy (if more than one is available);

f) dynamically decide where to get energy for charging (station battery, DSO, or a mix thereof according to availability and cost).

The interaction between the various providers (eMSPs, CPOs, and DSOs) occurs through uniform APIs. Thanks to this, an eMSP can interact with multiple CPOs and, on the other side, a CPO can interact with multiple eMSPs. Hence, users can exploit a large variety of charging options. Similarly, a CPO can interact with multiple DSOs and vice versa.

6 Project Scope

You have been contacted to define the RASD and DD for the eMall case. The task is different for groups of different sizes. More specifically:

- **Groups composed of a single student** will focus on the eMSP and the CPMS. The eMSP subsystem will offer the features from 1) to 5) listed in the previous section. The CPMS will offer the features a) and b). The eMSP will interact with the CPMS of a single CPO.
- **Groups composed of two students** focus on the eMSP and the CPMS. The eMSP subsystem will offer the features from 1) to 5) listed in the previous section. The CPMS will offer the features a) to f). An eMSP can interact with multiple CPMSs, each one owned by a different CPO. The interaction approach between eMSP and CPMSs (synchronous, asynchronous, a mixture between the two) will be a focus of the project as well.
- **Groups of three students** will focus on the same aspects addressed by groups of two. Moreover, they will also make the eMSP software smarter. This consists in the possibility for the eMSP to proactively suggest the user to go and charge the vehicle, depending on the status of the battery, the schedule of the user (this implies that the eMSP application can get access to the calendar of the user and his/her navigation system), the special offers made available by some CPOs, and the availability of charging slots at the identified stations.

Examples of companies working in this business area

Ionity https://ionity.eu/

Enel X https://www.enelx.com/n-a/en.html

Becharge https://www.bec.energy/en

Fastned https://fastnedcharging.com/en/

Atlante https://atlante.energy/

Interesting references

Platform for Electromobility. EV Charging: How to tap in the grid smartly? May 2022 https://www.platformelectromobility.eu/2022/05/17/ev-charging-how-to-tap-in-the-grid-smartly/

F. Campos, L. Marques, and K. Kotsalos, Electric Vehicle CPMS and Secondary Substation Management. 2nd E-Mobility Power System Integration Symposium, Stockholm, Sweden, 15 October 2018. https://mobilityintegrationsymposium.org/wp-content/uploads/sites/10/2018/11/4A_3_Emob18_024_paper_Filipe_Campos.pdf

Shu Su, Hui Yan, and Ning Ding. 2018. Machine Learning-Based Charging Network Operation Service Platform Reservation Charging Service System. In Proceedings of the 2018 International Conference on Signal Processing and Machine Learning (SPML '18). Association for Computing Machinery, New York, NY, USA, 1–5. https://doi.org/10.1145/3297067.3297078

Jan Mrkos, Antonín Komenda, and Michal Jakob. 2018. Revenue Maximization for Electric Vehicle Charging Service Providers Using Sequential Dynamic Pricing. In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '18). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 832–840.

7 The documents to be created

Each document you produce will include the following elements:

- **A FRONT PAGE** that includes the project title, the version of the document, your names and the release date.
- **A TABLE OF CONTENTS** that includes the headers of the first three levels of headings in your document, with the corresponding page number. At the beginning of this document, you find a table of contents that you can use as an example. Since in this document there are no level three headings (e.g., 3.1.1), they are not part of the table of contents.

The specific characteristics of each document are described in the next subsections.

7.1 Assignment 1 - RASD

The *Requirements analysis and specification document (RASD)* contains the description of the scenarios, the use cases that describe them, and the models describing the requirements and specification for the problem under consideration. You are to use a suitable mix of natural language, UML, and Alloy. Any Alloy model should be validated through the tool, by reporting the models obtained by using it and/or by showing the results of assertion checks. Of course, the initial written problem statement we provide suffers from the typical drawbacks of natural language descriptions: it is informal, incomplete, uses different terms for the same concepts, and the like. You may choose to solve the incompleteness and ambiguity as you wish but be careful to clearly document the choices you make and the corresponding rationale. You will also include in the document information on the number of hours each group member has worked towards the fulfillment of this deadline. As a reference structure for your document, you should refer to the one reported below that is derived from the one suggested by IEEE. Please include in the document information about the effort spent by each group member for completing this document.

1. **INTRODUCTION**
   A. *Purpose*: here we include the goals of the project

B. *Scope*: here we include an analysis of the world and of the shared phenomena
C. *Definitions, Acronyms, Abbreviations*
D. *Revision history*
E. *Reference Documents*
F. *Document Structure*

2. **OVERALL DESCRIPTION**
   A. *Product perspective*: here we include scenarios and further details on the shared phenomena and a domain model (class diagrams and state diagrams)
   B. *Product functions*: here we include the most important requirements
   C. *User characteristics*: here we include anything that is relevant to clarify their needs
   D. *Assumptions, dependencies and constraints*: here we include domain assumptions
3. **SPECIFIC REQUIREMENTS:** Here we include more details on all aspects in Section 2 if they can be useful for the development team.
   A. *External Interface Requirements*
      A.1 *User Interfaces*
      A.2 *Hardware Interfaces*
      A.3 *Software Interfaces*
      A.4 *Communication Interfaces*
   B. *Functional Requirements*: Definition of use case diagrams, use cases and associated sequence/activity diagrams, and mapping on requirements
   C. *Performance Requirements*
   D. *Design Constraints*
      D.1 *Standards compliance*
      D.2 *Hardware limitations*
      D.3 *Any other constraint*
   E. *Software System Attributes*
      E.1 *Reliability*
      E.2 *Availability*
      E.3 *Security*
      E.4 *Maintainability*
      E.5 *Portability*
4. **FORMAL ANALYSIS USING ALLOY:** This section should include a brief presentation of the main objectives driving the formal modeling activity, as well as a description of the model itself, what can be proved with it, and why what is proved is important given the problem at hand. To show the soundness and correctness of the model, this section can show some worlds obtained by running it, and/or the results of the checks performed on meaningful assertions.
5. **EFFORT SPENT:** In this section you will include information about the number of hours each group member has worked for this document.
6. **REFERENCES**

7.2 Assignment 2 - DD

The *Design document (DD)* must contain a functional description of the system, and any other view you find useful to provide. You should use all the UML diagrams you need to provide a full description of

the system. Alloy may also be useful, but not mandatory. You will also include information on the number of hours each group member has worked towards the fulfillment of this deadline. As a reference structure for your document please refer to the following one:

1. **INTRODUCTION**
   - A. *Purpose*
   - B. *Scope*
   - C. *Definitions, Acronyms, Abbreviations*
   - D. *Revision history*
   - E. *Reference Documents*
   - F. *Document Structure*
2. **ARCHITECTURAL DESIGN**
   - A. *Overview*: High-level components and their interaction
   - B. *Component view*
   - C. *Deployment view*
   - D. *Runtime view*: You can use sequence diagrams to describe the way components interact to accomplish specific tasks typically related to your use cases
   - E. *Component interfaces*
   - F. *Selected architectural styles and patterns*: Please explain which styles/patterns you used, why, and how
   - G. *Other design deci*sions
3. **USER INTERFACE DESIGN**: Provide an overview on how the user interface(s) of your system will look like; if you have included this part in the RASD, you can simply refer to what you have already done, possibly providing here some extensions if applicable.
4. **REQUIREMENTS TRACEABILITY**: Explain how the requirements you have defined in the RASD map to the design elements that you have defined in this document.
5. **IMPLEMENTATION, INTEGRATION AND TEST PLAN**: Identify here the order in which you plan to implement the subcomponents of your system and the order in which you plan to integrate such subcomponents and test the integration.
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**