

FAF.PSI16.1 Fall 2023 PBL Project Guideline

Mentors: Mariana Catruc, Gabriel Zaharia, Vasile Drumea.

Preliminaries

So far, during previous semesters each project has emphasized different aspects or concerns relevant for a software engineering project. Now, having this background it all can be used for creating the design of the final project of this learning program.

The term design is a fairly abstract one. It is usually associated with graphical design through visuals, diagrams and other stuff like that, but it includes more than that. The design phase of a project actually represents multiple things like coming up with the characteristics/requirements for a project either functional or non-functional. It is also important during this phase to do the necessary discovery and establish a plan of action. Later, for sure it is necessary to have visual representations of the structure and behavior of the system, preferably using the Unified Modeling Language (UML).

Below are listed the general requirements for this semester. In case of anything missing or misunderstandings reach out to the mentors to clarify everything ASAP.

1st Midterm Requirements

For the first midterm the students will be working on the following things:

1. Come up with a project proposal:

- Project title: It should contain or give details about the following:
 - Type of the system;
 - Problem statement;
 - Intended audience or domain (if applicable).
- Project goal: It is basically a statement which explains how the problem can be solved;
- Objectives: A list of objectives, not necessarily technical, that represent what needs to be accomplished at the end of the project.

2. Defining the requirements of the project:

- Functional (What the project does?): Functions of the system;
- Non-Functional (How the project does what it does?): Characteristics of the system (e.g. performance, usability etc.).

3. Creating a high level architectural model of the software:

- Use Case: how many are needed to cover all actors and use cases;

- Sequence: one for each relevant functionality of the system;
- Activity: same as ↑;
- State Machine: in case you have entities with state transitions;
- Class: in case you have data models with relations;
- Component/Deployment: how many the project needs;
- Other diagrams that you need.

2nd Midterm Requirements

The second midterm will be organized in the following way:

- Students need to take part in a project that is different from the one that was chosen for the Bachelor's thesis. The reason why we need to do this is to supposedly cover work time that should have been invested in the project, and also for diversity, which might help with achieving some of the goals of this subject (i.e. PSI), without putting too much emphasis on the thesis. Now, this extra effort can be done in the following 2 ways:
 - Taking part in the Hackathon for Anti Human Trafficking solutions (24/10/2023);
 - Working in teams on an additional project which is supposed to have the same weight as the projects developed during the Hackathon, according to the following requirements:
 - * Find a problem that needs to be solved. It can be the one of the problems from the Hackathon as well.
 - * Students will have to work in teams and present the following things on 5th December at the PSI class:
 - A usual project presentation. The slides should represent the project.
 - An MVP which is supposed to help solving the problem which initiated the project.
- In order to be admitted to the second midterm, the previous requirements has to be fulfilled. This also means that all the students who have taken part in the Hackathon are by default admitted.
- There will be also an online test, as the last step of the second midterm, which will also help the evaluators grade your theoretical knowledge on this subject. The way it will be organized will be announced next week.

Exam Requirements

TBA.

Project Documentation

The deliverable components for this project are the presentation slides, the report and the developed software. At the presentation, the students should explain what they did and how did they do it. Also, they should motivate the decisions that have been taken.

In order to keep the presentations interesting the presenter/-s should bring arguments and cover only the important stuff. The presentations should mimic the report structure.

Quoting a previous FAF-er, the general rule for the report of the project is "High quality content, with low levels of bullshit". There are no page limits (for now).

For the structure of the report, one can take inspiration from the previous sections where the tasks were described. Also, it should contain a title page and a references section. Not stating the sources is considered plagiarism and *will not be tolerated*.

This project requires the use of L^AT_EX for text editing, and a safe choice which also provides collaboration capabilities is [Overleaf](#)). Also, [Grammarly](#) could be used as it is a free tool that will help fix not only spelling but also bad grammar and awkward sentence formulations.

If the project can be demoed, demo it at the end of the presentations. The source code must be version controlled and available to the exam committee at the day of presentation. Any good code repository has a good README file, obviously.

Evaluation

The final grade for this project will be calculated as follows:

$$\begin{aligned} final_grade &= 30\% \times individual_midterm1_grade \\ &+ 30\% \times individual_midterm2_grade \\ &+ 40\% \times final_exam_grade \end{aligned}$$

Good Luck! Don't disappoint!!

Annex

The following text was taken from "FAF-17x PBL Project Guideline", authored by Alex Burlacu. If it helped polish up the project topic, don't forget to say thanks to him.

Any project that you design should solve a problem, don't forget that. That means, whatever you do, have a couple of arguments of why you're doing it. With sound arguments you could even make the committee believe a programming language that you designed is indeed a suitable Bachelor thesis project, in case you thought about it.

- **Topic** – Topic must convey the concrete deliverable, in technical terms, that is the result of the Bachelor thesis. An example of a fairly bad name is: "Platform for restaurants" or "Digital whiteboard". These names are bad because they do not explain exactly what will be made during the work on the Bachelor thesis project. On the other hand, names like: "A client-server based system for tracking inventory" or "Mobile-based secure chat application with support for short video messages" are considerably better, for the reasons outlined above;
- **Subject** – The subject should summarize what exactly will be done in the project. Think of it as a more extended version of the topic, as if someone asked "OK, tell me more" or "What should I expect of it?". Remember, it's not a sales pitch and thus be honest and don't promise the sea and the salt. Use this as an opportunity to define well what to expect of this project, who are the target auditory and what is its scope. A bad example would be: "Creation of a service to help people save money", where's for the same project, a better subject would be: "Design and development of a service and a web-based client that will provide analytics on weekly and monthly income/expenses, divided by categories, and also will make custom recommendations on how to save money based on expense patterns of users";
- **Objectives** – Objectives are not test scenarios, and neither a task list. On the other hand, objectives do describe what are the system capabilities. Therefore, a good list of objectives define what functionalities the system has, and not how they are realised, and neither how it will be developed. Lists containing things like stages of a project life cycle, tools and languages that are to be used in the project, and generic lines like "The system will use artificial intelligence" or "Build a platform for X", are not to be present in an objectives list. If you still want to brag about what a cool thing will it contain, rather write it like: "The system will use `insert type of the AI algorithm to be used to support some functionality`", for example.