FAF.BDA21.1 Fall 2023
PBL Project Guideline

Handed out: September 4, 2023
Updated: December 11, 2023

Preliminaries

According to the Oxford English Dictionary, a (software) application is a computer program designed to carry out a specific task, typically used by end-users. Because of it's abstract definition, the term (often times shortened to just "app") is used for everything, from the software running on your fridge to the Microsoft365 suite. What unites them? For most applications, one can define the following portions:

- A client-facing portion;
- A server portion;
- A data keeping / managing portion;
- A portion that ties all this together.

The client-facing portion of an app (aka. Presentation Tier) is usually responsible for presenting information to end-users and accepting inputs that a user might provide. Sometimes, this portion can validate user input or can allow for filtering or manipulating the presented info. The server portion's job (aka. Logic Tier) is to process requests made by a user (e.g. updating settings, downloading new or deleting old data etc.). Besides that, this tier is responsible for maintaining functionality of the whole app. The data keeping and managing portion's role (aka. Data Tier) is pretty self explanatory. This portion is in change of storing and accessing all data an app might use. All previous tiers need a way to communicate, which is exactly what the last portion should do. Main tasks of this portion include sending and receiving data sent by other tiers, usually via network, or pigeons. In general, applications come in many forms. Examples include:

- Web Applications;
- Mobile Applications;
- Desktop Applications.

Though many app classifications exist, the one which differentiates them by Platform is quite important. While always different in what they do, all apps in the end will need a platform to be run on, to be deployed. As such, this project aims at providing the students with an opportunity to experience developing multi-part applications that would be deployed on a platform of choice, while also taking part in a PBL environment, but not only (see next section).

Getting Started

During the first half of the project you will partake in an Internship at a company. Besides the well-known benefits [1], being at an internship implies that you will have 2 mentors – your university mentor and your company mentor. This will become important later.

To prepare for the internship, during summer you will have several activities, most notably Team Formation and Internship Search. Pay attention to any conditions the people responsible for internships of that year tell you. This will ensure good organization of and maximum benefits for those involved in the process.

After successfully gathering a team and creating a communication channel with the company, it is time to think about Choosing a Topic for your project. When choosing topics, it is best to search for any problems that your company providing the internship might have (and could be solved in the context of this project). This represents a good exercise in applying the PBL approach in the wild, while also being aligned with the company's interests.

After deciding on a topic for the project, you'll need to formulate the Project Proposal. The proposal form can be found, as usual, on Else. Make sure to validate it with both of your mentors. *It is worth remembering that, in case of any misunderstandings between you and a company, you should address the person responsible for Internships of that year.*

The Task: Midterm 1

For the first midterm you'll be working on:

1. Defining Requirements for your application;
2. Implementing an MVP;
3. Tracking your project development in a VCS.

As usual any project needs to be developed according to some requirements, otherwise there is no goal, no scope. Work with your company representatives to define functional and non-functional requirements for your project. Delimit which requirements will be implemented at the MVP stage, and which on other stages.

Continuing, the meat of the project – developing an MVP. Make sure to focus only on the most important feature of your application. When documenting your development, describe the technology stack, how every application portion works and how they communicate. To better align with your university courses, this project requires the use of an SQL Database. Overall, the focus of this project should be the development processes, not the number of an application's features.

During the development of your application, you will need a way to cooperate between developers and track your progress. As such, use a VCS to track your development. Make sure to write meaningful comments when committing, and overall, follow best practices. When nearing the end of development, write a README file for anyone interested in your application.

The Task: Midterm 2

For the second midterm, you'll be working on:

1. Validating your Idea;
2. Pivoting or Adding New Features;
3. Modelling your Application;
4. Deploying your Application.

After attending whatever feedback you had after Midterm 1, you'll need to do that without which one should never start developing anything (except, maybe, if "to develop" is the actual goal) and that is validating your idea. Make sure to cover all important parts of idea validation such as Problem Analysis, Customer Validation and Comparative Analysis.

Similarly to PCAIT, after idea validation you'll need to decide whether you will pivot or not. In case you do, pivot then adapt your current implementation of your app to better suit the new direction of the project. If not, continue to develop new features in accordance with the requirements defined previously. Remember that no work is wasted here, even if you decide to pivot. The road to glory is rarely only straight and only forward.

Though Software Modeling is most often done before developing a project, it can also be done during or after it's development, to provide a standardized view of the current architecture or functionalities of your system. As such, provide a thorough analysis of your system using UML diagrams, both from the Behavioural and Structural points of view. Make sure to cover all main processes, user interactions, message exchanges, class hierarchies, components and APIs and delivery options. [2] [3]

Finally, think about how you will be delivering your application. Decide on the deployment strategies and delivery tools that best suit your project and application. Regardless of the chosen method, make sure the application is accessible to your users (and your mentors).

The Task: Exam

For the exam you'll be working on:

1. Redacting / Typesetting your Project Report;
2. Preparing each other for the exam.

After receiving feedback from the second Midterm, it is now time to finalize your Project. Make sure to follow your mentor's instructions on how to enhance your Report, what needs to be added and what needs to be re-done. After applying all the changes, you will upload the final version of the report on ELSE, print it out and bring it with you to the exam.

For the exam itself, expect an individual interview with your mentors. Brush up on theoretical concepts learned during the project and make sure that every team member knows

about what other team members did in the project. The fact that a student did only part of the Project does not exempt them from needing to answer any question about any part of the Project.

Reporting

Because much of the work done in this project is hard to track (e.g. meetings, research etc.), the ability to efficiently record and report your work is very important. In this project, reporting your work should be done via so called deliverables - things written, coded and submitted for mentors to review. Your deliverables in this project are the Project Report, Project Demo, Repository Link, Midterm Presentations and Meeting Notes.

The report is the bread and butter of this project. It represents your team's semester of work, organised and easy to read. A Report Template can be found on Else Course and should provide everything you'd need to just start filling it in with text and info. The report's main part is the text you write there. You are required to use LaTeX for text editing (a safe choice is Overleaf). Also, consider using Grammarly – it's a free tool that will help you fix not only spelling but also bad grammar and awkward sentence formulations.

It is a good idea to record a project demo before presenting your project progress. This is much safer and will save you a lot of stress during your presentations, compared to presenting your progress live. Plus, you can always refer back to how the application worked before. While preparing the demo, don't forget to also share a repository link of your project. Make sure it is accessible by any of the mentors – this is the main way they will be reviewing your code. A private or inaccessible repository link will most likely bring penalties.

The aim of the midterm presentations is to familiarize the audience (your peers and the mentors committee) with the work presented in the report. During presentations, try to explain what did you do, how did you do it and why did you choose to do it like you did. Bring arguments, cover only the important stuff and keep your presentations interesting and visually appealing. Presentations should mimic your report structure.

Meeting notes are an integral part of this project. These should be done after each team meeting, with or without your mentor. Meeting notes aren't necessarily graded but can be used to determine the efficiency of teamwork, which is an important factor in grading the whole team. The Meeting Notes form can be found on Else Course. Remember that any effort that is not documented cannot be appreciated / graded, so report diligently.

Evaluation

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

$$\begin{aligned} final\_grade = {} & 20\% \times individual\_midterm1\_grade \\ & + 20\% \times individual\_midterm2\_grade \\ & + 15\% \times mentor\_grade \\ & + 15\% \times company\_mentor\_grade \\ & + 30\% \times final\_exam\_grade \end{aligned}$$

Both mentors grade and final exam grade are given to you individually. However midterm presentations are going to be team presentations for which you will receive a team grade. Individual midterm grades will then be calculated as follows:

$$individual\_midterm\_grade = team\_midterm\_grade + individual\_correction$$

For both Midterm 1 and Midterm 2 presentations, each team must submit a list of corrections in which they appreciate each member's contribution to the project. Below are 2 examples of such a list:

Team 1		**Team 2**	
Andronic G.	0	Popa A.	0
Popova Z.	0	Cusnir A.	+1
Zamfir I.	0	Petrov O.	+1
		Ivanov I.	−2

You will submit 1 list of corrections per team. The possible correction values are `[-2.5, -2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5]`, and their sum should always be equal to `0`. After the presentation, a student's individual midterm grade will be calculated by the formula above. If, for example, both Team 1 and Team 2 have received an `8` for their presentations, their individual marks will be as follows:

Team 1		**Team 2**	
Andronic G.	$8 + (0) = 8$	Popa A.	$8 + (0) = 8$
Popova Z.	$8 + (0) = 8$	Cusnir A.	$8 + (+1) = 9$
Zamfir I.	$8 + (0) = 8$	Petrov O.	$8 + (+1) = 9$
		Ivanov I.	$8 + (-2) = 6$

As such, these corrections could be used to appreciate teammates that had a significant input to the project during the half-semester. Conversely, the corrections could be used to penalize teammates which did not prove helpful in the team during the project.

Good Luck!

References

[1] ChatGPT. "Re. Internships". https://chat.openai.com/share/bd979a06-d2b9-4b8a-8edb-da1819aace0d.

[2] PlantUML. "PlantUML at a Glance". https://plantuml.com/.

[3] Kirill Fakhroutdinov. "The Unified Modeling Language". https://www.uml-diagrams.org/.