PROJECT ASSIGNMENT

LETI-ESOFT ▸ 2025-2026

1 Preamble

This document presents a simulated context in which DGS, a state-funded Portuguese healthcare system, intends to explore and validate an application designed to manage the vaccination process. To this end, the organization decided to start the development of a new software product (prototype) in collaboration with the faculty and students of LETI-ESOFT.

The software product described herein is, on the one hand, designed to promote:

1. The consolidation and acquisition of new competencies related to software development, by students.
2. The practice and internalization of the recommended working methods and best practices commonly adopted in the software industry.

On the other hand, the software product is used to evaluate students throughout the semester, in alignment with the approved course syllabus (cf. ISEP Portal).

For the project's development, students must form teams of four elements (exceptionally three) from the same lab class and communicate their composition to the LETI-ESOFT faculty (cf. formalized groups/teams on Moodle). Each team will operate as an independent supplier company, competing to be selected/hired to deliver the intended software product. To achieve this, all teams need to develop a prototype meeting all (or most) of the specified software requirements and demonstrate that their development process ensures high product quality, while employing appropriate working methods and industry best practices.

2 Intended Software Product

DGS is a state-funded Portuguese healthcare system seeking to develop an application to manage the vaccination process in Portugal. Given that the Coronavirus may not be the last pandemic we face in our lifetime, the application should be designed to easily accommodate future pandemic scenarios requiring large-scale population vaccination. Furthermore, the software application should be conceived with potential commercialization in mind, allowing it to be offered to other companies, organizations, or healthcare systems beyond DGS.

2.1 Business Context

Immunization is a global health and development success story, saving millions of lives every year. Vaccines reduce the risks of contracting diseases by working with the body's natural defenses to build protection. When a person receives a vaccine, the immune system responds accordingly. Vaccines are also critical for the prevention and control of infectious disease outbreaks, such as the Swine Flu and the Covid-19 pandemic.

DGS, the state-funded Portuguese healthcare system, is responsible for the National Vaccination Program and requires a software application to manage the vaccination process. The application must allow SNS users to schedule vaccination appointments and obtain vaccination certificates. To do so, SNS users must be registered in the system with the following data: name, date of birth, sex, postal address, phone number, email, citizen card number, and SNS user number.

The vaccination process is primarily carried out through two types of facilities: community mass vaccination centers and healthcare centers distributed across the country. Healthcare centers provide a wide range of medical services to citizens within a given area and can administer various types of vaccines (e.g., Covid-19, Influenza, Tetanus, Smallpox). In contrast, community mass vaccination centers are temporary facilities created specifically to administer a single type of vaccine in response to an ongoing disease outbreak (e.g., Swine Flu, Covid-19).

The vaccination workflow and the roles of personnel involved are largely similar across both types of centers. Nurses at healthcare centers can issue and deliver vaccination certificates on-site upon request by SNS users. Both types of centers are characterized by attributes such as name, postal address, phone number, email, website address, opening and closing hours, and the maximum number of vaccines that can be administered per hour.

Receptionists and nurses registered in the application will take part in the vaccination process. Both roles are defined by attributes such as automatically generated ID, name, postal address, phone number, and email. Given that staff allocation to vaccination centers may not be straightforward, by now the system might assume that receptionists and nurses can work at any vaccination center.

It is important to note that for each type of vaccine, multiple vaccine brands may exist. For example, the Covid-19 vaccine type includes Pfizer, Moderna, AstraZeneca, among others. Although each vaccine brand may require a distinct administration protocol depending on the age group (e.g., number of doses, dosage in milliliters, interval between doses), this version of the system will not record such detailed information.

To receive a vaccine, the SNS user must use the application to schedule an appointment. The user provides their SNS number, selects a vaccination center, date, time and vaccine type. By default, the system suggests the vaccine type associated with the current outbreak. The application then checks the center's capacity for the selected day and time. If available, the appointment is confirmed, and the user is notified to attend the selected center at the scheduled time. Some users (e.g., elderly individuals) may prefer to schedule their appointment in person at a healthcare center with the assistance of a receptionist.

On the scheduled day and time, the SNS user arrives at the vaccination center. A receptionist registers the user's arrival, verifies their SNS number, and confirms the scheduled appointment. If the information is correct, the receptionist marks the user as ready in the system and directs them to a waiting room.

A nurse responsible for administering vaccines uses the application to view the list of users present at the center to receive the vaccine. Users are typically vaccinated in order of arrival (FIFO), although operational constraints may occasionally alter this sequence. The nurse reviews the user's information and health conditions, along with the scheduled vaccine type and vaccination history, and receives system instructions regarding the vaccine to be administered (e.g., brand and dosage). It's worth mentioning that only nurses are authorized to access full health data of users.

After administering the vaccine, the nurse records the event in the system, including the vaccine type (e.g., Covid-19), brand (e.g., Pfizer, Moderna, AstraZeneca) and lot number. The user is then sent to a recovery room for a designated observation period (e.g., 30 minutes). If no adverse reaction occurs, the user may leave the center. If any reactions are observed, the nurse must record them in the system.

DGS administrators are responsible for managing the application, including the registration of vaccination centers and personnel (receptionists and nurses) involved in the vaccination process. They also configure the core data required for the system's daily operation, such as vaccines and related information.

Each vaccine type should include the following attributes: code, disease (i.e., the condition targeted by the vaccine), and a short description. To describe a specific vaccine, the following attributes must be registered: commercial name, brand, type, and the technology used. For reference, please consult the following web page to identify the six available vaccine technology types: https://www.pfizer.com/news/articles/understanding_six_types_of_vaccine_technologies

2.2 Envisioned Applications

It is envisioned that general information management will be handled through multiple dedicated User Interfaces (UI), including:

1. A Web Application targeted at DGS staff.
2. A mobile application targeted at SNS users.

At this stage, the focus is not on developing these UI, but rather on the development of the core applications. For demonstration purposes, a basic console-based UI is considered sufficient. Similarly, although all system users must be authenticated beforehand, the authentication process may be simulated/mocked for now.

Given the sensitivity and criticality of the core application, it is required to be developed in C++ using the CLion IDE, preferably.

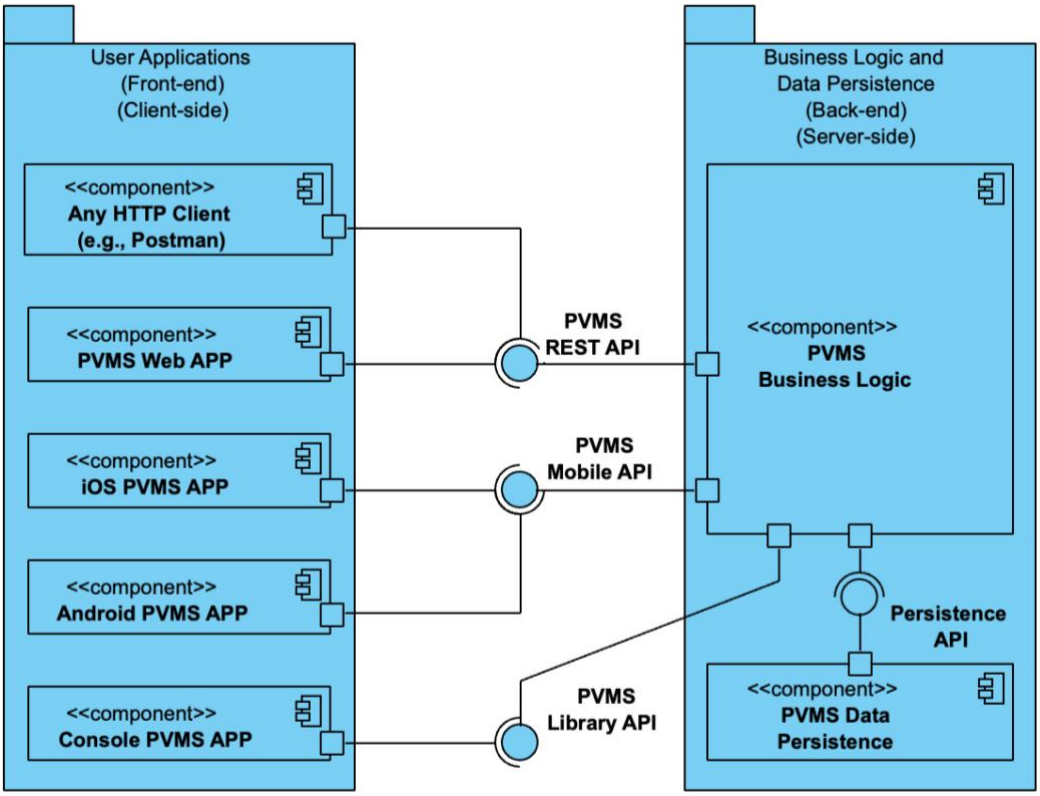

Figure 1. Envisioned System Applications and Components

To guide the development of the intended prototype, it is envisioned that the *Pandemic Vaccination Management System* (PVMS) will comprise (at minimum) the applications and/or components illustrated in Figure 1.

As depicted, the system consists of:

1. **A core server-side application**, responsible for maintaining and persisting all data required by the system (e.g., vaccination centers, vaccines, employees) as well as enforcing the applicable business logic (e.g., allowing/denying scheduling of a vaccine administration for a given day/time). Additionally, this application exposes multiple API to support the development of user applications. **Given its critical role in the overall system architecture, teams should prioritize its development**.
2. **Several client-side applications**, responsible for interacting with end-users such as DGS staff and SNS users. These applications consume one of the API provided by the server-side component. Although the system envisions at least one user application per platform (i.e., web, iOS, Android), their development is currently out of scope, **except for a basic console-based UI intended for demonstration purposes**. Furthermore, generic HTTP client tools (e.g., Postman) may be used to verify and validate the exposed API.

The system must start up in less than 10 seconds, and, in the event of failure, no data loss should occur. At peak times, the system is expected to experience high loads. To mitigate potential issues, it must be designed to ensure a maximum response time of 5 seconds, regardless of load conditions.

Overall system availability must exceed 99% annually. Additionally, the system should be designed to support data persistence across multiple target platforms, including relational databases, NoSQL databases, and in-memory databases.

All project artifacts, including source code, must be developed in English. The adoption of automated regression testing, preferably using the Google Testing Framework, is strongly recommended and highly valued.

3 Working Method

To ensure alignment with DGS's internal processes, supplier companies must make partial deliveries throughout the course of the project, following the schedule outlined in Table 1. Additionally, best practices must be adopted in areas such as requirements management and traceability, business analysis, software design, testing, and version control.

Table 1. Project Schedule

Sprint	Begin (week)	End (week)	Main Purpose / Focus
1	4	6	Elicitation and specification of requirements and business analysis
2	7	9	Development of the 1st software prototype
3	10	12	Development of the 2nd software prototype
4	13	15	Development of the 3rd software prototype
–	16	16	Project evaluation and final selection of the supplier

3.1 Sprints

According to Table 1, prototype development is organized into 4 sprints:

1. **Sprint 1** focuses on enhancing the teams' overall understanding of the underlying business ideas and gaining a broad overview of both functional and non-functional requirements. It also includes setting up the working environment adopted by each team (e.g., creating the GitHub repository and its structure, configuring tools to work with the repository, creating default C++ projects).
2. **Sprint 2** focuses on developing the first set of requirements. By the end of this sprint, it is recommended that a working prototype be available, enabling the team to demonstrate the implemented features and gather client feedback.
3. **Sprint 3** focuses on developing the second set of requirements and refining previously implemented features based on received feedback and any business changes.
4. **Sprint 4** focuses on developing the final set of requirements and further improving existing features in response to feedback and business updates. A working prototype must be available to demonstrate the team's achievements.

At the end of sprints 1 and 3, typically during the first week of the following sprint, each team will receive feedback on the submitted work. At the end of sprints 2 and 4, students are required to present the current state of the project and individually defend the submitted work. During the defense, questions of both theoretical and practical nature may be asked, including hypothetical scenarios related to the project.

Sprint deadlines are strict and non-negotiable. Each sprint begins on Monday of the first week and ends on Sunday at 23:59 of the final week. Before the sprint deadline, each team must submit a copy of their working repository on Moodle (cf. available links). The requirements to be addressed in each sprint will be clearly specified at its start.

It is strongly recommended that each team's working environment adopt a repository/project structure similar to the one presented in the Demo Project, available on Moodle. Additionally, please refer to the instructions provided in section 3.3.

3.2 Project Requirements Clarifications

In some LETI-ESOFT classes, a DGS representative will be available to answer questions regarding the intended software product.

Outside these sessions, any doubts about requirements must be addressed through the dedicated electronic forum created for this purpose on Moodle, accessible to all teams. A DGS representative will aim to respond within an average timeframe of 24 to 48 hours. **Before posting a new question, teams must verify whether it has already been answered in the forum.** Duplicate questions will be ignored.

To ensure the smooth progression of each sprint, DGS representatives will not respond to new questions during the final two days of the sprint. If such questions are critical to sprint completion, they should have been raised earlier, preferably during the first week. These occurrences may indicate deficiencies in the team's working methodology.

Please note that this forum is intended solely for questions related to DGS business and requirements. Technical questions are not within its scope.

3.3 Version Control System

To comply with DGS practices, teams must use GitHub as their version control system throughout prototype development. Only proposals demonstrating good collaborative editing practices (both code and other project artifacts) will be accepted.

In line with DGS guidelines, **all project artifacts must be stored in the version control repository**. These include, but are not limited to diagrams, documents, source code, configuration files, and images. For readability purposes, **all images must be in SVG format**.

A Base Project to assist in setting up the GitHub repository is available on Moodle. All team tasks must be properly planned and documented using the "TeamMembersAndTasks.md" template file provided within the project. According to this template, some tasks are to be carried out by all team members,

while others are primarily assigned to a single member. Nevertheless, active collaboration among all team members is strongly encouraged and highly recommended.

To streamline the process, the GitHub repository name must follow the format `leti-esoft-25-26-[LabClass][TeamNumber]`, where `[LabClass]` is replaced by `A`, `B`, or `C` according to the team's lab class, and `[TeamNumber]` is the identifier assigned by the faculty. For example, the repository `leti-esoft-24-25-A1` corresponds to team `1` from lab class `A`.

3.4 Proposal Evaluation Criteria

Each team's proposal will be evaluated by the LETI-ESOFT faculty based on the following criteria:

1. Compliance with the specified requirements.
2. Overall quality of the proposed solution.
3. Evidence of best practices in software development and team working methods.
4. The team's and individual members' ability to present and defend the proposal throughout the project development.

4 Revision History

Version	Date	Description
1.0	2025.10.06	Initial release