Integrative Project - Spring Semester 2021/2022
Pandemic Vaccination Management System

1. Introduction

The Degree in Informatics Engineering (LEI) adopts a teaching-learning process based on the development of a single project that fosters the integration and coordinated application of the knowledge and competencies covered by all the courses (UC) taught throughout the 2nd semester: ESOFT, PPROG, MDISC, MATCP and LAPR2.

The project is to be carried out by teams of four students. These teams remain the same across all the UCs (ESOFT, PPROG, MDISC, MATCP and LAPR2). It must be clear that this is a single project - not 5 separate projects! At the end, each team should come out with an integrated solution, an application, encompassing several interconnected modules.
The set of rules provided in this document cover the main existing scenarios. Situations other than those mentioned in this document should be reported to the LAPR2 course's coordinator (RUC), who will decide how to proceed together with the other RUCs.

Although this project adopts a Project-Based Learning (PBL) model, comprising the 2nd semester UCs, it is important to highlight that the project development must adopt all the best practices introduced both throughout the 1st semester and the 2nd semester - the adequate application of those best practices is part of the evaluation criteria of the project.

2. Application to be developed

DGS is a state-funded Portuguese healthcare system that wants an application to manage the vaccination process in Portugal to manage the Covid-19 vaccination process. As Coronavirus might not be the last pandemic in our lifetime, the application should be designed to easily support managing other future pandemic events requiring a massive vaccination of the population. The software application should also be conceived having in mind that it can be further commercialized to other companies and/or organizations and/or healthcare systems besides DGS.

2.1 Business context

Immunization is a global health and development success story, saving millions of lives every year. Vaccines reduce risks of getting a disease by working with your body's natural defenses to build protection. When you get a vaccine, your immune system responds. Vaccines are also critical to the prevention and control of infectious-disease outbreaks, such as Covid-19 disease.

DGS is responsible for the Vaccination Program in Portugal and needs a software application to manage the vaccination process and to allow SNS users to schedule a vaccine and obtain a vaccination certificate.

The DGS vaccination process is mainly carried out through community mass vaccination centers and health care centers distributed across the country. Different from the health care centers, which provide a wide range of healthcare services to citizens in a certain area, the community mass vaccination centers are facilities specifically created to administer vaccines of a single type as response to an ongoing disease outbreak (e.g.: Covid-19).

The vaccination process flow and employees enrolled in the vaccination process are almost the same in each kind of vaccination center. The main difference between the two kinds of centers is that a healthcare center is associated with a given ARS (*Administração Regional de Saúde*) and

AGES (*Agrupamentos de Centros de Saúde*), and it can administer any type of vaccines (e.g.: Covid-19, Dengue, Tetanus, smallpox). Moreover, nurses working in the healthcare centers can issue and deliver on site a vaccination certificate whenever a SNS user asks for it. Both kinds of vaccination centers are characterized by a name, an address, a phone number, an e-mail address, a fax number, a website address, opening and closing hours, slot duration (e.g.: 5 minutes) and the maximum number of vaccines that can be given per slot (e.g.: 10 vaccines per slot). In addition, each vaccination center has one coordinator. Furthermore, receptionists and nurses registered in the application will work in the vaccination process. As the allocation of receptionists and nurses to vaccination centers might be complex, by now, the system might assume that receptionists and nurses can work on any vaccination center.

Yet, it is worth noticing that for each type of vaccine, several vaccines might exist, each one demanding a distinct administration process. For instance, for the Covid-19 type, there is (i) the Pfizer vaccine, (ii) the Moderna vaccine, (iii) the Astra Zeneca vaccine, and so on. The vaccine administration process comprises (i) one or more age groups (e.g.: 5 to 12 years old, 13 to 18 years old, greater than 18 years old), and (ii) per age group, the doses to be administered (e.g.: 1, 2, 3), the vaccine dosage (e.g.: 30 ml), and the time interval regarding the previously administered dose. Regarding this, it is important to notice that between doses (e.g.: between the 1st and 2nd doses) the dosage to be administered might vary as well as the time interval elapsing between two consecutive doses (e.g.: between the 1st and 2nd doses 21 days might be required, while between the 2nd and the 3rd doses 6 months might be required).

An Administrator is responsible for properly configuring and managing the core information (e.g.: type of vaccines, vaccines, vaccination centers, employees) required for this application to be operated daily by SNS users, nurses, receptionists, etc.

To take a vaccine, the SNS user should use the application to schedule his/her vaccination. The user should introduce his/her SNS user number, select the vaccination center, the date, and the time (s)he wants to be vaccinated as well as the type of vaccine to be administered (by default, the system suggests the one related to the ongoing outbreak). Then, the application should check the vaccination center capacity for that day/time and, if possible, confirm that the vaccination is scheduled and inform the user that (s)he should be at the selected vaccination center at the scheduled day and time. The SNS user may also authorize the DGS to send a SMS message with information about the scheduled appointment. If the user authorizes the sending of the SMS, the application should send an SMS message when the vaccination event is scheduled and registered in the system. Some users (e.g.: older ones) may want to go to a healthcare center to schedule the vaccine appointment with the help of a receptionists at one vaccination center.

On the scheduled day and time, the SNS user should go to the vaccination center to get the vaccine. When the SNS user arrives at the vaccination center, a receptionist registers the arrival of the user to take the respective vaccine. The receptionist asks the SNS user for his/her SNS user number and confirms that he/she has the vaccine scheduled for the that day and time. If the information is correct, the receptionist acknowledges the system that the user is ready to take the vaccine. Then, the receptionist should send the SNS user to a waiting room where (s)he should wait for his/her time.

At any time, a nurse responsible for administering the vaccine will use the application to check the list of SNS users that are present in the vaccination center to take the vaccine and will call one SNS user to administer him/her the vaccine. Usually, the user that has arrived firstly will be the first one to be vaccinated (like a FIFO queue). However, sometimes, due to operational issues, that might not happen. The nurse checks the user info and health conditions in the system and in accordance with the scheduled vaccine type, and the SNS user vaccination history, (s)he gets system instructions regarding the vaccine to be administered (e.g.: vaccine and respective dosage considering the SNS user age group). After giving the vaccine to the user, each nurse registers the event in the system, more precisely, registers the vaccine type (e.g.: Covid-19), vaccine name/brand (e.g.: Astra Zeneca, Moderna, Pfizer), and the lot number used. Afterwards, the nurse sends the user to a recovery room, to stay there for a given recovery period (e.g.: 30 minutes). If there are no problems, after the given recovery period, the user should leave the vaccination center. The system should be able to notify (e.g.: SMS or email) the user that his/her recovery period has ended. If the nurse identifies any adverse reactions during that recovery period, the nurse should record the adverse reactions in the system.

After taking the vaccine, any SNS user can request the issuance of the EU COVID Digital Certificate (this feature of the system will be implemented later by the DGS's IT department).

Each vaccination center has a Center Coordinator that has the responsibility to manage the Covid-19 vaccination process. The Center Coordinator wants to monitor the vaccination process, to see statistics and charts, to evaluate the performance of the vaccination process, generate reports and analyze data from other centers, including data from law systems. The goal of the performance analysis is to decrease the number of clients in the center, from the moment they register at the arrival, until the moment they receive the SMS informing they can leave the vaccination center. To evaluate this, it proceeds as follows: for any time interval on one day, the difference between the number of new clients arrival and the number of clients leaving the center every five-minute period is computed. In the case of a working day, with a center open from 8 a.m. until 8 p.m., a list with 144 integers is obtained, where a positive integer means that in such a five-minute slot more clients arrive at the center for vaccination than clients leave with the vaccination process completed. A negative integer means the opposite.

Now, the problem consists in determining what the contiguous subsequence of the initial sequence is, whose sum of their entries is maximum. This will show the time interval, in such a day, when the vaccination center was less effective in responding. So, the application should implement a brute-force algorithm (an algorithm which examines all the contiguous subsequences) to determine the contiguous subsequence with maximum sum. The implemented algorithm should be analyzed in terms of its worst-case time complexity, and it should be compared to a benchmark algorithm provided. The computational complexity analysis (of the brute-force algorithm and any sorting algorithms implemented within this application), must be accompanied by the observation of the execution time of the algorithms for inputs of variable size, in order to observe the asymptotic behavior. The worst-case time complexity analysis of the algorithms should be properly documented in the user manual of the application (in the annexes). The user manual must be delivered with the application.

The DGS has Administrators who administer the application. Any Administrator uses the application to register centers, SNS users, center coordinators, receptionists, and nurses enrolled in the vaccination process.

2.2 Technological requirements

The application must be developed in Java language using the IntelliJ IDE or NetBeans. The application graphical interface is to be developed in JavaFX 11. All those who wish to use the application must be authenticated with a password holding seven alphanumeric characters, including three capital letters and two digits. Only the nurses are allowed to access all user's health data. The application must support, at least, the Portuguese and the English languages.

During the system development, the team must: (i) adopt best practices for identifying requirements, and for OO software analysis and design; (ii) adopt recognized coding standards (e.g., CamelCase); (iii) use Javadoc to generate useful documentation for Java code.

The development team must implement unit tests for all methods, except for methods that implement Input/Output operations. The unit tests should be implemented using the JUnit 5 framework. The JaCoCo plugin should be used to generate the coverage report.

All the images/figures produced during the software development process should be recorded in SVG format.

The application should use object serialization to ensure data persistence between two runs of the application.

3. Project operating mode

It must be clear that it is not intended at the end of the Integrative Project to obtain 5 separate projects but rather an integrated solution, an application, encompassing several modules. The focus should always be on the project as a whole and not on each UC individually.

3.1 Work teams

The students are organized in teams of 4/5 members. The teams are the same in all the UCs of the 2nd semester. Each team will work as an independent company to compete in the development of the required application.

The project takes place during the semester in a subset of classes of each UC. Each UC introduces concepts and helps students in the development of the project, giving support to, focusing on specific areas as follows:

- ESOFT - Software development process;
- PPROG - Java OO programming;
- MDISC - Worst-case time complexity of sorting and maximum subsequence algorithms;
- MATCP - Linear regression and prediction tasks;
- LAPR2 - Team management, working methodology, integration of the different modules, English written skills.

3.2 Sprints

The semester is divided according to Table 1. The detailed requirements of each sprint will be presented in a separate document, just before the sprint.

Table 1: Timetable of the semester

Sprint	Start (week)	End (week)	General objective	UCs involved
-	1	3	• Group formation process	LAPR2
A	4	6	• Acquisition of basic skills of ER, AOO, DOO and COO. • Introduce concepts of Software Testing, Continuous Integration and Code Quality. • Introduce Agile working methodology. • Prepare students for writing technical documentation in English.	ESOFT LAPR2
B	7	9	• Acquisition and application of more advanced skills of ER, AOO, DOO and COO. • Practice of programming in JAVA language. Develop a subset of USs. Implement a console user interface. • Prepare students for writing technical documentation in English.	ESOFT PPROG LAPR2
C	10	12	• Acquisition and application of more advanced skills of ER, AOO, DOO and COO. • Practice of programming in JAVA language. Develop a subset of USs and update (if needed) the USs developed in Sprint B. Implement a console user interface. • Prepare students for writing technical documentation in English.	ESOFT PPROG LAPR2
D	13	15	• Acquisition and application of more advanced skills of ER, AOO, DOO and COO. • Study computational complexity and linear regression. • Practice of programming in JAVA language. Develop all USs not addressed in Sprints B and C and, if needed, update the USs previously developed. Implement a Graphical User Interface (GUI) using JavaFX. • Prepare students for writing technical documentation in English.	ESOFT PPROG MATCPMDISC LAPR2
Evaluation	16	16	• Evaluate the work developed during the semester (project documentation, code and work methodology). The evaluation of each UC is independent, with its own criteria.	ESOFT PPROG MATCP MDISC LAPR2

4. Revision History

Date	Description