Integrative Project - Spring Semester 2020/2021
Many Labs

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

Many Labs is an English company that has a network of clinical analysis laboratories and that wants an application to manage the clinical analyses performed in its laboratories.

2.1 Business context

Many Labs is a company that operates in the English market, it has headquarters in London and has a network of clinical analysis laboratories in England where analysis of blood (samples are collected) are performed, as well as Covid-19 tests. In England, *Many Labs* has exclusivity for Covid-19 tests throughout the territory, which means that no other company can perform this type of testing. All *Many Labs* clinical analysis laboratories perform clinical blood tests, and a subset of these laboratories also performs Covid-19 tests.

The set of *Many Labs* clinical analysis laboratories form a network that covers all England, and it is responsible for collecting samples and interacting with clients. The samples collected by the network of laboratories are then sent to the chemical laboratory located in the company's headquarters and the chemical analysis are performed there.

Typically, the client arrives at one of the clinical analysis laboratories with a lab order prescribed by a doctor. Once there, a receptionist asks the client's citizen card number, the lab order (which contains the type of test and parameters to be measured), and registers in the application the test to be performed to that client. Then, the client should wait until a medical lab technician calls him/her to collect the samples required to perform a given test.

In case of a new client, the receptionist registers the client in the application. To register a client, the receptionist needs the client's citizen card number, National Healthcare Service (NHS) number, birth date, sex, Tax Identification number (TIF), phone number, e-mail and name.

All the tests (clinical blood tests and Covid-19 tests) performed by the network of laboratories are registered locally by the medical lab technicians who collect the samples. The samples are sent daily to the chemical laboratory where the chemical analyses are performed, and results obtained. When sampling (blood or swab) the medical lab technician records the samples in the system, associating the samples with the client/test, and identifying each sample with a barcode that is automatically generated using an external API.

At the company's headquarters, the clinical chemistry technologist receives the samples (delivered by a courier) and performs the chemical analysis, recording the results in the software application. After completing the chemical analysis, the results of all chemical analyses are analysed by a specialist doctor who makes a diagnosis and writes a report that afterwards will be delivered to the client.
To facilitate and simplify the validation work performed by the specialist doctor, the application uses an external module that is responsible for doing an automatic validation using test reference values.

After the specialist doctor has completed the diagnosis, the results of the clinical analyses and the report become available in the system and must be validated by the laboratory coordinator. To validate the work done, the laboratory coordinator checks the chemical test/result and associated diagnosis made and confirms that everything was done correctly. Once the laboratory coordinator confirms that everything was done correctly, the client receives a notification alerting that the results are already available in the central application and informing that he/she must access the application to view those results. The client receives the notification by SMS and e-mail. At the same time the results are also available in the central application where the medical lab technicians who collect the samples, the clinical chemistry technologist, the specialist doctor, and the laboratory coordinator can check them.

To facilitate the access to the results, the application must allow ordering the clients by TIF and by name. The ordering algorithm to be used by the application must be defined through a configuration file. It is intended that the choice of the ordering algorithm is based on the algorithm complexity (mainly the execution time). Therefore, at least two sorting algorithms should be evaluated and documented in the application user manual (in the annexes) that must be delivered with the application.

Moreover, *Many Labs* is a company that needs to be continuously evaluating and improving its internal processes to achieve excellence and to beat the competition. Therefore, the company wants to decrease the number of tests waiting for its result. To evaluate this, it proceeds as following: for any interval of time, for example one week (6 working days with 12 working hours per day), the difference between the number of new tests and the number of results available to the client during each half an hour period is computed. In that case, a list with 144 integers is obtained, where a positive integer means that in such half an hour more tests were processed than results were obtained, and 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 week, when the company was less effective in responding. So, the application should implement a brute-force algorithm (an algorithm which examines each subsequence) to determine the contiguous subsequence with maximum sum, for any interval of time registered. The implemented algorithm should be analysed in terms of its worst-case time

complexity, and it should be compared to a provided benchmark algorithm. The algorithm to be used by the application must be defined through a configuration file.
The complexity analysis must be accompanied by the observation of the execution time of the algorithms for inputs of variable size in order to observe the asymptotic behaviour. The time complexity analysis of the algorithms should be properly documented in the application user manual (in the annexes) that must be delivered with the application.

Considering that *Many Labs* has the exclusivity to perform Covid-19 tests, and that the contract with the NHS in England requires *Many Labs* to summarize and report Covid-19 data, the company needs to: identify the number of Covid-19 tests performed, identify all positive results to Covid-19 tests, report the total number of Covid-19 cases per day, per week and per month of the year, and send the forecasts for these same time horizons (the number of Covid-19 cases for the following day, next week and next month). The company is also required to generate daily (automatic) reports with all the information demanded by the NHS and should send them to the NHS using their API. To make the predictions, the NHS contract defines that a linear regression algorithm should be used. The NHS required that both simple linear and multiple linear regression algorithms should be evaluated to select the best model. The accuracy of the prediction models should be analysed and documented in the application user manual (in the annexes) that must be delivered with the application. The algorithm to be used by the application must be defined through a configuration file.

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 specialist doctor is allowed to access all client data. The application must support the English language only.

During 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 methods that implement Input/Output operations. The unit tests should be implemented using the JUnit 4 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.

2.3 Clinical examinations data

Many Labs performs two types of tests. Each test is characterized by an internal code, an NHS code, a description that identifies the sample collection method, the date and time when the samples were collected, the date and time of the chemical analysis, the date and time of the diagnosis made by the specialist doctor, the date and time when the laboratory coordinator validated the test, and the test type (whether it is blood test or Covid test).

2.3.1 Blood tests

Blood tests are frequently characterized by measuring several parameters which for presentation/reporting purposes are organized by categories. For example, parameters such as the number of Red Blood Cells (RBC), White Blood Cells (RBC) and Platelets (PLT) are usually presented under the blood count (Hemogram) category,

2.3.2 Covid tests

Covid tests are characterized by measuring a single parameter stating whether it is a positive or a negative result.

2.3.3 Other tests support

Despite being out of scope, the system should be developed having in mind the need to easily support other kinds of tests (e.g., urine). Regardless, such tests rely on measuring one or more parameters that can be grouped/organized by categories.

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 to be organized in teams of 4 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:

- 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 starting the sprint.

Table 1: Timetable of the semester

Sprint	Start (week)	End (week)	General objective	UCs involved
A	1	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 MATCP MDISC 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