

## Integrative Project - Spring Semester 2021/2022

### Project Requirements

This document describes the work to be developed by the teams during the semester. Most of the software application features are informally described from an end user's perspective. The Integrative Project will be developed in four sprints:

#### Sprint A. Integrative Project Analysis

- *Requirements Specification and Analysis*
- *Configure the Bitbucket Repository*

#### Sprint B: User Stories to develop

- US3: As a receptionist, I want to register a SNS user.
  - **Acceptance criteria:** The SNS user must become a system user. The "auth" component available on the repository must be reused (without modifications).
- US9: As an administrator, I want to register a vaccination center to respond to a certain pandemic.
- US10: As an administrator, I want to register an Employee.
  - **Acceptance criteria:** Each user must have a single role defined in the system. The "auth" component available on the repository must be reused (without modifications).
- US11: As an administrator, I want to get a list of Employees with a given function/role.
- US12: As an administrator, I intend to specify a new vaccine type.
- US13: As an administrator, I intend to specify a new vaccine and its administration process.

#### Sprint C: User Stories to develop

- US1: As a SNS user, I intend to use the application to schedule a vaccine.
  - **Acceptance criteria:** A SNS user cannot schedule the same vaccine more than once. The algorithm should check if the SNS User is within the age and time since the last vaccine.
- US2: As a receptionist at one vaccination center, I want to schedule a vaccination.
  - **Acceptance criteria:** A SNS user cannot schedule the same vaccine more than once. The algorithm should check if the SNS User is within the age and time since the last vaccine.
- US4: As a receptionist at a vaccination center, I want to register the arrival of a SNS user to take the vaccine.
  - **Acceptance criteria:** No duplicate entries should be possible for the same SNS user on the same day or vaccine period.
- US5: As a nurse, I intend to consult the users in the waiting room of a vaccination center.
  - **Acceptance criteria:** SNS Users' list should be presented by order of arrival.
- US14: As an administrator, I want to load a set of users from a CSV file.

- **Acceptance criteria:** The application must support importing two types of CSV files: a) one type must have a header, column separation is done using ";" character; b) the other type does not have a header, column separation is done using "," character.
- US18: As a center coordinator, I want to get a list of all vaccines.
  - **Acceptance criteria:** The vaccines should be grouped by their type and then alphabetically listed by its name.

## Sprint D: User Stories to develop

- US6: DGS wants to record daily the total number of people vaccinated in each vaccination center.
  - **Acceptance criteria:** The algorithm should run automatically at a time defined in a configuration file and should register the date, the name of the vaccination center and the total number of vaccinated users.
- US7: As a nurse, I intend to record adverse reactions of a SNS user.
- US8: As a nurse, I want to record the administration of a vaccine to a SNS user. At the end of the recovery period, the user should receive a SMS message informing the SNS user that he can leave the vaccination center.
  - **Acceptance criteria:** The nurse should select a vaccine and the administered dose number.
- US15: As a center coordinator, I intend to check and export vaccination statistics. I want to export, to a csv file, the total number of fully vaccinated users per day.

The goal of this US is to explore this data jointly with data obtained from external sources to estimate the number of new cases and the number of deaths using information about the *Reproduction Rate, Number of ICU Patients, Number of In-Hospital Patients, Number of New Cases, Positive Rate and Number of People Fully Vaccinated*. In this study, simple linear and a multi-linear regression models should be developed to find the linear relationship between: 1- each independent variable and each dependent variable; 2- all six independent variables and each dependent variable; 3- three selected variables and each dependent variable. As the application is under development it is not possible to generate all data required for the analysis. Therefore, and to simulate a production system, we made available in moodle a file containing all the information required for this study. The regression analysis should be made outside the application and should be performed using Microsoft Excel spreadsheet program. The regression analysis should be documented in the application user manual (in the annexes) that must be delivered with the application. The report should include day and week (observed and estimated) values, the regression model used to estimate each value, R(SLR), R<sup>2</sup> and R<sup>2</sup> adjusted for SLR and MLR, confidence intervals and hypothesis tests for regression coefficients and significance model with Anova.

- US16: As a center coordinator, I intend to analyze the performance of a center.

The goal of this US is to implement a procedure that, for a specific day, and time intervals of  $m$  minutes (for  $m = 30, 20, 10, 5$ , and  $1$ , for example) chosen by the coordinator of the center, with a daily work from 8 a.m. to 8 p.m., the procedure creates a list of length  $720/m$  (respectively, lists of length  $24, 36, 72, 144, 720$ ), where the  $i$ -th value of the list is the difference between the number of new clients arriving and the number of clients leaving the center in that  $i$ -th time interval.

Then, the application should implement a brute-force algorithm (an algorithm which examines all the contiguous sublists of the input one) to determine the contiguous sublist with maximum sum. The output should be the input list, the maximum sum contiguous sublist and its sum, and the time interval corresponding to this contiguous sublist (for example, for time intervals of 1 hour, a list of length 12 is created; if, for example, the maximum sum contiguous sublist starts at the 2nd and ends at the 5th entries of the input list, with a sum  $s$ , it means that the vaccination center was less effective in responding from 9 a.m. to 1 p.m., with  $s$  clients inside the center).

The performance analysis should be documented in the application user manual (in the annexes) that must be delivered with the application. Also in the user manual, the implemented algorithm should be analyzed in terms of its worst-case time complexity. The complexity analysis must be accompanied by the observation of the execution time of the algorithms, and it should be compared to a benchmark algorithm provided, for inputs of variable size  $m$ , with  $m = 24, 36, 72, 144, 720$ , in order to observe the asymptotic behavior.

- US17: As a center coordinator, I want to import data from a legacy system that was used in the past to manage centers. The imported data should be presented to the user sorted by arrival time or by the center leaving time. The name of the SNS user and the vaccine type *Short Description* attribute should also be presented to the user.
  - **Acceptance criteria:** Two sorting algorithms should be implemented (to be chosen manually by the coordinator), and worst-case time complexity of each algorithm should be documented in the application user manual (in the annexes) that must be delivered with the application. The center coordinator must be able to choose the file that is to be uploaded.

#### **Other Requirements:**

- The users should use a graphical user interface to access the features introduced in Sprint D.
- The application should use object serialization to ensure persistence of the data between two runs of the application. Serialization must be applied to all classes developed in all sprints.