SYSC 3120: Software Requirements Engineering
Module 1 Lab Assignment

Winter 2022

Dr. J. Jaskolka
Carleton University
Department of Systems and Computer Engineering

Posted: January 9, 2022

Due: January 23, 2022

Due on Sunday, January 23, 2022 by 11:59PM

This assignment contains 4 pages (including this cover page) and 5 problems. You are responsible for ensuring that your copy of the assignment is complete. Bring any discrepancy to the attention of your instructor.

Special Instructions:

1. **Do as many problems as you can.**
2. Start early as this lab assignment can be much more time consuming than you might initially think!
3. The burden of communication is upon you. Solutions not properly explained will not be considered correct. Part of proper communication is the appearance and layout. If we cannot "decode" what you wrote, we cannot grade it as a correct solution.
4. You may consult outside sources, such as textbooks, but *any use* of *any source* **must** be documented in the assignment solutions.
5. You are permitted to discuss *general aspects* of the problem sets with other students in the class, but you must hand in your own copy of the solutions.
6. Your lab assignment solutions are due by 11:59PM on the due date and must be submitted on Brightspace.
   - Late assignments will be graded with a late penalty of 20% of the full grade per day *up to 48 hours past the deadline.*
7. You are responsible for ensuring that your lab assignment is submitted correctly and without corruption.

Problem	1	2	3	4	5	Total
Points:	6	8	10	8	8	40

The purpose of this lab assignment is to exercise your understanding of functional and non-functional requirements and writing high-quality requirements. It aims to assess your ability to explain and identify the different types of requirements and to document high-quality requirements to describe the visible behaviour of a software system.

The following example system, which describes an application for managing vaccinations, patient records, and appointments, illustrates the kind of informal specifications that a software or systems engineer must translate to a computable form.

CoVax: A COVID-19 Vaccination Information System

The *CoVax* system to be developed is intended to support the day-to-day operations of medical clinics offering support for patients requiring or requesting COVID-19 vaccinations. Most patients that require vaccinations do not require dedicated hospitals but need to attend specialist or ad hoc clinics that can administer the needed vaccinations. *CoVax* is intended to be a medical information system that maintains information about patients and the vaccinations that they require or have received. To make it easier for patients to receive their vaccinations, these clinics are not just run in hospitals. They may also be held in local medical practices, community centres, or local pharmacies. Therefore, the system must be able to accomodate many simultaneous users. *CoVax* aims to generate management information that allows health service managers to assess performance against local and government targets, as well as to provide medical staff with timely information to support patient care. Currently, clinics offering such support operate by accepting walk-in patients (which sometimes contributes to long wait times) and maintaining paper records stored in large filing cabinets in administrative offices. This make its difficult to share information across clinics and diminishes the level of patient support that can be offered.

Many patients that choose to be vaccinated against COVID-19 simply need to receive the vaccination and a record needs to be kept about their history and the type of vaccine (e.g., Pfizer, Moderna, AstraZeneca, etc.). However, in some cases, patients may have specific medical conditions that require them to receive certain vaccines. These patients are considered to be part of the "*at risk*" group. Conversely, some patients may have specific medical conditions or allergies to certain substances that prevent them from safely receiving certain vaccines.

Users of the system will include clinical staff such as doctors, nurses, and pharmacists. Non-medical users will include receptionists who make appointments, medical records staff who maintain the records system, and administrative staff who generate reports.

The *CoVax* system needs to record information about patients (name, address, age, etc.), medical conditions or allergies, and past vaccines. Reports are generated at regular intervals for medical staff and health authority managers. Typically, reports for medical staff focus on information about individual patients, whereas management reports are anonymized and are concerned with conditions, vaccine supplies and demands, etc.

Clinicians should be able to create records for patients, edit the information in the system, view patient vaccination history, and so on. The system must support the generation of data summaries so that doctors who have not previously met a patient can quickly learn about the patient's COVID-19 vaccine record and existing conditions. The system should issue warnings if possible problems in records are detected. These checks should be executed automatically on a fixed period. One of the most important elements of the monitoring system is to keep track of patients who are in the "*at risk*" group to ensure that their vaccination records are up-to-date.

The system must generate monthly management reports showing the number of patients vaccinated at each clinic, the number of patients "*at risk*", the vaccines offered, etc. The system should display this information in a simple, clean, and understandable form.

CoVax must comply with relevant laws on data protection that govern the confidentiality of personal information and health laws that govern vaccinations in a given jurisdiction.

As in all medical systems, privacy is essential. It is essential that patient information is confidential and is never disclosed to anyone apart from authorized medical staff and the patient themselves. *CoVax* is also a safety-critical system. Some vaccines cause patients to have severe allergic reactions. Wherever possible, the system should warn medical staff about potentially incompatible patients with requested vaccines. *CoVax* must be available when needed and requests for patient information must be handled immediately; otherwise safety may be compromised, and it may be impossible determine if a vaccine is safe to be administered to a patient.

It should be possible for the *CoVax* system to be accessed and used from sites that do not have secure network connectivity. When the local systems have secure network access, they use patient information in a central database, but they can download and use local copies of patient records when they are disconnected. The system is not a complete medical records system and so does not maintain information about other medical conditions. However, it may interact and exchange data with other clinical information systems.

The overall budget for the development of this new system is $525,000. The system must run on Windows and MacOS and mobile devices (such as Android and iOS tablets), all of which are used in the clinics. It must be ready as soon as possible.

Important Notes

The primary challenge in this lab lies in identifying the relevant details from an informal description of a system. Often software engineers are provided with a messy collection of details for a problem and they must make decisions about which details are important and which are not. They need to make important judgements to decompose complex problems into a form that is comprehensible by a variety of system stakeholders. In practice, there is no person that is able to tell a software engineer precisely how many requirements, operations, and stakeholders there are, or should be, for a particular system. For this reason, a software engineer needs to be systematic in their approach, and develop a system that captures the most important details that must be conveyed to the relevant stakeholders.

Submission Requirements

Please read the following instructions very carefully and follow them precisely when submitting your assignment!

The following items are required for a complete assignment submission:

1. **PDF Lab Assignment Solutions**: Submit a detailed document that carefully and concisely describes your solutions to the problems in this lab assignment. Please ensure that your document is well-organized and that the problem numbers and sub-parts are clearly provided to facilitate the grading of your solutions.

Grading Notes

An important part of this lab assignment is following instructions. As such, the following grade **penalties** will be applied for failure to comply with the submission requirements outlined above:

- Failure to submit the Lab Assignment Solutions will result in a grade of 0 for the assignment.
- Failure to submit the Lab Assignment Solutions in the required format (PDF) will result in deduction of 5% of the full grade of the assignment.

Problem 1 [6 points]

With respect to the *CoVax* description on Page 2:

(a) [2 points] Identify the System-as-is.

(b) [2 points] Identify the System-to-be.

(c) [2 points] Identify the Software-to-be.

Problem 2 [8 points]

With respect to the *CoVax* description on Page 2, identify **at least eight (8)** functional requirements of the system-to-be. Remember that some of the functional requirements may not necessarily be stated explicitly in the provided system description! Because you may have to add missing information or clarify requirements, try to always make reasonable assumptions.

Problem 3 [10 points]

With respect to the *CoVax* description on Page 2, identify **two** non-functional requirements that are reasonable for each of the following categories. Remember that some of the non-functional requirements may not necessarily be stated explicitly in the provided system description! Because you may have to add missing information or clarify requirements, try to always make reasonable assumptions.

(a) [2 points] Interface Requirements

(b) [2 points] Performance Requirements

(c) [2 points] Architectural Requirements

(d) [2 points] Compliance Requirements

(e) [2 points] Development Requirements

Problem 4 [8 points]

Consider the *CoVax* description provided on Page 2 stating some of the system requirements. Referring to the guidelines for writing high-quality requirements discussed in Lecture 1.4, list as many problems as you can.

NOTE: Although the description is rather short, do not rush! Read it over and over many times and identify statements that are not verifiable, vague, ambiguous, incomplete, missing, lacking metrics, etc. There are at least 8 to 10 problems in the stated requirements.

Problem 5 [8 points]

Rewrite each of the statements of requirements of the *CoVax* description from Problem 4, solving each of the problems you have found. Use the standard form for writing requirements statements discussed in Lecture 1.4. Because you will have to add missing information or clarify requirements, try to always make reasonable assumptions.

END OF LAB ASSIGNMENT