

VAXSYS

1. INTRODUCTION

This project is about designing, describing, implementing a proof-of-concept, and evaluating the software architecture of a vaccination management system (VMS).

In the rest of the document, both the project and the product are mnemonically-named **VAXSYS**.

It is expected that by the end of project, the students will be able to **understand a non-trivial domain, namely Health Informatics, to understand the needs of the stakeholders of that domain, and to become aware of the size and complexity** of one class of significant software for that domain.

2. PROJECT-LEVEL CONCERNS

A team is expected to make use of **project management body of experiential knowledge**, such as **collaboration patterns, thinklets, and articles in the book “97 Things Every Software Architect Should Know”**. For example, this can be done through practice and documenting such practice.

The work on VAXSYS has been divided into a collection of deliverables.

DELIBERATION

A team must ensure that the work towards VAXSYS is **visible**. It is imperative that each team **meet** regularly (at least twice a week), as well as use **Social Software**. For example, **GitHub/GitLab, Google Docs, Microsoft Teams, WhatsApp Group, and Wiki Hosting Service** are such social software. The selected social software should be used by a team for **communication as well as collaboration**. The **roles and responsibilities** to be carried out by team members pertaining to each deliverable should also be posted on the selected social software. Let $S(X)$ be social software being used by Team X . Then, the roles and responsibilities to be carried out by team members pertaining to each deliverable must be made available publicly as early as possible (and no later than May 31, 2022 for **all** the deliverables).

ELUCIDATION

A team must commit to a uniform basis for terminology across its members. To do that, a **glossary** related to the project must be created and maintained throughout the project. It is that the glossary be developed alongside (specifically, in **parallel** with) other activities. It is preferable to use “official” or otherwise recognized definitions in the glossary.

ABSTRACTION

A team must aim for a uniform working environment throughout the project. To do that, a **specific UML version** should be committed to, and a **single UML modeling tool** must be selected and used for all the artifacts that require the use of UML.

ATTRIBUTION

A team must strive for the highest standard of academic ethics. To do that, a team must **cite and reference appropriately** any and all non-original work (that is, any work external to that team). A comprehensive collection of resources on citing and referencing is available¹. For example, **ACM, APA, and IEEE** provide standard formats for citing and referencing. It is important not to make claims that cannot be substantiated, and not to copy others’ work verbatim regardless of whether it is cited. A copied work does not earn any credit.

3. PEOPLE-LEVEL CONCERNS

VAXSYS is a team project involving **both** individual and communal work.

4. PROCESS-LEVEL CONCERNS

VAXSYS does not follow any particular software process model per se. However, it acknowledges that large-scale software development is based on a systematic **acquisition, distillation, and communication of knowledge**. Therefore, VAXSYS progresses through **iteration and incrementation**, follows a number of **agile practices**, is informed by experiential knowledge (such as patterns), and is driven by artifacts.

VAXSYS must be **general** but at the same time also **feasible**, by being mindful of the given time and resource constraints. To do that, VAXSYS could be **scoped** in a number of ways. For example, VAXSYS could focus only on specific location (say, clinic or hospital based in Montreal), or on specific type of mobile devices (say, smartphones). However, scoping should not inhibit the ability (specifically, reusability and extensibility) of VAXSYS to be used as a **product line**². For example, if a distributed version of VAXSYS has been developed, then to develop a non-distributed version of VAXSYS should not mean repeating from scratch the design and implementation of VAXSYS, and conversely. Furthermore, VAXSYS could serve as a variant for a general VMS.

¹ URL: <http://library.concordia.ca/help/howto/citations.html> .

² URL: http://www.sei.cmu.edu/productlines/frame_report/index.html .

5. PRODUCT-LEVEL CONCERNS

VAXSYS is expected to have several capabilities, including the following:

- It will allow people using essentially any computing device, at any time, and from anywhere to schedule an appointment for a vaccination, cancel an appointment, and provide pertinent personal details for such an appointment (including, but not limited to, name, address, age, any current medications, and any allergies).
- It will have information on the **types of vaccines** for **multiple, common, infectious diseases**.
- It will have information (address, contact information, and so on) on vaccine centers.
- It will provide **instructions** to potential vaccine recipients for vaccination.
- It will be able to **remind** people of their vaccination appointment.
- It should allow an individual to receive mixed doses of different types of vaccines for the same infectious disease.
- It should **not** allow an individual without a valid personal identification to receive a vaccine.
- It should **not** allow an individual to receive more than the number of authorized doses of a vaccine for a particular infectious disease.
- It will be able to provide vaccine recipients with a **proof of vaccination record**.
- It will allow authorized personnel to enter pertinent information (including, but not limited to, any side-effects of vaccination) about a vaccine recipient.
- It will allow information to be **archived, modified, navigated, searched, and retrieved**.
- It must be **credible**, by avoiding misinformation and by avoiding the potential for **infodemic**.
- It must be **usable, in general, and accessible, in particular**, within Canada by people using a mobile device.
- It must aim to support **privacy** of information of vaccine recipients.
- It must aim to provide **security** to all its users (including authorized personnel, system administrators, and vaccine recipients).
- It must be **maintainable**. This is important because VAXSYS is expected to evolve over time. For example, vaccine center's opening hours or recipient's personal information might change.

VAXSYS could be **inspired** by other (commercial or non-commercial, open or closed source) VMS³. However, cloning of an existing VMS is **not** expected.

³ URL: [https://www.google.com/search?q=\"vaccination management system\"](https://www.google.com/search?q=\).

6. DELIVERABLE 1 (D1)

PART 1: VISION [20 MARKS]

This involves giving the **business goal** of VAXSYS.

The business goal of VAXSYS could be decomposed into sub-goals, if necessary. This could, for example, be done using the **GQM Framework**.

PART 2: STAKEHOLDERS AND CONCERNS [40 MARKS]

- (a) This involves eliciting a **mind map of stakeholders** and, based on which, giving a **list of stakeholders** of VAXSYS, possibly organized (categorized) in some manner if deemed necessary.
- (b) This involves giving a list of concerns corresponding to each stakeholder. It also involves giving a list of quality attribute(s) corresponding to each product-related concern. (In other words, there must be a clear **stakeholder ↔ product-related-concern ↔ quality attribute mapping**.)

PART 3: PROBLEM [40 MARKS]

This involves giving the **technical definition** of VAXSYS.

The technical definition of VAXSYS should consist of software requirements, as outlined by the **ISO/IEC/IEEE 29148** Standard.

These software requirements must include requirements pertaining to desirable quality attributes, namely **credibility, maintainability, privacy, security, and usability**, among others.

PART 4: VIEWPOINTS AND VIEWS [40 MARKS]

This involves giving a list of views relevant to VAXSYS. The inclusion of each view must be **rationalized** briefly. The relationship of a view to a viewpoint model, if any, must be given.

This must conform to the **ISO/IEC/IEEE 42010** Standard.

PART 5: ARCHITECTURAL DECISIONS [40 MARKS]

This involves identifying and listing architectural knowledge (available explicitly, say, in the form of heuristics, patterns, principles, styles, and/or tactics) to be used in the software architecture of VAXSYS. The selection of such knowledge must be **rationalized** briefly.

PART 6: PROOF-OF-CONCEPT [40 MARKS]

This involves implementing N elements, one per team member, of the software architecture of VAXSYS, where N is the team size. The selection of such elements will be decided as the course progresses.

NOTES

Parts 1-2 should focus on the **‘why’** and the **‘what’**, not on the **‘how’**. The purpose of the **‘why’** is related to **viability assessment**. The purpose of the **‘what’** is related to the **problem** (namely, requirements, which may be stated informally or formally).

Parts 3-5 must conform to the **ISO/IEC/IEEE 42010** Standard, must be **informed by** each other, and be **consistent with** each other.

Part 6 must be carried out using technologies agreed collectively by the team.

In general, the diagrams, if any, pertaining to the software architecture description, must be expressed in UML, and in appropriate UML diagram types. If UML is deemed not a viable option, then the diagrams could be expressed in a “box-and-lines” form, such that each notational construct included in the diagrams is defined explicitly.

It is important that all critical arguments are **rationalized**, which may require citations to references, as necessary.

There should be a note pointing out to any **principles**⁴ and/or **practices**⁵ of **Agile Modeling**⁶ that were deployed during conceptual modeling.

The quality of both the **representation** and the **presentation** of information will be determinants in marking. In particular, **syntactic, semantic, and pragmatic** concerns of artifacts are significant.

⁴ URL: <http://www.agilemodeling.com/principles.htm> .

⁵ URL: <http://www.agilemodeling.com/practices.htm> .

⁶ URL: <http://www.agilemodeling.com/> .

The deliverable must include the names of the team members and the details of contribution that are verifiable and accessible to the entire team.

7. DELIVERABLE 2 (D2)

PART 1: CHECKLIST FOR EVALUATION [20 MARKS]

This involves preparing a checklist of criteria for evaluation of software architecture description, a scale for satisfying each criterion, a weight for each criterion such that the weights add up to 100.

PART 2: RECIPROCAL PEER REVIEW [30 MARKS]

This involves conducting a reciprocal peer review, that is, a systematic review of D2 of another team that is based on the checklist in Part 1. It is important that the review aims to be **accurate**.

NOTES

The checklist must be based on the framework (of questions) given in the following publication: **Question Framework for Architectural Description Quality Evaluation**⁷.

The criteria in the checklist must be **rationalized**.

The quality of both the **representation** and the **presentation** of information will be determinants in marking. In particular, **syntactic, semantic, and pragmatic** concerns of artifacts are significant.

The deliverable must include the names of the team members and the details of contribution that are verifiable and accessible to the entire team.

⁷ URL: <http://dx.doi.org/10.1007/s11219-008-9068-1> .

8. DELIVERABLE 3 (D3) [30 MARKS]

This involves, in a poster, slides, or video presentation, defending major decisions made in D1, providing a rebuttal on the comments received from D2, and reflecting on the review received from D2 (such as, by giving a perspective on the lessons learned).

NOTES

The **team in the audience** must **prepare (in real-time or otherwise) two questions and pose those questions** to the team that is presenting.