SOFTWARE ARCHITECTURE AND QUALITY

Assignment - 2

Group 8

Sree Kavya Ganja-20000217T122

Prudhvi Nath Naidu Cherukuri-199812125533

Jeevan Sai Koushik Masavarapu-200002047512

MVVS Varma-19990310T195

Khalil Anjum

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PART-A

DOCUMENTATION BEYOND VIEWS

DOCUMENTATION ROADMAP:

The first part of the document provides an insight into how the document has been structured. First, how the views in Part B are documented i.e., their structure, and what is discussed in each view. Next, the system is discussed from a high-level perspective without going into specifics. Then, the chosen architectural pattern is discussed along with how the system satisfies the quality attributes provided in the system description. Then, the rationale behind the chosen architectural pattern is explained.

The second part of the document discusses the two views of the system: Module View and Component & Connector View. The module view is aimed at providing a modular view of the system while the component-connector view shows the system in a runtime point of view. Both the views have pictorial representations, an element catalogue that discusses each component in the diagram, along with their relations.

HOW ARE THE VIEWS DOCUMENTED?

In this document, two views are discussed:

1. Module View
2. Component & Connector View

Module view provides a pictorial representation of the modular composition of the system. Then, an element catalogue explains about the elements, relationships, and their properties. It also describes element interfaces and their behaviors. Context diagram explains the boundary between the system and its environment, showing how the system interacts with outside entities. Variability guide helps to show the variation points that are part of the modular view of the architecture. Rationale explains the reasons behind various decisions taken for the system.

Component & Connector View provides a runtime view of the system and its various components. The C&C view has the similar structure as the module view discussed earlier.

SYSTEM OVERVIEW:

As the nation health authority has been instructed to develop a new system for citizens which gives verifiable covid vaccine certificates for international travel so, COVTRA is the new proposed system. Attribute-driven design will be used in architecture of the COVTRA system. COVTRA system has three kinds of users – citizens, vaccinators, and immigration officers. Citizens can request certificates, vaccinators can update dosages, and immigration officers can validate the certificates for travel. It is a greenfield system in mature domain. The motive of this document is to provide information about design and design decisions about the system.

APPLIED ARCHITECTURAL PATTERN:

The architectural pattern used for this system is microservice oriented architecture. Where each of the service provide necessary functionalities to the system. Microservice is a distributed system, and all the components can be deployed independently. The reason behind choosing this architectural patter for the problem at hand as this pattern offers increased modularity making applications easier to develop and deploy. The changes and the maintenance of the system is made simple in this style as these services are separated into smaller segments which are interlinked using light weight protocols to maintain less dependency. The application will be broken down into different segments each having a different functionality which is not possible using other architectural patterns. This results in increased resilience. The advantages of micro service architecture are:

1. This architectural pattern enables the developer to continuously deliver and deploy the large complex applications.
2. Maintainability: it is easy to maintain the task model as the system will be divided into smaller components.
3. Fault isolation: if there is a problem in one service component then the system will continue to work normally.
4. Scalability: the scalability of a particular functionality is easy and made possible in microservice architecture as each functionality is divided into separate component and the scalability of a particular functionality is independent and does not affect the rest of the components of the system.
5. Easy debugging and maintenance: as this contain small components of functionalities it is easy to debug and test these functionalities to enable continuous delivery of these components.

As per the prioritize quality attributes that have been selected in the architectural drivers of the system, the quality attributes are mapped with advantages of selecting microservice architecture mentioned above.

Availability: as the services are independent, if there is a problem in one service the system will run as usual without having to stop all the services. The problem will be resolved during the downtime.

Performance: the services are scalable and deployed independently therefore can handle a large amount of user requests concurrently.

Security: we can resist attacks by adding an authentication layer and encrypting the data.

Modifiability: any service can be replaced easily without having to change the entire system.

Usability: it is easy to maintain the task model as the system will be divided into smaller components.

RATIONALE:

The microservice architecture was chosen as it seemed to fit the system description best. Since the system needs to be able to accommodate future changes easily, the microservice architecture is well suited. A common database interface was deemed a good choice instead of having the system communicate with individual databases independently because NCTRS would be decommissioned shortly and having a common database interface ensures that only a few changes need to be made to the interface instead of having to change each service/component of the system when NCTRS is decommissioned. The services in the architecture are scalable and therefore, can be easily made to accommodate more and more service requests. An additional authentication layer ensures security of the services and encryption of data protects it from attacks.

PART-B

MODULE VIEW DOCUMENTATION

SECTION-1 PRIMARY PRESENTATION

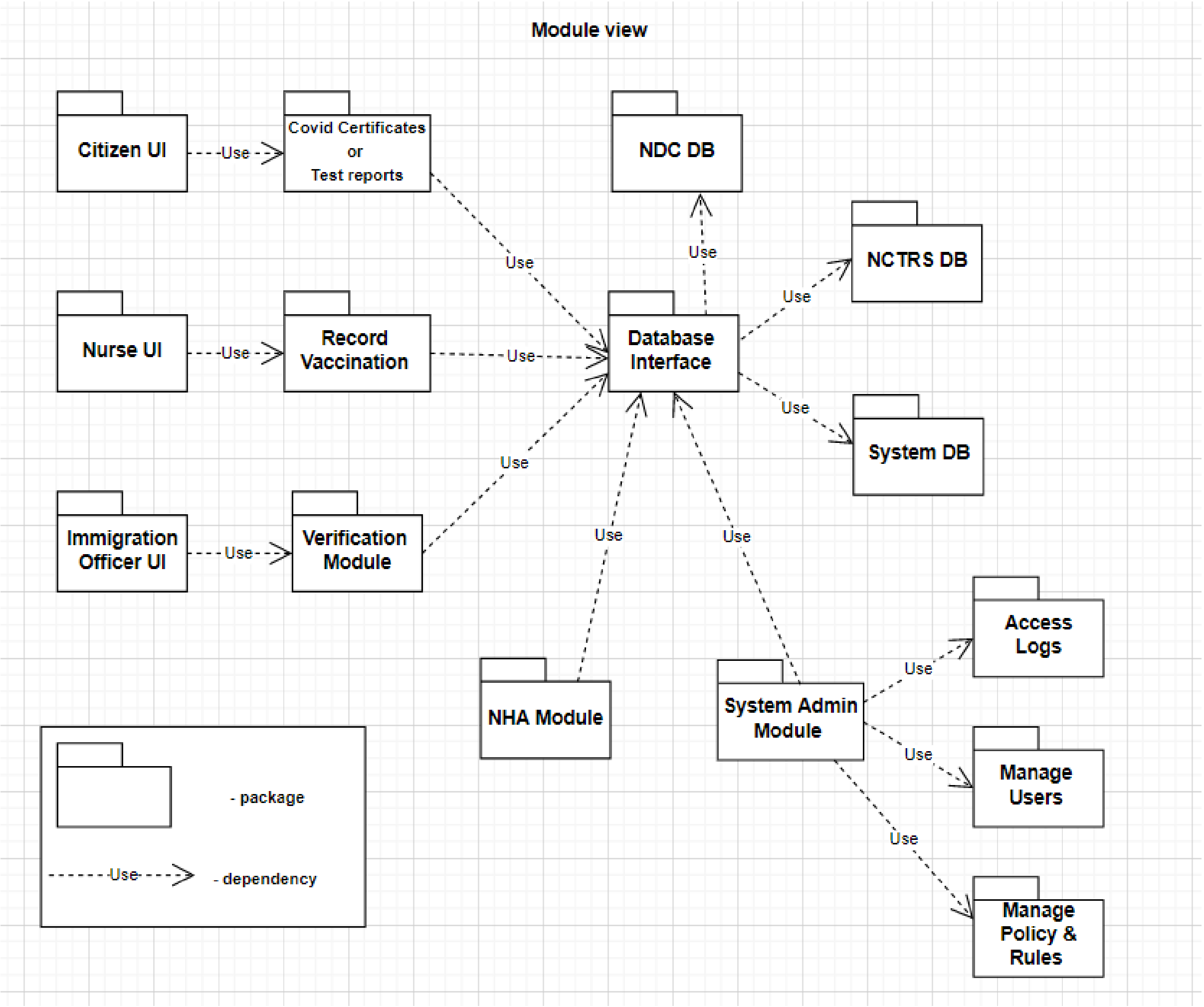


Figure 1. Primary Representation of Module View

SECTION – 2 ELEMENT CATALOGUE

Section 2a Elements and their properties

Elements	Properties
Citizen UI	This serves as the interface between the system and the citizen. This has information regarding the certificates and test reports.
Covid certificates or test reports	This module generates covid vaccination certificates and test reports and stores them in the data base.
Nurse UI	This module is used by the nurse to record the vaccination dosages and other information regarding vaccination and stores them in the data base.
Record vaccination	Whenever the nurse updates the vaccination details, this module helps commit the vaccination data to the data base.
Immigration officer UI	The immigration officer uses this interface to verify the vaccination certificates or test reports for travel.
Verification module	This module enables the immigration officer to verify the certificate.
NHA module	The complaint staff will access the database through this module to follow up on the complaint tickets.
System admin module	This module helps the system admin to communicate with the system and database to access logs, manage users, manage policy and rules.
Access logs	This module helps the system admin to access the system logs, event logs, audit logs.
Manage users	This module helps the system admin to manage covtra users, and their privileges.
Manage policy and rules	The system admin uses this module to manage the policies and rules to issue verifiable certificates for travel.
Database interface	A common interface that interacts with all the databases i.e., NDC, NCTRS and the system database according to the requests received.
NDC	This module contains the data related to the citizen.
NCTRS	This module has covid related information and it will also decommission in the future this module will be replaced by NHA -HMIS.
System database	It contains all the system logs, audit logs, event logs, vaccination records and all the generated covid vaccine certificates will be stored in this database.

Table 1. Elements and their Properties - Module View

Section 2b Relations and their properties

Relations	Properties
Citizen UI – covid test reports and vaccine certificates	The relation relationship between these two modules is dependency. Citizen UI module uses the covid vaccination or test reports module to generate vaccine reports or test reports.
Covid test reports and vaccine certificates – database interface.	The module test reports, and vaccination certificates depends on the database interface to generate certificates based in the data from the system database and test reports from the NCTRS database. The citizen can download the certificate or test report from UI.
Nurse UI – Record Vaccination	The module nurse UI will record vaccination information using the record vaccination module.
Record vaccination – database interface	The data recorded by the nurse will be stored in the system database and when the user requests for a covid certificate then the certificate will be generated based on the stored information.
Immigration officer UI – Verification module	The immigration officer uses the verification module to verify the vaccination certificates or test reports of citizens for travel. This verification module will also accept the verification requests from the national and EU systems through the API.
Verification module – database interface	The verification module depends on the database to verify the vaccination certificates which are stored in the system database and test reports stored in the NCTRS database.
NHA module – database interface	The NHA module depends on the database interface to carry out the complaint tickets regarding the covtra data to check and correct information in the database.
System admin module – database interface	The system admin module depends on the database interface to carry out its main functionalities accessing logs, managing users, and managing policies and rules.
Database interface – NDC data Database interface- NCTRS Database interface- system database.	NDC, NCTRS, and system database are separate databases through the database interface and based on the requests received the interface redirects to that database.
System admin module – access logs System admin module – manage users System admin module – manage policy	Access logs, manage users, and manage policy are the main functionalities of the system admin and through the system admin module the admin performs the required action.

Table 2. Relations and their properties - Module View

Section 2c Element Interfaces

1. Citizen UI

- Interface identity: This interface provides the user with functionalities like generating a covid vaccination certificate or test reports to facilitate travel after authentication from the user.
- Locally defined data types: In this interface the datatype string is used other than the data types used by the programming language.
- Quality attribute: This interface should provide better usability and performance in terms of quality attributes.
- Requirements of the element: This element uses the data from the database interface to generate test reports and vaccination certificates.
- Rationale and design issues: The main motivation behind this interface is that the citizen UI helps the citizen to communicate with the system to get covid certificates and test reports providing different features like changing language, complaint services to have a good user experience.

2. Nurse UI

- Interface identity: this interface is mainly used by the nurse to record vaccination dosages and other vaccination related information and store to the database.
- Locally defined data types: the nurse UI mainly uses strings and integers to record the data.
- Quality attributes: the interface provides better usability, performance, modifiability to the nurse.
- Requirements of the element: the nurse enters the data of the citizen and stores it in the system database.
- Rationale and design issues: the main motivation here is to register the vaccination dosages of the citizen into the covtra system.

3. Immigration officer UI:

- Interface identity: this interface is used by the immigration officer to verify the covid test reports or vaccination certificates during travel.
- Locally defined datatypes: this interface mainly uses the string data type to verify the certificates or test reports of the citizen based on their credentials.
- Quality attributes: the interface must provide better performance.
- Requirements of the element: the immigration in uses the data related to the vaccination certificate or test report from the database interface.
- Rationale and design issues: the main motivation is to provide the immigration officer to have an interface where the officer will verify the certificate or test report using the data base interface.

4. NHA:

- Interface identity: this interface is used by the complaint staff to manage the complaints received from the user. The complaint staff can also change or correct the data in the system as requested by the user.
- Locally defined datatypes: the datatypes that can be used in this interface are strings.
- Quality attributes: modifiability, performance.
- Requirements of the elements: NHA will use all the data from the system.
- Rationale and design issues: the main motivation is to update, correct or delete information based on the ticket raised by the user accessing all the data present in the system.

5. System admin:

- Interface identity: this interface is used by the system admin to access system logs, audit logs and event logs, to manage users and their access privileges, to manage policy and rules in the system.
- Quality attributes: performance, security, modifiability.
- Requirements of the elements: this interface will depend on the database interface completely to manage users, accessing logs and also managing policy and rules for issuance of the certificate.
- Rationale and design: this interface will be designed mainly to carry out system admin functionalities.

6. Database interface:

- Interface identity: this interface is the common interface for the databases that are used in the system NDC, NCTRS, and system database. The accessing of the database is based on the request of the user or other stakeholders.
- Quality attributes: security, modifiability, accessibility
- Rationale and design: this interface are designed to provide a level of abstraction to make it easy for the future updates. If we change the data in a single module we don’t have to change in the whole system.

Section 2d Element behavior

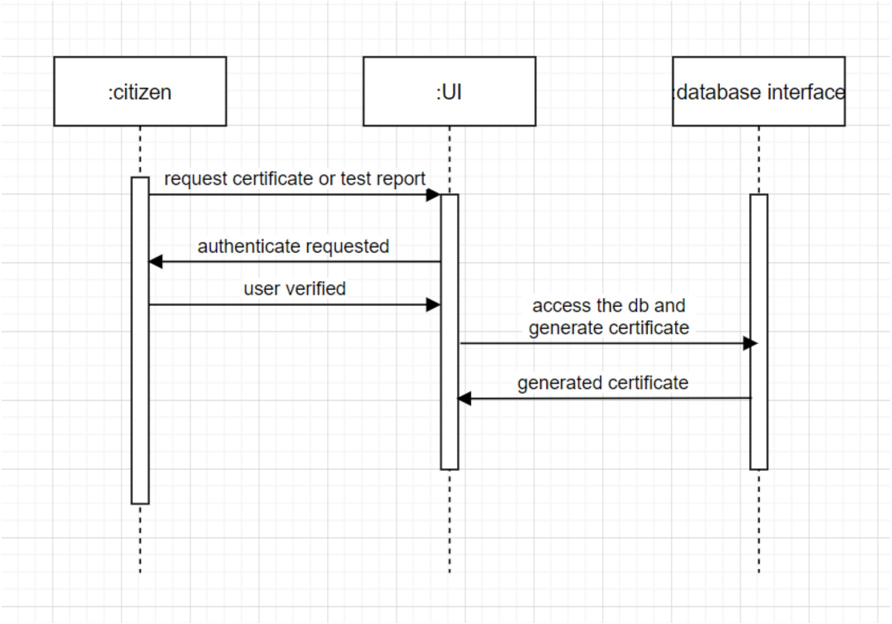


Figure 2. Element behavior of citizen subsystem - Module View

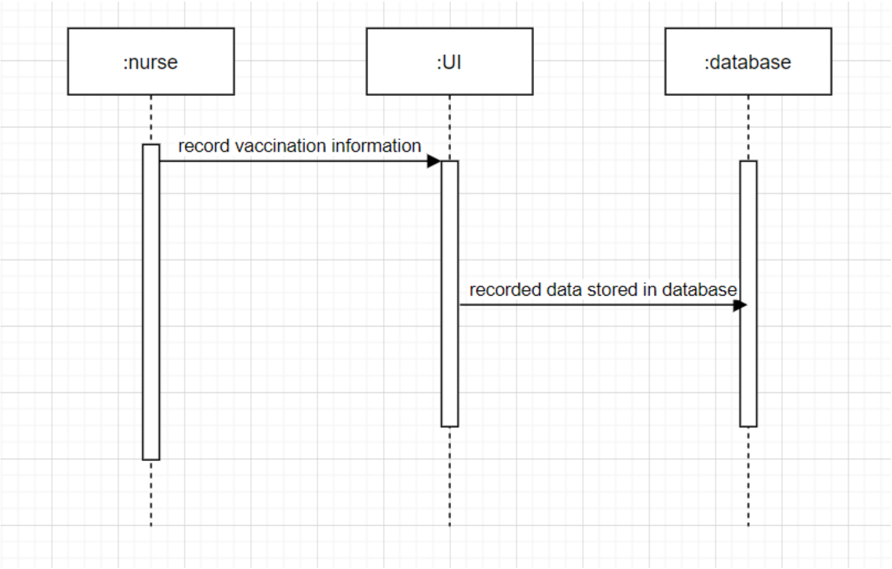

:nurse
:UI
:database
record vaccination information
recorded data stored in database

Figure 3. Element behavior of nurse subsystem - Module View

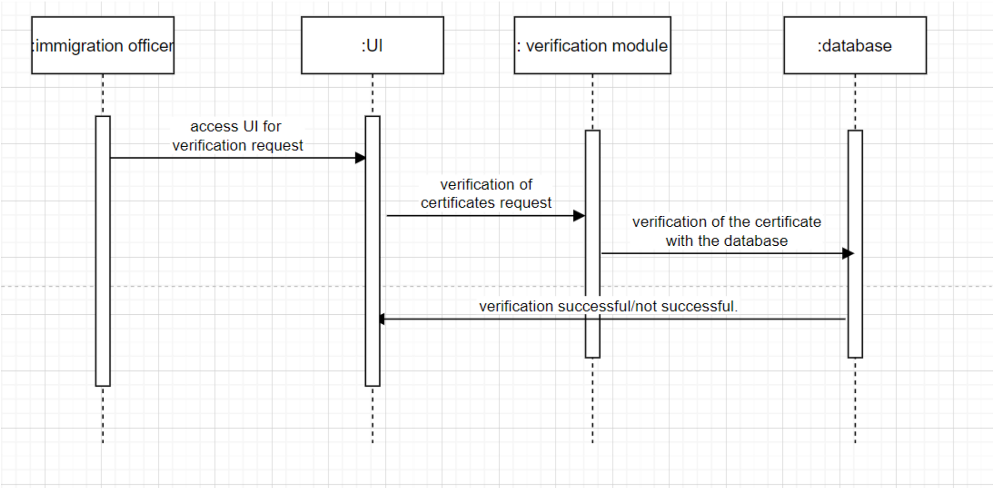

:immigration officer
:UI
: verification module
:database
access UI for verification request
verification of certificates request
verification of the certificate with the database
verification successful/not successful.

Figure 4. Element behavior of immigration officer subsystem - Module View

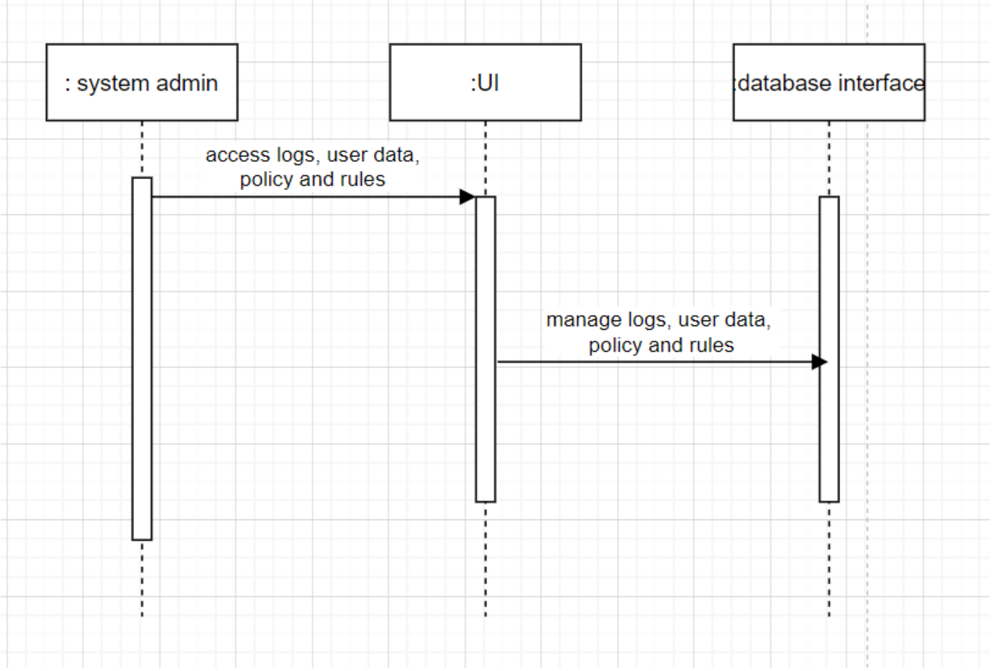

: system admin
:UI
database interface
access logs, user data, policy and rules
manage logs, user data, policy and rules

Figure 5. Element behavior of system admin subsystem - Module View

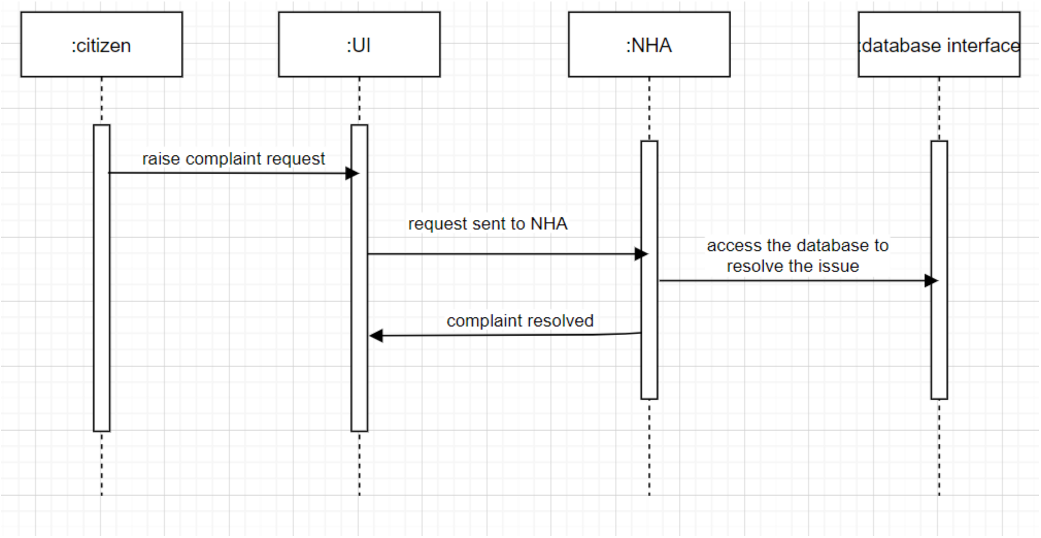


Figure 6. Element behavior of NHA subsystem - Module View

SECTION – 3 CONTEXT DIAGRAM

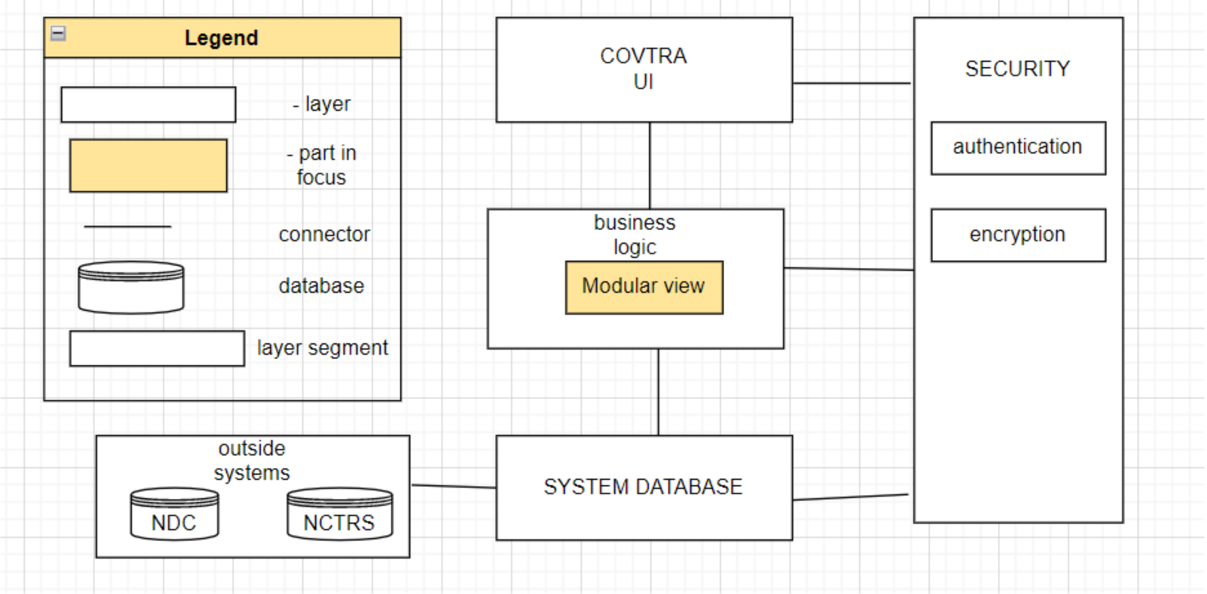


Figure 7. Context Diagram - Module View

SECTION – 4 VARIABILITY GUIDE

Variation point	Affected element or relation	variants	condition	Binding time
VP1: customer authentication information	Generation of certificate	Text data	If the entered data is incorrect or if it does not exist in the system, then the user will not get the certificate.	Run time
VP2: nurse recorded data	Recording vaccination information.	Dosage number, date, user id	If the data entered is not correct, then the nurse cannot record the vaccination information.	Runtime
VP3: NHA/admin authentication information	Authentication information	Password or user id	If the data entered is wrong, then the NHA/admin cannot carry out their respective operations.	Runtime

Table 3. Variability guide in Module View

SECTION – 5 RATIONALE

In the module view, the view is completely designed based on the primary functionality from the architectural drivers document. The decisions on the modules were taken based on actors and use cases. The user can access the covtra system using the Citizen UI module and get covid certificates or test reports after authentication, this generation of reports or certificates depend on the database interface which will communicate to the respective database module. The decision of using a data base interface is taken to introduce abstraction and improve the modifiability of the system. The nurse UI module will record the vaccination dosages in record vaccination module which will be stored in the system database which will be accessed when the user requests for a vaccination certificate. The immigration officer UI will communicate with the verification module to verify the certificates or test reports from the database to provide safe travel. NHA employees use the database interface to read and write changes to the system. Administrative employees are supposed to maintain the system and manage the users from the database interface. Different types of users have different UI modules to avoid confusion and to increase performance and usability. This way it is easy to maintain the task model and it will be easy to understand, if there is a problem in specific module the problem can be easily tracked down and resolved.

C&C VIEW DOCUMENTATION

SECTION-1 PRIMARY PRESENTATION

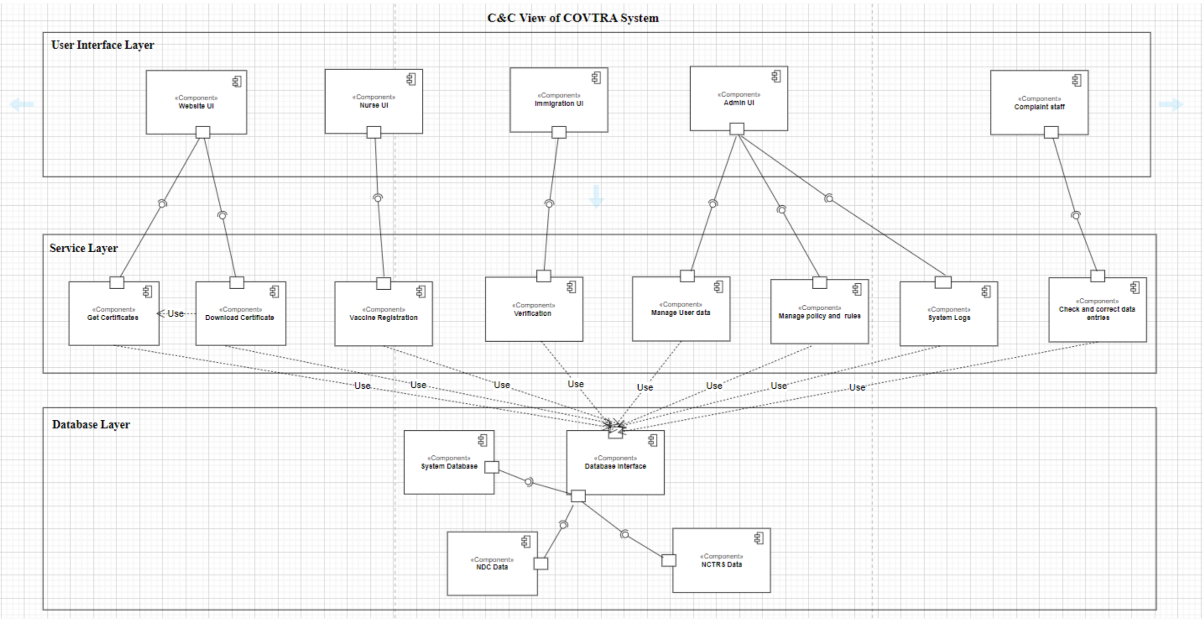
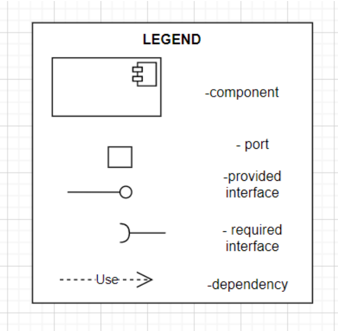

C&C View of COVTRA System
User Interface Layer
«Component» Website UI
«Component» Nurse UI
«Component» Immigration UI
«Component» Admin UI
«Component» Complaint staff
Service Layer
«Component» Get Certificates
Use
«Component» Download Certificate
«Component» Vaccine Registration
«Component» Verification
«Component» Manage User data
«Component» Manage policy and rules
«Component» System Logs
«Component» Check and correct data entries
Use
Use
Use
Use
Use
Use
Use
Use
Database Layer
«Component» System Database
«Component» Database Interface
«Component» NDC Data
«Component» NCTRS Data

Figure 8. Primary presentation of C&C View



SECTION – 2 ELEMENT CATALOGUE

Section 2a Elements and their properties

Elements	**Properties**
User interface layer components	*Citizen UI component* – during runtime the user communicates with the system using this UI component. *Nurse UI component* - this component will communicate to the vaccine registration component in the service layer to register the vaccination information. *Immigration UI component* – this component will communicate with the verification component in the service layer during runtime of this component to verify the vaccination certificates or test reports provided by the citizen. *Admin UI*- the admin UI component communicates with the manage user data, manage policy and rules and system logs in the service layer. *Complaint staff*- the complaint staff will check and correct the data in the service layer to solve user complaints.
Service layer components	*Get certificates*- during runtime this component communicates with the database layer and get the required data upon the user request. *Download certificates:* this component helps the user to download the certificate or the test report from the data obtained from get certificate component. *Vaccine registration:* this component will help the nurse UI component to store data in the database. *Verification:* this component is used by the immigration UI of the user interface layer for verifying the vaccination certificate and test report from the database. *Manage user data:* the admin UI component used this component to manage the users and their access privileges using the database. *Manage policy and rules:* the admin UI will manage the policy and rules for issuance of the certificate using the database. *System logs:* the admin UI will communicate with this component to manage sys logs. *Check and correct data:* the complaint staff (NHA) will use this module to update the data, correct the data or delete the data based on the user request accessing the database.
Database layer	*Database interface:* the database interface component is the main component that will communicate will other components in this layer based on the data request during runtime.

	System database: the system database component stores the data from the nurse, updates from complaint staff component in this component during runtime. *NDC database:* it is an external data component to the system containing all the citizen data. This component will be used during runtime when the user requests certificates. *NCTRS database:* it also an external database like NDC containing all the information related to citizen covid related data this component will be used when the user request for test report.

Table 4. Elements and their properties - C&C View

Section 2b Relations and their properties

Relations	**Properties**
Website UI – Get certificates Website UI- download certificate Download certificate – get certificate	The website UI and the components in service layer are connected using require interface where the website UI is the provider interface for the citizens to access the covtra system. The download certificate component depends on the get certificate component.
Nurse UI – vaccine registration	The nurse UI component is the provided interface to the nurse and vaccination registration component is connected using required interface in the service layer to record the vaccination dosages of the citizen.
Immigration UI – verification	The immigration UI component is the provided interface to the immigration officer to verify the vaccination certificates through the verification component using required interface.
Admin UI- manage user data, manage policy and rules, system logs	The admin UI component is the provided interface to the system admin of the covtra connected to the required interface in the service layer components.
Complaint staff – check and correct data entries	The complaint staff component is the provided interface to the complaint staff and component in service layer using required interface to resolve the citizen complaints.
Service layer components – database interface	All the service layer components are dependent on the database interface to access the required data and carry out the required requests.
Database interface component – NDC data, NCTRS data, system database	The database interface component acts as the main provided interface for all the other components in the database layer with the required interfaces to fulfill all the covtra system needs.

Table 5. Relations and their properties - C&C View

Section 2c Element interfaces

User Interface layer:

Interface identity: this interface layer consists of components related to the interaction with the covtra. It consists of the following components and the element interface specifications are discussed below for all the interface elements.

- Website UI: this component is the main provided interface for the citizen to get verifiable vaccination certificate or test report using the service layer components that are connected using required interface after successful authentication.
- Nurse UI: this component helps the nurse to record the vaccination dosages by using the required interface in the service layer component.
- Immigration UI: this component provides the user interface immigration officer to verify the test reports or vaccination certificates by using the required interface connected to the service components.
- Admin UI: the admin UI component helps the system admin to perform all the admin level services to maintain the system using the required interface in the service layer components.
- Complaint staff UI: the complaint staff UI is the interface for the complaint staff to manage the complaint tickets raised by the customer.

Resources provided: the website UI component provides the citizens to generate a download the covid vaccination certificates and test reports.

Locally defined data types: stings, integers

Quality attributes: modifiability, security, usability, performance.

What the element requires: these elements use the data from the database interface to fulfill the need to different users.

Rationale and design issues: the user does not have direct access to the services so, we use the separate UI components for different users.

Database layer interface:

Interface identified: this layer consists of database interface component that is the main interface identified in this layer that connect to the respective database components using the provide/required interface connection to complete the respective tasks requested by the users of covtra system.

Quality attributes: this interface provides the modifiability, security, performance quality attributes.

What the element requires: this element interface requires the request commands and the proper authentication by the user to access the database using the database interface component.

Rationale and design issues: this layer has the database interface component to facilitate and ease the usage of database respectively according to the user's request.

Section 2c Element behavior

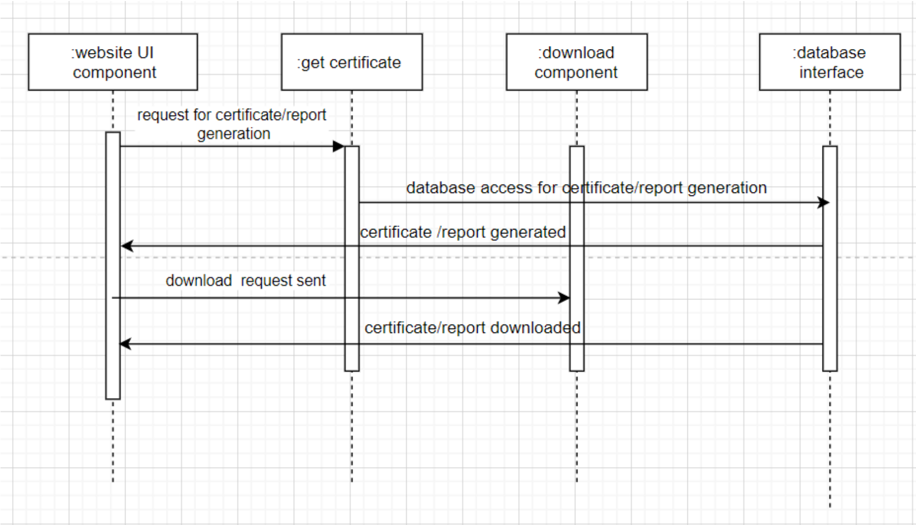

:website UI component
:get certificate
:download component
:database interface
request for certificate/report generation
database access for certificate/report generation
certificate /report generated
download request sent
certificate/report downloaded

Figure 9. Element behavior of citizen UI - C&C View

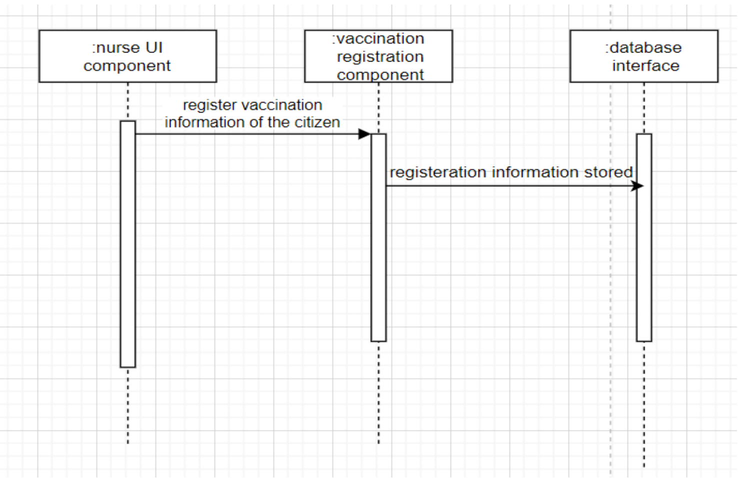

:nurse UI component
:vaccination registration component
:database interface
register vaccination information of the citizen
registeration information stored

Figure 10. Element behavior of nurse UI - C&C View

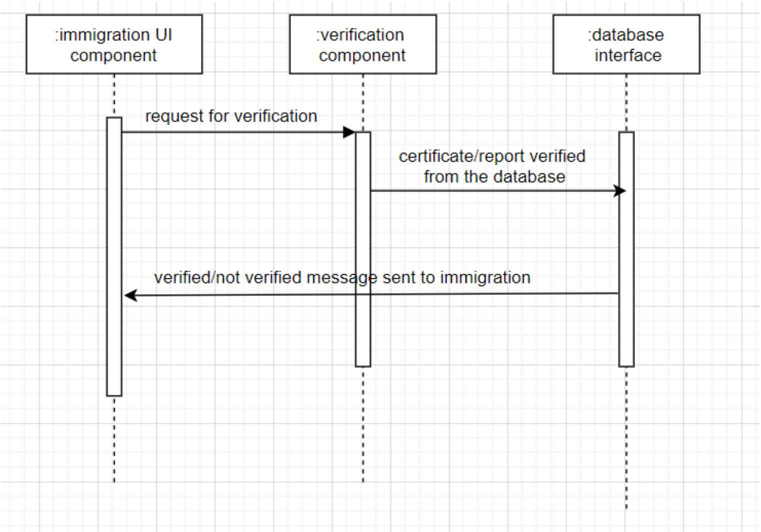

:immigration UI component
:verification component
:database interface
request for verification
certificate/report verified from the database
verified/not verified message sent to immigration

Figure 11. Element behavior of immigration officer UI - C&C View

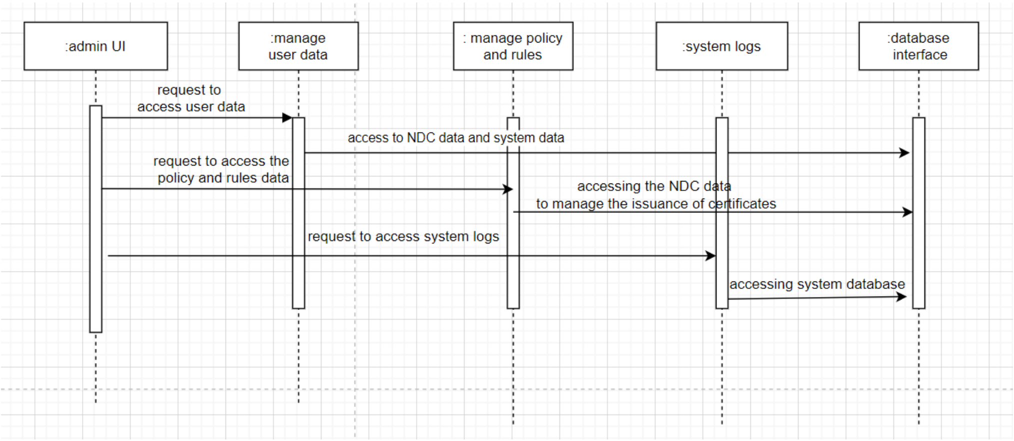

:admin UI
:manage user data
: manage policy and rules
:system logs
:database interface
request to access user data
access to NDC data and system data
request to access the policy and rules data
accessing the NDC data to manage the issuance of certificates
request to access system logs
accessing system database

Figure 12. Element behavior of system admin UI - C&C View

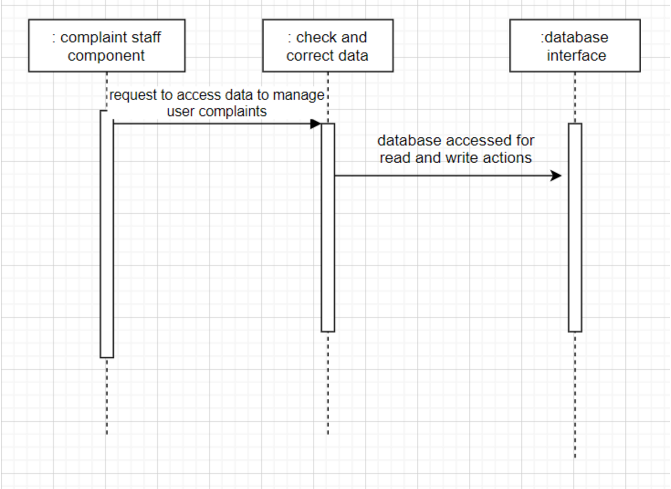

: complaint staff component
: check and correct data
:database interface
request to access data to manage user complaints
database accessed for read and write actions

Figure 13. Element behavior of NHA UI - C&C View

SECTION – 3 CONTEXT DIAGRAM

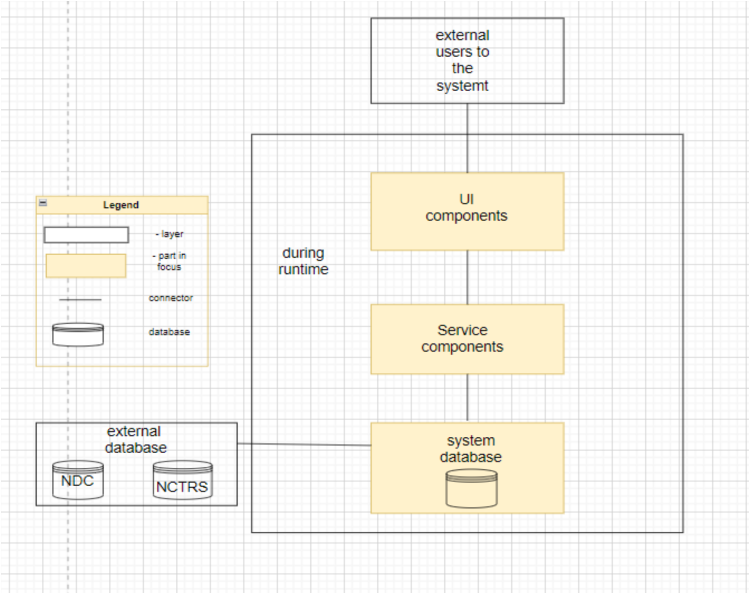


Figure 14. Context diagram of C&C View

SECTION – 4 VARIABILITY GUIDE

Variation point	Affected element or relation	variants	condition	Binding time
VP1: data base	External database system NCTRS	Change in data storage format from JSON to XML.	change in database when we decommission NCTRS.	downtime

Table 6. Variability guide of C&C View

SECTION – 5 RATIONALE

The component and connector view mainly discusses about the communication between the components during runtime. There are three layers in the C&C view namely, user interface layer, service layer, and database layer. As we have used the micro service architectural pattern, we have used different components for different services carried out by different actors. As previously discussed in the module view, we have different UI components for different actors so that external users who may cause harm to the system will not have access to the complete database this way security is provided to the system.

The C&C view documentation is designed by keeping in mind of quality attributes, in following ways. The availability of the system is ensured in such way that if there is any issue or error occurred in the system by this C&C view, we can easily track down the problems in the system this address the availability quality attribute. The resources of the covtra system are placed in the service layer where most of the functionalities of the system are implemented and interconnected to the other components in the user interface layer and database layer to increase performance as the management of resources will be at ease. The security of the system in not compromised as it is divided into several layers and the communication of the functionalities is only possible using valid user authentication methods. The modifiability of the system is made easy as the components are placed in their respective layers to avoid confusion to the developer or the stake holder who intend to interact with the system. This component and connector view is designed to reduce coupling and decrease the binding time of the modification. The implementation of coupling among the components is carefully designed as the NCTRS DB will decommission and if that happens the system will be able to carry out the similar functionalities as usual and uses the NHA-HMIS DB instead of NCTRS to store the data related to vaccination certificates and test reports data in future. In the designed C&C view it is easy to maintain the task model and have better usability of the system.

Appendix

Software Architectures and Quality

Assignment 1

Group 8 members:

Sree Kavya Ganja – srga20@student.bth.se – 20000217-T122

Prudhvi Nath Naidu Cherukuri – prch20@student.bth.se – 19981212-T292

Jeevan Sai Koushik Masavarapu – jems20@student.bth.se – 20000204-T077

Veera Venkata Subrahmanaya Varma Munduri – vemu20@student.bth.se - 19990310-T195

Anjum Khalil - anki21@student.bth.se

Contents

List of Figures

1. Introduction:

Attribute-driven design (ADD) is the practice of designing or creating software architecture for a system from a set of quality attributes. In this context, quality attributes are defined as the properties of a system that can objectively indicate how well the needs of the stakeholders are met.

As the nation health authority has been instructed to develop a new system for citizens which gives verifiable covid vaccine certificates for international travel so, COVTRA is the new proposed system. Attribute-driven design will be used in architecture of the COVTRA system. COVTRA system has three kinds of users – citizens, vaccinators, and immigration officers. Citizens can request certificates, vaccinators can update dosages, and immigration officers can validate the certificates for travel.

It shall be accessible throughout the day with less than 5 minutes of downtime. The system collaborates with two existing software systems the National Digital System Register (NDC) which contains the personal details of citizens and the National COVID19 Test Record System (NCTRS) which contains the Covid test results of the citizens to issue new certificates and to verify the validity of certificates.

In this main purpose of this report is to identify the architectural drivers. Architectural drivers distinctly explain what the purpose of the system is and why should we design architecture for a particular system. In the following sections all the drivers design purpose, quality attributes, primary functionality, architectural concerns, and constraints will be discussed clearly. These drivers play an important role in the architecture they drive and shape the architecture.

2. Assumptions

- The National Digital Citizen Register (NDC) system is available 24 hours a day.
- The National COVID19 Test Record System (NCTRS) is available 24 hours a day.
- The signing of vaccination certificates and test reports by NDC takes less than 10 seconds.
- Generating certificates means automatically we are getting a downloadable file
- If we are generating certificates, then we are getting a signed certificate from NDC.
- QR code for verification is assumed to be available in the vaccination certificate PDF.

3. Design purpose:

The purpose of this design is to develop a new system that manages the vaccination and test reports related to Covid to facilitate safe travel for the citizen. The system imports data from two different systems, one of which is projected to change in a few months. The COVTRA system should be ready to work with the modified system, as well.

The main business goals are:

- Providing verifiable covid vaccination certification to ensure safe travel of the citizens.
- Recording the covid vaccine dosages by the administration.
- Modification of any functionality within the system should be made available for future purposes.

The system is expected to run in a dynamic environment where modifications to modules are very common. The system is, therefore, expected to accommodate changes quickly. The system can be considered a case of Greenfield system in a mature domain because the system is being developed from the scratch but based on some ideologies of similar website like Covid Bevis in Sweden to have an idea of what exactly the system does and design accordingly to our system[1].

4. Primary Functionality

Use case diagram:

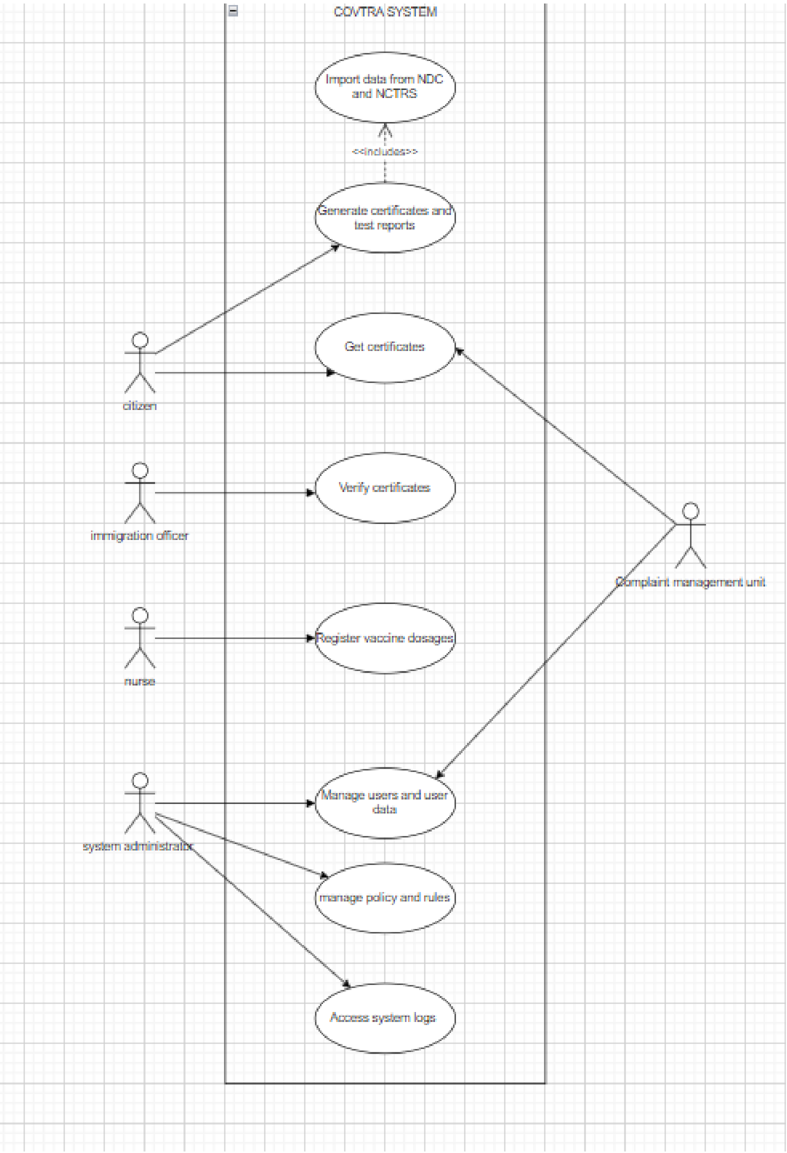


Figure 15. Use case diagram for Primary Functionality

Use case description:

1. **UC-1 Import data from NDC and NCTRS:**
   To generate the vaccination certificates or test reports the system uses the data from the already existing systems NDC and NCTRS. From NDC system gets the citizens data and NCTRS has all the covid test reports. It is important for the system to import this data because based on this information the system will generate certificates or test reports which are signed by the NDC, the complaint management system also has

access to this user data because they are responsible for the users' complaints and update the data with correct information**.**

2. **UC-2 Generate certificates and test reports:**
   The citizen would want to get a covid test report or if the citizen is vaccinated the citizen would request the system to get a vaccination certificate for travel. This use case main functionality is to provide vaccination certification certificates and test reports to the citizens. The citizen can access this system to generate a valid vaccination certificate or test report for travel.

3. **UC-3 Get Certificates:**
   The user can download the generated certificates. These certificates also have a QR code for verification. These certificates are also accessed by the complaint management system for managing the user complaints and correct the wrong information. The main reason to include this use case as primary functionality is that the citizen can view the generated vaccinated certificates or test reports by accessing this functionality in the system.

4. **UC-4 Verify Certificates:**
   The immigration officer is responsible for verifying the vaccine certificate/ test report the citizen has submitted for travel. The immigration officer has access to the data related to the user via NDC. The main reason for using this as a primary functionality is that it helps the citizen to get required permission to travel in or out of the country using the vaccination certificate or the test report that has been verified by the immigration officer.

5. **UC-5 Register vaccine dosages:**
   The nurse/vaccinator is responsible for registering the citizen when he/she gets vaccinated like date, time, dosage, type of vaccine., etc. they have the ability to modify the data related to vaccine dosages given to the citizen. The validity of the vaccine certificates is only limited to two weeks after the date of final dose. This chosen as one of the main functionalities of the system as this helps in storing the data related to vaccine dosages given to the citizen which helps in generation of certificates.

6. **UC-6 Manage users and user data:**
   The system administrator is responsible for managing the users and user data who uses the functionalities within the system. The complaint management staff also has access to this data as he/she is responsible for solving the complaints/tickets received from the user. This is an important functionality because the administrator and complaint management staff have to access this data to maintain the system without any errors.

7. **UC-7 Manage policy and rules:**
   The system administrator is responsible for managing the policies and rules for issuance of vaccine certificates and reports. To issue a vaccination certificate these rules and policies are important to validate the certificate.

8. **UC-8 Access System logs:**
   The system administrator is responsible for access the system logs because the administrator manages the system errors, faults, crashes, and failures. As this functionality helps to identify systems errors and rectify them which is important for the proper functioning of the system.

The main business goal behind this system is to accommodate changes to the working environment, it should be ready to work with the modified system in the future such as when NCTRS system is combined with the NHA-HMIS system where it contains covid related testing alongside of the existing standard medical diagnosis. Through this system the business goal will be realized by changing the received XML format to JSON format in the future after the decommission of the NCTRS system. The mail goal of the system is issuing verifiable covid vaccination certificate or test reports to the citizens.

5. Quality Attributes:

Quality is the ability of the system to work as per the requirement. Quality attributes are the desired attributes of the software. In this system the quality attributes are discussed below.

a. Availability
b. Performance
c. Security
d. Modifiability
e. Usability

a. Availability:

When there is a need for a particular functionality then the system should be able to carry out the functionality when it needs to be [2]. This system should be able to provide covid test reports after gathering the data from NDC and NCTRS within reasonable amount of time, the download time for the certificate be less than 15 seconds based on the existing data. The system should be available all the time with minimum downtime of 5 minutes per day. The use cases 3 and 2 are related to this attribute. Based on these non-functional requirements availability is chosen the most relevant quality attribute of the system.

General Scenario: [2]

Source: internal system or external to the system
Stimulus: Incorrect timing, crash
Environment: normal operation
Artifact: process
Response: notify the users and be temporarily unavailable or mask the fault.
Response measure: downtime, availability percentage

Concrete Scenario:

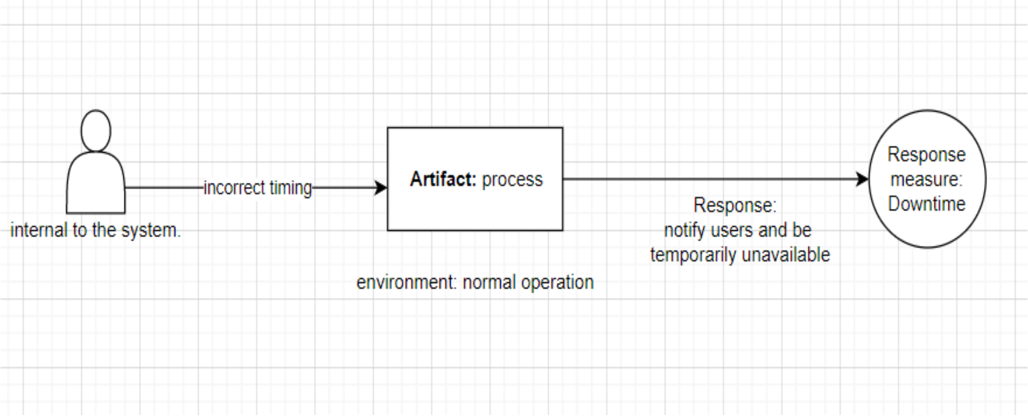


Figure 16. Concrete scenario diagram of availability attribute

b. **Performance:**

When an event takes place like requests, messages., etc. the system should be able to use the respond to that event in time [3]. In this system, COVTRA should be able to handle many different requests from multiple users and generate correct vaccine certificates and tests to the users. For example, the system should be able to issue 20000-50000 users their certificates and test reports.

General Scenario[3]:

Source: internal system or external to the system
Stimulus: periodic event, an event which occurs in patterns, an event which occurs in some probabilistic distribution.
Environment: normal operation, overload,
Artifact: system and its components
Response: process events which causes the system to change its events.
Response measure: latency, throughput.

Concrete Scenario:

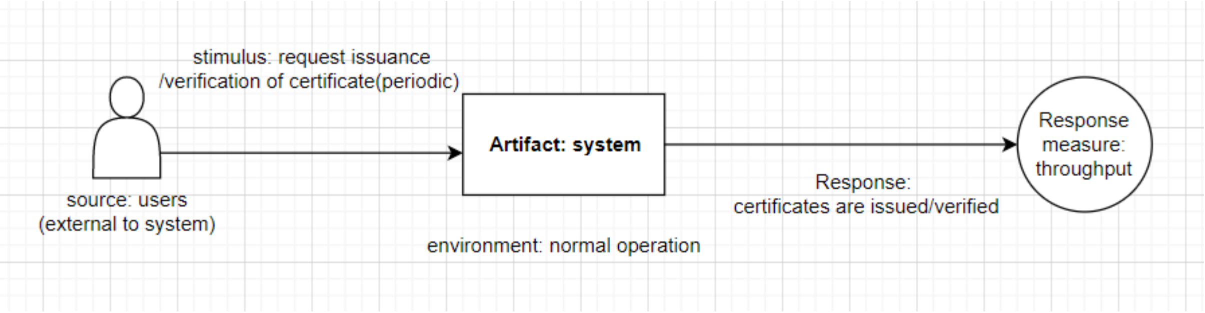

stimulus: request issuance
/verification of certificate(periodic)
Artifact: system
Response measure: throughput
source: users
(external to system)
Response:
certificates are issued/verified
environment: normal operation

Figure 17. Concrete scenario diagram of performance attribute

c. Security:

The ability of the system to protect the data from unauthorized systems[4]. In COVTRA as mentioned in the description, the system administrators and the complaint management staff should be able to login into the system using Active directory server.

General Scenario[4]:

Source: A human, another system, or a hacker.
Stimulus: unauthorized attempt to access the data or change the data.
Environment: normal operations, partially operational, not operational.
Artifact: data within the system, data consumed by the system
Response: protection of data and services.
Response measure: how long does it take to recover from an attack, how much data is vulnerable.

Concrete Scenario:

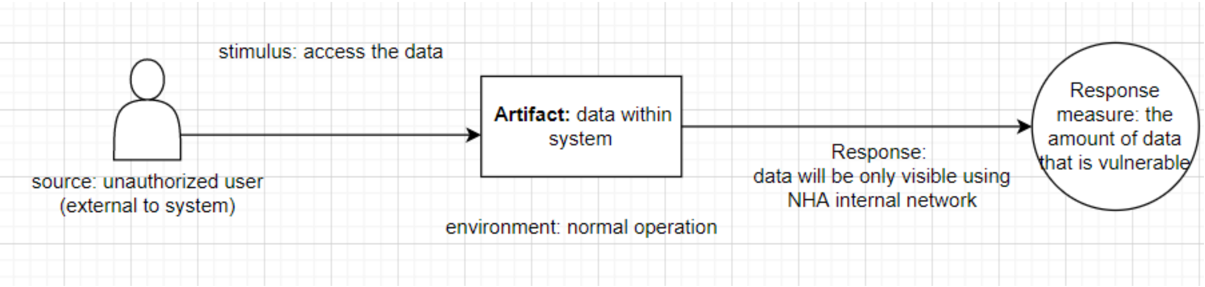

stimulus: access the data
Artifact: data within system
Response measure: the amount of data that is vulnerable
source: unauthorized user
(external to system)
Response:
data will be only visible using
NHA internal network
environment: normal operation

Figure 18. Concrete scenario diagram of security attribute

d. Modifiability:

when we want to add new functionalities, changing the input systems in such changes should be modifiable in the system[5]. In this system, the NCTRS will decommission and merge into the standard medical system NHA-HMIS this change must be accommodated in the system by modifying a few things. If there is change in the policy rules, data modification, such modifications will be made in the system.

General Scenario[5]:

Source: developer, system admin, end user.
Stimulus: addition of a function, deleting a function, modify any function.
Environment: design time, run time, compile time.
Artifact: platform, user interface, environment.
Response: make change, test, and deploy.
Response measure: time and money

Concrete Scenario:

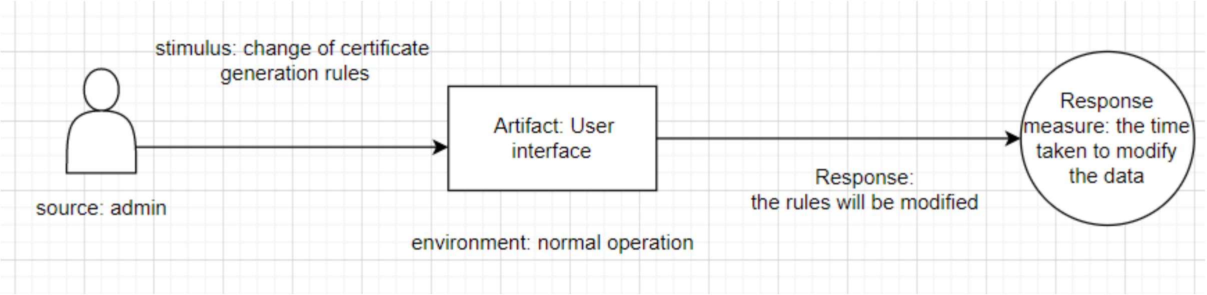

stimulus: change of certificate generation rules
source: admin
Artifact: User interface
environment: normal operation
Response: the rules will be modified
Response measure: the time taken to modify the data

Figure 19. Concrete scenario diagram of maintainability attribute

e. Usability:

Usability quality attribute explains about the overall user experience of any system, this quality attribute is considered as an important attribute in our system as the system should be easy to use, and the language used in the system should be in English and the user should be provided with the option to change the language of the system to any EU languages.

General Scenario[6]:

Source: end user.
Stimulus: end user trying to understand and use the system
Environment: run time, normal operations.
Artifact: system.
Response: provide the features that the user can understand quickly.
Response measure: task time, number of errors, tasks accomplished.

Concrete Scenario:

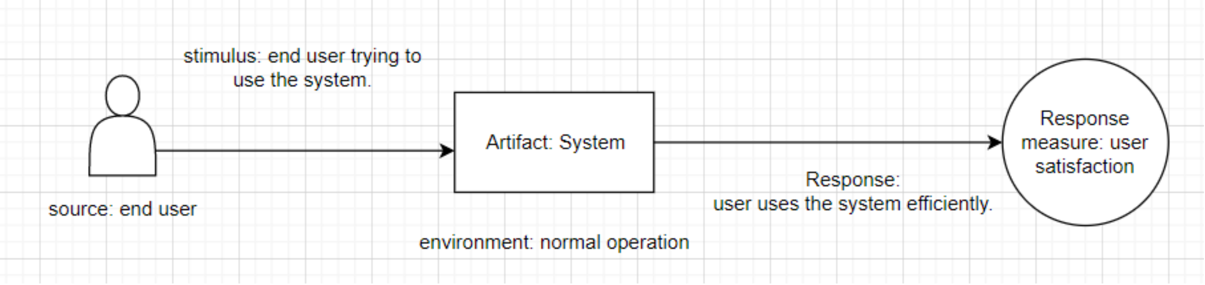

stimulus: end user trying to use the system.
source: end user
Artifact: System
environment: normal operation
Response: user uses the system efficiently.
Response measure: user satisfaction

Figure 20. Concrete scenario diagram of usability attribute

Utility Tree for Quality Attributes:

This is the representation of quality attributes based on their priority level by considering the bussiness goals of the system.

The priorites can be marked as Low(L), Medium(M), High(H) by the customers and the architects of the system.

In our system the quality attributes are prioritized according to the importance of the attribute towards the success of the system and the technical risk involved in the implementation of the functionality in the system.

The quality attributes selected in the sytems are availability, performance, usability, maintainability, and security attributes are selected and prioritized using the uitility tree diageram.

The below figure show the detailed level of prioritization of quality attributes.

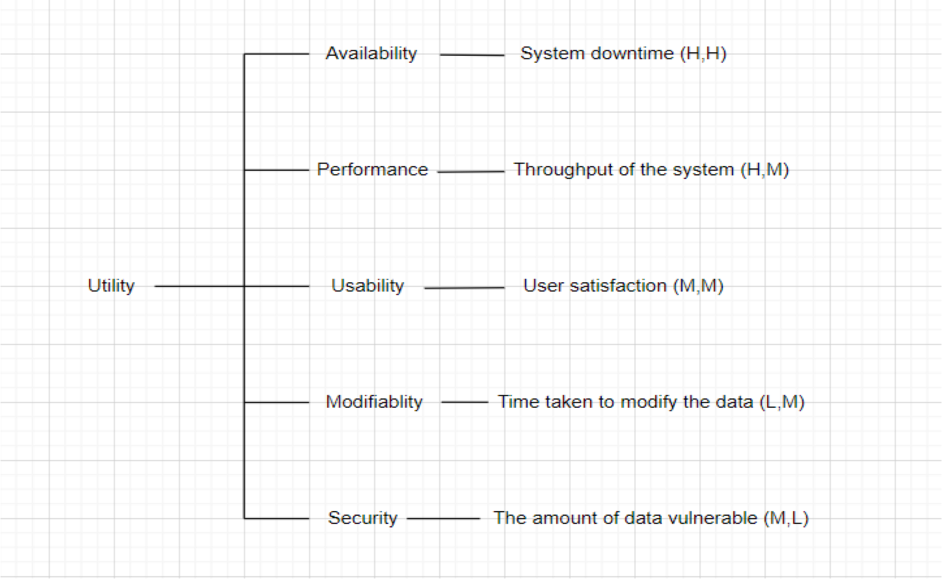


Figure 21 Utility tree of quality attributes

6. Architectural Concerns:

General concerns: the context diagram gives the information regarding the general concerns with the system.

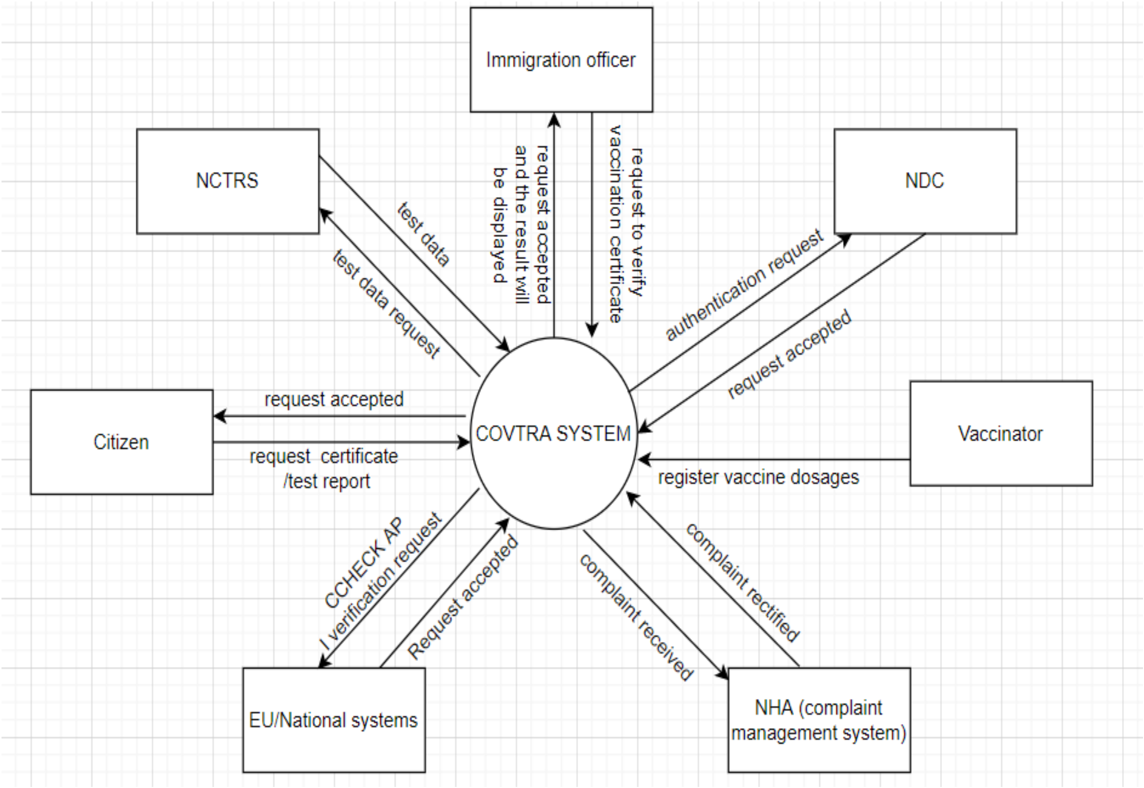


Figure 22. Context diagram of general concerns

Specific Concerns:

- Establishing an initial overall structure as this is a greenfield system.
- Leverage the team’s knowledge of .Net technologies.
- The single sign on authentication for NHA employees could complicate system.
- There may be some discrepancies in NDC and NCTRS.

7. Architectural Constraints:

- The system should be able to handle 20000-50000 requests per day.
- The system must be accessible from a web browser (Chrome, Firefox, Edge) through internet on different platforms: Windows, Linux, and MacOS.
- Connections to the NDC and NCTRS systems must be fast and reliable.

- System failures and errors should be logged. And the system should be able to recover from said failures or errors.
- NHA employees' login is done using a single sign-on (SSO) service which will be handled by integration with the Active Directory server.
- NHA employees' log, internal reports and data should only be available within NHA's internal network.
- Public servant login (Immigration Officers and Nurse) is done by NDC's public servant digital ID.
- Signing of certificates and test reports is done by NDC only after successful verification.
- The system should have easily adaptable and replaceable modules. [7]

The system will not be able to perform in the expected manner without following the above listed constrains, so the if the system is capable of executing the above constrains then it is considered as fully functional system.

8. References

[1] *2. Architectural Design*. Accessed: Nov. 21, 2021. [Online]. Available: https://learning.oreilly.com/library/view/designing-software-architectures/9780134390857/ch02.html

[2] *5. Availability*. Accessed: Nov. 21, 2021. [Online]. Available: https://learning.oreilly.com/library/view/software-architecture-in/9780132942799/ch05.html

[3] *8. Performance*. Accessed: Nov. 21, 2021. [Online]. Available: https://learning.oreilly.com/library/view/software-architecture-in/9780132942799/ch08.html

[4] *9. Security*. Accessed: Nov. 21, 2021. [Online]. Available: https://learning.oreilly.com/library/view/software-architecture-in/9780132942799/ch09.html

[5] *7. Modifiability*. Accessed: Nov. 21, 2021. [Online]. Available: https://learning.oreilly.com/library/view/software-architecture-in/9780132942799/ch07.html

[6] *11. Usability*. Accessed: Nov. 21, 2021. [Online]. Available: https://learning.oreilly.com/library/view/software-architecture-in/9780132942799/ch11.html

[7] *4. Case Study: FCAPS System*. Accessed: Nov. 21, 2021. [Online]. Available: https://learning.oreilly.com/library/view/designing-software-architectures/9780134390857/ch04.html