ISAFE: TRACKING COVID RISK

Assignment 2: Advanced Requirements Modeling and Specification

OCTOBER 28, 2021

PRAMODA GUNARATHNE
CURTIN ID: 20531218

Table of Contents

LIST OF FIRGURES

Report Process

The application is used to demonstrate and slow down the spread of the widely infected pandemic as each day the vaccine and the population of the world ignoring and underestimating the virus grows rapidly. The main objective of the problem is to develop an active and withstanding application to showcase the importance of social distancing and provide users with efficient updates by maintaining a risk profile of their health as symptoms are being recorded each day which will be used in the analysis process when healthcare authorities verify the depth of spreading and document the evaluation of the application as well as the current situation of the country.

The main approach to end up with a quality and diverse application is having a complete Software requirement specification(SRS) document where the developers, project managers, and stakeholders can always refer to during the software development life cycle. The SRS serves as a basis for later enhancement of the finished product. As well as maintaining a valid and verified SRS document, the elicitation process is also crucial in collecting information and thoroughly understanding the concept behind the developing software from stakeholders to make the development process more reliable and efficient. Both concepts will be elaborated on in further sections in this document, where the main concept of elicitation techniques used for this project will be discussed here as it is the first raw approach in starting the development. This project uses a combination of methodologies to approach elicitation as no single requirements elicitation method can provide all the software needs (Chowdhury et al. 2020), where the process continues until a successful development of the application is achieved.

The first technique used is ***Interviews***, which is the basic method carried out in the early stages of the process. The first meeting was held with the stakeholders, end-users, and project manager to get the basic evaluation of the system. The use of open-ended interviews yielded the greatest outcomes. The next interviews were carried out with the healthcare authorities, vaccine department authorities, National Governors Association, ICT managers of the technology companies. Open-end interviews are possible to conclude as more akin to dialogue analysis than genuine interviewing. The most efficient strategy was to establish a list of subjects intended to cover during the interview without putting any constraints on the process, allowing for free (but organized) conversation among the participants, with the sequence of interaction determined in real-time. The most difficult aspect of this strategy is staying on point during the interview and not letting the "conservation" veer off into undesired detours that will lead you nowhere.

In this project domain, reading ***existing documents*** such as job descriptions, task descriptions, and quality assurance guides, among other types of documents, might be quite useful. Examining World Health Organization (WHO) documents, literature reviewing international bodies which are used as one of the strategies for eliciting requirements in this case, and Effective Vaccine Management Criteria and reading interacting system reports of past software was effective to evaluate an idea of the system the clients expected. Most of the documents were not available to read due to integrity reasons which could have contained prominent details needed for the system.

JAD (Joint Application Development) was used to evaluate the were mostly used to resolve issues rather than to collect health care domain expertise. Issues such as the amount of investment in the system, how the system can easily provide a user's health analysis through a physician, and the different varying aspects the system must provide to accomplish the objective of the software. Later in the development stages, JAD sessions were used to get good results in dealing with conflicting requirements. Since the stakeholders have only a general image of the solution, so it was difficult to know the actual requirements of the application.

Focus Workshops and Groups of 6 - 10 current stakeholders to evaluate the reasonable aspects of the system acquired that is if the system requires all users to be logged in during contact tracing or can the user not be on the app and still be notified of close contact alerts. Brainstorming sessions can be used to acquire in the process of how to achieve this requirement. The goal is to present a variety of options or ideas that will aid in determining what is

blocking the group from moving forward and enable creative thinking approaches to the problem at hand. Standardized ***Surveys/Questionnaires*** were distributed during the first meetings of each week the requirement elicitation process continued. End-users and other stakeholders will be asked questions in the app allowing for scientific response analysis and evaluation. It enabled the gathering of massive amounts of data and replies from a wide range of stakeholders, where the results were further discussed in more workshops, benchmarking, and individual interviews with the ministry of health (MOH) and clients.

Benchmarking was used to analysis and compare the strength, weaknesses of the iSafe application against the other existing applications. Processes, operations, and strategies are the subject of benchmarking. Identifying the area to be studied and evaluating the development from there is critical, identifying the leaders and the age groups that are more likely to use the app, evaluating the survey results accordingly, and documenting what's the best choice of implementing a particular functional requirement such as risk profile creation, finally developing a project proposal to put the best practices into action. However, benchmarking can be quite vague and bombard us with too many requirements, hence the process should be carried out with awareness to put a stop after the crucial objective of the technique is gathered. ***Requirement workshops*** with the healthcare authorities, nurses in charge, security clerks, and IT managers can help the team members in identifying the actors, rules that govern the tracking covid application, scope, the prioritizes of the application, and swiftly come to an agreement on requirements. Because of its high productivity rate, this technique is being used to ensure active communication, trust, and clear understanding among all workshop participants. A neutral moderator or team member will keep track of it, and the process will be documented and recorded. Final feedback is received to conclude the requirement stage and agree upon the software to be made. Conforming to the most acquired security maintenance is crucial and it is achieved at this stage of the requirement gathering process. ***Prototype Analysis*** is used to overcome confusion and have a broad idea of the application interface more thoroughly, making the process reliable and easy. Each prototype brought to the clients in the meeting is respectively analysed and concludes any changes or modifications it requires. Protocol analysis is attending to the problem as cognitive activities and directly verbalized for revealing interaction problems with existing systems more authenticity.

Even though the app claims to retain encrypted location data, there is a possibility of cyber hacking or government surveillance of citizens. Furthermore, no clear information exists about who has access to data held in data centres or how long the data would be retained. Rather than gathering domain expertise, workshops might be utilized to resolve issues. We can have a third party explain the crucial ways around the document reading component, and we can have a healthcare domain specialist interview the health authorities for more accurate data collecting. Rather than being used as a means of acquisition, observation can be used to corroborate some of the information acquired. Modeling requirements as goal modeling and strategic dependency and strategic rationale models are included in the SRS Appendix to showcase the system's requirements in more depth. As a lesson learned goal modeling should be drawn during the early stages and further developed to handle the problems encountered later such as when adding non-functional requirements. The organisation of the SRS document created is described as follows. The aim and scope of the project, as well as key definitions and acronyms used throughout the text, are all included in the introduction section. The product perspective, product functions, user characteristics, restrictions, assumptions and dependencies, and proportionate requirements are all included in the overall description section. The third section of the SRS list functional requirements for the product discussed in the elicitation process. The fourth section broadly describes the External Interface Requirements, and the fifth section lists other non-functional requirements of the project, where the sixth section is for any other requirements needed and the seventh section concludes the reference list used for this document. The appendix: Analysis model is where extra models of the system can be found. In this development of the SRS document, the top-down approach is used assuming that the system requirement is defined before the software requirement.

1. Introduction

1.1. Purpose

The purpose of this SRS document is to describe the software requirements for an easy-to-use mobile application that tracks infected COVID-19 risk users and keeps track of the user's risk levels by tracking the potential contamination through whereabouts and activities. This document will be used as a means of communication between the development team and the customer to ensure that the iSafe: COVID-19 application meets the business requirements. The document is also designed for the project managers, marketing staff, testers, and documentation writers of the system to understand how the system interacts with the user, the types of scenarios the system can manage, and the requirements and limitations that control its design. The document can also aid in the generation of further requirements and constraints before the implementation stage is initiated. Finally, the document will include use case scenarios, the modelling analysis process as well as prototypes of the system (wireframe diagrams) to show how the system works as intended.

1.2. Project scope

Following its first reported outbreak in December 2019, COVID-19 has already become a global pandemic affecting over 200 countries. Public health authorities have taken several steps to stop it from spreading. Identifying people who have had close contact with sick people and then isolating them (so-called contact tracing) has proven to be very effective. As a result, contact tracing has emerged as a critical tool in the fight against COVID-19, and with the implementation of this application, the contact tracing process is a lot more convenient and reliable for the health care authorities to analyse the bigger problem at hand that is stopping the spread of the virus. And because of their widespread use and mobility, smartphones and related digital technologies have the potential to give a better approach. Hence the implementation of this application will broadly mitigate the spread of the virus if half the nation's population decides to install and follow its objective in mitigating the risk. The main objective of the system is to maintain and keep track of the user's risk level by tracking potential contamination through whereabouts and activities, whilst maintaining a decent risk profile that acts as the doorway for the healthcare department when producing analysis of the spread. The goal of the project is to help the necessary authorities accomplish the world back to normal. With this motive in mind, the system can analyse the risk at a social touchpoint, at a specific location, providing travel suggestions if the risk assessment performed by the application shows a higher risk, and notify users when they encounter an infected person. As additional requirements to specify the user-friendly environment, the system also consists of a vaccine test that can be done to verify the user to request, if they are yet to be vaccinated. The system also provides updates of COVID-19. In determining the depth of the system, it is crucial to get the scope of the system maximized, and hence this application is being recommended for nationwide implementation only. And in future updates or generating of other similar systems, this system (and the SRS document) can be used as a reference.

1.3. Definitions, acronyms, and abbreviations

MOH - MOH is the ministry of health involved mostly in the vaccine supplement.

Close contact – Close contact is known when infected people with the application encounter other users that uses the application. The users will be notified of any close contacts in ***red alerts*** through notifications.

Red alerts – When an infected close contact is made, the system distributes an alert message to all the users that were within the vicinity of the infected person.

Token ID – Token is the unique code distributed or given to each user which is used to analyse close contacts, typically done by comparing the token ids of users. These unique identifiers are a way of maintaining the objective

of the system, but this can be a great security threat and hence the system automatically generates and changes the ID every 24 hours minimizing cyber-attacks.

Timestamp – The user can choose to share their date, time, and location/ event all at once is known as the timestamp.

Social interaction touchpoint – This is a public interaction point that the user may visit during their day, such as bank ATM, shopping mall, grocery shops, etc.

Covid check-in – This is an option available in the system where the users can perform a self-analysis of their current health status and update their risk profiles which can be useful later.

API(Application programming interface)- This is a software interface offering a service to other software particularly in coding.

API specification- This document describes how to use or build the interface accordingly to our software.

Private Set Interaction Cardinality(PSI-CA)- Permits two parties, the sender and receiver, to calculate the intersection's cardinality.

Bluetooth low-energy (BLE) protocol- The Bluetooth protocol that needs to be available in the mobile phone for the application to perform close contact tracing.

Transport Layer Security (TLS) – This is a cryptographic technology for providing communications security over a computer network.

1.4. Overview and Reading suggestions

The rest of the SRS document is intended to show off the system's features and guiding requirements. The SRS comprises diagrams and models that visually depict the system's implementation, as well as prototypes that show how the system navigates through example scenarios. The objective and scope of the project are described in the introduction section. The product perspective, product functions, user characteristics, restrictions, assumptions and dependencies, and functional requirements are all included in the overall description section. And the rest of the chapters are mentioned above under the report process section of the document. Exclusively reading through the document can be of advantage as the whole problem domain is captured and the solution is provided meaningfully. Reading the document from the top to bottom can provide the audience with a futuristic and broad detail of the system's purpose as well as its intention. The functional requirements under chapter 2, overall description and non-functional requirements described under chapter 3 are of extreme importance for the developers of the application and the project managers, and they are the most crucial sections of the SRS document. The following components were used to create appropriate specifications for the proposed system: use case diagrams, sequence diagrams, activity diagrams, state diagrams, class diagrams, and goal modelling. The project manager or anyone reading the whole documentation will have a depth understanding of the objective behind the implementation of the COVID-19 iSafe application. Each non-functional requirement is categorised accordingly and reading each part associated with the specific requirement will be important during the implementation of the system, hence this section must be read by the main software engineers assigned by the company.

1.5. References

- The following is the reference to a document that was used in benchmarking as a reference to existing systems already deployed in other countries.

Chowdhury, Mohammad Jabed Morshed, Md Sadek Ferdous, Kamanashis Biswas, Niaz Chowdhury, and Vallipuram Muthukkumarasamy. 2020. "COVID-19 Contact Tracing: Challenges And Future Directions". *IEEE Access* 8: 225703-225729. doi:10.1109/access.2020.3036718.

- The following link is for the API specification that can be used as a reference.

Kulsresth, Keshav, and Shivam Shasheesh. 2021. "Covid 19 Tracker Using REST API Android App". *Ieeexplore.Ieee.Org*. https://ieeexplore.ieee.org/document/9514988.

- The IEEE code of ethics

"IEEE Code Of Ethics". 2021. *Ieee.Org*. https://www.ieee.org/about/corporate/governance/p7-8.html.

2. Overall description

The system's general factors affecting the product and its requirements are broadly being discussed here. The second section will go over the system as a product and explain why it exists. The functions of the system and their high-level descriptions, as well as the limits of safety-critical abnormalities within the system, will be described. The assumptions and dependencies of the software, as well as its environment and intended user, will be covered here.

Overall, the functionality of the system is to manage risk profiles, evaluate risks associated with locations, alert during close contacts and be able to get the analysis of a location before traveling there. Performing risk assessment is a crucial aspect of this project and implementing that functionality can be challenging.

Although the primary aspect is travel monitoring and risk assessment, the system must be able to update the user of current news on COVID-19, able to request vaccines and provide eligible travel suggestions.

Correspondingly the system must be able to get an analysis of their health through the application and update their risk profiles.

The system must allow for personal information updates about the users whenever they want, overall the application must be flexible and user-friendly when considering this aspect.

2.1. Product Perspective

The system that is being implemented is a new and self-contained product. However, there are many existing documents available that have already implemented similar applications throughout 2019-2020. And in recent times the world's most powerful mobile platforms, Android and ISO have released the API(Application programming interface) specification to the public. This is the only external component that the developer reference during the development process. The reference to the documents can be found under **1.5. References** section.

- The iSafe application must control the following information:
  - Personal information of all users including healthcare authorities, data clerks, system administrators
    - Name
    - Email
    - Address
    - Password
    - Contact details
  - Contact tracing
    - Get token ids created
    - Store token ids
  - Risk assessment and travel suggestions

    - Risk level analysis
    - Calculated risks
    - Risk social touchpoint locations
- Health records
    - Store uploaded health records
    - Store analysed reports done by physicians
- Vaccine
    - Get vaccine test results
- Share location
    - Location saved

➢ Health care authorities' performances or any evaluation done must be stored in the backend server/ system such as:

    - Get secure code for location sharing
    - Store any analysis or risk assessment reports done by the allocating physicians

Please refer to the block diagram under Appendix.

2.2. Product features

The system depends on contact tracing protocol to implement efficient tracing which is an important aspect that is needed to perform risk assessment to get traveling suggestions and used for contact close alerts where alerts would not be transported if the protocol was not sufficient. The following paragraph explains how this can be acquired.

A decentralized contact tracing tool can be used as a unique protocol with excellent privacy guarantees. The system makes use of Private Set Interaction Cardinality to provide privacy (PSI-CA). PSI (Private Set Interaction) is a multi-party computation-based encryption technique (MPC) (Chowdhury et al. 2020). PSI allows two parties with two separate sets of data to compare their encrypted token IDs to compute the size of their intersection without revealing the contents of their sets to the other party. PSI-CA is a constrained form of PSI in which just the cardinality of the intersection set is computed rather than the intersection set itself (Chowdhury et al. 2020). Hence, the integrity of information is secured. At the initial state, the program receives a private key pair and is assigned to each user that installs the application. The system then generates random tokens over time specifically after the 24-hour creation of a token. When a user approaches a contact, their applications use Bluetooth to exchange tokens, where the servers then compare the tokens received and send red alerts to the user that is at risk.

When users are tested positive with COVID-19 (as certified by the healthcare authority), the user encrypts the server's generated random token, which is subsequently submitted to the healthcare provider with the user's permission. At the end of each seed, the server re-generates the tokens, representing the list of tokens distributed by each infected user over a period of time.

The system also relays in GPS and Bluetooth when sharing tokens where it uses a Bluetooth-based proximity tracking implementation. Without this implementation, the system cannot perform close contact analysis and send out alerts where needed. New mobile phones acquire Bluetooth low-energy (BLE) protocol, where this protocol uses Received Signal Strength Indicator (RSSI) that is used to estimate the proximity between users. Bluetooth devices do not store an individual's exact location; instead, they record whether there are any Bluetooth devices within radio range. It can't tell you where that interaction took place on its own. As a result, it gives greater protection against location privacy (Chowdhury et al. 2020). Also, the backend server must co-exist with the system's decentralized server. Further analysis of the requirements stage is carried out in chapter 3.

Getting the risk level associated with the specific location is critical to getting the risk assessment report/ analysis. Travel monitoring is done as the risk levels are compared and if the current location's risk level is greater than the threshold, notifications are sent to users about the risk. The users can then request the report for further analysis.

The following use case diagram and sequence diagram explain the solution above in a modeling analysis state.

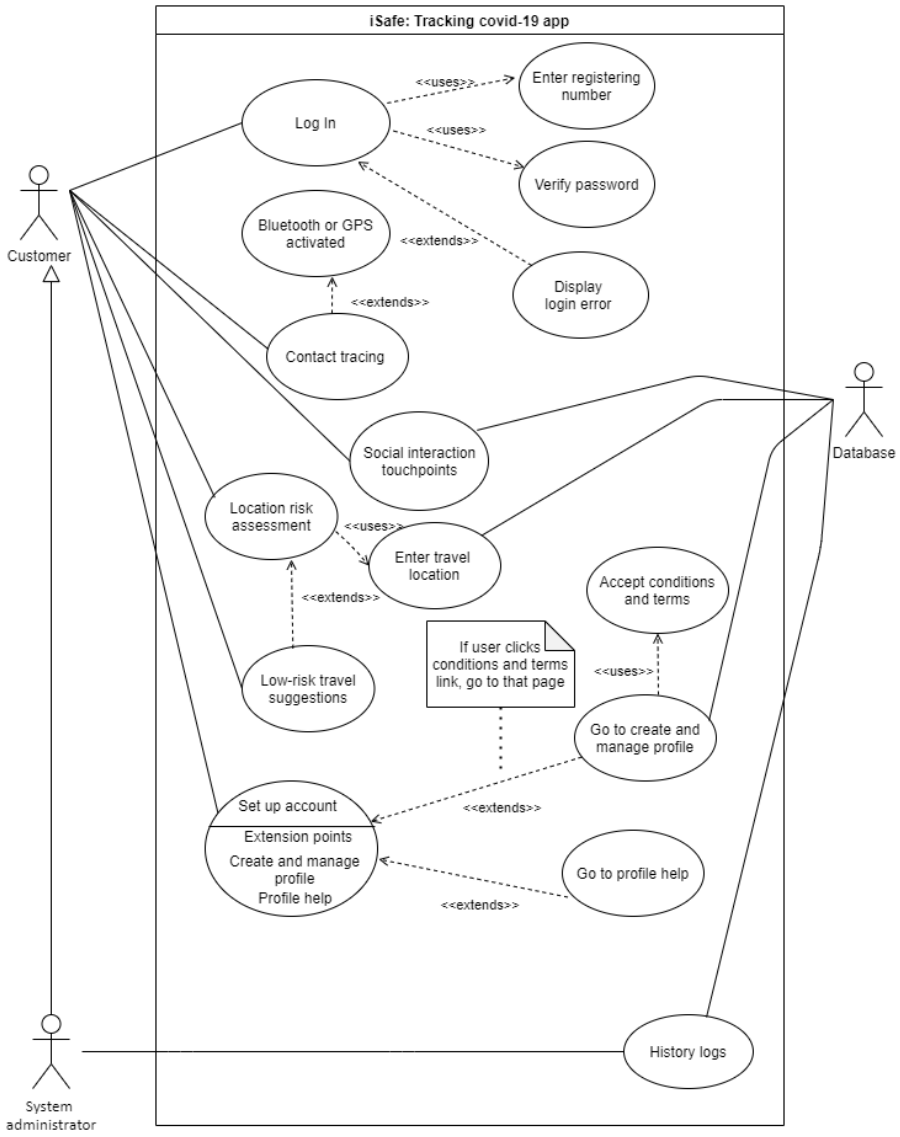

Figure 1- Use case diagram for the risk assessment and contact tracing acquired by the system

Referenced from Assignment 1: Modeling requirements

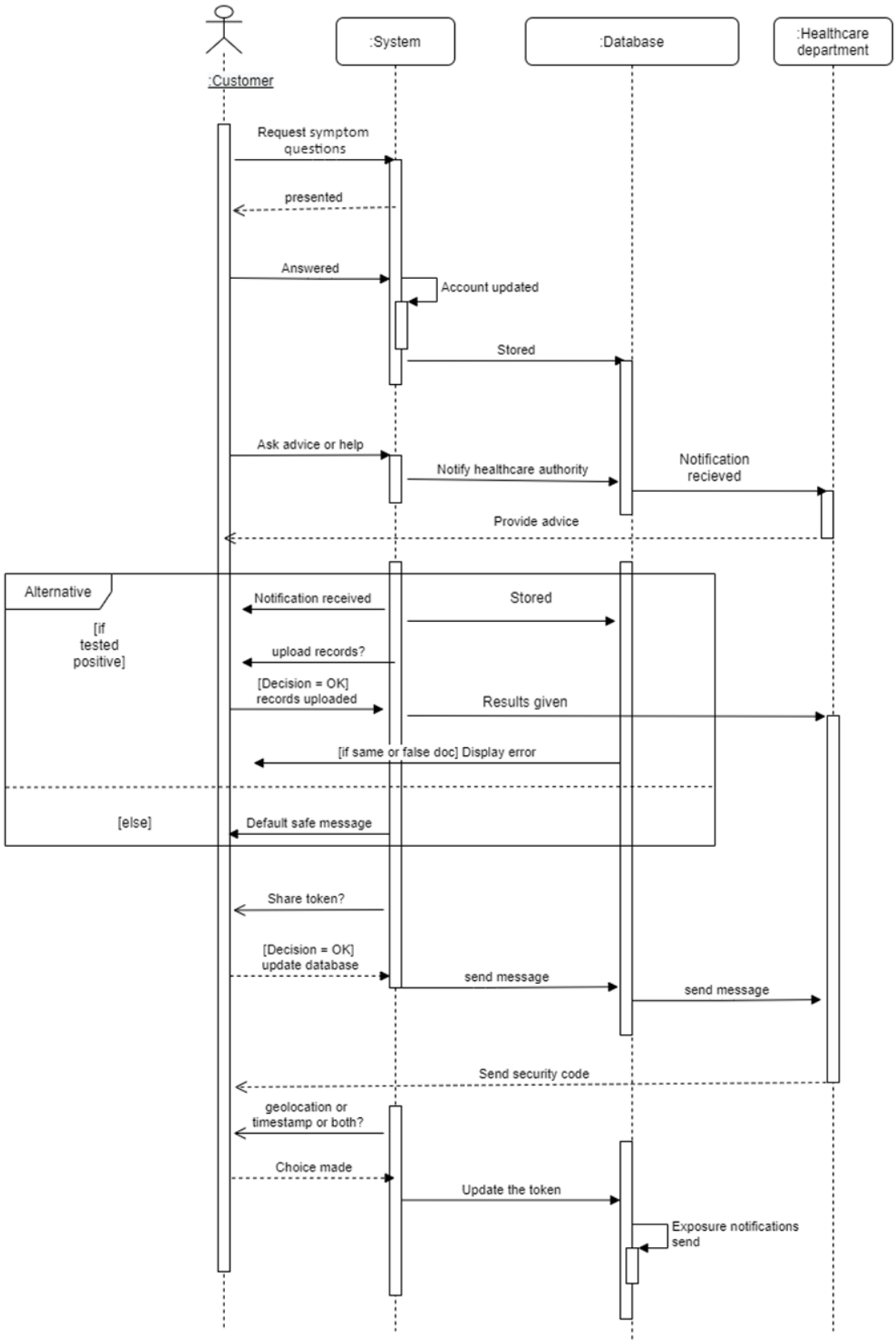
:Customer
:System
:Database
:Healthcare department
Request symptom questions
presented
Answered
Account updated
Stored
Ask advice or help
Notify healthcare authority
Notification recieved
Provide advice
Alternative
[if tested positive]
Notification received
Stored
upload records?
[Decision = OK] records uploaded
Results given
[if same or false doc] Display error
[else]
Default safe message
Share token?
[Decision = OK] update database
send message
send message
Send security code
geolocation or timestamp or both?
Choice made
Update the token
Exposure notifications send

Figure 2 – Sequence diagram for the risk assessment and contact tracing done by the system

Referenced from Assignment 1: Modelling Requirements

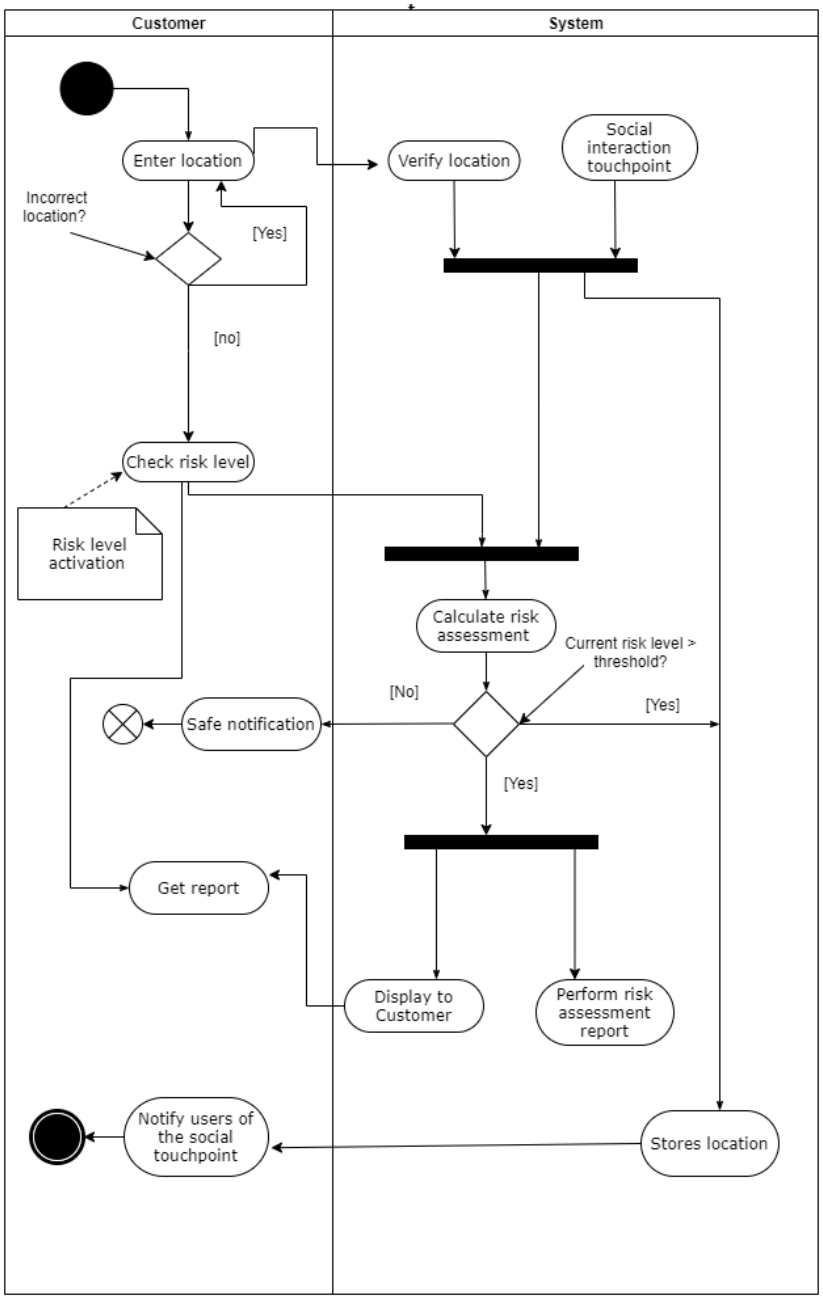

Figure 3 – Activity diagram for risk assessment and perform travel suggestions accordingly

Referenced from Assignment 1: Modelling Requirements

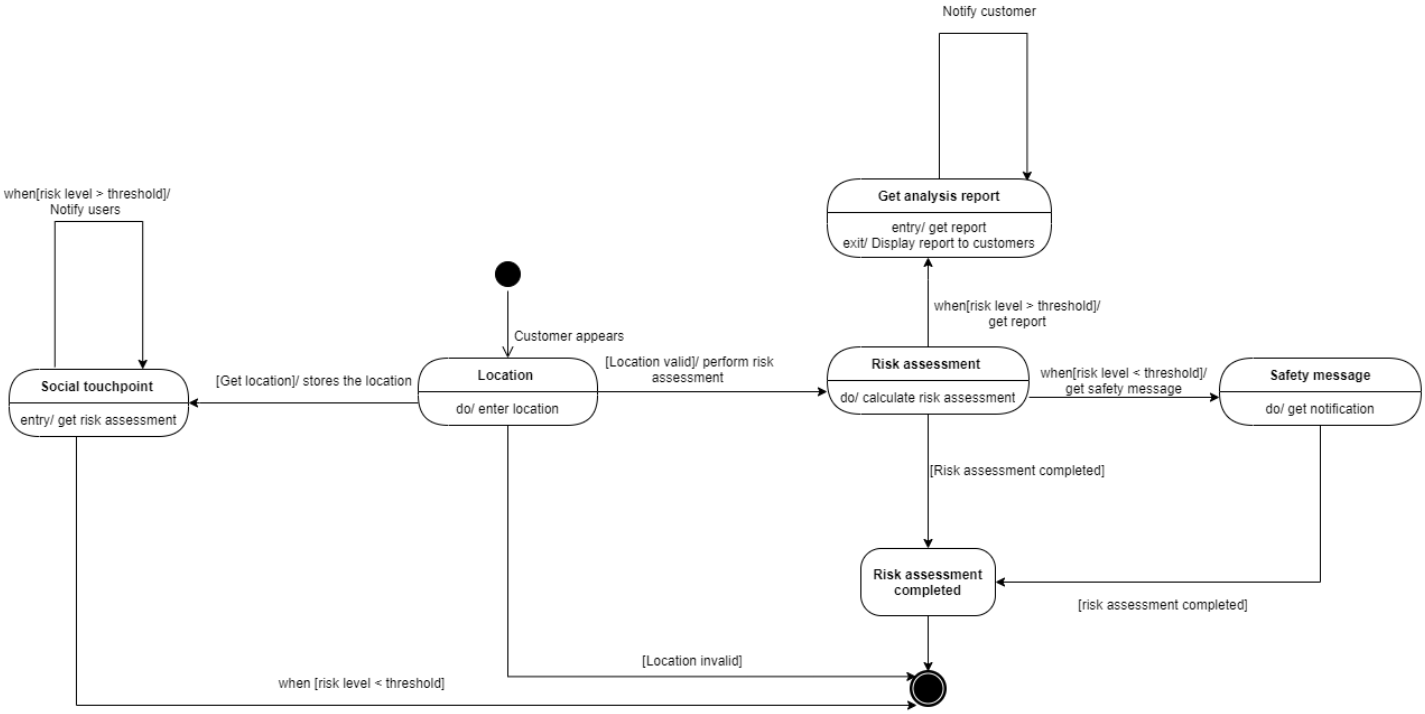

Figure 4- State diagram that utilizes the risk assessment process

Referenced from Assignment 1: Modeling Requirements

Another functionality is the requesting of the vaccine from the system where the system needs to acquire it from the health care authorities. But in general, the users must be able to request vaccines if they pass the non-vaccinated test produced by the system. The vaccine can be an alternative/ additional feature implemented into this system that exceptionally increases user-friendliness. The vaccine department and cargo handling supplier centre are both very impactful departments where the vaccine department contains all the relevant information regarding the vaccines, producing a vaccine order sheet that is to be given to the health care authority. The cargo handling centre could be associated with the vaccine department or a separate entity. Being under the vaccine department can be advantageous as this is likely to be easy during the delivery stage as outside entities require more time and manageable difficulties.

The following use case diagram and sequence diagram are used to get a more depth understanding of the solution.

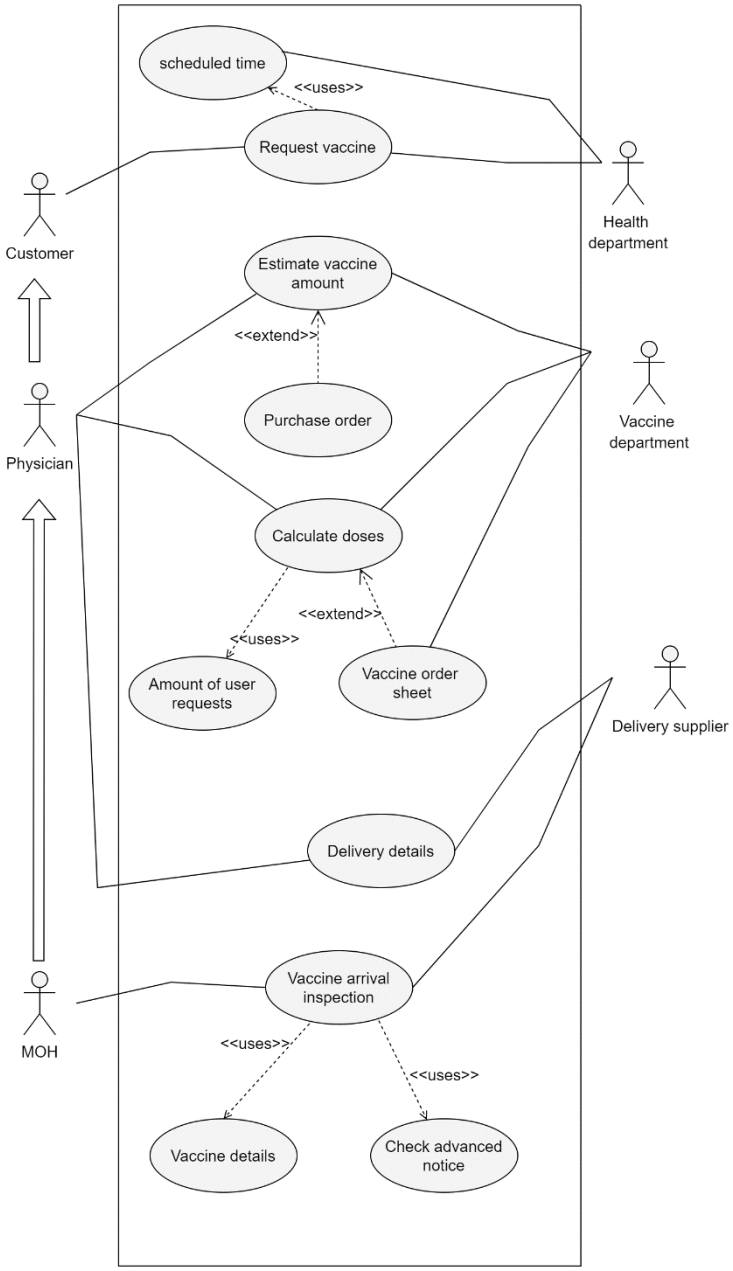

Figure 5- Use case diagram for requesting the vaccination

Assumptions

- Assuming that the non-vaccinated test is a pass hence the users can request a vaccine.
- Assuming that the vaccine department and delivery supplier are separate entities from the healthcare authority centre.

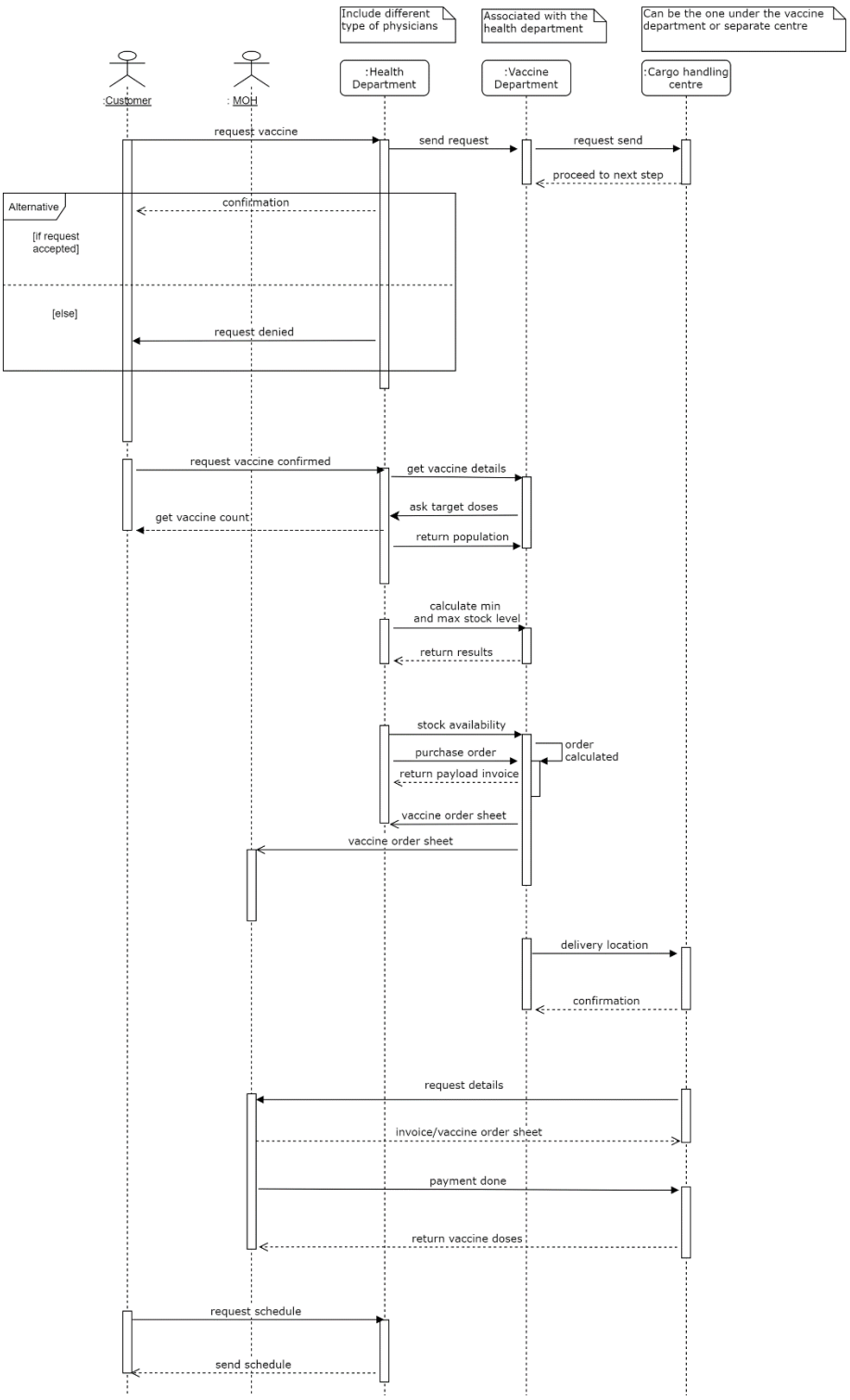

Figure 6- Sequence diagram for acquiring the vaccine through the system and healthcare authorities

- *<u>Assuming</u> that the final schedule received by the users is the schedule analysed by the physicians.*

2.3. User Classes and Characteristics

The user characteristics of the system are a vital part of the application. Fulfilling this part is crucial and must be one of the most prioritized aspects that need to be taken care of. When considering the technical expertise the users are the IT managers, IT staff, and system admins/ administrators that protect and govern the system accordingly. The system administrators can be used in getting access to close contact for system audits. The end-user of the system is the customer that this system is originally built for where they perform all the required functionality created by the system. The security levels are governed by data clerks and security administrators who perform frequent and vigorous testing to analyse the performance and check for any cyber-attacks. In the experience and educational aspect, the healthcare domain is the most important in this project as analysis of risks, and assessing user records is a vital process in accomplishing the requirements of this software application. The vaccine department and the cargo supplying classes are also vital educational expertise that is used in the use case/ functionality of getting the vaccine successfully.

Methods in each class are broadly showcased in the UML class diagram that can be found under the appendix section of the SRS document. Each associated class has its own attributes that may aggregate from the superclass or be generalized from the abstract class.

The backend server that stores all the data, history logs can be also found in the UML class diagrams.

Assumptions

- That all the user classes are generalized from the users abstract class.
- The account class is an inner class inside the health authority class.
- The object diagrams are represented by the dash lines extending between a specific relationship.
- The arrows show what the association link states.

Please refer to the Appendix for the class diagrams and their assumptions.

2.4. Operating Environment

The operating environment is important for the application to be run. The system should be compatible with both ISO and Android operating systems. Mainly Android smartphones need to have the latest operating system that is OS 11 to support the installation of this system.

For ISO all the operating systems support the installation of the application without any trouble and hence the hardware constraints aren't that severe as when Andriod downloads are considered.

The main operating environment that this system runs is under **Microsoft Windows** if the system is considered to be installed in computers.

2.5. Design and Implementation Constraints

When designing following constraints must be followed to achieve the system's desirable outcomes, the following constraints must be met for the application to be successfully completed:

Hardware limitations

Windows platform: The constraint is that the main system is expected to work on a windows platform and so is dependent on these computer systems being installed and must also work within the confines of the mobile phone's latest operating system version. When installing the application is of size 67MB. The backend server must be a distributed system of computers of about 40 networks together working to store the available data correspondingly over real time.

Memory limitations

Memory: When considering the memory needed for the application, we only have to focus on the memory that is taking up space, which is the decentralized server. Approximately having a primary memory size of 16GB for the decentralized server and initially 200GB can be used as secondary memory is valid. This memory is needed during the design process and is efficient usage of memory.

corporate or regulatory policies

Further constraints include regulatory governmental constraints including adherence to privacy, integrity, and confidentiality of data that is crucial for the act in relation to the right users where they have their personal information remaining private and out of reach of unauthorized users. Also, in relation to the salary process in the employees, it must be ensured that the correct tax rates are applied to all transactions or works they have done, calculating salary due to the amount of work they had done during the day. Taxes applied must be following the government income tax laws.

When considering the adherence to policies in the workplace, the employees can follow the IEEE code of ethics (attached in references). Maintaining professionalism in requirement engineering is crucial as competence, confidentiality, intellectual property rights, and data protection are all contributing to one goal that is maintaining policies and ethics in the workplace.

Copyright issues can be serious; hence copywriting laws must be followed during the implementation process as critical as possible and cite any work that is being referenced even your previous work.

Specific tools and protocols

The tools that are needed are the contact tracing tool/ protocol and the Bluetooth proximity protocol. These will be inbuilt within the decentralized server and capture the system in its environment in a more legible way. Using this approach is reasonable and authentic giving more references to the system itself.

Parallel operations

The system must be able to able to handle multiple access of the same data simultaneously. Hence, the system must allow multitasking. As tokens are being constantly shared the system must be able to allow multitasking. Tokens should be automatically generated after 24-hours of generation and hence this means of parallel operations must happen over time, therefore the performance of the decentralized server must be fast and efficient.

Language requirement

Since the scope of this project is a nationwide implementation, the language barrier is not considered hence the application will be developed in the English language with no issues.

Communication protocols

The only communication that is critical is the sharing of encrypted tokens with the system and users. This protocol is done by HTTP, a connectionless protocol, devices don't need to stay connected to the cloud IoT(internet of things) core. Alternatively, they can send requests and receive responses accurately. HTTP is used is primarily used for communication, storage, encryption, networks, decryption, security, and user login management, among other things. But for this project, it is mainly used as a means of communication protocol for decrypting and encrypting shared/ transferred token IDs.

Programming standards

The client's and stakeholder's organization are responsible for maintaining the delivery of the software application iSafe. The programmers can effectively be allocated to the standards of the particular organization and continue their work as joint programmers from suggested companies can help in the implementation process.

2.6. Assumptions and Dependencies

The following factors are not designed constraints but rather assumptions on which the requirements have been based. Any changes to these stated assumptions could affect the requirements stated in the SRS here within. If these assumptions are false, these requirements cannot be considered valid. Hence, it is said that due to these reasons the requirements are dependent on these assumptions.

The following assumptions have been made for the project:

Windows platform: The software is to be designed using the windows SDK and with windows look a feel. It is expected that a windows operating system needs to be at least Windows 2000 to maintain the application. The decentralized server must be sustainable enough to withstand the operations performed by the software. The decentralized server contains more than 20 network computers analyzing and storing data at the same time. The network is based on wireless connection; hence the server is more dynamic, meaning the servers can join and leave with the underlying performance of the network changing continuously.

Bluetooth requirements for communication protocols: As turning on Bluetooth is crucial for contact tracing, BLE must be available in all smartphones that this application is to be installed.

Latest OS available for Android phones- In order to be downloaded and to have highly flexible software the smartphones must have the latest operating version that is 11 in android(which is mandatory) and 14 in ISO (optional). The software works perfectly fine with lesser versions of the ISO operating system.

HTTP available – The HTTP available must be consistent and decrypt messages simultaneously as it is received and sent back to users. The performance must be kept up with the HTTP protocol and must have an alternative protocol available ready to be used if something happens to the current one as without HTTP contact tracing process cannot be carried out and hence the value of the applications is not maintained.

Upback backend server- All the information is stored in the back-end server, no one has access to its content. It's merely a distributed system that stores the current data evaluated from the software application and saves the data for later use. The system cannot co-exist without the backend server, which acts as a backup plan for the whole system as well if any cyber-attacks were to happen. In the modelling analysis diagrams, the name used for decentralized server and backend server is backend server/ server. This was for simplicity and should not be confused between the two.

Privacy – The users depend on the system to preserve their privacy when timestamp or geolocation is shared. Hence having a good privacy checking tool is of vital importance to the system. The network is assumed to include firewall-protected at all times for both the backend server and the decentralized server.

When considering the dependencies:

- Close contact alerts cannot be produced because the Bluetooth proximity protocol isn't working.
- Contact tracing is not activated therefore close contacts cannot be shared.
- Risk assessment and travel monitoring cannot be performed hence no travel suggestions are made.

Hence the prominent aspect of the system is the implementation of the contact tracing protocol and having the default Bluetooth proximity protocol. Without the proper implementations of these two concepts, the application has no use. Therefore, every functionality depends on the implementation of these two aspects.

3. Specific Requirements

The goal of this study is to examine the functional and non-functional requirements of users to continue developing the iSafe application. This specification helps the developers be more consistent and confident with the final product during its development as the functional requirements are stated and visible in a more punctual manner.

Assuming that all the figures that say referenced means they were directly taken from assignment 1: Modelling requirements.

3.1. Setting up the account

This is the first user interface the user comes across.

Functional requirement FR_1	
Title	*Setting up the account*
Description	The user must accept the conditions and terms before setting up the application for the first time and should be able to follow on-screen instructions to set up (like changing privacy settings and allowing exposure notifications activated).
Justification and details	It is a must-have requirement as it contains clauses that set out the rules, conditions, restrictions, and limitations that a user must accept to use the mobile app.
User story	As a user, I must be able to accept the conditions and terms before signing up for the first time so I can have an idea of the rules and allow certain permissions to be activated.

Referenced

3.2. Login to the application

This is where the users can log in to the system by verifying their passwords. Users include all the personnel that uses this system such as a physician, system admins etc.

Functional requirement FR_2	
Title	*Login*
Description	Users can only log in to the software application using the registered number and provided password.
Justification and details	The first page a user sees after logging in is the login home page. This will allow users to log into their accounts, check updates, reset their details or password, and check in to their risk profiles without any delay.
User story	As a user, I must be able to log in to my account so that I could access my information and update my details.

Referenced

Verification message printed to the screen if correctly logged in and for incorrect logins the system output an error message.

Please refer to the UML class diagram under the appendix

3.3. Contact tracing activated

The contact tracing should be activated by the users, this is an option to be carried out by the users as having contact tracing on as a default setting wear offs the battery life of the phones.

Functional requirement FR_3	
Title	*Contact tracing activation*
Description	The users must be able to activate and deactivate the contact tracing feature anytime they want
Justification and details	It's a must need priority as the user may or may not want the tracing to be activated due to various reasons. The contact tracing is activated by ticking the mark. If the contact tracing is not activated, the users can access it by the application settings and fix the issue.
User story	As a user, I must be able to activate and deactivate the contact tracing tab anytime I want so that I can save the mobile battery when I'm not using the application.

Referenced

3.4. Bluetooth activated

The program creates a unique identifier/token that it broadcasts to adjacent devices when it uses *Bluetooth*. The user's phone then saves the identifiers of other phones that it has been nearby. If a user becomes infected, their unique identifiers can be compared to those saved by other users to see who the infected person has met

Functional requirement FR_4	
Title	*Bluetooth or GPS*
Description	The system will turn on Bluetooth and GPS automatically as the contact tracing is activated.
Justification and details	No trouble needing activating the GPS or Bluetooth
User story	The system will activate GPS and Bluetooth once contact tracing is activated.

Referenced

3.5. Location sharing

Checking risk levels before going to a particular event.

Functional requirement FR_5	
Title	*Travel monitoring*

Description	The user should be able to check the risk levels of a social event before traveling to that place.
Justification and details	This is an effective way to save people's time and keep them safe. When the threshold level is higher than the normal level, it is more likely that this location is not safe so users can avoid traveling here. Uses an automatic risk checker in public settings like shopping malls and office buildings to reduce community transmission.
User story	As a user, I should be able to check the risk levels before traveling to a social event, so I can be safe and save time.

Referenced

3.6. Travel suggestions

Firstly, the system administrator checks if there are any history logs of the particular location available in the back-end server.

Functional requirement FR_6	
Title	*Travel Location access*
Description	The system administrator must be able to look at the history logs of the locations of the customers and disclose an infected person's contacts to the close contacted people.
Justification and details	This is a high priority that requires the system administrator.
User story	As a system administrator, I want to be able to trace history logs from the database, so I can act and inform close contacted people if they were contacted with an infected person.

Referenced

Then, if it doesn't contain any history about the location, the system will perform a risk assessment through physicians and recommend travelling destinations.

Functional requirement FR_7	
Title	*Travel suggestions after a risk assessment*
Description	The users must be able to receive the low-risk level places suggested by the application as a travel suggestion.
Justification and details	This is a medium feature added to the application where the users can have the suggestions of low-risk analysis of certain places the application can detect nearby.
User story	As a user, I would like to receive suggestions of low-risk level places, so I can travel to those places instead.

Referenced

3.7. Social interaction touchpoints

This is where the users can get the different risk levels of a public place before going to that place such as an ATM, shopping mall. First, the user must scan the QR code available at the places.

Functional requirement FR_8	
Title	*QR code scan*
Description	The system must be able to scan the QR code available at the stores and social places either using their phone's camera or from the system itself
Justification and details	It is a high priority requirement that the system should have which can minimize the spread of the virus even more.
User story	As the system, the QR code scan can be used to evaluate the place's risk levels so it can satisfy the users by providing the risk level notification.

Then the system must be able to automatically store the location to the backend server.

Functional requirement FR_9	
Title	*Social interaction touchpoints*
Description	The system must be able to record the location of a contaminated social/public interaction point the user interacts with automatically such as an ATM, contaminated private transports, cashier points, shopping malls, or banks.
Justification and details	It is a high priority requirement that the system should have which can minimize the spread of the virus even more.
User story	As the system, the location of a certain contaminated place should be recorded in the database, so this can be notified to the people using the application.

Referenced

3.8. Covid check-in questions and feedback

Functional requirement FR_10	
Title	*Covid check-in*
Description	The users can check in and help by recording any symptoms they have to the system where they can keep track of their updated profiles.

Justification and details	Once the system asks the basic questions and the symptoms are recorded, the application advises on what to do next, giving the users the options between view history and read more advice options and other. This way of recording symptoms helps the healthcare authorities/ physicians see how covid is affecting people across the nation.
User story	As a user, I want to record any symptoms I might have, so I could get advice from the healthcare executives or read more advice.

Referenced

Please refer to the UML class diagram under the appendix

The application assumes that there is an option to get a personal physician if needed for this sole purpose. The other option is where the users can get help from doctors if they are talking to them or being assisted by one or can get help from the contact tracing team.

Functional requirement FR_11	
Title	*Covid check-in by physicians to confirm users*
Description	The physicians should be able to assist the customers who have requested their help by accessing their risk profiles and calling the respective customers if they have asked for help.
Justification and details	Confirming the users are very important as they will be waiting for confirmation.
User story	As a physician, I should be able to access the relevant risk profiles and check their well-being before confirming if they are positive or not, so I could do my job.

Referenced

The physicians can check back on information by accessing their risk profiles and updating back on the customers of their results.

Functional requirement FR_12	
Title	*Sending notification from the physicians*
Description	The physicians can update back on the results to the respective users as a notification confirming the symptoms to be covid-19, preparing them for precautions. If they are negative, the physicians can send a default message to the users saying they are safe.

Justification and details	This is a medium-high priority functionality as they are important as the end-users.
User story	As a physician, I should be able to contact the users by notifying them of the results, so the users can get their results free and continue with their daily life or take start taking precautions.

Referenced

3.9. Tested positive

The users are tested positive, and there are different precautions that can be taken through the application to minimize the impact. The users can choose to upload health records to 100% confirm from the physicians' online than having to travel to a doctor physically.

Functional requirement FR_13	
Title	*Upload health records*
Description	The users can choose to upload any records anonymously to alert close contacts that they met within the 14 days.
Justification and details	The uploaded results confirm that the users are covid-19 positive, where the doctors analyse the data more efficiently.
User story	As a user, I should be able to upload my records to the system, so I can anonymously alert close contacts.

Referenced

Any false, spam uploads or duplicated uploads will be ignored and produce a warning message/ error message

Functional requirement FR_14	
Title	*False notifications updates*
Description	The system must be able to issues a confirmation page by having the users press a button before changing status over time.
Justification and details	It is highly necessary to have the warning/ confirmation button so the same users cannot update false status and prevent false notifications.
User story	As the system, there should be a warning page, so users cannot provide false information.

Referenced

Please refer to the UML class diagram under the appendix

3.10. Release timestamp or location

The users can share their timestamp or geolocation to alert the close contact list.

Functional requirement FR_15	
Title	*Release timestamp or geolocation*
Description	The users who tested positive could have the option of reporting their close contacts the timestamp of when that close contact happened, and their geolocation.
Justification and details	This is a medium requirement and should be provided as the users should always decide whether their information should be shared or not.
User story	As a user, I should be to choose between the options of reporting close contact their timestamp, and geolocation, so I can have my choice to my likeness.

Referenced

3.11. Send secure code to users

For the users to do that the physicians must send the security codes.

Functional requirement FR_16	
Title	*Secure code sent*
Description	Upon receiving the uploaded records, the users receive a notification asking them to share the code with the healthcare authorities.
Justification and details	The send secure unique code by text is to be entered by the users hence, this is the mechanism for sharing the random token with the healthcare authorities helping them slow down the spread of covid-19.
User story	As a user, I must be able to receive the unique code giving a head start to share the token with the physicians, so I can help the ones I have been in close contact with.

Referenced

3.12. Close contact alerts

The users can receive risk level notifications when they come in close contact with an infected person.

Functional requirement FR_17	
Title	*Risk level notifications /close contact alert*

Description	The users must be able to receive alerts/notifications when they come in close contact of 2m of proximity with an infected person.
Justification and details	This is a high-priority requirement and allows users to receive notifications faster.
User story	As a user, I should be notified if I encounter an infected person as soon as possible, so I can take precautions beforehand.

The users can have the phone call me button activated so the team or public health can call them to alert them about the close contacts if they were unable to see the alerts.

Functional requirement FR_18	
Title	*Phone call you*
Description	The users can receive phone calls from the public healthcare contact tracing team when they come in close contact with an infected person.
Justification and details	This is an effective way to minimize the spread of the virus as most people tend to ignore their notifications especially elderly people. This mechanism can only be activated provided the users have given their phones numbers to the system.
User story	As a user, I should be able to receive a phone call from the tracing team, so I get the message somehow even if I miss my notifications.

3.13. Buttons/ features available

The users have a variety of different options available when they are needed help.

1. Ask for advice button

Functional requirement FR_19	
Title	*Ask for advice*
Description	The users can ask for advice from public healthcare or from the contact tracing team.
Justification and details	This is an effective way to minimize the spread of the virus by having additional advice.

User story	As a user, I should be able to get advice from physicians, so I can be more on the safe side.

2. The users can direct ambulance support when they click the emergency button.

Functional requirement FR_20	
Title	*Link to the direct-dial ambulance*
Description	The users must be able to direct themselves to the direct-dial-in ambulance service and view details of emergency numbers.
Justification and details	During an emergency, the users must have the ability to dial in ambulance service and view emergency contacts during a time of need
User story	As a user, I should be able to view emergency contact and dial for an ambulance, so I can be safe during an emergency

3.14. Additional requirements

The following are additional features available :

1. Get updated news of covid-19

Functional requirement FR_21	
Title	*Updates of covid-19*
Description	The users can receive easy access to additional and latest updates about the virus.
Justification and details	Additional updates contain the latest figures and facts about covid-19 in the country.
User story	As a user, I can receive access to the latest updates of the virus, so I can be up to date with the current situation of the country.

Referenced

2. Deactivate the account feature

Functional requirement FR_22	
Title	*Deactivate Account*

Description	The users must be able to deactivate their accounts whenever they want.
Justification and details	This may be due to many reasons but the freedom of deactivating the user's accounts should always be an option provided to them. This process involves the deletion of data from the cloud database too.
User story	As a user, I should be able to deactivate my current account, so I can recreate a new account (or for more reasons).

3.15. Vaccine procedures / functionalities

1. The users can request the vaccine from the system

Functional requirement FR_23	
Title	*Request vaccine*
Description	The users must be able to request vaccination date and time (schedule) as the non-vaccinated test is passed
Justification and details	This ensures that all users will be vaccinated hence helping the analysis process for the physicians when they analysis the spread of the covid-19 pandemic
User story	As a user, I should be able to request the vaccine's scheduled time and date through notification by physicians, so that I can be vaccinated.

2. *The healthcare authority can request the vaccine order sheet from the vaccine department*

Functional requirement FR_24	
Title	*Get vaccine order sheet*
Description	The physicians request for vaccine order sheet to confirm the order.
Justification and details	The vaccine order sheet gives a broad understanding of vaccine details before ordering. The vaccine department generates the vaccine order sheet accordingly to the information received before this process from the physicians.

User story	As a physician, I must be able to request the vaccine order sheet from the vaccine department, so that I can have a broad idea of the vaccine details before ordering.

Please refer to the UML class diagram under the appendix

3. *Get the cargo to the delivery details*

Functional requirement FR_26	
Title	*Send to cargo handling centre*
Description	The vaccine department sends delivery details to the cargo supplier.
Justification and details	The cargo supplier could be someone from the vaccine department or an external entity.
User story	As a cargo supplier, I must be able to receive delivery details, so I can deliver the vaccines to the destination.

Please refer to the UML class diagram under the appendix

4. Get the vaccines from the cargo supplier

Functional requirement FR_26	
Title	*Get vaccines*
Description	The physicians request vaccines from the cargo supplier.
Justification and details	The cargo supplier requests the vaccine invoice that was given to physicians during the generating of the vaccine order sheet.
User story	As a physician, I must be able to get the vaccines I requested from the supplier, so I can send them to vaccine centres to be able to be used on customers

3.16. Privacy checks

While data-driven solutions are highly effective, citizens are concerned that information such as location or proximity, when combined with other personal data, might be used by governments to impose monitoring. The low adoption rate of such apps due to a lack of trust cast doubt on their usefulness, therefore performing security audits and checks is critical to maintaining the integrity and confidentiality of data.

Functional requirement FR_21	
Title	*Privacy checks*
Description	The national security data executive can check the system for the accuracy of maintaining the privacy of the users.
Justification and details	As privacy is one of the main concerns, it should always be looked after and so the security data executive can do penetration testing, application audits to check the login details and analyse the data thoroughly.
User story	As a security data executive, I should be able to check the accuracy of the system, so confidentiality and integrity are maintained.

4. External Interface Requirements

4.1. System interfaces

The system interacts solely with the backend server which is a decentralized server and as it is not part of a bigger system there is little need for specifying software requirements. However, there are a few functionalities of the software that needs to be achieved in order to accomplish the system requirements. These aspects are described below:

- The software needs to be developed in an interactive stage where the users can perform a risk assessment and get the evaluation of the report.
- The software has to logically do contact tracing within the proximity of 5m and be able to alert users within 10seconds after being encountered with an infected close contact.
- The software needs to be able to deploy new covid check-in questions after every question is answered so the system can evaluate these and update the risk profile history.
- The software must be able to produce error messages at the relevant times, explicitly stating what was done wrong, for example, if the user uploads a false health record or try to upload a duplicate record, the system produces an error message.
- The software must be able to validate all users in order for the system to do any requirement.
- The software recommends updates to be made for the system to perform the various functions with ease.

All these mini- functionalities must be achieved for the system to perform at its maximum value.

4.2. User interfaces

Each of the user interfaces needs to be achieved to maintain a user-friendly environment.

1. Login screen

The app can be accessed by entering the user's password and email. The time the user registers to the app, the user must set up the application. A verification code is sent to their phone number via SMS when first registering to the app as shown in figure 1.

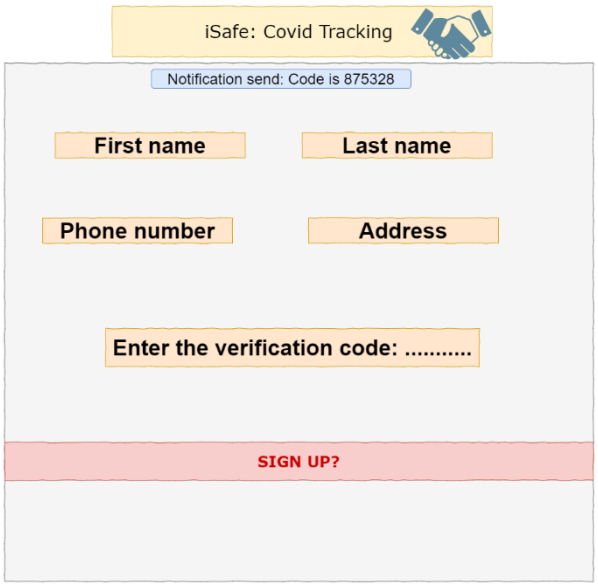

Figure 8 – Wireframe diagram one

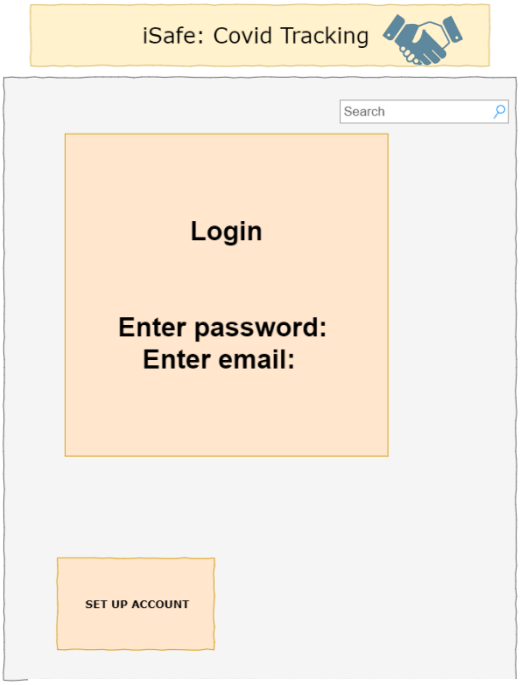

Figure 9- Wireframe diagram two

2. The next is setting up the account of where the users have to read and accept the terms and conditions, confirming that they are 18 and above as shown in figure 3, this needs to be done before setting up the risk profile. And figure 4 shows the page after setting up the profile.

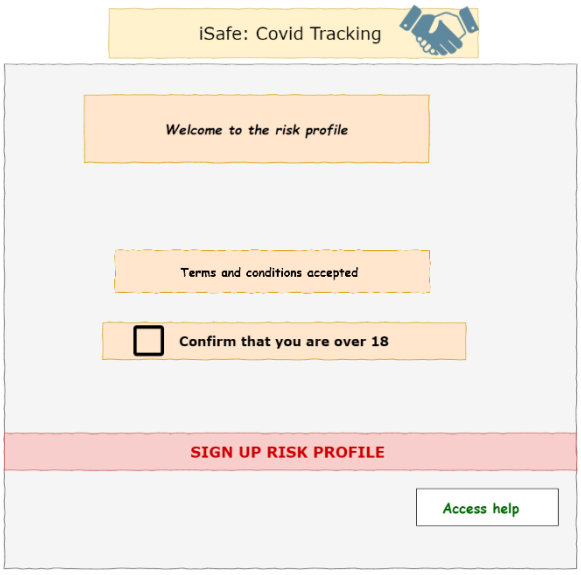

Figure 10- Wireframe diagram three

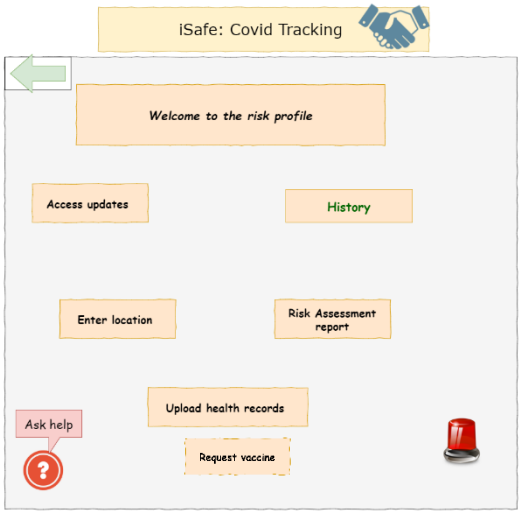

Figure 11- Wireframe diagram four

3. Activate the contact tracing option in the tab as shown below in figure 5. This is a crucial aspect of the whole system and must be active at all stages in order to get the maximum usage of the software

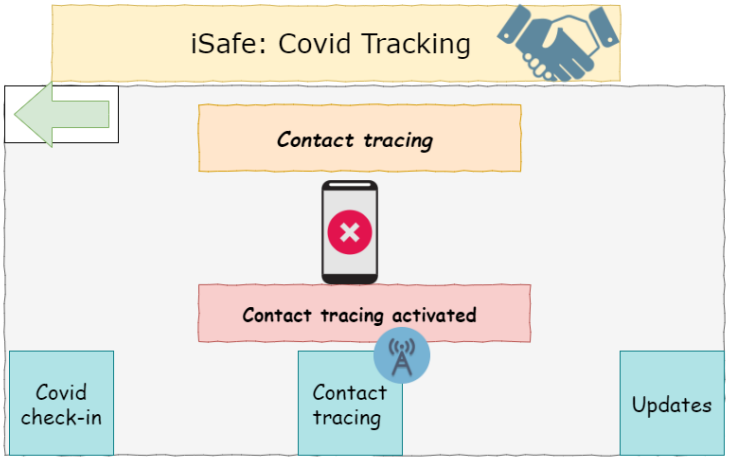

Figure 12- Wireframe diagram five

4. The users have the option to view history under risk profile basically, this option shows all the recorded symptoms evaluated by the daily covid check-in questions that can be done by the users.

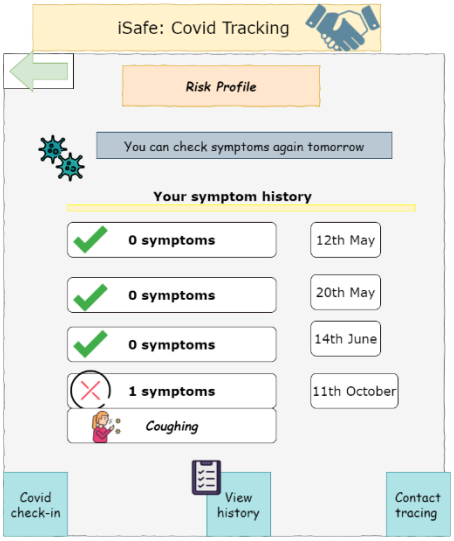

Figure 13- Wireframe diagram six

4.3. Software interfaces

All the third-party interfaces are captured within the system itself, such as contact tracing tool/ protocol and Bluetooth protocol that will be available in smart devices developed since 2015. The decentralized server needs to be supporting all these internal frameworks effectively. The acquiring of a decentralized server can be done by acquiring the decentralized application, that is a computer application that runs decentralized computing systems.

Under section **3.1.1. System interfaces**, most of the software functionalities are described for the accomplishment of the overall system requirement. In conclusion, there is a very little amount of software interfaces that need to be achieved in the project as the application will be built as a new product where most of the system requirements will be the same as the software functionality. The contact tracing protocol, as well as the close contact alerting, does

not require additional systems/software as these tools will be internally implemented during the design process. Hence, software interfaces become system interfaces as well.

4.4. Hardware interfaces

The system supports any device with the BLE protocol as the Bluetooth proximity function needs to be Bluetooth low energy for the contact tracing process to work. This is a possible hardware interface required as the application is smartphone-based, much of the interfaces and functionality depends on the system and its internal software. There are no explicitly needed hardware interfaces for this system to work and hence this section is not that important in the SRS document. The decentralized server must co-exist with the distributed backend server, where all the information will be stored concurrently. The backend server and the decentralized server are the same only the number of networks and the usage is different. The backend server is solely used for storing the data, it acts as a backup system to the decentralized server.

4.5. Communication interfaces

The system acquires an email or phone number to communicate when the user first logs in to the system, this is used to send the one-time verification code to the users. The IT providers are the ones that are responsible for the communication interface. The HTTP is used to access small files such as shared health records from the server by physicians. And accessing the user profiles by physicians is also easy and requires only HTTP. No large files are being uploaded by the users or physicians hence FTP is not required. These requirements for the communication interface are not that hard to implement providing a gateway for developers to focus on more difficult functional requirements.

5. Other Non-functional Requirements

Most of the non-functional requirements are already discussed in the above sections very broadly especially under the external interface requirements. And here, short discussions about the system's relevant attributes are described and taken into account.

5.1. Performance Requirements

5.1.1. Follow up instructions and scanning

The users must be able to view the follow-up instructions within 2seconds of accepting the conditions and terms and navigate on them. This should be developed alongside the deployment of the creation of the accounts of users. The system must be able to scan the tokens of the users who are at a proximity of 5 meters from each other helping the Bluetooth proximity protocol to be achieved successfully.

5.1.2. Close contact alerts

The system must be able to scan the close contacts tokens using Bluetooth in under 700 milliseconds and upload them to the backend in under 800 milliseconds. As the system needs to get the tokens to the decentralized server the accessible time slot of the tokens to proceed in comparing must be shorter as taking greater time to send alerts to users could be a dangerous move. Hence, this performance rate is vital. The system scans and decrypts at least 6.700.000 message tokens (approximately equalling to 70Mbyte of data being transferred) per day. This is crucial as each day more than 10000 tokens will be shared so the analysis and comparisons must be quick. Within 700 milliseconds of meeting covid-19 victims, the system must be able to notify users. Time taken to notify users are also critical and hence shorter periods must be allocated for this process as well.

5.1.3. Risk calculation and secure code sent

Within 3 seconds of the user entering the location, the system must be able to assess and compare the risk levels of the social events. After receiving records, the system must be able to communicate the security code to users within

5 seconds. Physicians will have 12 hours to report back on the user's symptom state, with an alert sent out before the last hour. The process is quite tedious, but the decentralized server must be able to fulfil its requirements efficiently.

5.2. Safety Requirements

5.2.1. Personal information

The system must be able to store and keep the user's private information in the backend server of the system. No one can access the backend server unless the requirements are met. The decentralized server sends out all the personal information once received to the backend. The access is granted to the backend only through strict access control mechanisms.

5.2.2. Error messages or warning pages and confirmations

The system must provide an error message for every wrong log-on attempt and restrict attempts after 5 continuous wrong attempts. The warning pages are distributed when false uploading of health records is made into the system. The system must limit the user to report a sensible number of notifications/ updates to the system and also prevent the application from being flooded with the notifications. (At most the application can contact a maximum of about 1000 users).

5.2.3. Keeping back up of data

The system must ensure that all the data are backed up to a server, this server could potentially be the same backend server or a different one, but it is crucial to have a good and strong backup server available in case of disruptions.

5.3. Security Requirements

Considering the whole application domain:

The system must be able to detect malicious uploads to the system's server. Multiple attempts to log into the system will result in a warning notice being displayed on the screen, as well as the screen is locked for 50 seconds. Unencrypted passwords must not be stored in the system. Every 24 hours, the system must supply unique IDs/tokens to all users. The system must use TLS (Transport Layer Security) encryption for all network communications. To prevent malicious attackers from tracking the user's location, the system must be able to remove all malicious advertising packets. Data confidentiality, integrity, and availability must all be guaranteed by the system.

Codes of ethics can help people make better decisions when it comes to computer use. Employees who have obtained some level of qualification or professional accreditation, on the other hand, maybe stopped from engaging in unethical behaviour by the possibility of losing their accreditation or certification due to a code of conduct violation. The loss of certification or accreditation can have a significant impact on one's marketability and earning potential. It is the responsibility of security professionals to operate responsibly and in accordance with their employers' rules and procedures, as well as the laws of society. Hence, the security certificates and privacy certificates must be acquired by the employees as well as for the system.

5.4. Software Quality Attributes

5.4.1. Availability

The application should be accessible seven days a week, 24 hours a day. (To store its data, the application will use a decentralized cloud system.) The system should be able to hold enough data to store all of the encrypted messages. Both ISO and Android operating systems should be supported by the system. The system’s user registration component should be available 99.99% of the time. The system’s uploading status should be available 99.999% of the time.

5.4.2. Reliability

The application must have a mean time to failure of at least one year. Hence, it should work without any bugs and crashes for the application to be recognized by the nation. Notifications from the system should not fail more than 0.1 percent of the time. Within 0.1 percent of attempts, the system should receive the uploaded records from the relevant users without failure. The travel monitoring feature of the system should not fail more than 0.2 percent of the time. Every 1000 hours, the system must not send out more than one inaccurate alert message. Every 100 hours, the system cannot produce more than one inaccurate low-risk trip proposal. When users try to dial a public number, the system must not make more than two incorrect direct calls to the contact tracing team.

5.4.3. Usability

Users should be able to change their passwords whenever they desire using an easy-to-use interface. The system must allow infected users with a page of options where they can decide to share their geolocations and timestamp or not. The system must allow the user to enter the destinations to be visited to check risk levels and recommend the location with the lowest risk level to the users. The user should be able to query their historical data with no more than two mouse clicks. When consumers try to update their profiles, the software must help them choose the correct records to upload at least 99 percent of the time.

5.4.4. Interoperability

The system must be able to communicate with MySQL version 8.0 for database storage. Android and IOS operating systems with version 11 and higher will be able to execute the program.

5.4.5. Maintainability

The user's device will receive a new update request. The upgrade will fix any flaws in the system, as well as improve it and add new features. Maintaining the system after its development is critical and can be done by having a security audit team or the team members perform regular check-ups on the system.

6. Apportioning of requirements

In the future, the system requirements can change over time as new updates are made or when the client suggests additional features all these various ways to alter the system are very likely to change the application domain. A few of the suggestions and updates that can be done to the system to make it more powerful is by using both GPS and Bluetooth protocol mechanisms, taking advantage of each technology's strengths. Also, new algorithms can be developed that limit the likelihood of small or false exposures being flagged which can help in the reliability of the system.

7. Appendix

All the extra analysis models are concluded in this section such as block diagrams, state-transition diagrams, and activity diagrams, use case diagrams, sequence diagrams, goal modelling, and finally strategic dependency and strategic rationale diagrams exclusively.

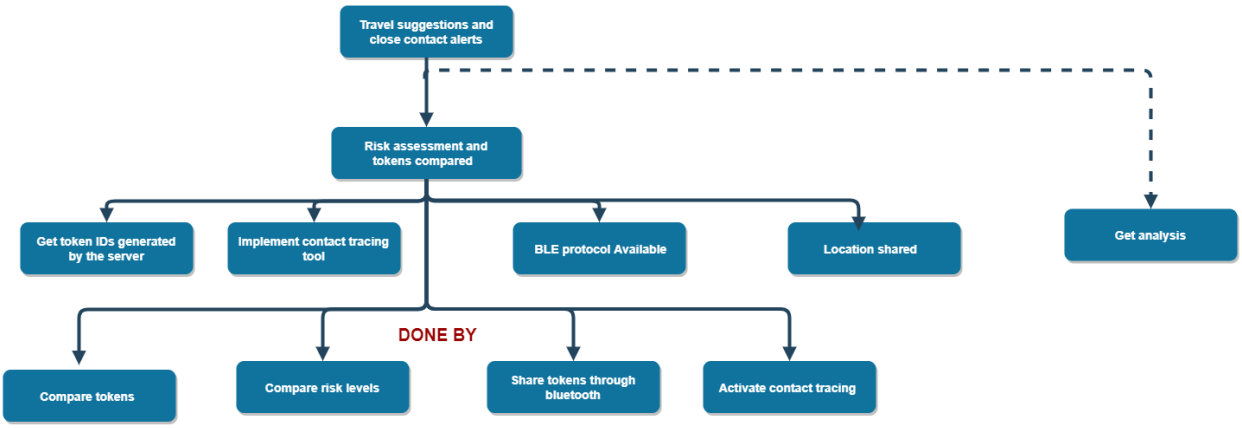

Figure 14- The block diagram for the product perspective

UML class diagrams

- The UML class diagram below showcases the relationships between users classes, where it shows the connections between the risk profile account of customers and the healthcare authority's account class.

Assumptions

- Assuming that enumerations class is just to show the order of status the customer class can have.
- The account class is a private inner class inside the healthcare authority's superclass.

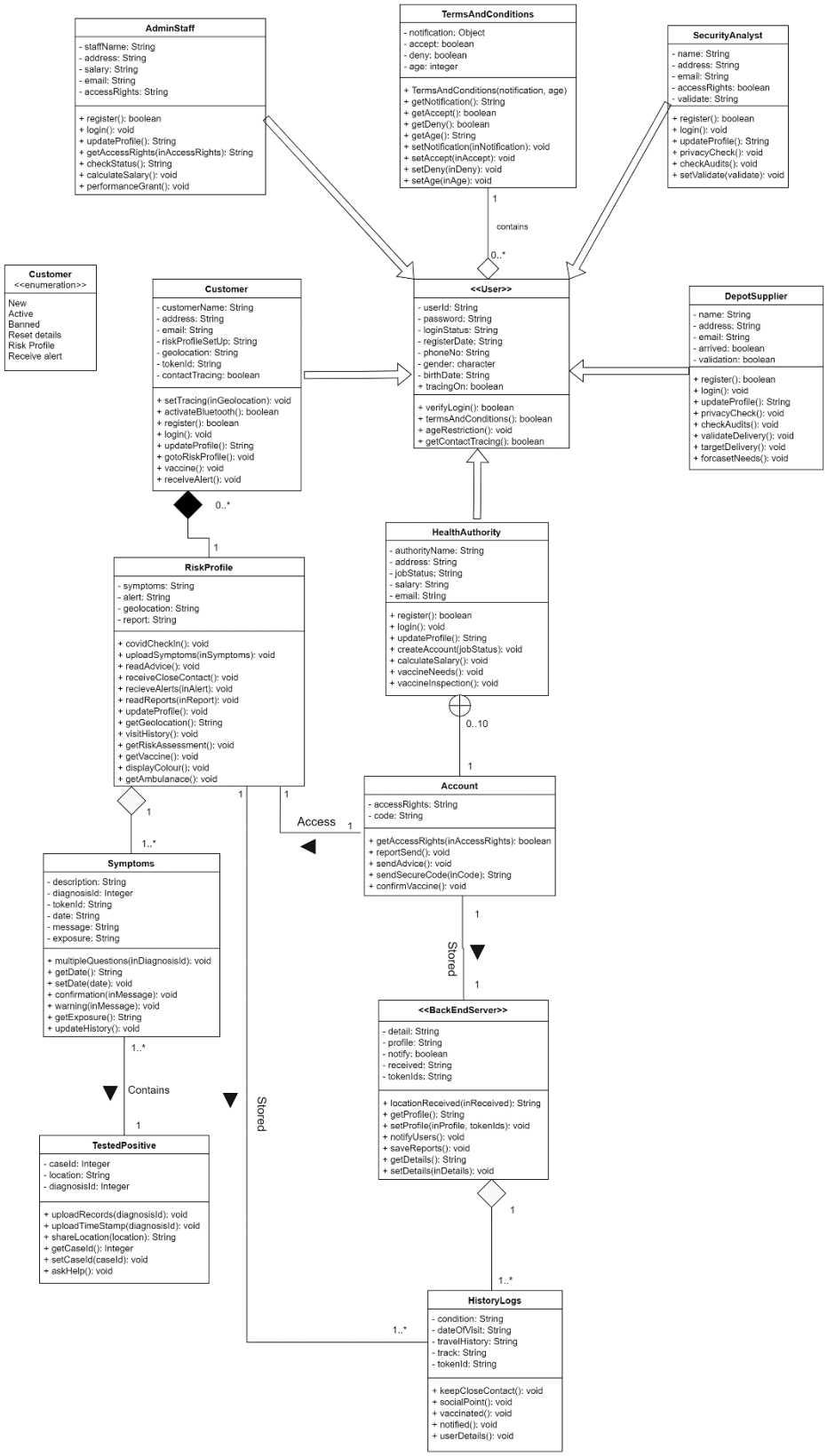

Figure 15- The first UML diagram that demonstrates the relationships between the user classes and its subclasses

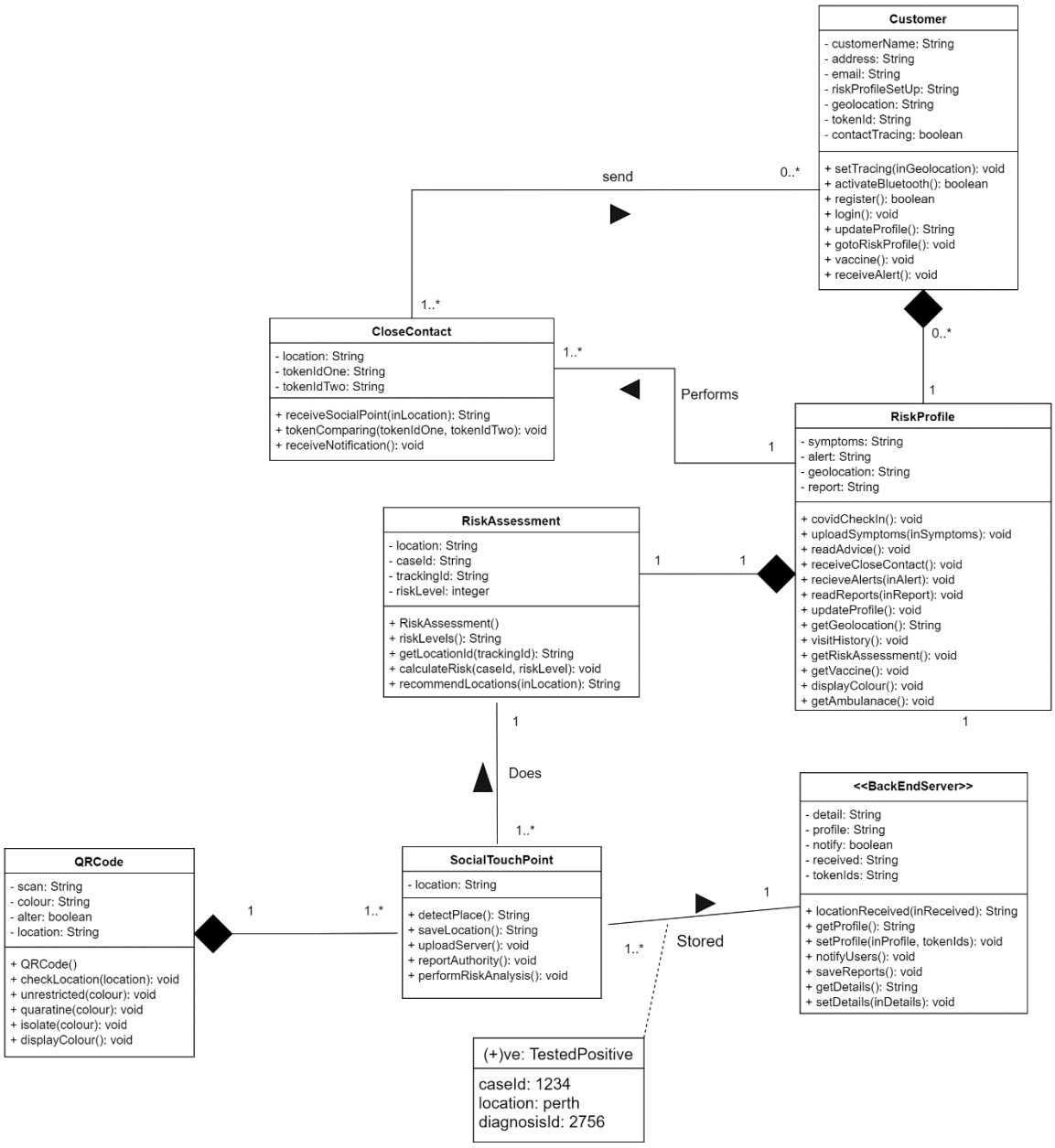

Figure 16- The second UML diagram that showcases the relationships in the problem domain

- <u>The assumptions are:</u>
  - Composition is a whole relationship
  - Most diagrams are drawn here

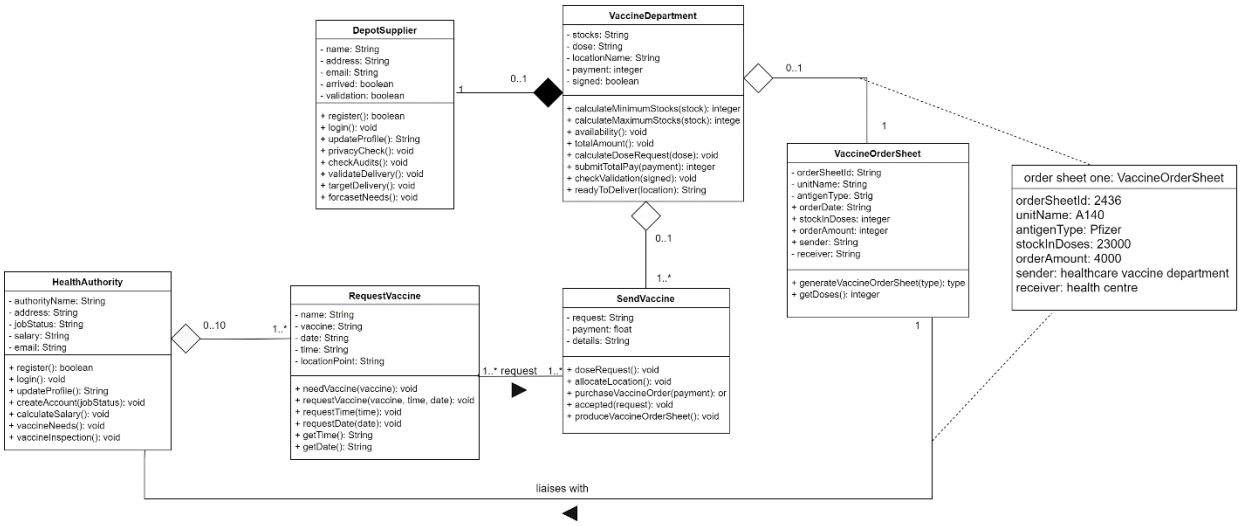

Figure 17- The Third UML diagram that shows the relationships when vaccine order is involved

- The assumptions are:
    - The vaccine is being requested by the customers
    - Object diagram is shown

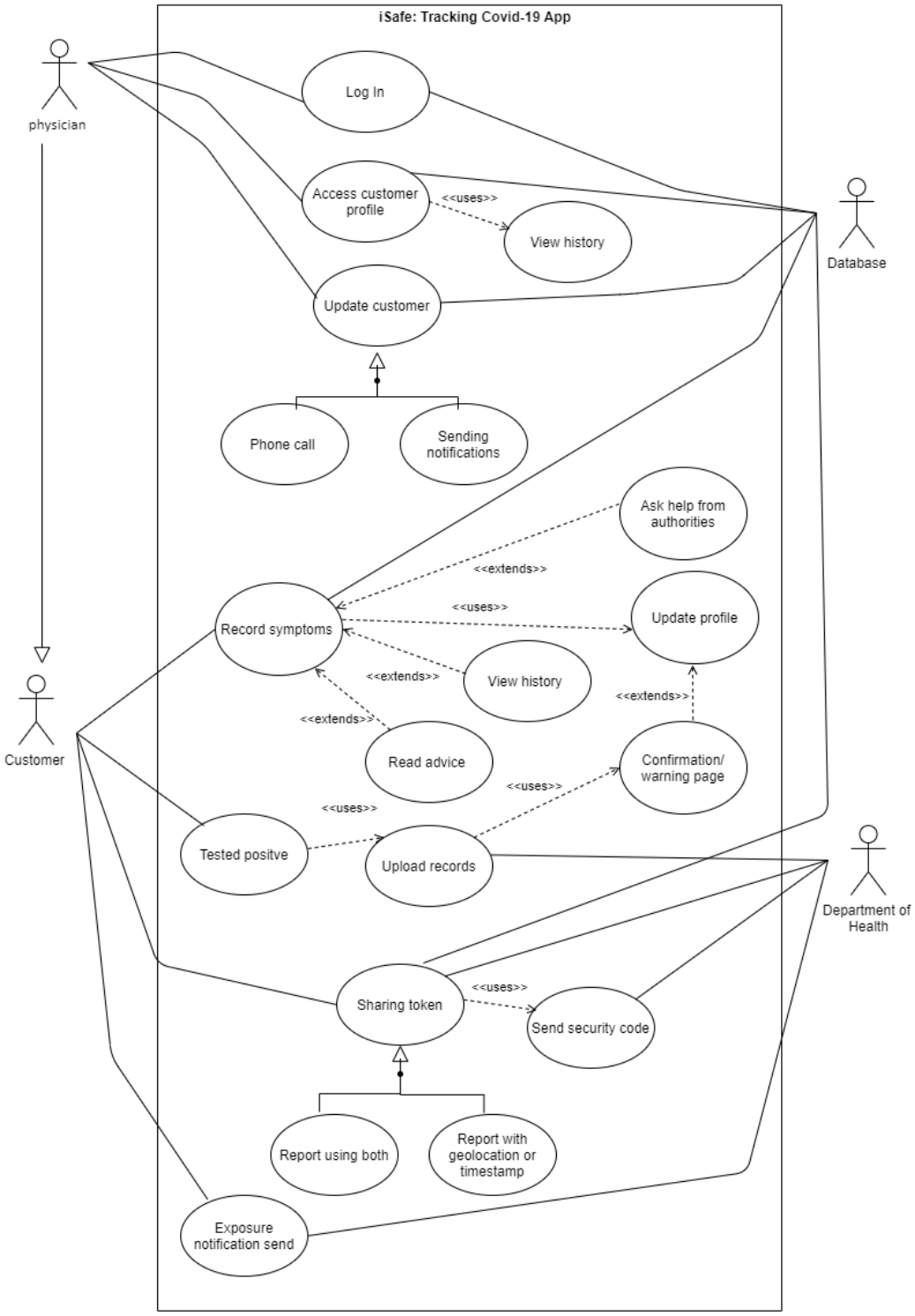

Figure 18-Use case diagram that shows the uploading of health records and sharing of token ID or geolocation

Referenced from Assignment 1: Modelling requirements

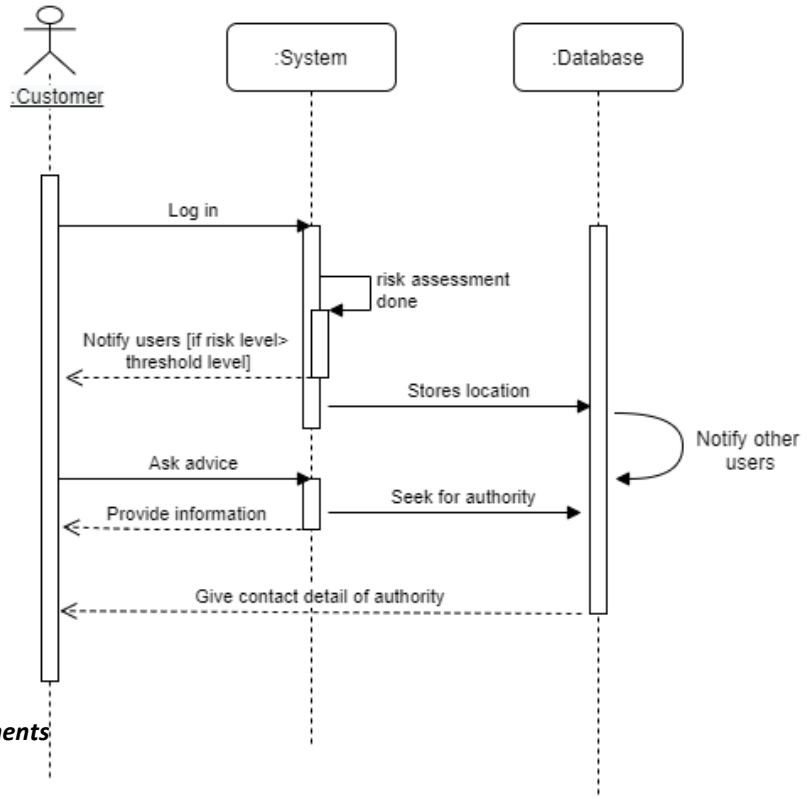

Figure 17 – Sequence diagram for how the social touchpoint is recorded in the application using risk assessment.

Referenced from Assignment 1: Modelling requirements

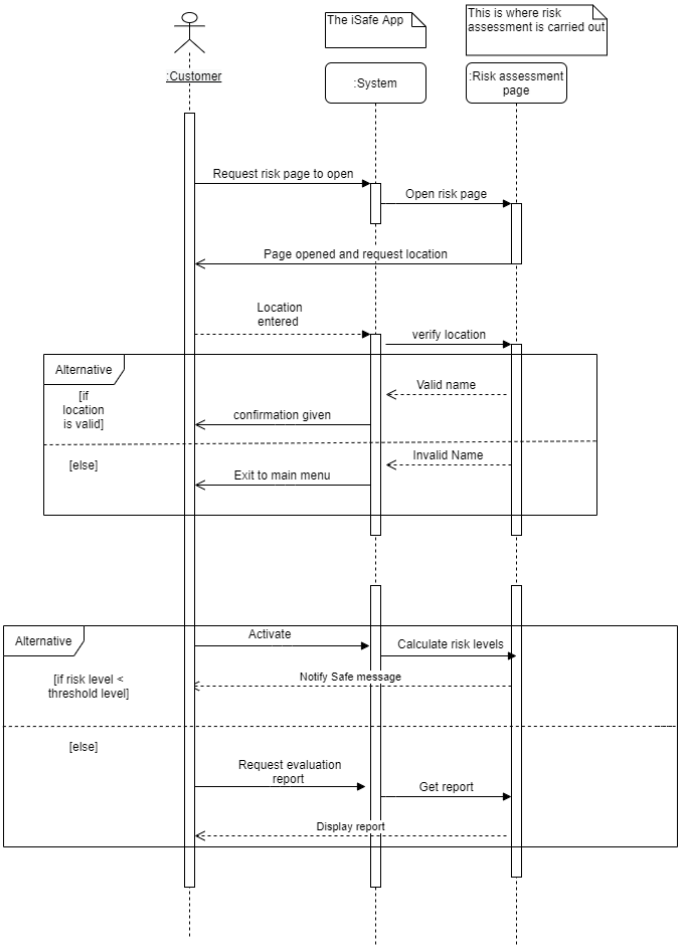

Figure 18- Sequence diagram in how the risk assessment analysis process is done

Referenced from Assignment 1: Modelling requirements

Goal modelling analysis models

The Goal modelling analysis diagrams are concluded from here onwards:

1. The first diagram showcases the goals achieved by the tracing activate and its soft goals.

Assumptions:

- Update password is when the user wants to get a new password which is considered a hard goal here.

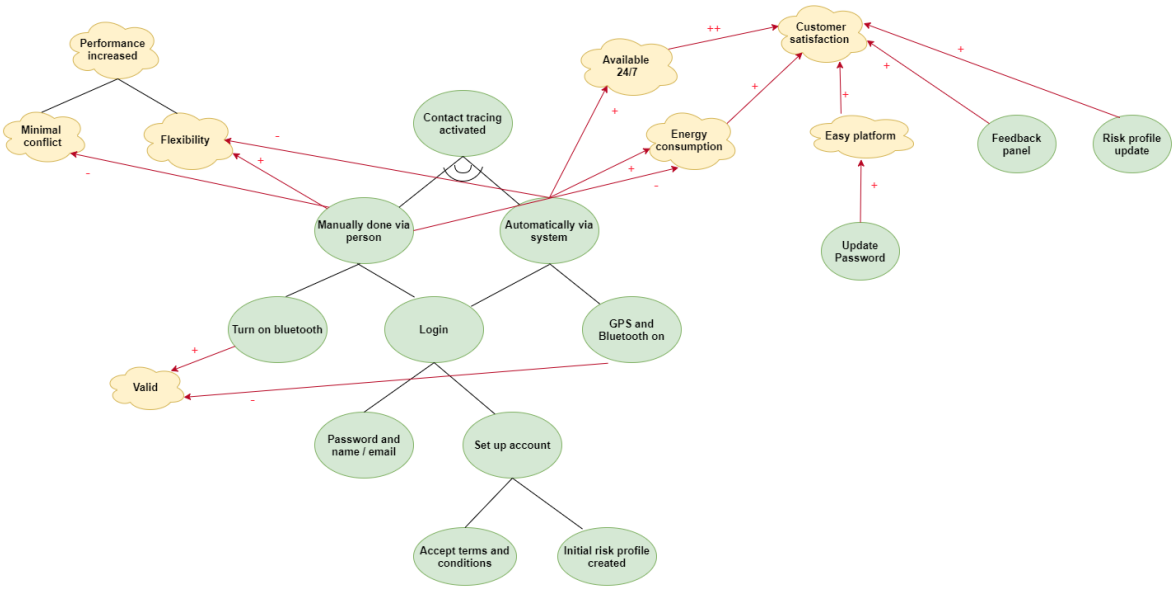

Figure 19- Goal modelling diagram One

2. The second Goal modelling diagram showcases how the goal ” travel suggestions” can be achieved by its sub-hard gaols and the corresponding soft goals as distributed below.

Assumptions are:

- Both the hard goals report generated by physician and user diagnosis is used to get a clear understanding of the diagrams.

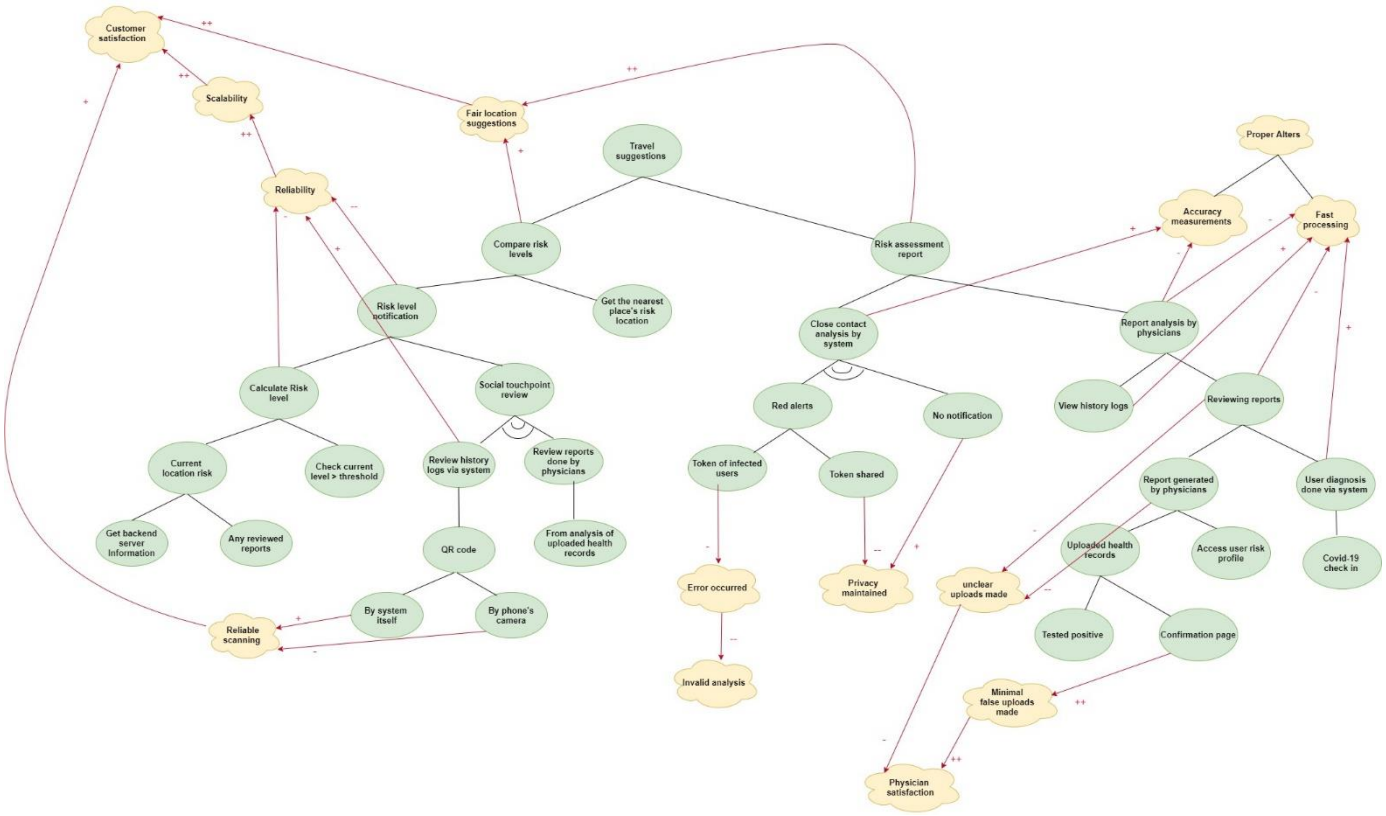

Figure 20- Goal modelling diagram Two

3. This model describes how the goal of “sharing geo location or timestamp ” can be achieved.

Assumption is evidence means sharing geolocation **or** timestamp (date, time and location, hence all)

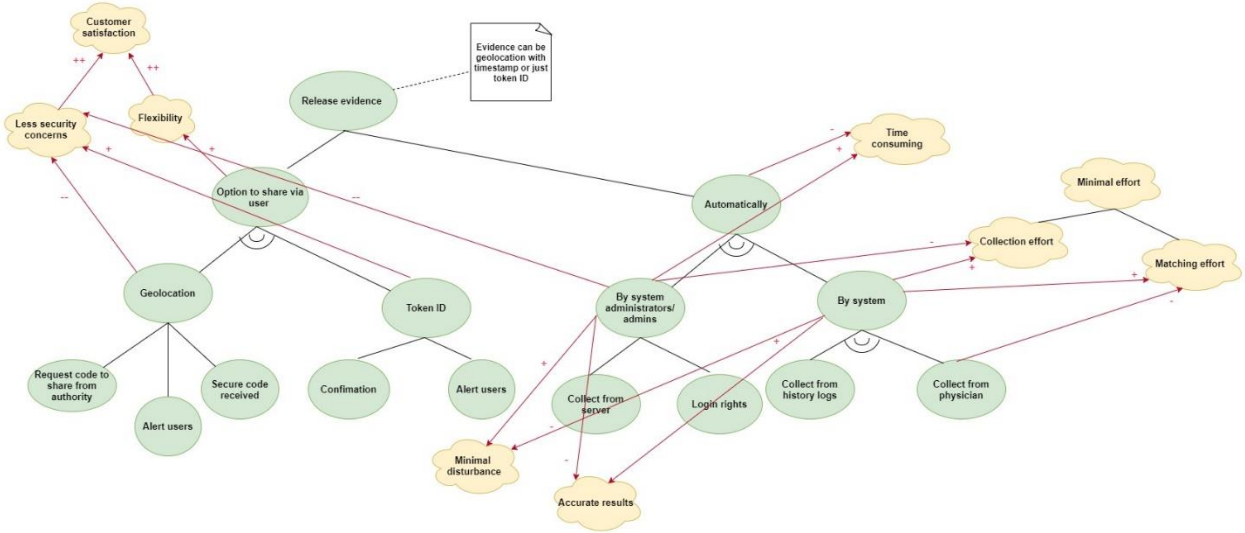

Figure 21- Goal modelling diagram three

4. This model describes how the vaccine is delivered to the health care authorities.

Assuming that scheduled time to get the vaccine for the users is associated with the health care's provided time frame.

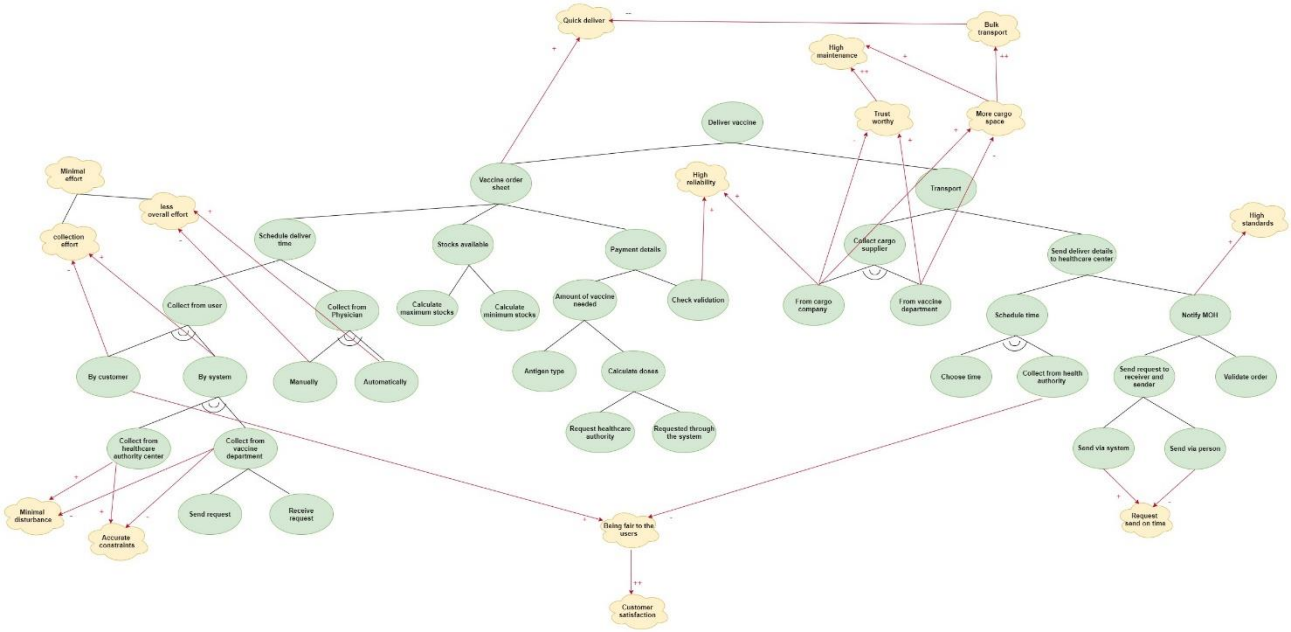

Figure 22- Goal modelling diagram Four

5. This model showcases how the soft goals that is the non-functional requirements are achieved when the system stays with the current application.

Assuming that the this model is achieved early on in the software requirement specification/ elicitation process

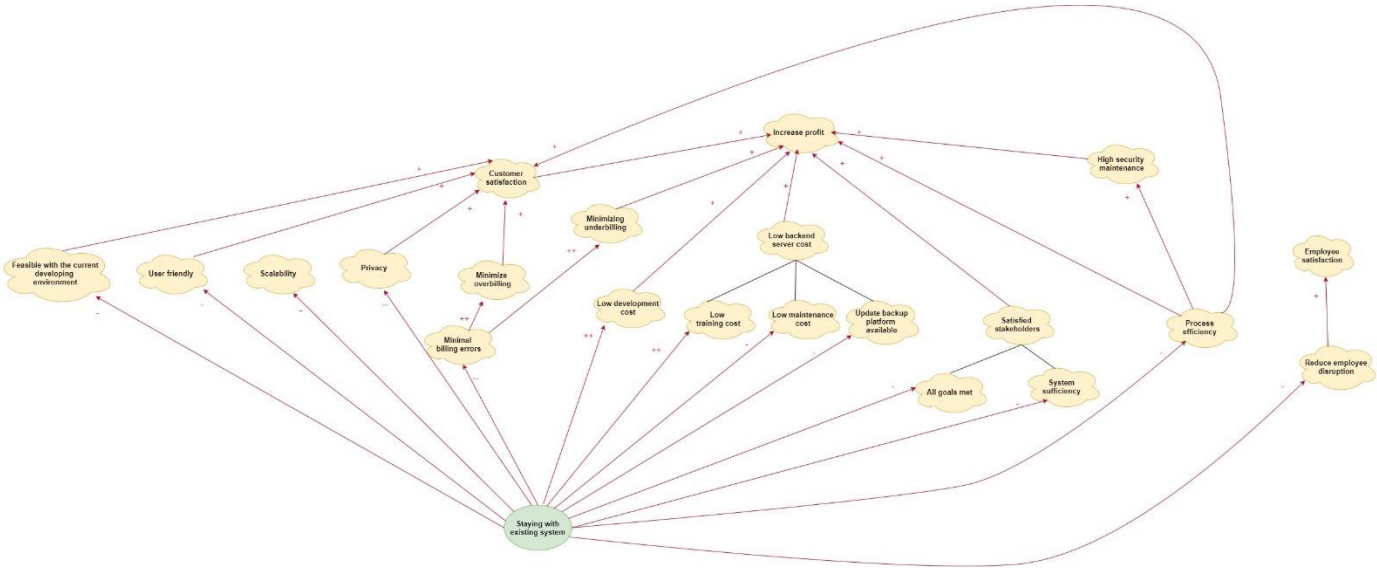

Figure 23- Goal modelling diagram Five

6. This model showcases how the soft goals that is the non-functional requirements are achieved when the system updates from its current state.

Assuming that changes are made accordingly.

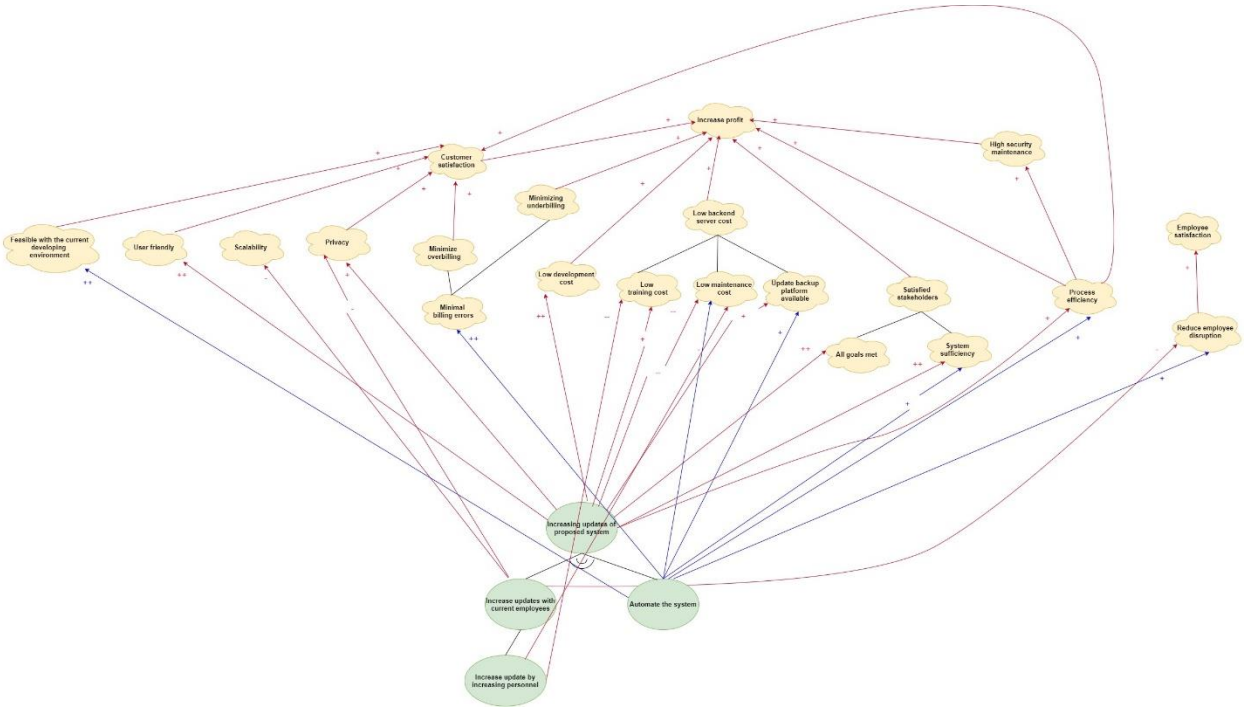

Figure 24- Final goal model diagram

i star modelling analysis

The i star modelling is associated with the enterprising level and is used to give a more depth understanding of the problem domain, its goals and the dependencies, constraints, and such.

1. The following strategic dependency model is used to show case how each requirement is dependent on each actor. Here the actors are customer, backend server and health care authority.

   Assumptions are:

   The backend server is the decentralized system itself.

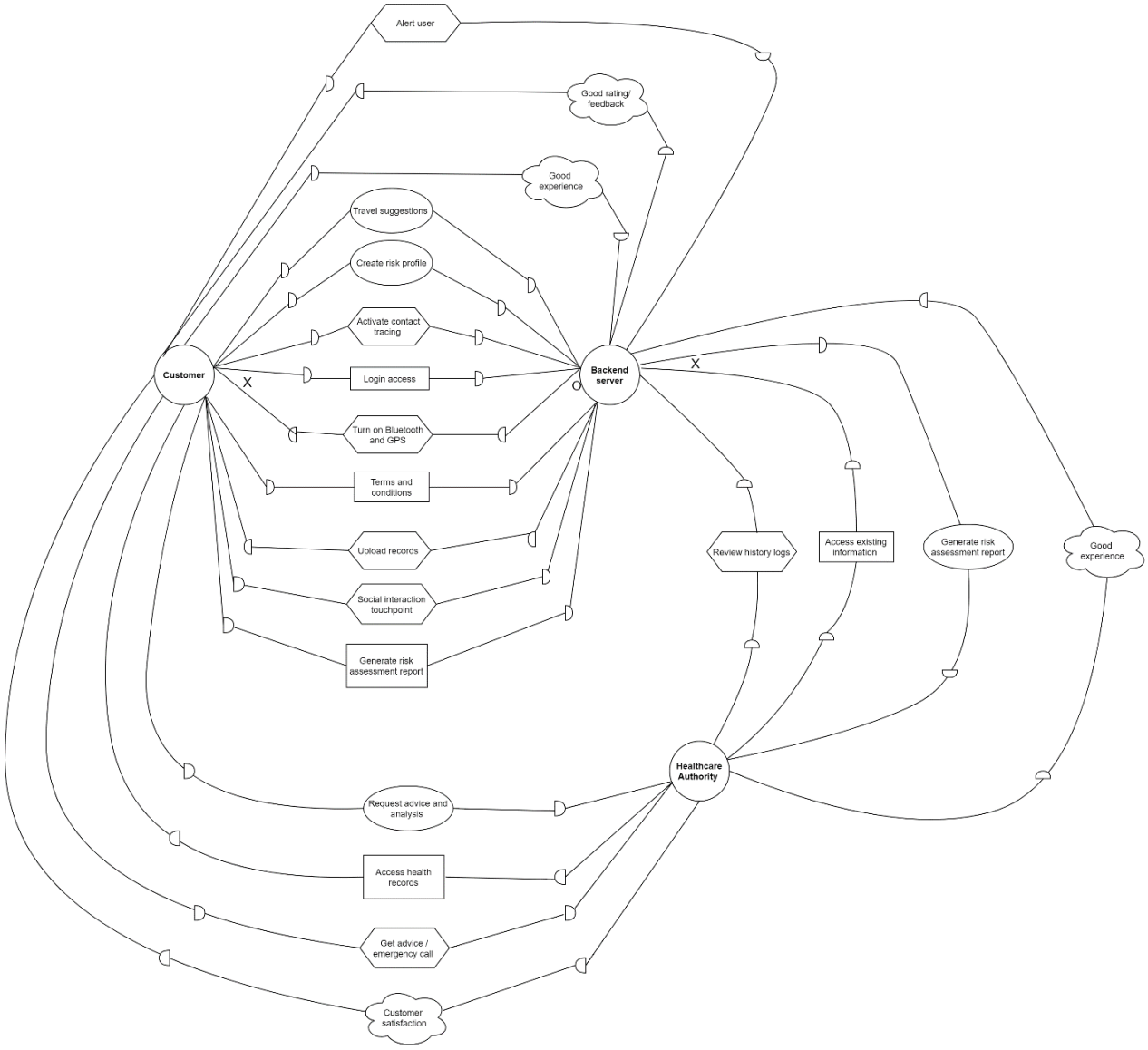

Figure 25- Strategic dependency model one

- The corresponding strategic Rationale models can be drawn for each actor boundary.
  1. Gets the customer as the boundary actor and get the rationale model of both the healthcare authority as well as the backend server.

Assumptions are:

The backend server is the decentralized system itself.

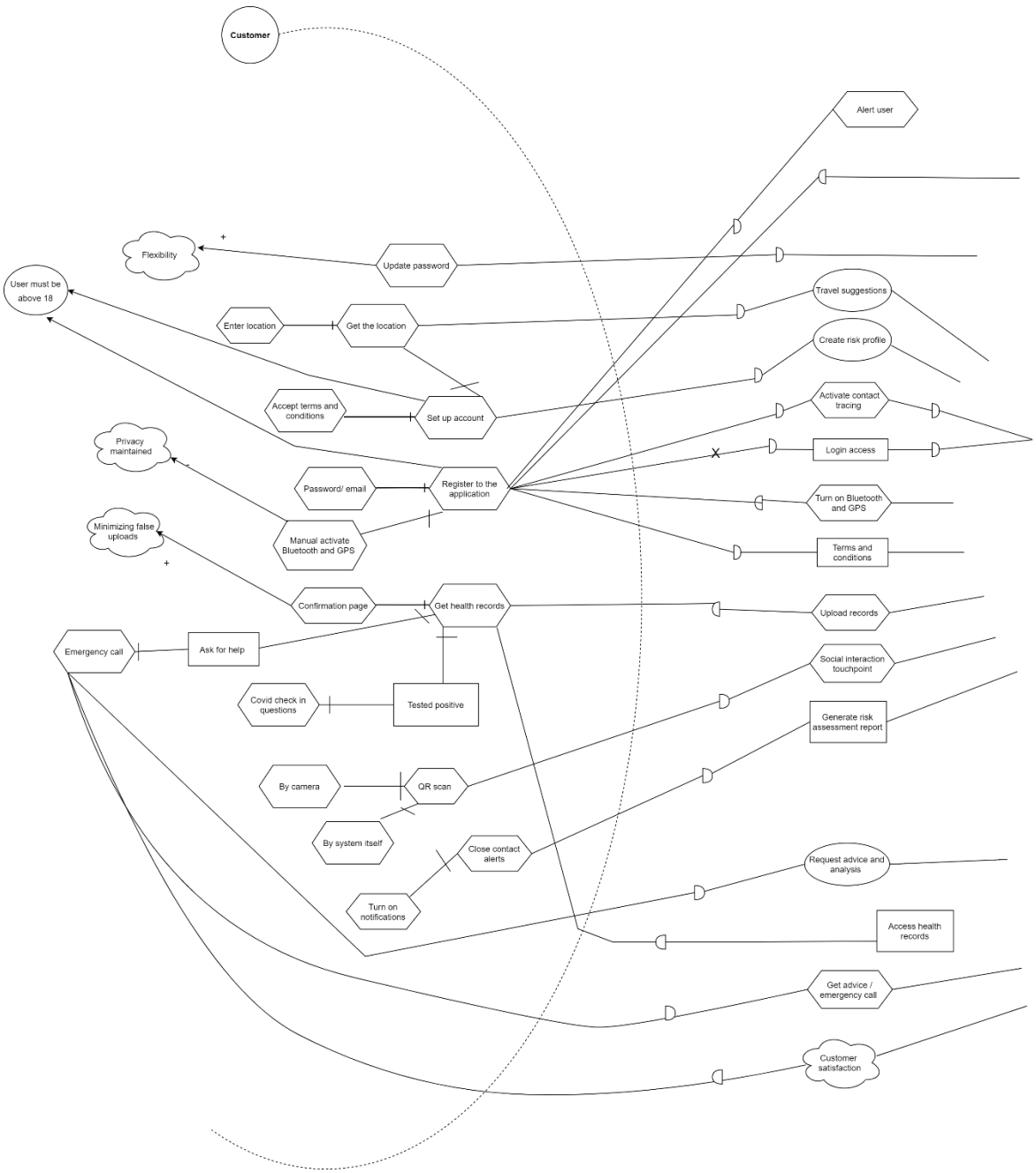

Figure 26- Strategic Rationale model where customer is the actor boundary

2. Gets the health care authority as the boundary actor and perform the analysis of tasks for customer and backend server.

Assumptions are:

The backend server is the decentralized system itself.

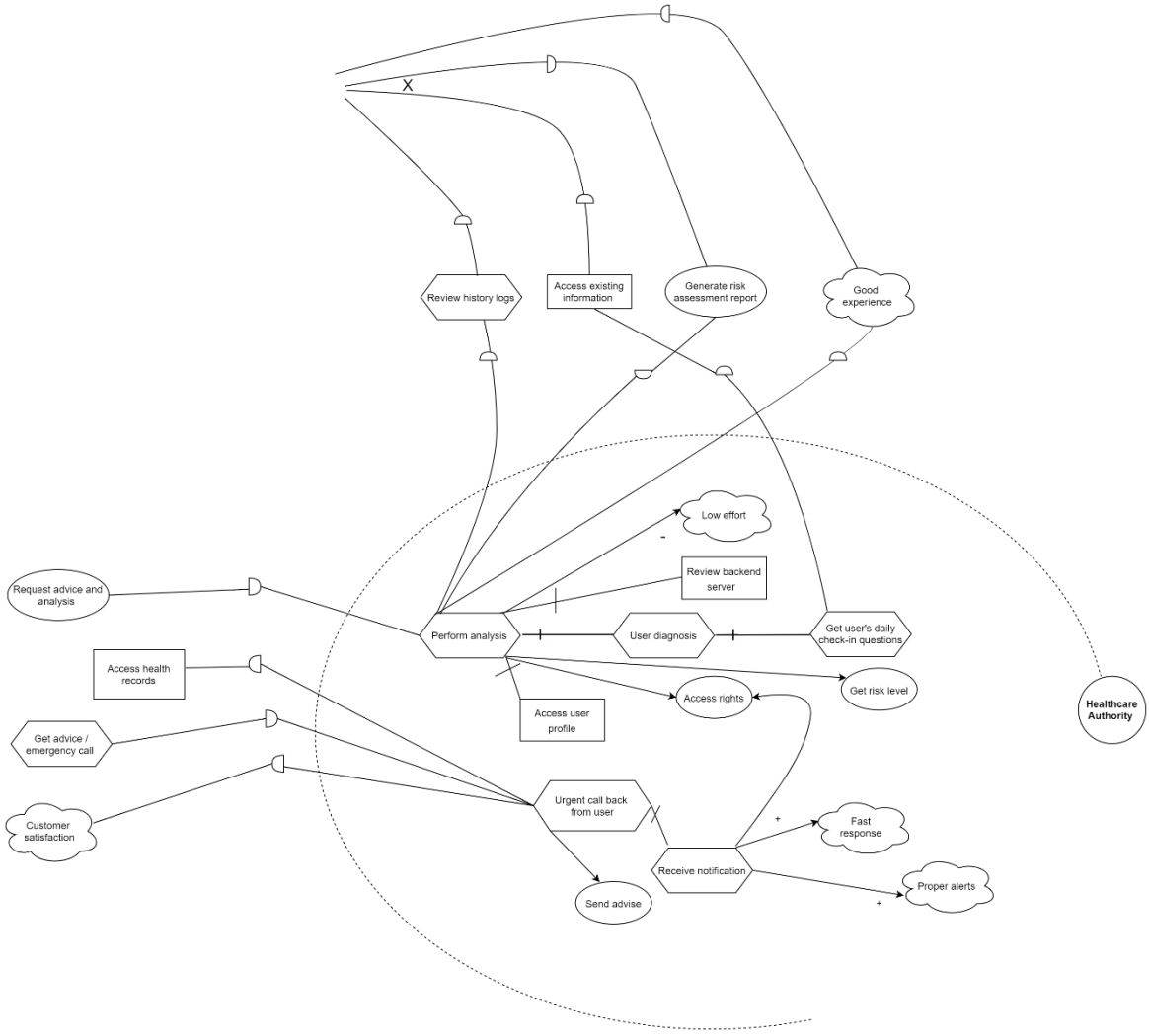

Figure 27- Strategic Rationale model where health care authority is the actor boundary

3. Gets the back-end server as the boundary actor and perform the analysis of tasks for customer and healthcare authority.

Assumptions are:

The backend server is the decentralized system itself.

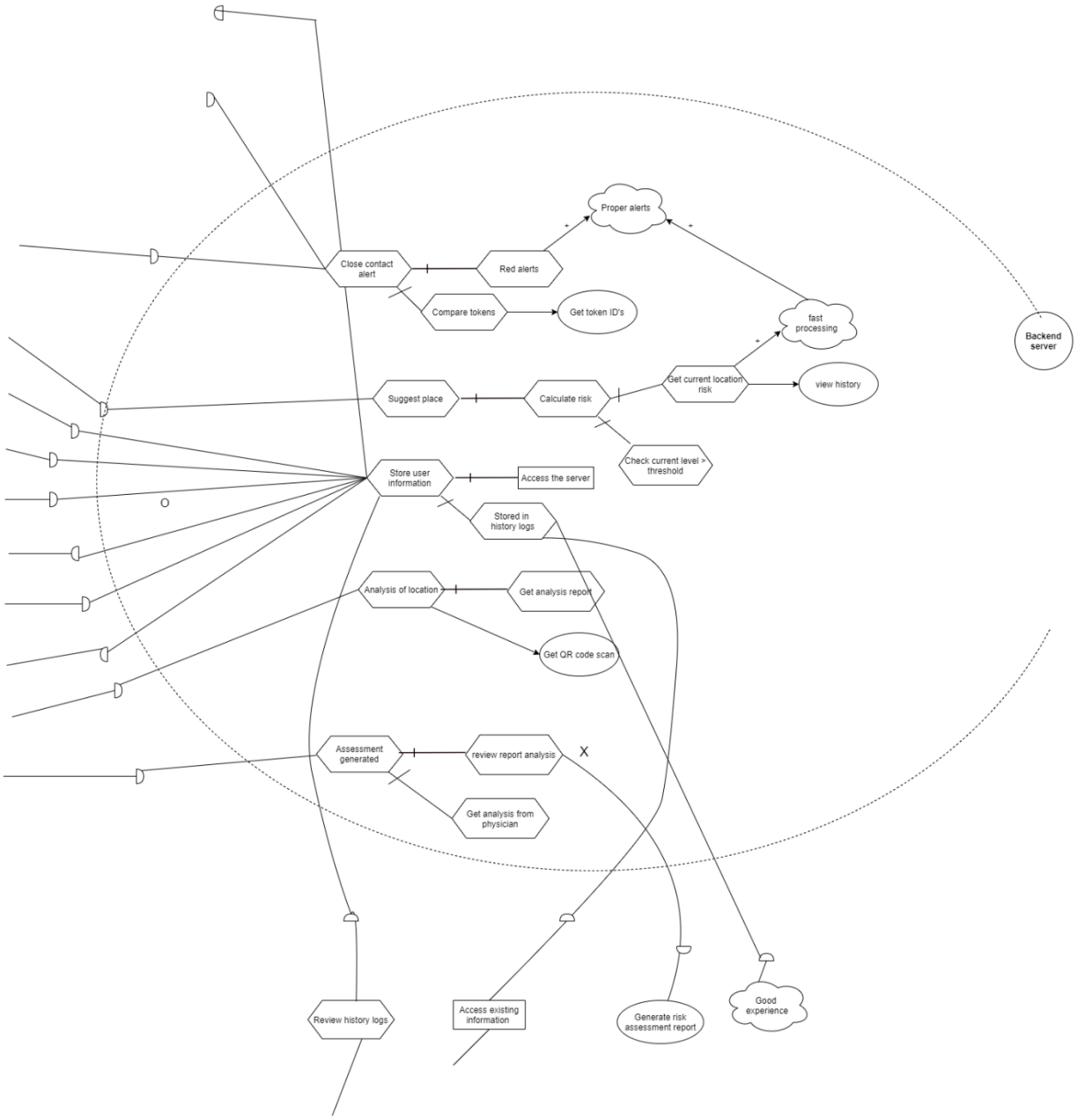

Figure 28- Strategic Rationale model where backend server is the actor boundary

2. The next strategic dependency model is used to show case how each requirement is dependent on each actor. Here the actors are customer, backend server and health care authority and MOH, Vaccine department and cargo supplier centre.

Assumptions are:

- The backend server is the decentralized system itself.
- The healthcare authority and MOH are consider one actor (MOH is included as a healthcare authority actor).

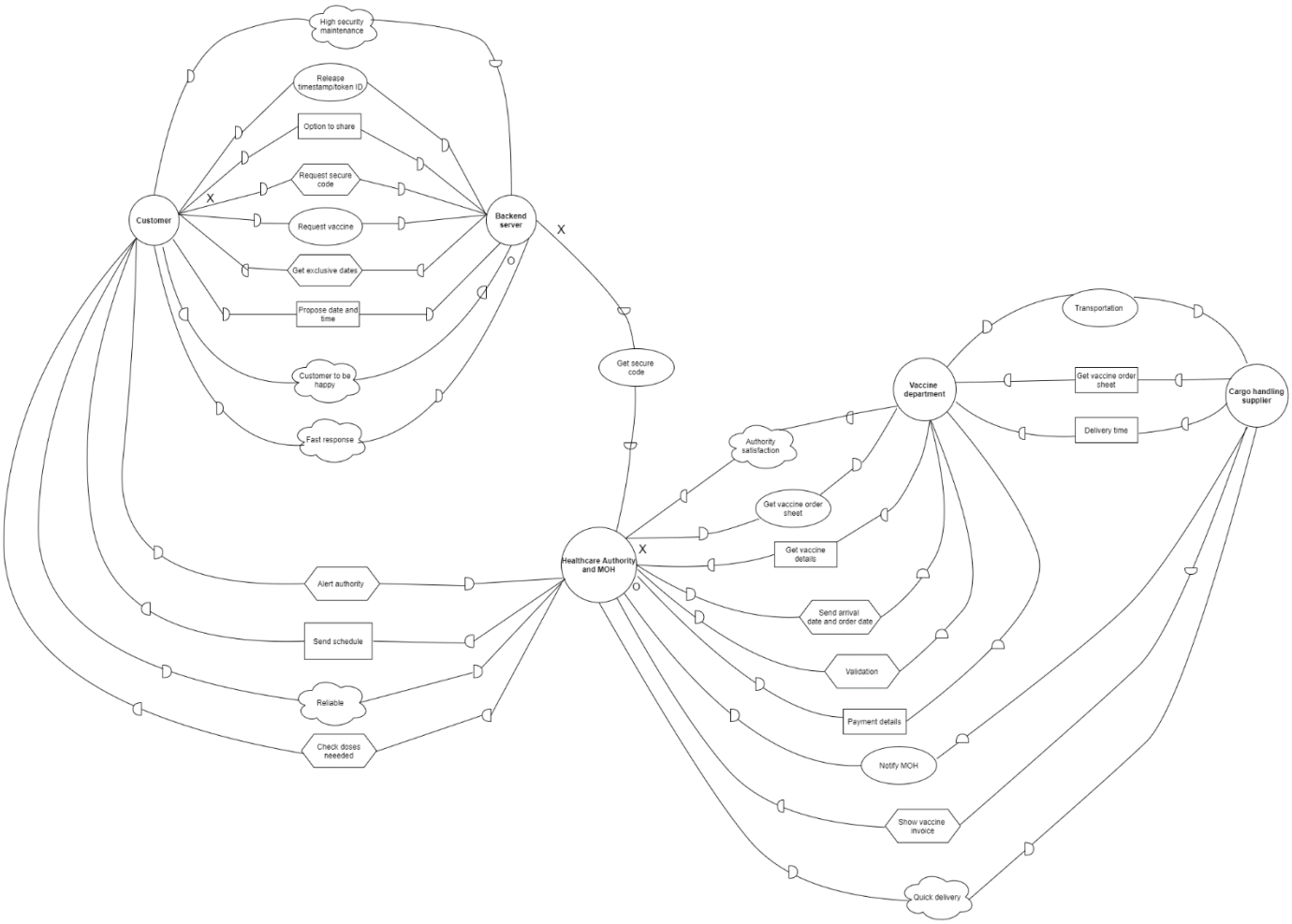

Figure 29- Strategic dependency model Two

- The corresponding strategic Rationale models can be drawn for each actor boundary.
    1. Gets the customer as the boundary actor and get the rationale model of both the healthcare authority & MOH as well as the backend server.

<u>Assumptions are:</u>

- The backend server is the decentralized system itself.
- The healthcare authority and MOH are consider one actor (MOH is included as a healthcare authority actor).

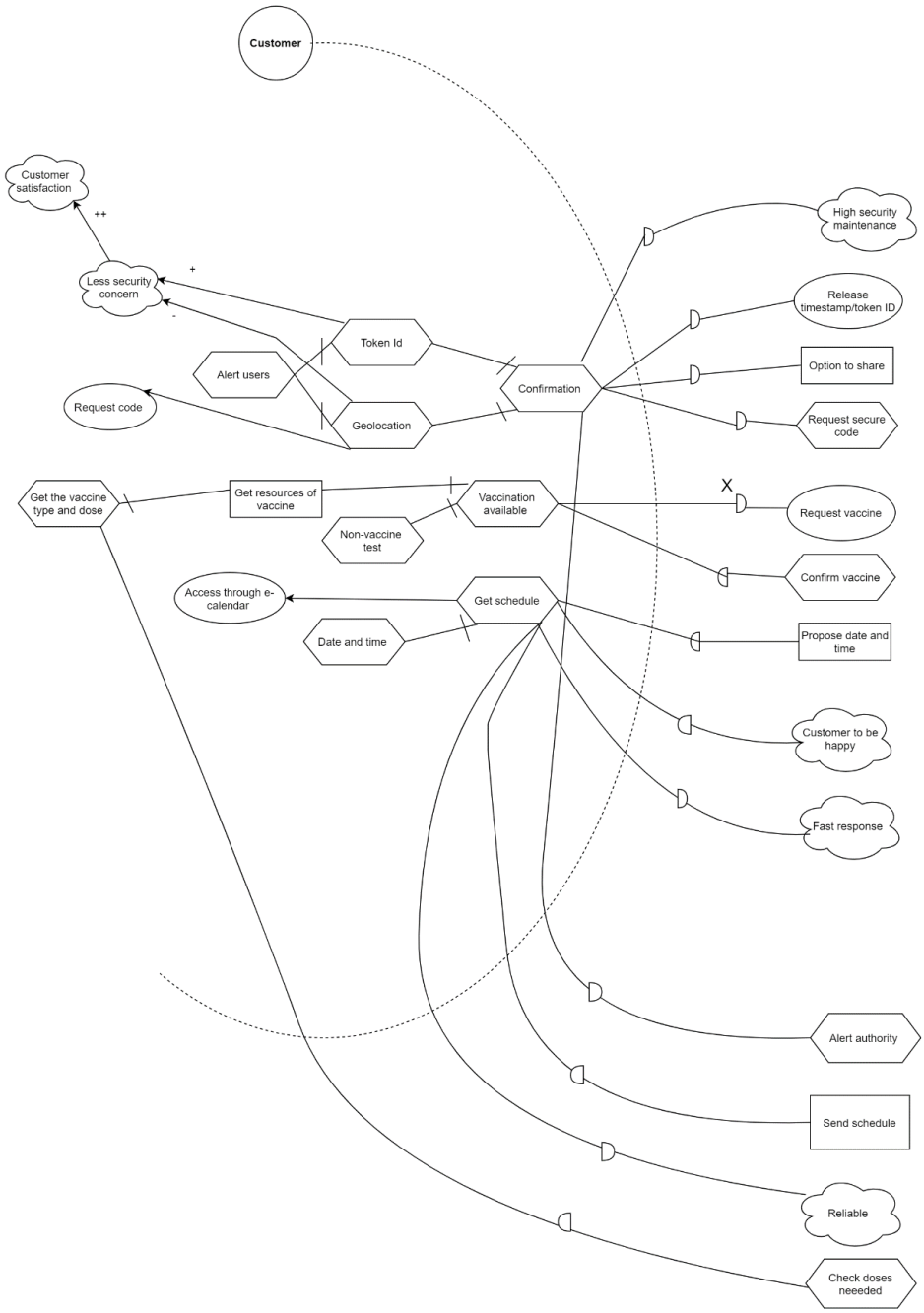

Figure 30- Strategic Rationale model where customer is the actor boundary

2. Gets the health care authority and back end server as the boundary actors and perform the analysis of tasks for customers.

Assumptions are:

The backend server is the decentralized system itself.

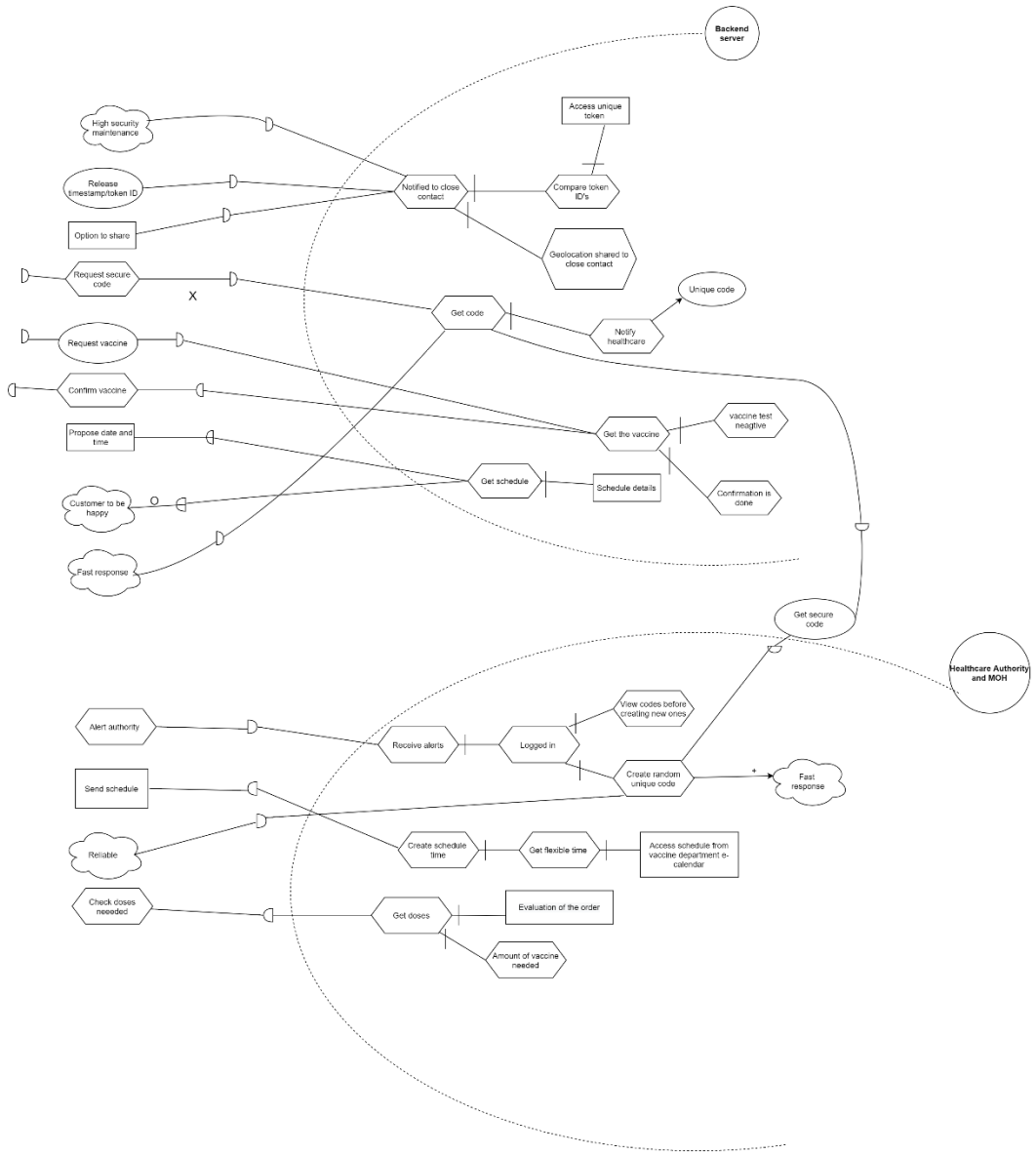

Figure 31- Strategic Rationale model where health care authority and backend server are the actor boundaries.

3. Get the health care authority as the actor boundary and perform tasks analysis for vaccine department

Assumptions are:

- The backend server is the decentralized system itself.
- The healthcare authority and MOH are consider one actor (MOH is included as a healthcare authority actor).

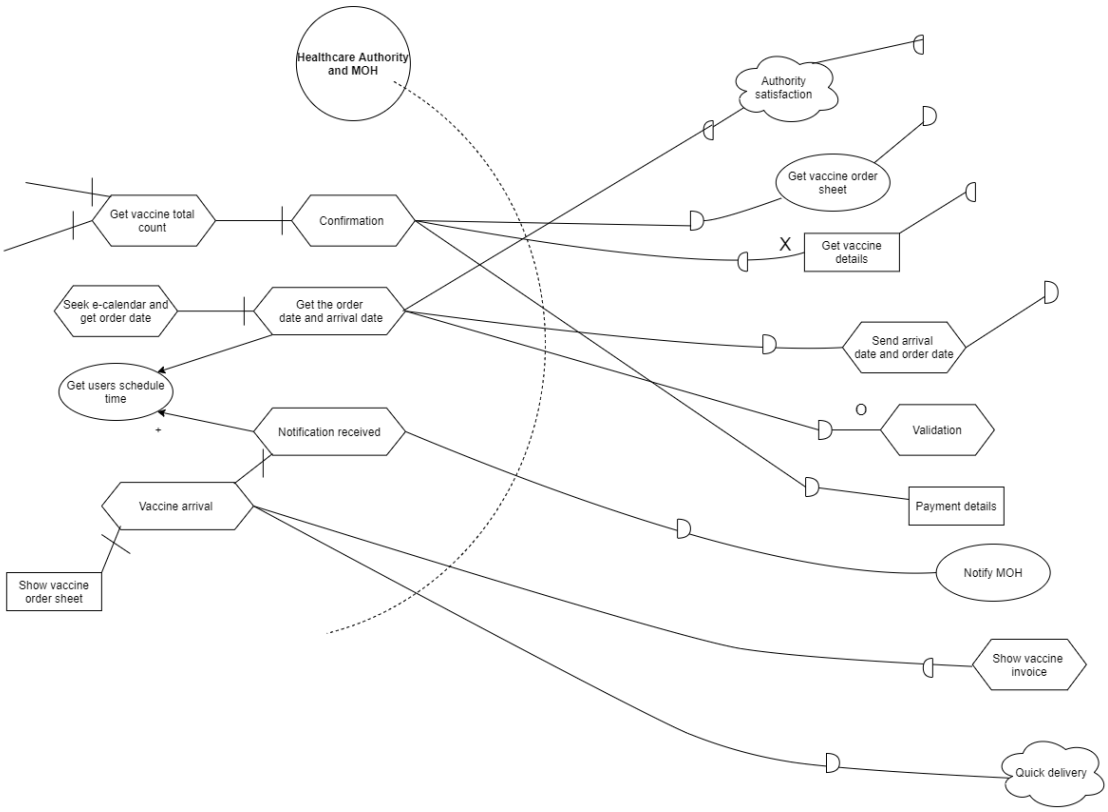

Figure 32- Strategic Rationale model where health care authority is the actor boundary

4. Get the vaccine department as the actor boundary and perform tasks analysis for health care authority.

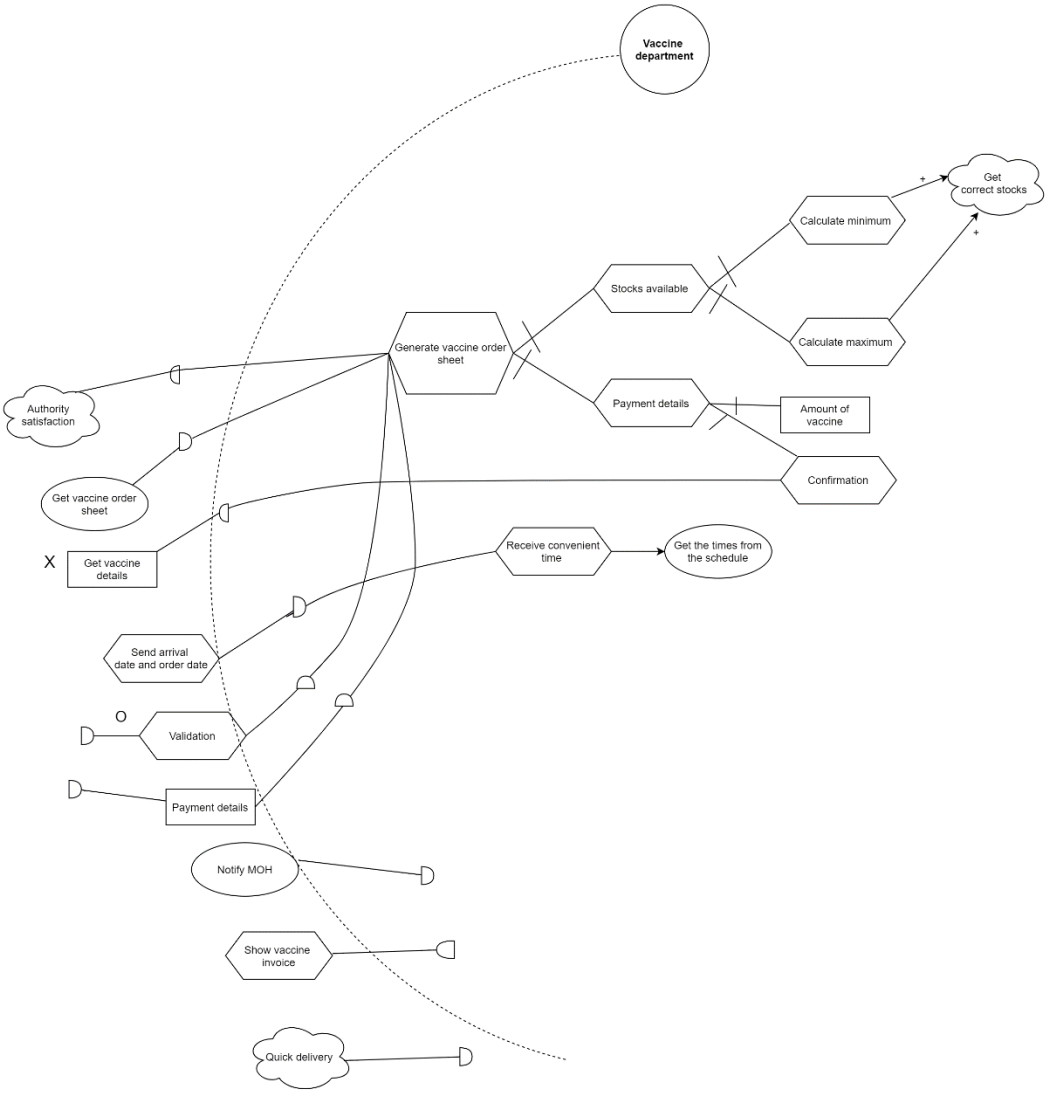

Figure 33- Strategic Rationale model where vaccine department is the actor boundary

5. The final strategic rationale model is vaccine department and the cargo handling supplier are the actor boundaries.

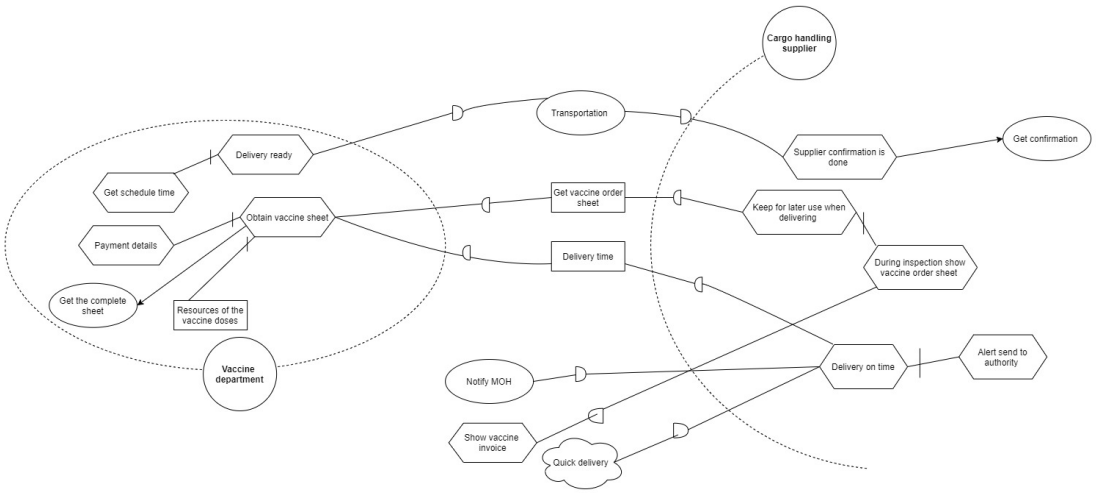

Figure 34- Strategic Rationale model where vaccine department and cargo handling supplier are the actor boundaries.