ISAFE: TRACKING COVID RISK

Assignment 1: Modelling Requirements

SEPTEMBER 16, 2021

PRAMODA GUNARATHNE
CURTIN ID: 20531218

Table of contents

1. Overview

The overall problem being addressed is the development of software that can do contact tracing to easily monitor citizens' travel locations, social interactions, risk assessment, notification, and travel suggestions after assessing risks. The problem can be addressed by using a range of elicitation methodologies to gather requirements from the perspectives of various stakeholders interested in the project to create a common aim that benefits all users. The goal of this technical report is to provide an overview of the project scope for user requirements and modeling requirements. The assumptions considered during the modeling process will be mentioned under each diagram.

2. Elicitation Techniques

Elicitation is the process of acquiring all system requirements and thoroughly understanding the concept behind the developing software from stakeholders to make the development process more reliable. System developers and the project team maintain close contact with end-users to learn more about the problem at hand, this creates a connection between stakeholders and developers early on to eventually produce successful software. This project uses a mix of methods to approach the elicitation because no single requirements elicitation method can provide all of the software needs (Yousuf and M.Asger 2015). The following methods can be used to accomplish the requirements stage, involving the establishment of the overall organizational objectives, and affirming the system requirements.

Reviewing the background information such as company reports, organization charts, policy manuals, documentation of existing systems from released Application programming interfaces that enable interoperability between Android and iOS devices. 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 if any (Mindila et al. 2019). ***Protocol analysis*** can be done attending to the problem as cognitive activities are directly verbalized for revealing interaction problems with existing systems more authenticity.

The ***literature review process*** is held as weekly workshops before the interviews to research the vastest topics in more detail for the whole elicitation period of 3 months, where during this time the members write white papers of the topics discussed that are important in the requirement evaluation process later in the development procedure. The process also considers the Internet of Things as one of the main topics that will broadly be operated throughout the development of the application (Mindila et al. 2019). It conveys the aspect of having a decentralized network as the database, considers entities of objects in the application environment such as its capabilities like contact tracing, data storage, exposure notifications, and privacy-preserving aspect.

In phase three, the project team members prepare ***surveys/questionnaires*** to be distributed in the upcoming workshops and interviews. The survey includes standardized questions allowing quick analysis and access to responses scientifically. It enables the collection of a large number of statistical data from a wide range of responses.

Focus Group workshops(JAD principals) to be carried out on different days with each involved coordinating company. Workshop with the department of the Ministry of Health (MoH), clients, and project manager on the first day, workshop 2 can be held with the relevant Public Health Authorities and the National Governors Association guided by survey questionnaires designed in phase three and creating documentation after each session. Workshop 3 can be held with the Identity Providers and service providers as a full day ***brainstorming session*** where its 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. Workshops can be held with the IT support, end-users, and maintenance companies along with the National Data Privacy and Security Legislation to address and discuss preserving security, maintaining the best interest of IT support, policy, and Regulations from companies.

Individual Interviews are conducted with the ICT manager of the technology companies and the public health Authorities where ***hard data,*** including facts, figures, survey results, marketing data, and reports analysed in the workshop stages are used for decision making more effective. Open-ended and structured interviews are carried out to gather information qualitatively which makes the analysis process more functional. Interviews that are being conducted allow for a clear business goal and shared understanding (Mindila et al. 2019), between the project team and the head departments to efficiently evaluate the social interaction touchpoints and how to upload the risk profiles with the help of health physicians. All interviews can be recorded after permission is received from the participants. ***Sampling*** can be done to determine the depth of gathering the correct amount of data using its different techniques to satisfy the requirements such as using simple sampling.

Requirement workshops with the healthcare authorities, nurses in charge, data 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 (Mulero 2015). *The* ***card sorting technique*** is used here to investigating how people organize items to create frameworks that increase the likelihood of users being able to locate items, where the cards provided to the participants contain the specific aspects of the application like the quality of the risk profiles to be raised, reliably monitoring, Bluetooth and GPS connection, privacy-preserving, scalability, energy consumption, and notification panels. The participants were then asked to fill out cards with their opinions. The participants were divided into four groups after being asked four questions. They'd next talk about their responses and pin up a summary of the results. This procedure was carried out with a total of 15-20 participants. Following the group sessions, the data were analysed in a plenary session (Mindila et al. 2019). This method is like ***meetings*** where the final feedback is received to conclude the requirement stage and agree upon the software to be made.

Observation is the process of collecting data through paying great attention, researching, monitoring, and observing human behaviour. This utilizes to research persons and other stakeholders in their environment to see how they respond to user requirements for the proposed software development, allowing accurate data recording documents visually (Mulero 2015).

3. Conclusion

The solutions and lessons learned during the elicitation and modelling stages are vividly informative for future projects. Subsequently, gathering needs from the stakeholders can be quite challenging as they could say conflicting things, and analysing these later can be problematic. Open-ended interviews are opposing to the actual interview as it feels more akin to conversation analysis. The most effective strategy is to establish a list of subjects intended to be covered during the interview without putting any constraints on the process, allowing for free (yet organized) interaction among the participants. Workshops can be used to resolve conflicts rather than to collect domain expertise. In protocol analysis, we can have a third party explain the important ways around the document reading part and can have a healthcare domain expert interview the health authorities for more accuracy of data collection(Cysneiros, Breitman, and Luiz 2016). Observation was utilized to corroborate some of the information gathered rather than as a method of acquisition. Modelling requirements can be carried out during the elicitation process and although use cases are useful for providing an overview of the entire process, dealing with details and exceptions are more time-consuming, so scenarios or sequence diagrams are more appropriate (Cysneiros, Breitman, and Luiz 2016). Stakeholders feel more comfortable validating scenario descriptions than use case models in general. It's justifiable to say that this covid-19 tracking application will consequently aid in preventing the spread of the virus if the population comes as one and install it on their devices. Because mobile phones are the way of the future, this easy-to-use application has the potential to evolve and grow into a valuable tool for all individuals across the country. It will save users' time, keep them safe, and help the planet recover from the virus faster than normal. Hence, the implementation of this application can provide betterment for the world.

4. Appendix

4.1. Functional Specification

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.

4.1.1. Functional Requirements

The software's functions – what it should do – are dealt with in functional requirements. Functional requirements are characteristics that enable a system to perform as intended. To put it another way, the system will not function unless the functional requirements are met. Product features are referred to as functional requirements, and they are centred on the needs of the user.

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.

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.

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.

There could be many more reasons as to why the user might want to deactivate this feature. But if this feature is deactivated it is quite useless to use this software, then because it is intended to track the risk assessment of covid-19 positive pass-byers as the users travel to their destinations.

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.

By capturing each user's time-stamped *GPS location*, GPS-based tracing can establish a "location trail" for them. If someone contracts COVID-19, they can disclose their location trace with the appropriate authority—a health worker, public health official, or government official. After then, the authority makes some or all the location trail available for other users to compare maintaining privacy as a core task (Vepakomma et al. 2020). Because it can capture the user's location history with a tiny quantity of data, it's easier to scale and use in areas with high data prices.

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 in close proximity to. 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 (Vepakomma et al. 2020).

If the iSafe application will integrate both approaches, allowing the user to take advantage of the benefits of both while avoiding some of the drawbacks.

Functional requirement FR_5	
Title	*Location sharing*
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.

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.

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.

Functional requirement FR_8	
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.

Functional requirement FR_9	
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.

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_10	
Title	*Login in for external entities*
Description	The doctors/physicians, system administrators, data clerks, and Identity providers should be able to login into the application.

Justification and details	This is a high-priority functionality as they are important as the end-users.
User story	As a physician, system administrator, data clerk, and identity provider, I should be able to login into the system, so I can attend to my necessary deeds within the system.

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 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 customer profiles and check their well-being before confirming if they are positive or not, so I could do my job.

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.

Functional requirement FR_13	
Title	*If tested positive*
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.

Functional requirement FR_14	
Title	*User requirement*
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.

Functional requirement FR_15	
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.

Functional requirement FR_16	
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.

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.

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.

Functional requirement FR_19	
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

Functional requirement FR_20	
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.

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.

Functional requirement FR_22	
Title	*Validation from the project managers*
Description	The project manager must be able to check that all the workstations are functioning properly.
Justification and details	As a validation from the project manager is important to acknowledge that the business goal of the application is maintained and achieved.
User story	As the project manager, I should be able to check whether all the relevant workstations of the application are working, so that I could validate the business goals are achieved.

Functional requirement FR_23	
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).

4.1.2. Non-functional Requirements

Non-functional requirements are concerned with the characteristics of the software – what it should be. The attributes or characteristics (not functions) that the system must have, are described by Non-functional requirements. They may apply to specific parts of the system, specific functions, or the entire system.

The non-functional requirements are:

i. ***Availability requirements***

1. The application should be available 24 hours a day, 7 days a week. (The application will be utilizing the decentralized cloud system to store its data.)
2. When the users enter the wrong password, the system will generate an error message.
3. The system should have enough data storage capacity for all the encrypted messages being stored.
4. The system should be compatible with both ISO and Android operating systems.

ii. ***Performance requirements***

1. The users must be able to view the follow-up instructions within 2seconds of accepting the conditions and terms and navigate on them.
2. The system must be able to scan the tokens of the users who are at a proximity of 2 meters from each other.

To determine the distance between devices, the framework employs a Received Signal Strength Indicator (RSSI). The RSSI indicates the signal's intensity.

3. The system must be able to scan the close contacts tokens within 700ms using Bluetooth and upload them to the backend within 800ms.
4. The system must be able to notify the users of close contact within 700ms of meeting them.
5. The application must be able to locate and store the location of the social interaction touchpoints within 5secs of the user's interaction periods.
6. The system must be able to calculate and compare the risk levels of the social events within 3 seconds after the user enters the location.
7. The system must be able to scan and decrypt at least 6.700.000 message tokens (approximately equalling to 70Mbyte of data being transferred) per day.
8. The system must be able to have a fast-working database (such as an oracle Rdb database).
9. The system must be able to send the security code to users within 5 seconds upon receiving records.
10. The system will provide physicians with 12 hours to report back to the user's symptom status and sent out an alert before the last hour.
11. 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).

iii. ***Usability requirements***

1. The system must allow an easy-change platform for the users to change their passwords whenever they want.
2. The system must allow infected users with a page of options where they can decide to share their geolocations and timestamp or not.
3. The system must allow the user to enter the locations to be travelled to check the risk levels and get the lowest risk level place and suggest it to the users.
4. The system must allow the user to query their history records with at most 2 mouse clicks.
5. The software must facilitate customers choosing the correct records to upload when trying to update their profiles at least 99% of the time.
6. The system can provide a bar where the application developers can receive feedback from the users about the application for further development of the software.

7. The system allows the project managers to check all the workstations without any delay.
8. The system's deactivating the account feature should not take more than 48 hours.

iv. *Reliability requirements*

1. *MTTF/MTBF (Mean Time To/Between Failure)*- 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.
2. *AVAIL (Availability)* –
    1. The system's user registration component should be available 99.99% of the time.
    2. The system's uploading status should be available 99.999% of the time.
    3. The system's ability to store the unique tokens from customers using the Received Signal Strength Indicator (RSSI) should be available 99.9999% of the time.
3. *POFOD (Probability of Failure on Demand)* –
    1. The system's notification feature should not fail more than 0.1% of the time.
    2. The system should receive the uploaded records by the respective users without failure within 0.1% of attempts.
    3. The system's travel monitoring feature should not fail more than 0.2% of attempts.
4. *ROCOF (Rate of Occurrence of Failure)*-
    1. The system must not make more than 1 incorrect alert messages every 1000 hours
    2. The system must not make more than 1 incorrect low-risk travel suggestion every 100 hours.
    3. The system must not make more than 2 incorrect direct callings to the contact tracing team when the users try to dial the public number.

v. *Security requirements*

1. The system must be able to filter out malicious uploads to the backend of the system.
2. Multiple log-in attempts to the system will produce a warning message to the screen and lock the screen for 50 seconds
3. The system must not store unencrypted passwords.
4. The system must provide unique IDs/ tokens every 24 hours to all the users.
5. For all network communications, the system must employ TLS encryption.
6. The system must be able to delete all the malicious advertising packets to prevent malicious attackers from tracking the user's location.
7. The system must make sure to maintain the confidentiality, integrity, and availability of data.

vi. *Maintenance requirements*

A new update request will be sent to the device of the user. The upgrade will address any bugs, as well as improve the system, and include new features.

vii. *Interoperability*

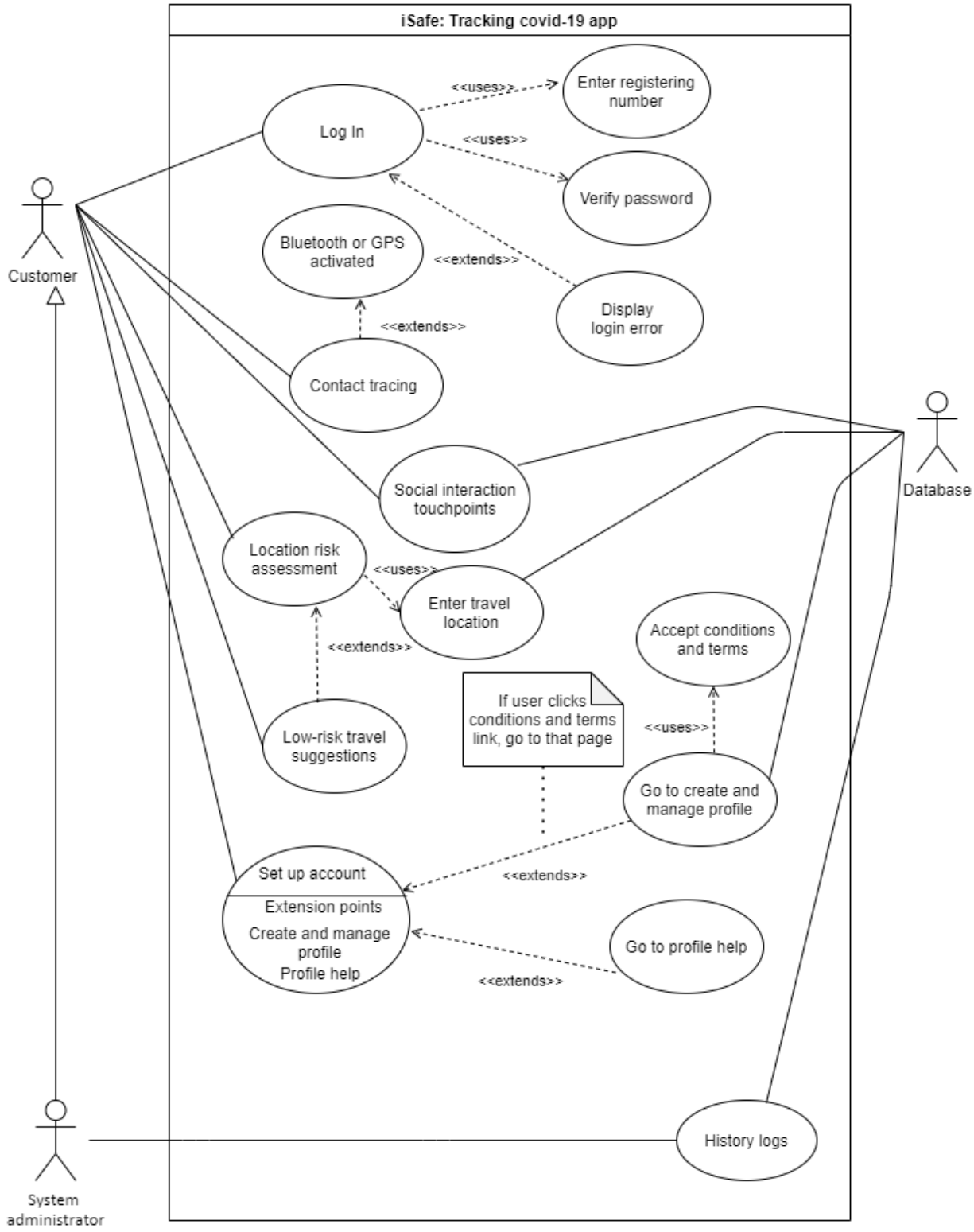
1. For database storage, the system must be able to communicate with MySQL version 8.0.
2. The application will run on Android operating systems and ISO operating systems with version 10 and more.

4.2. UML Use Case Diagrams

Use Case diagrams aid in the identification of actors and the definition of system behaviour using use cases during the requirement analysis process (Geppert and Schmid 2002). The use case diagram also helps to describe the functions from a user's perspective. Use Case diagrams can be used to describe common and variable behavioural features of a Software Product Line, as well as to model functional variability (Geppert and Schmid 2002).

4.2.1. First UML Use Case Diagram

This diagram describes the first quarter of functionalities in the application.


iSafe: Tracking covid-19 app
Log In
Enter registering number
Verify password
Bluetooth or GPS activated
Display login error
Contact tracing
Social interaction touchpoints
Location risk assessment
Enter travel location
Low-risk travel suggestions
Accept conditions and terms
If user clicks conditions and terms link, go to that page
Go to create and manage profile
Set up account
Extension points
Create and manage profile
Profile help
Go to profile help
History logs
<<uses>>
<<extends>>
Customer
Database
System administrator

Assumptions:

1. The contact tracing takes place within the application domain, hence no connection to the database.
2. Bluetooth and GPS are automatically activated after contact tracing is activated.
3. Social interaction touchpoints are the common social gathering places that the users will be most active.

Use case description 1:

Title: Setting up a customer account

Description: This use case describes a process through which the customer first registers/ set up their profiles.

Primary Actor: Customer

Preconditions: The customer is logged into the application.

Postconditions: The customer has their user account/profile activated.

Trigger: The customer clicks the activate account button.

STEPS:

1. The customer arrives at the account activation page and is asked to accept the conditions and terms to continue to the next page.
2. The customer can click the create and manage profile link.
3. The customer now can create their account by providing their email address, full name, and phone number.
4. The customer will have to accept and continue the privacy options and confirm to giving consent.
5. The customer will be asked to turn on certain setting options like turning on exposure notifications.
6. Then the customer can go back to the main menu.
7. The customer can click the profile help link to receive assistance.
8. The system stores the profile and the details in the database.

ALTERNATIVES:

3a. - The customer has provided an incorrect email address or phone number.

3a1. The system asks the customer to enter a valid email address or phone number.

3a2. The use case resumes at step 3 after providing the correct details.

4a. - The privacy consent was not accepted and pressed the continued button.

4a1. Accept the privacy consent option and continues to the next page.

EXTENSION POINTS:

STEP 1. The customer arrives at the account page.

Use case description 2:

Title: Activating contact tracing

Description: This use case describes a process through which the customer activates the contact tracing from the contact tracing tab.

Primary Actor: Customer

Preconditions: The customer is logged into the application.

Postconditions: The customer has activated the contact tracing option which alternatively activates the exposure alerts when the user comes in contact with an infected person.

Trigger: The customer enters the contact tracing tab

STEPS:

1. The customer arrives at the contact tracing page and is asked if they want to view details on how the contact tracing works.
2. IF the customers say yes to view the details, the application will redirect them to the detail page
   ELSE the customer can exit the detail page.
3. The users click the tick button next to the contact tracing option in the tab.
4. This activation automatically turns on Bluetooth and GPS.
5. The system saves this action.

ALTERNATIVES:

NONE.

EXTENSION POINTS:

STEP 1. The customer arrives at the contact tracing page.

Use case description 3:

Title: Location risk assessment

Description: This use case describes the process of how the application tracks the social event risk levels and updates the users.

Primary Actor: Customer

Preconditions: The customer is logged into the application and has turned on contact tracing.

Postconditions: The customer can receive an evaluation of the risk levels of an event before travelling to that place.

Trigger: The customer loads the risk assessment page.

STEPS:

1. The customer is asked to enter the specific location.
2. The system evaluates the risk levels and provides an analysis of the results to the users.
3. IF the risk level is higher than the threshold,
   THEN the customers can select the low-risk travel suggestion menu to find a nearby place that is of low risk.
4. But IF the risk level is lower than the threshold,
   THEN the customer gets a notification confirming that it's safe to travel to this particular place.
5. The customer can go back to the main menu.

ALTERNATIVES:

1a. - The customer has provided an incorrect location that does not exist.

3a1. The system asks the customer to enter a valid location to proceed.

3a2. The use case resumes at step 1 after providing the correct location.

EXTENSION POINTS:

STEP 1. The customer receives an accurate evaluation of the risk levels before traveling to the social event.

Use case description 4:

Title: Social interaction touchpoints

Description: This use case describes how the system stores the location of the social interaction touchpoints.

Primary Actor: Customer

Preconditions: The customer is logged into the application and has turned on contact tracing.

Postconditions: The location of the high-risked touchpoint is stored on the database and is notified to the user.

Trigger: The customer has the contact tracing turned on.

STEPS:

1. When the customer arrives at a contaminated touchpoint, the system detects it if they are logged into the application.
2. The system automatically stores the location and notifies the customers of the risk.
3. Then the application suggests advice on how to take immediate precautions to be on the safe side.
4. The users can choose to read the advice and contact a healthcare authority to ask for more accurate advice.

ALTERNATIVES:

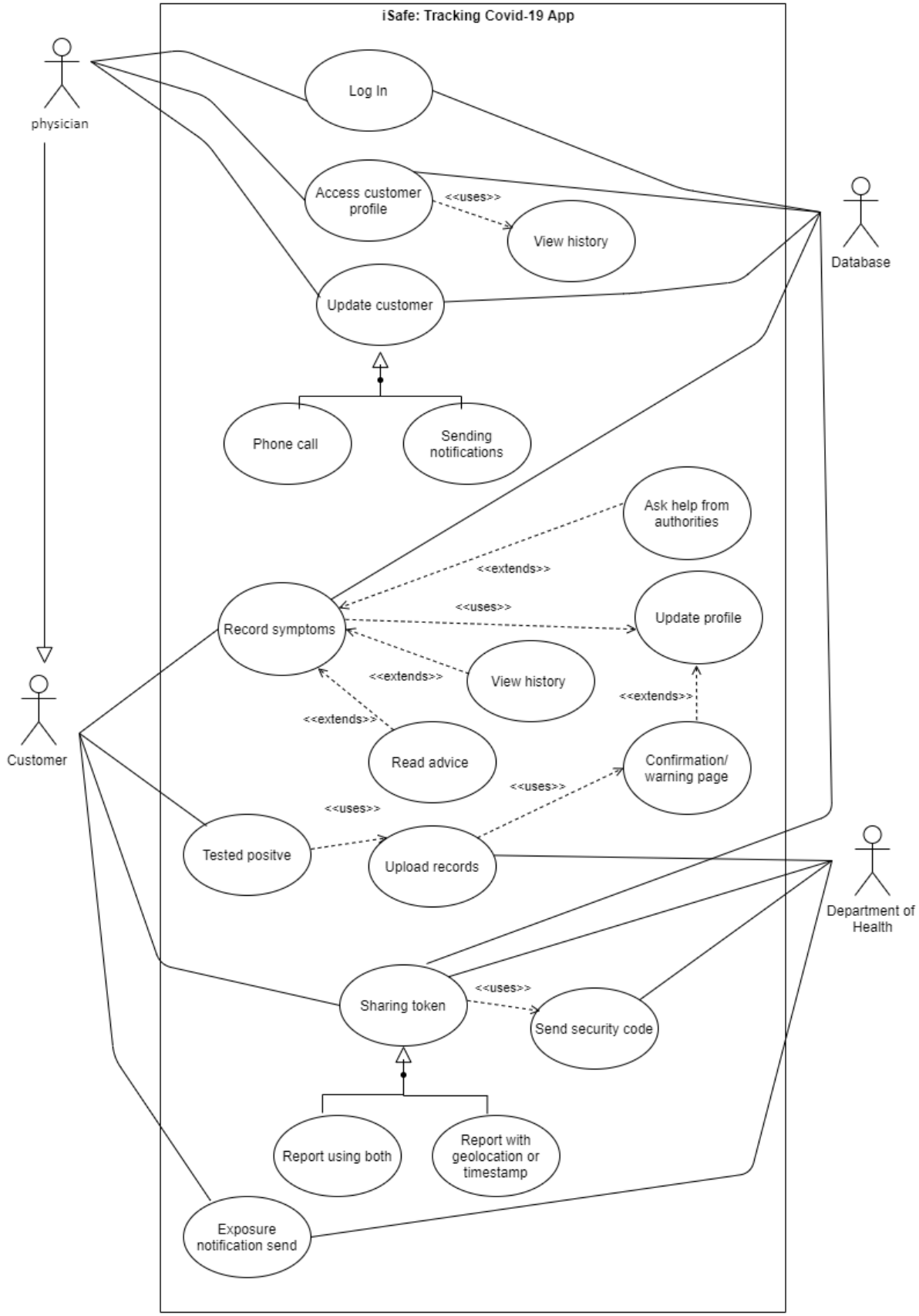
NONE.

EXTENSION POINTS:

NONE.

4.2.2. Second UML Use Case Diagram

This diagram contains the second half of the application functionality.



Assumptions:

1. Assuming that the physician can do the same functionalities as the customer from the first UML use case diagram, they are generalized from customers.

Use case Description 5:

Title: Records symptoms

Description: This use case describes a process through which a customer maintains and updates an individual risk profile.

Primary Actor: Customer

Preconditions: The customer is logged into their account and has a strong internet connection.

Postconditions: The customer has their profiles updated with their latest health concerns to keep track.

Trigger: The customer clicks the daily symptom checking menu.

STEPS:

1. The system will have the symptoms asked in multiple-choice questions and open-ended questions.
2. The system will give out a warning/ confirmation page every time the user uploads or records symptoms.
3. The system will update the current profile with the newly entered data.
4. The customers can ask for help from health authorities or read advice to be on the safe side.
5. IF the customer has tested positive according to the advice section, they will be sent a confirmation from the system after analysis by physicians.
   THEN the system will ask the user to upload the relevant records to be confirmed by an authority.
6. The system asks the customer to share their tokens with the backend to inform all the close contacts.
7. IF the customer accepts, then the healthcare department sends a security code.
   THEN the user has the freedom to select the option of sharing their last geolocation and timestamp.
   ELSE the user has selected the option of only sharing either one.
8. The system updates all the close contacts by sending them exposure notifications.

ALTERNATIVES:

1a. - The customer provides wrong symptoms

1a1. GOTO step 1.

5a. - The customer gets a warning message for uploading false or same documents.

5a1. The system asks the customers to upload the correct document again.

5a2. The use case resumes at step 5.

EXTENSION POINTS:

STEP 1. The customer links back to the multiple-choice question section.

Use case Description 6:

Title: Update customers after accessing the customer's uploaded records

Description: This use case describes a process through which the physician updates a customer after accessing their profiles.

Primary Actor: Physician

Secondary Actor: Customer

Preconditions: The physician is logged into the application.

Postconditions: The customer receives an updated confirmation about the uploaded records.

Trigger: The physician clicks the notifications that arrived at the application and enters the needed help page.

STEPS:

1. The physician opens the notification and is redirected to the needed help page where the relevant records uploaded by the customers are visible.
2. The physicians can review the past symptoms along with the current ones.
3. The physician writes out a result evaluation analysis.
4. The system asks the physician to send the evaluation to the customer.
5. IF the customer is positive the physicians can do either of the two steps below:
    1. The system will update the customers by sending out notifications of the results initiated by the physician.
    2. IF the customer has preferred phone call as the updating method

    THEN the system will provide an automatic call to the relevant customers when the physicians click the phone call button.

    ELSE send out a default notification saying they are safe from Covid-19.
6. The system stores all the current result evaluations in the database.

ALTERNATIVES:

4a. - The physician sends out the wrong result evaluation to the wrong customer

4a1. The system asks the physician to choose the correct analysis before sending it out.

4a2. The use case resumes at step 4.

EXTENSION POINTS: 43xzxx5

STEP 1. The physician accesses the needed help button to view new updates from customers.

4.2.3. The Final UML Use Case Diagram

This diagram is the final UML use case diagram that concludes the problem statement.

Assumptions:

1. Data security executive can perform the same functionalities as the customer from the first UML use case.
2. Assumes that the customer may want to deactivate the account.
3. Validation checking is done from the backend side of the system.

Use case description 7:

Title: Close contacts exposure

Description: This use case describes a process through which a customer receives notification when they come in contact with an infected person.

Primary Actor: Customer

Preconditions: The customer is logged into their account and has contact tracing turned on.

Postconditions: The customer is sent out an exposure alert.

Trigger: The customer has the application running in the background.

STEPS:

1. The system will always track down the close contacts with Covid-19.
2. The applications connect with Bluetooth and exchange the tokens of people with a proximity of 2 meters.
3. IF close contact is positive, the notifications are sent out within 2milliseconds.
   ELSE phone calls will be received by the public authority.
4. Users can read advice to take precautions.
5. The system will store all the unique tokens in the database.
6. Users can receive the latest updates of Covid-19 if they click the additional news page.

ALTERNATIVES:

2a. – Due to some reason Bluetooth cuts off and no tokens are exchanged.

2a1. The customers fail to receive notifications and the application sends out a notification asking the user to turn on Bluetooth again.

2a2. GOTO step 1.

EXTENSION POINTS:

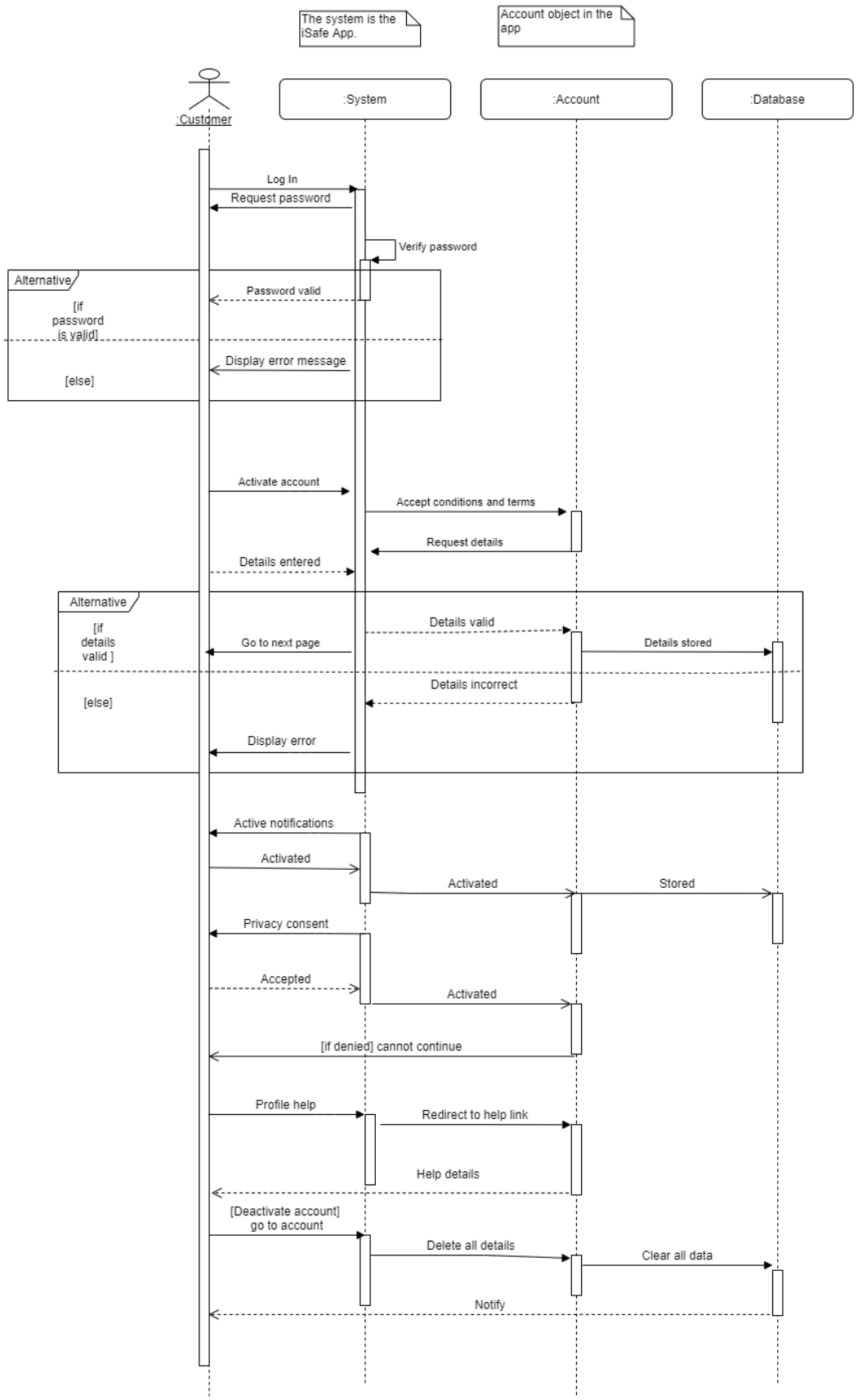
STEP 6. Customers can receive the latest updates on Covid-19.

4.3. UML Sequence Diagrams

Sequence diagrams describe what's involved in a use case step by step, where it is a type of interaction diagram that emphasizes the order in which communications are delivered. Sequence diagrams show how actors interact with use case descriptions in detail, and they can also be used to depict external business processes that the iSafe system will support.

4.3.1. First use case description's Sequence Diagram

This is the first use case description's sequence diagram showing the interaction between the customer and system over time when setting up an account.

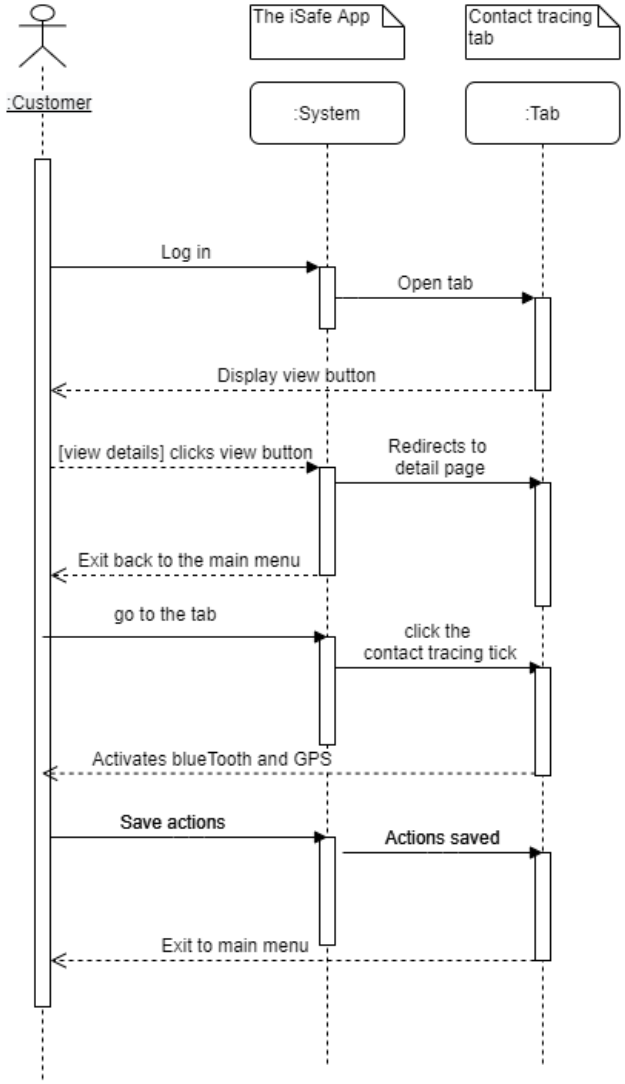

The system is the iSafe App.
Account object in the app
:Customer
:System
:Account
:Database
Log In
Request password
Verify password
Alternative
[if password is valid]
Password valid
[else]
Display error message
Activate account
Accept conditions and terms
Request details
Details entered
Alternative
[if details valid]
Details valid
Go to next page
Details stored
[else]
Details incorrect
Display error
Active notifications
Activated
Activated
Stored
Privacy consent
Accepted
Activated
[if denied] cannot continue
Profile help
Redirect to help link
Help details
[Deactivate account] go to account
Delete all details
Clear all data
Notify

Assumptions:

1. Anything that is returning from an object is denoted by dash lines. And IF ELSE conditions are written inside alternative frames for broader understanding.
2. Notes were written at the top of two objects just to make it easier for the stakeholders to understand.
3. Considering that the user wants to delete the account, the last interaction shows this process.

4.3.2. Second use case description's Sequence Diagram

This diagram shows the interaction between the customer activating the contact tracing tab and what happens over time during this period.

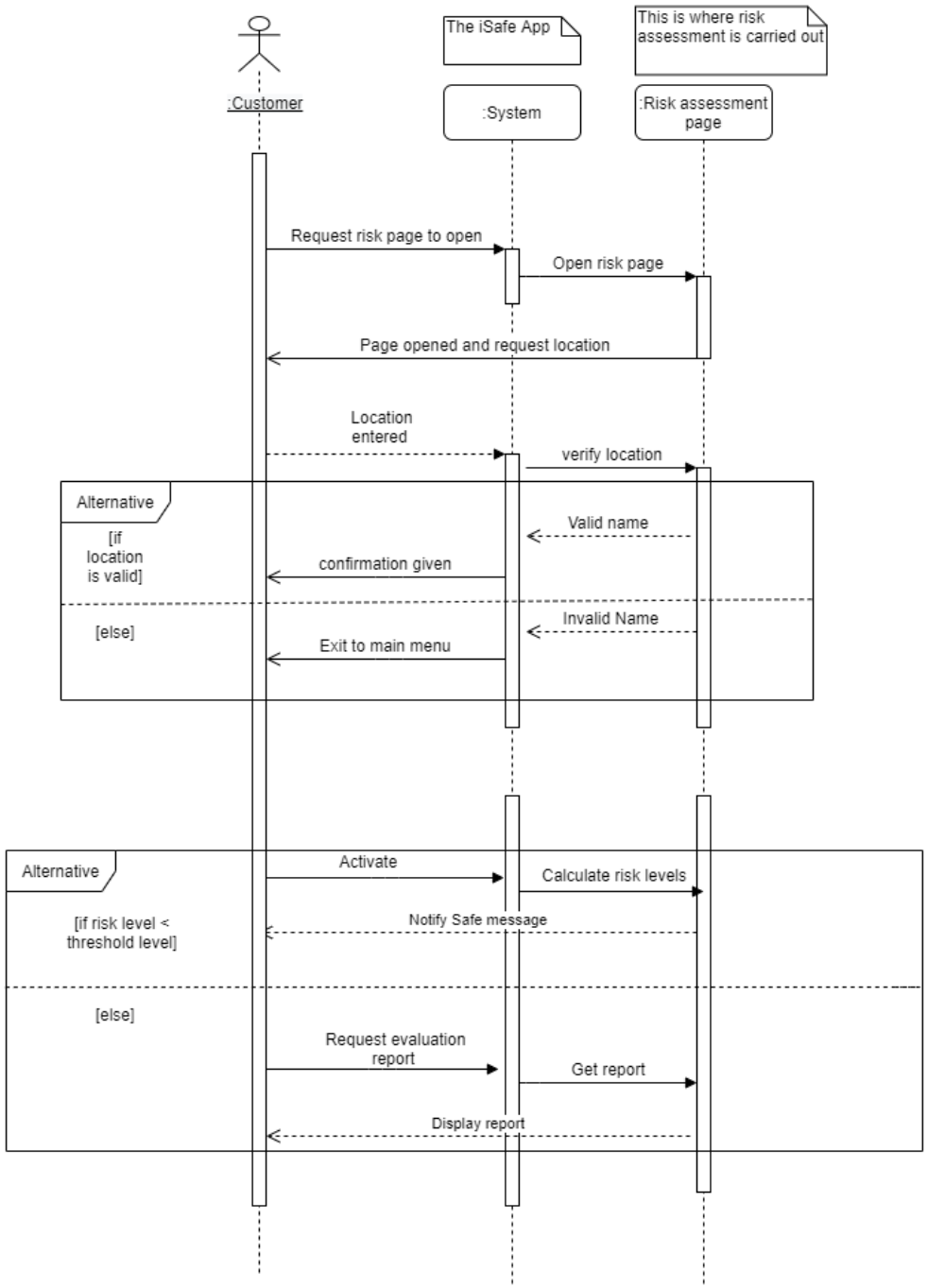


Assumptions:

1. [View details] is the condition if the user wants to view the details so they can click the view detail button to learn about the contact tracing software being used
2. Exit to the main menu means the user immediately goes back to the main page after activating contact tracing.
3. Assuming that the contact tracing tab cannot be open directly without accessing the system first, which might seem obvious but subtle.

4.3.3. Third use case description's Sequence Diagram

This diagram demonstrates the interaction of the customer with the risk levels when a specific location of a social event is given.

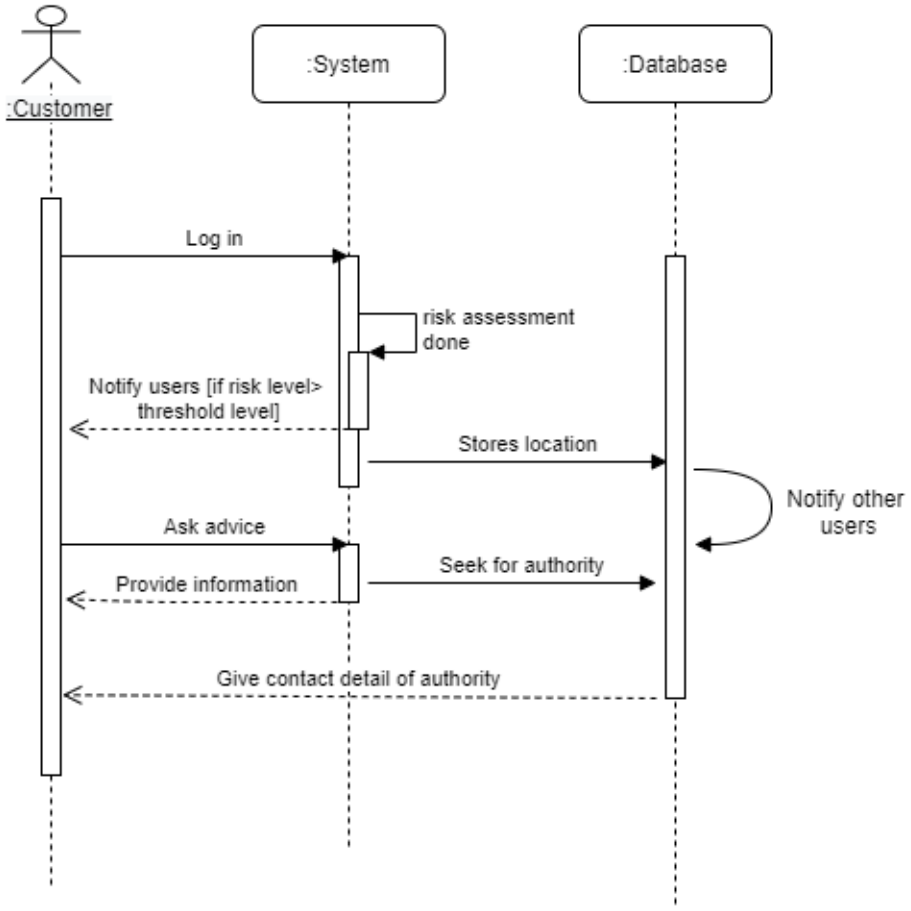


Assumptions:

1. Once the customer enters an invalid location the system exits to the main menu where the customer can enter the valid location again.
2. The risk assessment page is where the risk assessments are calculated.

4.3.4. Fourth use case description's Sequence Diagram

This diagram provides how the interaction between the customer and a contaminated social touchpoint is recorded in the application using risk assessment.



Assumptions:

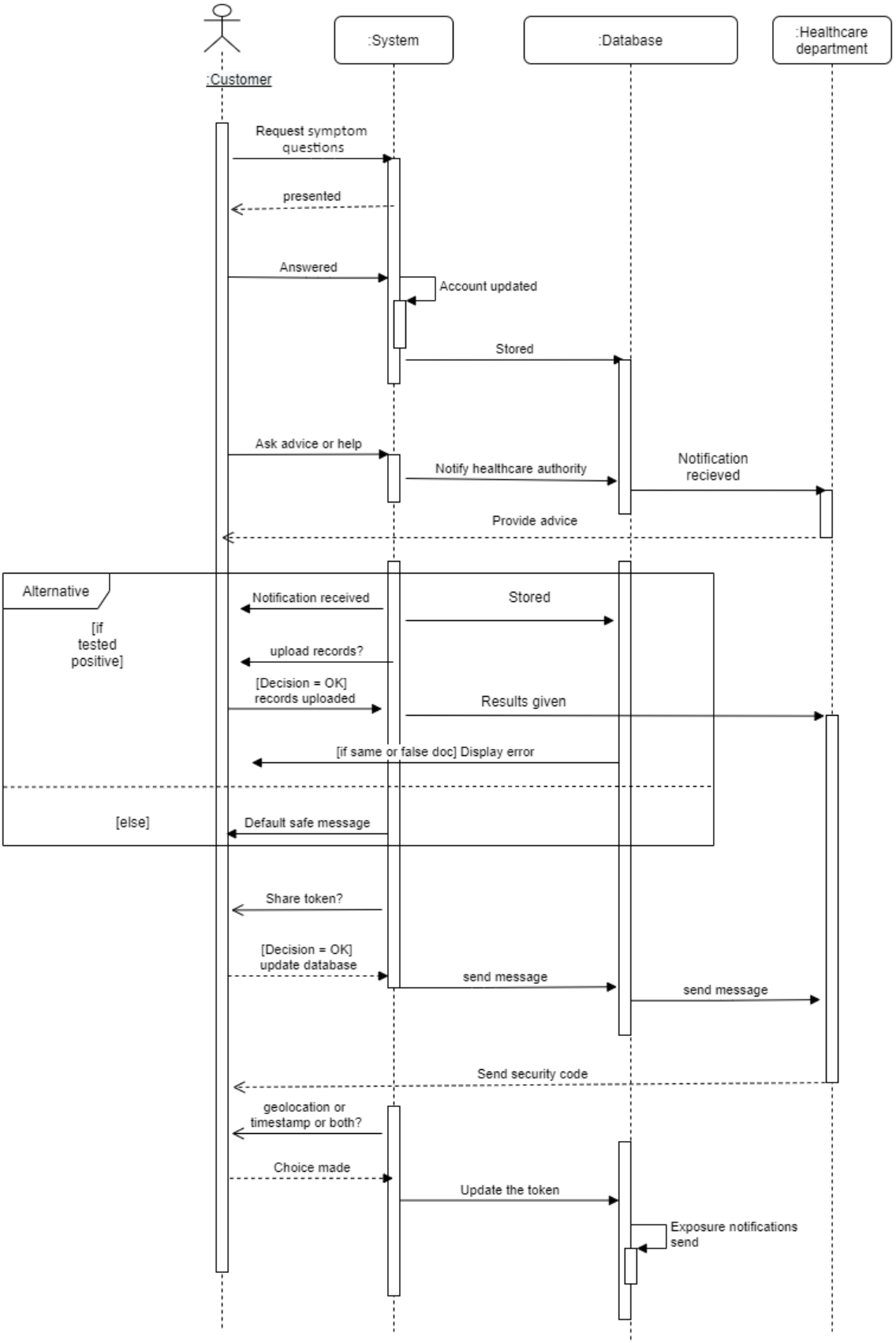
1. Once the users ask for advice from the application the system will provide information and seek for an authorities' contact detail to be provided to the customer.
2. Social touchpoints could be any triggering social place, such as a bank, ATM, private transport, and shopping malls.
3. The database stores the location in the database to inform other contacts of the possible risk here.

4.3.5. Fifth and sixth use case descriptions' Sequence Diagram.

This diagram shows the interaction of customers, system, database, and healthcare department for example (the doctor, physician) over time when the customer tests positive for covid-19.

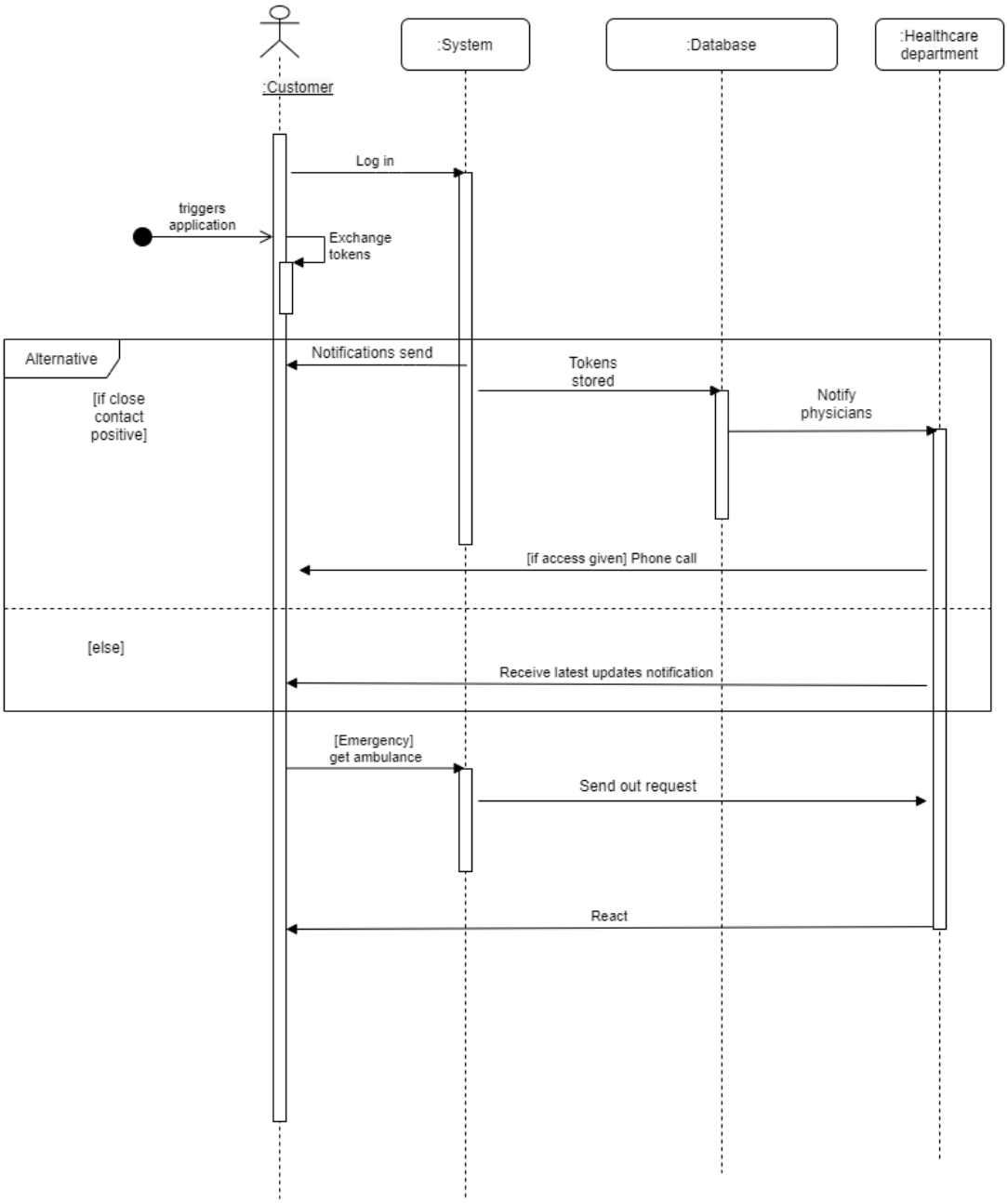
Assumptions:

1. The fifth and sixth use cases are combined and drawn in one sequence diagram.
2. The customer decides to share the token hence the message is sent to the health department from the database.
3. The choice where the user is asked to share geolocation or timestamp. Here, it is considered that the user's choice was both.


: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

4.3.6. Seventh use case description's Sequence Diagram

This diagram is about the interaction between the system, database, and healthcare department over time when a customer comes in close contact with a covid-19 patient.



Assumptions:

1. Notifications/ alerts are sent immediately after the system detects positive close contact people.
2. Doctors/ contact tracing teams can call customers if they have activated that option from their end.

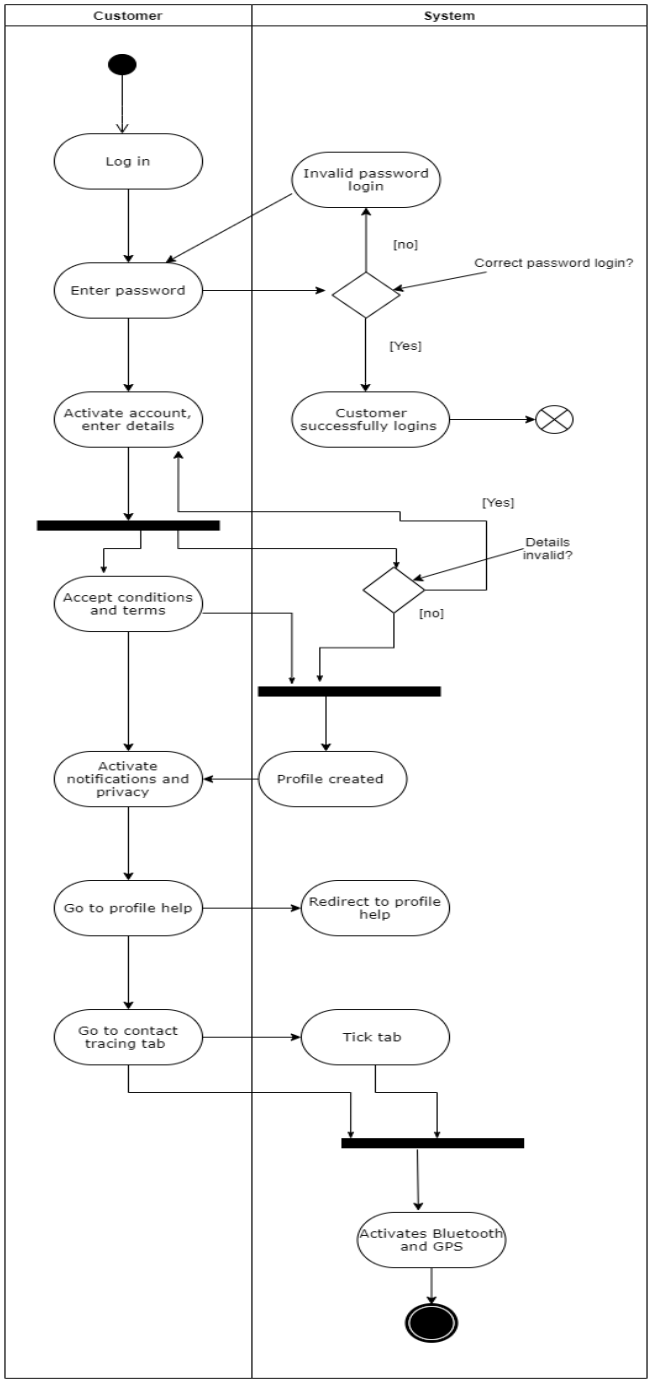
There are many more sequential diagrams that can be drawn for each use case. However, the main diagrams are drawn above for the main principal scenarios of the application.

4.4. UML Activity Diagrams

Activity diagrams are a type of diagram that is used to depict procedural logic, business processes, and workflow pointing out the dynamic aspects of the system. The following activity diagrams explain the workflow from the conceptual perspective.

4.4.1. First Activity Diagram

The following Activity diagram is a combined form of use case descriptions 1 and 2, where the customers first verify the login in detail and set up their risk profile to get the maximum out of the application. This also contains the activation of contact tracing steps. All the steps are demonstrated as a procedural workflow.

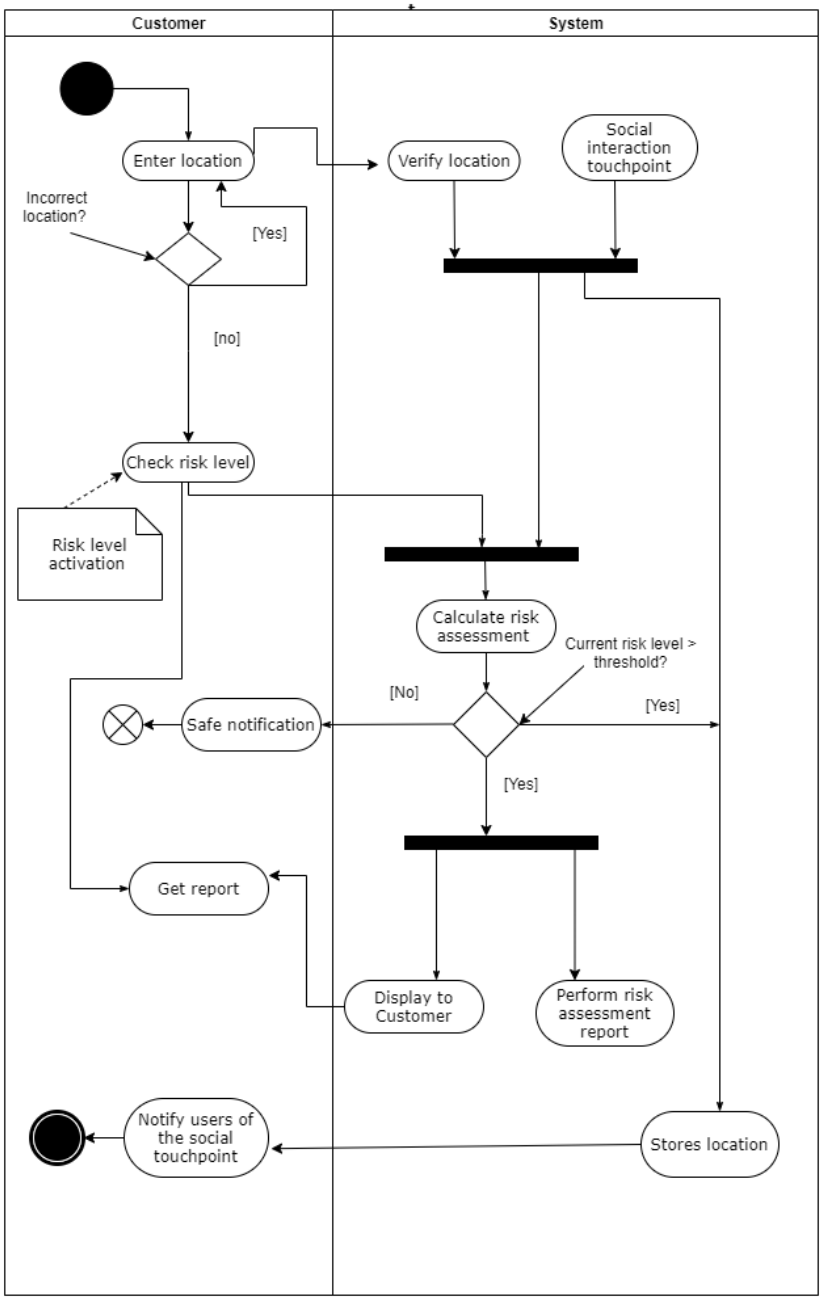


Assumption:

Assuming that the system is an actor that performs elementary behaviours.

4.4.2. Second Activity Diagram

This diagram is a combination of use case descriptions 3 and 4, where the customer checks the risk assessment level of a social event before travelling and the system automatically checks the risk assessment level of a social touchpoint and notify users, this workflow is shown in a more logically and dynamic manner in the following diagram.

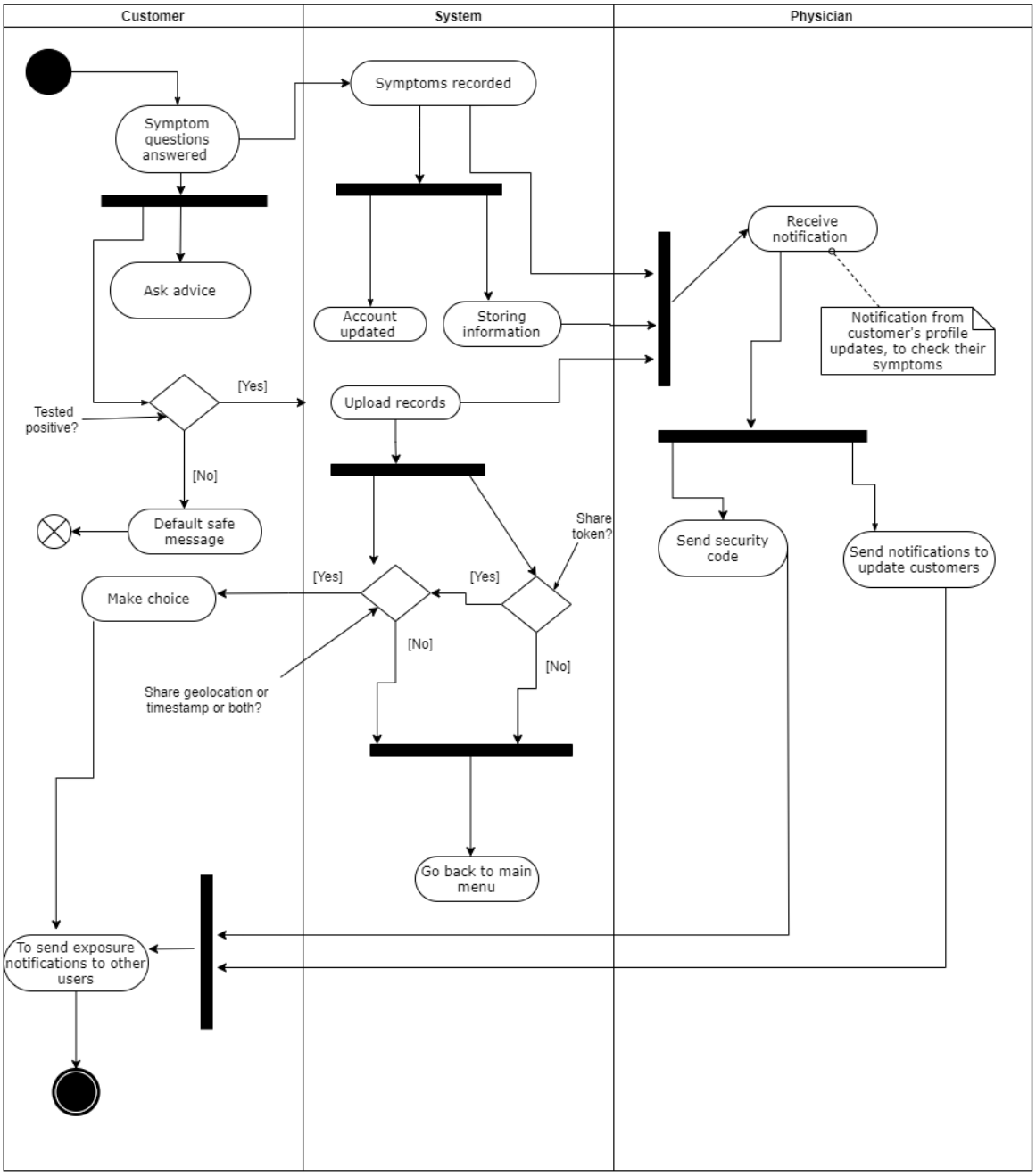


Assumptions:

1. Assuming that the system is an actor that performs elementary behaviours.

4.4.3. Third Activity Diagram

This diagram shows the logical flow of events when the customer's risk profile shows, they are positive.

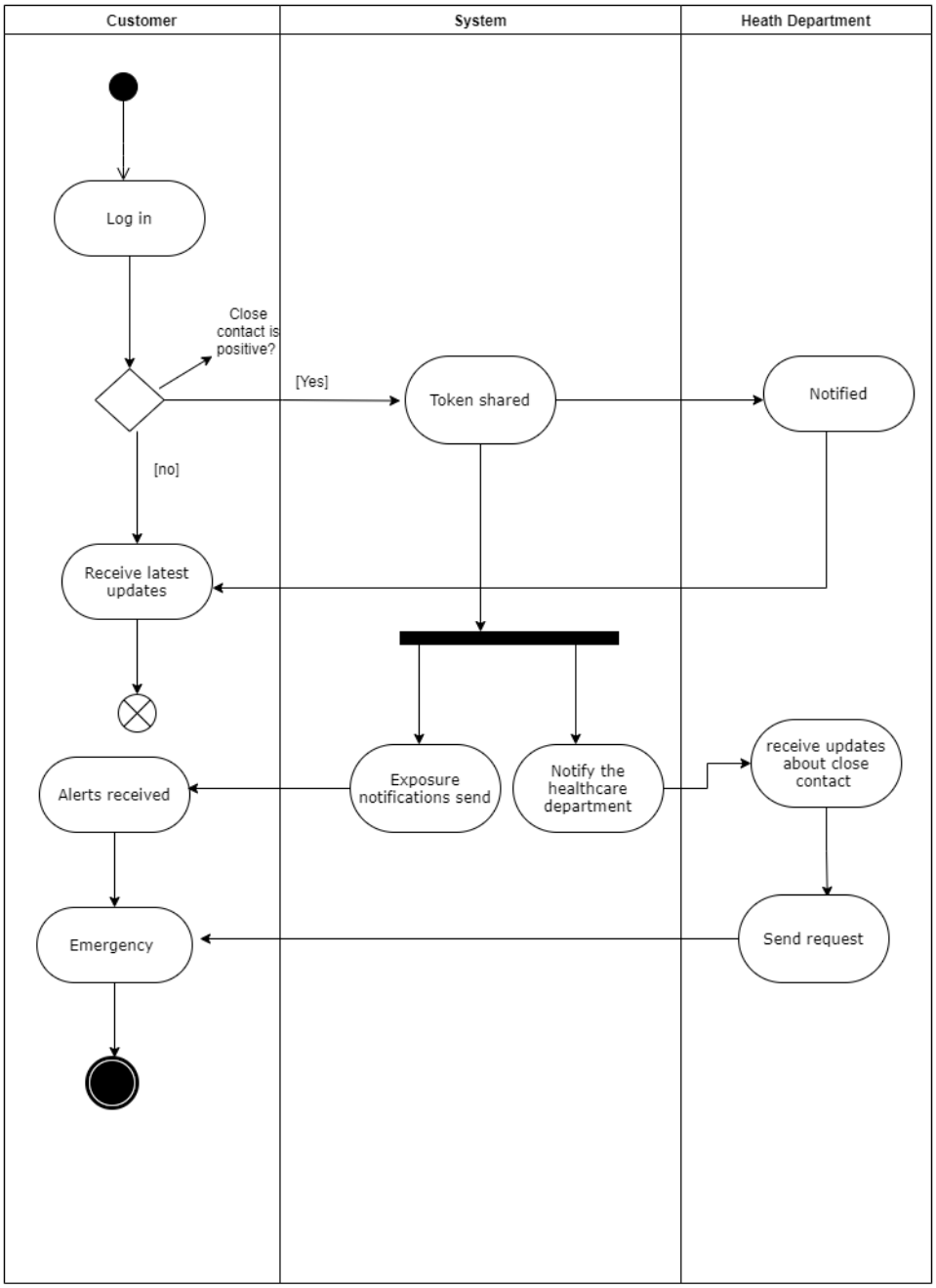


Assumptions:

1. Considering that make choice activity is sharing both geolocation and timestamp both.
2. The system is an actor that is involved in dynamic modelling.

4.4.4. Final Activity Diagram

This diagram shows the logical flow of the events when the customer encounters an infected person.



Assumptions:

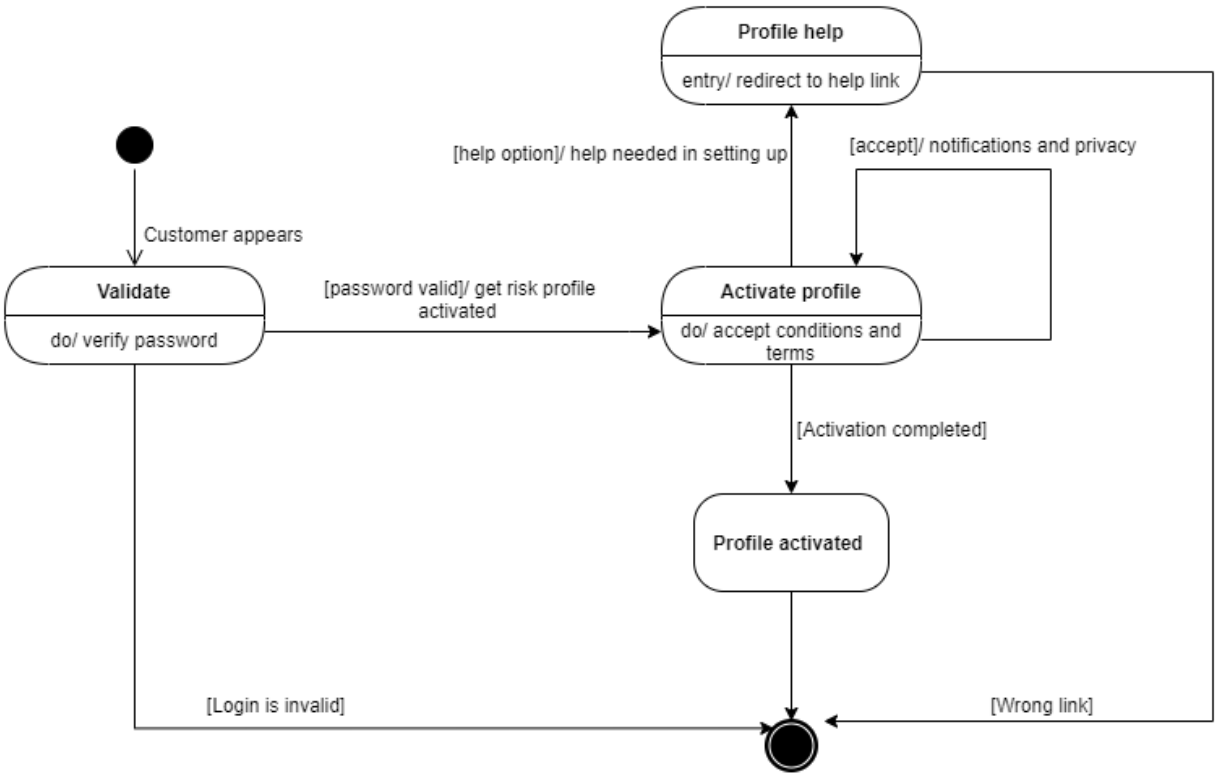
1. When tokens are shared the health department is automatically notified.
2. The system and the healthcare department are considered to perform elementary behaviours.

4.5. UML State Machine Diagrams

State Machine Diagrams depict the various states of an entity and are used in dynamic modelling . State machine diagrams can also be used to depict how an entity responds to different events by changing states.

4.5.1. First State Diagram

This diagram shows the state of the object when the customer is going to create the risk profile. It shows that validation process should be valid in order to activate the risk profile



Assumptions:

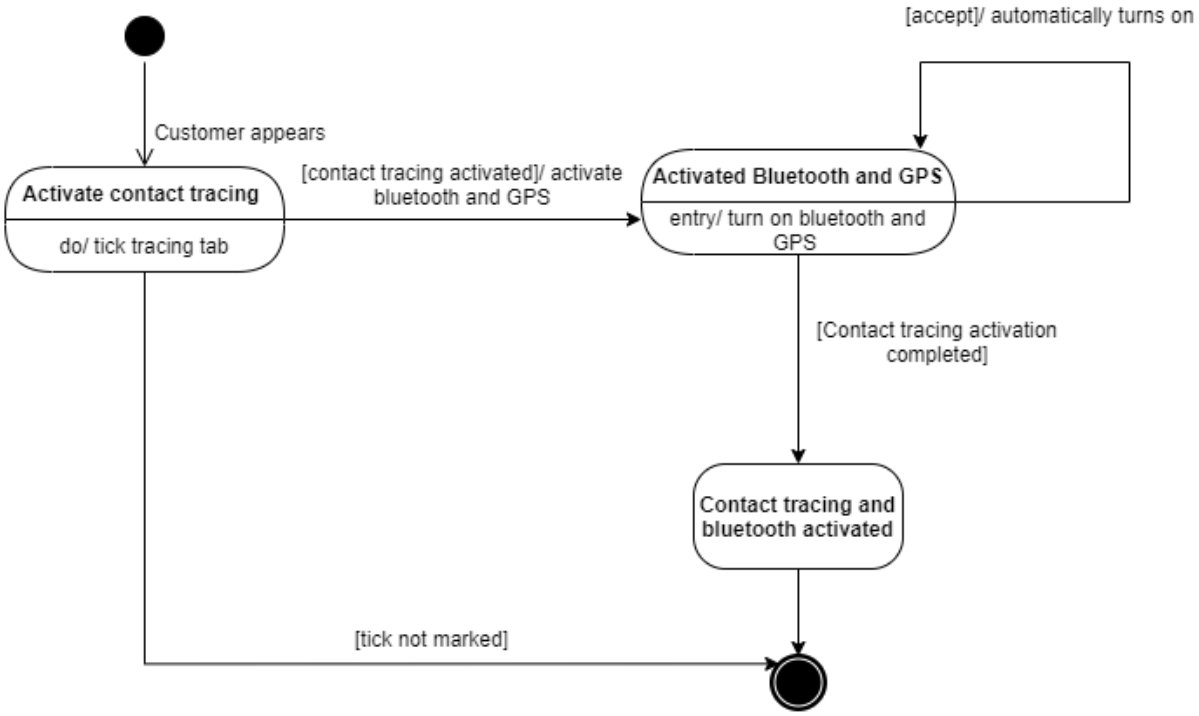
During the process, the customer appears at the beginning and after the process the customer has an activated profile, considering it is their first time creating one.

4.5.2. Second State Diagram

The following diagram gives the states of the objects when the customer activated the contact tracing option by ticking the option. The objects are not in many states because the activation process does not require many states or objects.

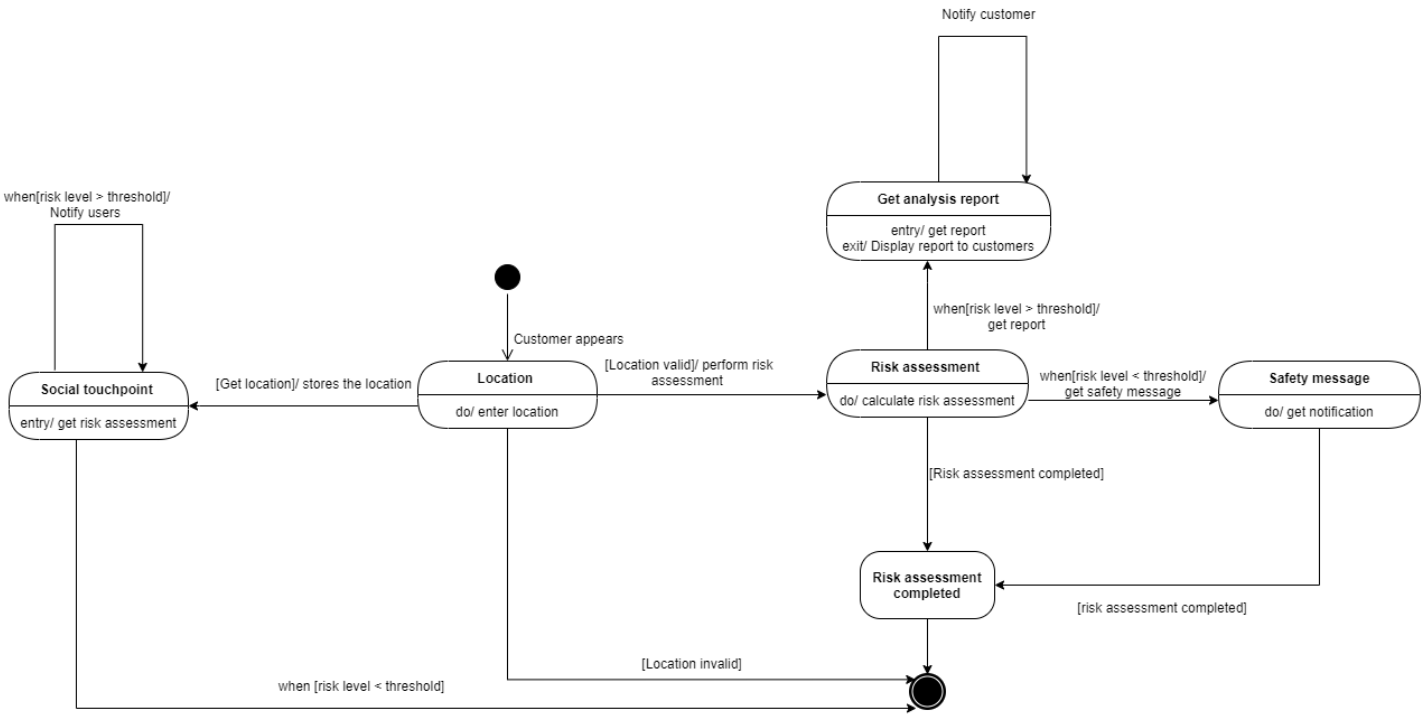
Assumptions:

1. The customer initiates the process.
2. GPS is always turned on, but Bluetooth will be automatically turned on after activating the contact tracing option.



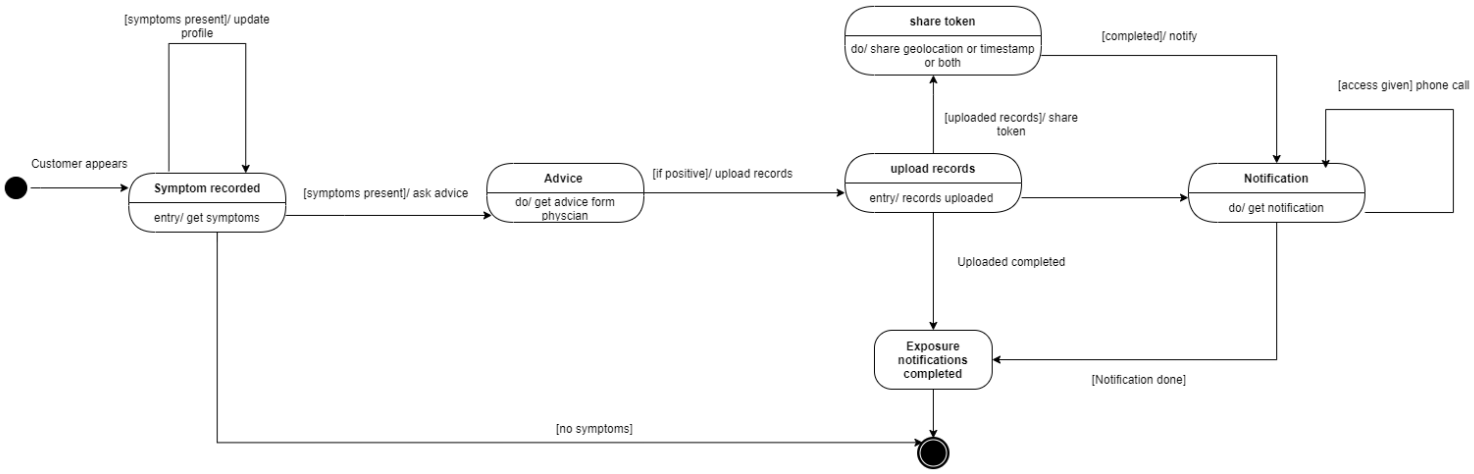
4.5.3. Third State Diagram

The diagram shows the states when the customer completes the risk assessment of a specific entered location.



4.5.4. Final State diagram

The following diagram shows the states of the objects when a covid-19 infected customer decides to upload their relevant document to the system so the public healthcare authorities can have an idea of this rapid spreading.



In conclusion, the sequential, activity and state diagrams can be broadly used in the requirements engineering domain because visually solving a problem is way more informative and interesting but as a downside most diagrams are not so accurate when it comes to reality. In this report, the sequential diagrams are done for each use case descriptions written, that is 7. Activity diagrams are done for the main scenarios encapsulating the whole requirement space.

5. References

Cysneiros, Luiz Marcio, Karin Koogan Breitman, and Marcio Luiz. 2016. "Lessons Learned On Requirements Elicitation Of Health Care Software Systems". *Researchgate*, 2016. https://www.researchgate.net/publication/268367842_Lessons_Learned_on_Requirements_Elicitation_of_Health_Care_Software_Systems.

Geppert, Birgit, and Klaus Schmid. 2002. "Requirements Engineering For Product Lines - An Overview - ". *Citeseerx.Ist.Psu.Edu*. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.2206&rep=rep1&type=pdf#page=25.

Mindila, Agnes N., Joseph M. Wafula, Harriet A. M. Ratemo, Collins Tabu, Joyce Charo, and Catherine Silali. 2019. "Requirements Elicitation For A Blockchain Vaccine Supply Chain Management Web/Mobile Application". *Gates Open Research* 3: 1420. doi:10.12688/gatesopenres.12877.1.

Mulero, Akeem. 2015. "REQUIREMENTS FOR A NCI STUDENT MOBILE APP DEVELOPMENT". National College of Ireland BSc Honours in Technology Management. http://norma.ncirl.ie/2351/1/akeemmulero.pdf.

Vepakomma, Praneeth, Robson Beaudry, Matt Gee, Jay Summet, Rachel Barbar, and John Werner. 2020. "COVID-19 Contact-Tracing Mobile Apps: Evaluation And Assessment For Decision Makers". *Arxiv.Org*. https://arxiv.org/ftp/arxiv/papers/2006/2006.05812.pdf.

Yousuf, Masooma, and M.Asger M.Asger. 2015. "Comparison Of Various Requirements Elicitation Techniques". *International Journal Of Computer Applications* 116 (4): 8-15. doi:10.5120/20322-2408.