SOFTWARE ENGINEERING
ASSIGNMENT PART 2
Requirement Specification Document

By Lean Bernardo, Harry Bishop, Daniel Brown, Luke Butterick, Vince Verdadero

Table of Contents

1. Introduction

1.1 Background

COVID-19 has had an enormous effect all over the world on health, the economy, and industries. The UK Government has introduced the NHS Test and Trace Program to help them trace the spread of the coronavirus and isolate new infections.

1.2 Purpose of the Document

This is a Requirement Specification document for our company Successful Partners Software (SPSoft*) which is tasked to create a Test and Trace software platform namely TTS.COVID-19 for the NHS. We aim to quickly reach potentially infected individuals to prevent the spread of the COVID 19 infection. The requirement is outlined in the project proposal section and extracted from https://www.gov.uk/guidance/nhs-test-and-trace-how-it-works#contents

1.3 Scope of the Document

This document will specify requirements for a simple application of the software system to gain a better understanding of what the system is and its functionalities along with the problems associated with Non-Functional Requirements.

Our group have used the MammoGrid use-case template to write our use-case specifications document. We have also used Astah to update any use cases that were identified in the project proposal.

1.4 Overview of the Document

This document will be organised as follows:

Section 2. Actors – An updated version of the actors identified in the project proposal and a short description of each actor specifications.

Section 3. Specifications of the core use cases – Each member has used the MammoGrid use-case template to write his use-case specifications document.

Section 4. System Models -

4.1 Class diagram - to model inheritance, association, aggregation, composition, and dependency, where available, to include entity classes and their attributes.

4.2 Use-case views – This section has two components:

> 4.2.1 System-level use-case model - the system level use-case model illustrates core use-cases and actors' interactions, whether human or non-human actors.

> 4.2.2 Individual team member use case model – Each member has produced the corresponding use-case view, showing interaction with respective actors and relationships (<<extend>> and <<include>>) with other non-core use-cases.

4.3 Sequence Diagrams for one main sub-flow that were specified in the specification core use cases.

Section 5. Non-Functional Requirements (NFRs) – we have identified and specified NFRs that may be related to the product, process, external requirements, and problems in this project assignment.

Section 6. Glossary – contains glossary terms that are related to our project.

2. Actors

NHS contact tracer

Responsible for bridging contact between the track and trace services and the user who tested positive for Covid – 19 infection.

Contact Sharing Service

The service that is responsible for securely sending the proximity contacts of the user who tested positive to the track and trace service and NHS contact tracers.

SMS service

The service that is responsible for receiving SMS requests and sending them out accordingly.

Email service

The service that is responsible for receiving email requests and sending them out accordingly.

User with a positive result

The user with symptoms tested positive for COVID 19 infection.

Track and Trace Service

The main service of the system, which is responsible for sending data between other services and end user-users

User with Symptoms

Anybody who is using the application to report symptoms of COVID 19.

Establishment owner

Anybody who owns a business that houses multiple people at once, such as a restaurant owner. In the use cases, it is assumed that the establishment owner records the contact details of whoever enters their establishment.

User in contact

Anybody who is contacted by an NHS contact tracer (via text/email) due to being in contact with someone who has tested positive for COVID-19.

TTS Admin

The Admin of the entire system. Is responsible for managing the system and can manage their account

Lab member

Person who receives and examine the test kits to know whether person is positive with COVID-19

3. Specification of the core use cases

3.1 Use case 1 Specification: Harry Bishop – Manage Account

1. Brief Description

This use case describes the tasks that the System Administrator can undertake upon logging into the system. Once they have successfully logged in with the correct credentials, they will have access to many options for updating their account, as well as managing the many sub systems and the overall system itself.

2. Actors

TTS Admin: The Admin of the entire system. Is responsible for managing the system and can manage their account.

Track and Trace Service: The main service of the system, which is responsible for sending data between other services and end user-users

3. Flow of Events

If the administrator chooses to log in to the system

UC1.1 The system executes Log In.

If the administrator logs in with the correct details

UC1.2 The system executes Authenticate Login Details and successfully logs the administrator in.

If the administrator logs in with incorrect details

UC1.3 The system executes Authenticate Login Details and informs the administrator of an invalid login attempt.

If the administrator chooses to update their account details

UC1.4 The system executes Update Account Details from which they can change their email and password.

If the administrator chooses to manage services

UC1.5 The system executes Manage Services and allows the administrator to access the SMS service, Track and Trace service, Contact sharing service, and Email service.

4. Non-Functional Requirements

System must log administrators in with correct details in less than 1 second

System must update details in less than 1 second when required

System must allow the administrator to access any of the services that need managing within 1 second

5. Pre-Conditions

5.1 Log in

The administrator already has an existing account.

5.2 Update Account Details

The administrator must be logged into a valid account before they can update their information.

5.3 Manage Services

The administrator must be logged into a valid account before they can manage any services.

6. Post-Conditions

6.1 Administrator Logs In

Authenticate the log in details and authorise a successful login

6.2 Update Account Details

Appropriate details are updated and saved

6.3 Manage Services

Appropriate changes to the services are updated and saved

7. Extension Points

7.1 Authenticate Login Details

UC1.1.1 Successful log in will log the administrator in
UC1.1.2 Unsuccessful log in will inform administrator as such

7.2 Update Account Details

UC1.2.1 Update Email
UC1.2.2 Change Password

7.3 Manage Services

UC1.3.1 Manage SMS service
UC1.3.2 Manage Email service
UC1.3.3 Manage Contact Sharing service
UC1.3.4 Manage Track and Trace service

3.2 Use case 2 Specification: Lean Bernardo – A person with possible COVID 19 report symptoms to the Track and Trace Service

1. Brief Description

This use case represents how a user with COVID-19 symptoms would get guidance on isolation when they have symptoms. The app will check if the user reports symptoms and ask for the date these symptoms started. If the user remembers, it will tell them to self-isolate 10 days from when they started and order a test from the NHS. If they do not, it will tell them to self-isolate immediately and order a test from the NHS.

2. Actors

User with symptoms: Anybody who is using the application to report symptoms of COVID-19
Track and Trace Service: The main service of the system, which is responsible for sending data between other services and end-users

3. Flow of Events

3.1   Basic Flow

Once a user login on the app, the app will provide a question if the user is unwell or not.

The user will then select if he or she does have the symptoms of COVID-19

If the user selects "I don't have any symptoms" then use case ends.

If the user inputs different kinds of symptoms of COVID-19, the app will ask the user the date when the symptom/s showed up.

The app will then remind them to self-isolate for 10 days from when they started to feel the symptoms of COVID-19 and order a test from the NHS.

The app will them report them symptoms to the Track and Trace Service to send data to other services and end-users.

3.2   Alternative Flows

3.2.1   No Data Entered

Problem: The user chose unwell, but did not state any symptoms, then:
The system will then generate an error message stating that at least one symptom should be entered before proceeding.

3.2.2   Invalid Field Edit

Problem: The patient attempts to edit non-editable fields.
The system generates an error message stating that the current field cannot be edited

3.2.3   Wrong Data Entered

Problem: The patient entered an invalid date.
The system then generates an error message stating that the date entered is invalid and should show the reason why. The date entered should only be 2 weeks ago from the present date.

4. Non-Functional Requirements

The user will provide different symptoms of COVID-19 and when he or she started feeling them, this will be then sent to the Track and Trace Service to share accurate information to other services and end-users.

All information given by the user will be protected and will be stored in a secure database. The user has the right to check every information stored about him or her and has the right to update every incorrect information.

The app will only be available for use in England, Wales, Scotland, and Northern Ireland.

5. Pre-Conditions

5.1 User Authentication

To create an account, the user will need to pre-register to the app. The user will be then asked if they agree on the terms and conditions. This will be then authorized to confirm user authentication which will then be approved or disapproved if the user is valid or not.

6. Post – Conditions

The system will tell the user to self-isolate from the date the user entered on when he or she had symptoms. The system user will be then told to book a test with the NHS and to follow different government guidelines for the safety of everyone.

7. Extension Points

7.1 Confirming User do not have symptoms

The app will ask the user the different symptoms he or she is feeling if unwell, if the user do not have any symptoms, the system will then generate message the confirm that the user do not have any symptoms at all.

7.2 User unsure with the date from when he/she started feeling symptoms

After the user finished entering the different symptoms, the app will then ask when the user started feeling the symptoms. If the person does not remember, it will extend to a button where it says, "Do not remember date".

3.3　Use case 3 Specification:　Luke Butterick – A user (With Symptoms of COVID-19) Creates an account

1 - Brief description

This use case details what happens when a user requests a COVID-19 test kit through the app. This includes all the actions that lead up to sending the test kit, as well as the response that the app gives the user.

2 – Actors

User with Symptoms: Anybody who is using the application to report symptoms of COVID-19

Track and Trace Service: The main service of the system, which is responsible for sending data between other services and end-users.

Email Service: The service that is responsible for receiving email requests and sending them out accordingly.

SMS Service: The service that is responsible for receiving SMS requests and sending them out accordingly.

3 - Flow of Events

3.1 - Basic Flow:

> The user attempts to create an account by inputting their full name, email address, phone number and home address.
> If there is not an existing account with the user's details:
> > - UC3.1 Authenticate the user's account as specified in extension point 7.1.
> If there is already an existing account with the user's details:
> > - UC3.2 Inform the user of their existing account and request for them to log in, as specified in extension point 7.2.
> Else, the use case ends.

3.2 - Alternative Flows:

> 3.2.1 - Invalid information inputted during account creation
> > If the User with Symptoms attempts to create an account with invalid information, then:

- UC3.3 Show a message stating that the information that was inputted is incorrect and ask the user to re-input the data.

The user changes their inputs as such that it is now valid data:

- UC3.3.1 The system allows the user to create an account again.

3.2.2 - Missing information during account creation

If the User with Symptoms attempts to create an account with missing information, then:

- UC3.4 Show a message stating that some key information is missing and ask the user to input valid data into the field.

The user inputs valid data into the missing field:

- UC3.4.1 The system allows the user to create an account again.

4 - Non-functional Requirements

4.1 - Requires interoperability with NHS Contact Tracers. This is because the contact tracers are responsible for contacting the lab and requesting the preparation of a test kit and providing them with the correct information.

4.2 - The system must be able to account for the fact that users might be using different email services or mobile network providers.

4.3 - To keep in line with the data protection act, user data will be stored for no longer than 1 month.

4.4 - Information must be kept securely to protect the information of the users.

5 - Pre-Conditions

5.1 – User Agreement

The user agrees to the terms and conditions of the app and service before creating an account.

6 - Post-Conditions

6.1 - Collection and Storage of User Data

The user's data will be collected and stored upon a successful execution of extension point 7.1. See the class model for a full description of the User Account class.

6.2 – Submit Request to Email Service

A request for an email to be sent will be submitted upon a successful execution of extension point 7.1.1.

6.3 – Submit Request to SMS Service

A request for an SMS to be sent will be submitted upon a successful execution of extension point 7.1.1

6.4 – Submit Request to Lab Member

A request for test kit prep will be sent to an NHS Contact Tracer upon a successful execution of extension point 7.1.1

7 - Extension Points

7.1 - Authenticate the User's Account

7.1.1 - Account Authenticated Successfully

The user inputted valid information during the account creation process and successfully authenticated their account.

UC3.5 A confirmation of account creation is sent to the email that was specified during account creation.

UC3.6 A confirmation of account creation is sent to the phone number that was specified during account creation.

UC3.7 A request for test kit prep is sent to an NHS Contact Tracer via the Track and Trace service.

UC3.8 Show a message informing the user that a test kit will be prepared and delivered to them as soon as possible.

The user will be taken back to the main screen of the app.

7.1.2 - Account Unable to Be Authenticated

The user inputted valid information during the account creation process, but their account was not authenticated.

UC3.9 Show a message informing the user of the unsuccessful account authentication.

The user will be taken back to the account creation screen and may attempt to make an account again.

15

7.2 - Account Already Exists with the Specified Details

The user inputted valid information during the account creation process but an account with one or more of the same details already exists.

UC3.10 Show a message informing the user of an account that already exists with the details specified and inform them that they can log in.

The user will be taken back to the account creation screen and may attempt to make an account again.

3.4   Use case 4 Specification: Vince Verdadero - NHS contact Tracer informs the user who tested positive for COVID – 19 infection

1. Brief Description

This use case describes how the NHS Contact Tracer will inform the user with symptoms who tested positive for COVID 19 infection and instructions on how to share his/her proximity contacts.

2. Actors

NHS contact tracer: Responsible for bridging contact between the track and trace services and the user who tested positive for Covid – 19 infection.

Contact Sharing Service: The service that is responsible for securely sending the proximity contacts of the user who tested positive to the track and trace service and NHS contact tracers.

SMS service: The service that is responsible for receiving SMS requests and sending them out accordingly.

Email service: The service that is responsible for receiving email requests and sending them out accordingly.

User with a positive result: The user with symptoms tested positive for COVID 19 infection.

3. Flow of Events

3.1 Basic flow

When the NHS Contact tracer has been authorised to execute this use-case, then he/she will be able to choose which of the tasks below to perform:

If the NHS contact tracer creates a message to send an Email and SMS to the user, then:

UC4.1. The system executes to send SMS and Email informing of a positive covid lab result and advice to isolate following government guidelines.

Else, If the NHS contact tracer creates a message to send SMS and Email instruction to share proximity contact details of the user who tested positive, then:

UC4.2. The system executes to send SMS and Email as specified in extension point 7.1.

Else, the system will request the proximity service to alert affected app users then:

UC4.3. The system executes to alert affected app users as specified in extension point 7.2.

Else, the use-case ends.

3.2 Alternative flows

3.2.1 If the user who has tested positive for COVID 19 is not responding within 24 – 48 hours.

The NHS contact tracer will create a message to contact the local authority then:

UC4.4. The system will send an Email and SMS to pass the user's information to the local authority to follow up the case by phone or in person. (Under 18's will have a guardian/parent to help)

Else, if the User who has tested positive for COVID 19 has no internet access or poor connection then:

UC4.5 The system will generate an error message to state that the Email and SMS service is unavailable.

3.2.2 If lab result is negative

When the NHS Contact tracer has been authorised to execute this use-case, then he/she will be able to choose which of the tasks below to perform:

If the NHS contact tracer creates a message to send an Email and SMS to the user, then:

UC4.6. The system executes to send SMS and Email informing of a negative covid lab result and give advice that he/she no longer needs to self-isolate if well and follow government advice.

3.2.3 Invalid username/password entered

The User with a positive result enters an invalid username/password on the TTS COVID app, then:

UC4.7 The system generates an informative message stating that an invalid username/password has been entered.

The system prompts the user to re-enter the correct username/password.

The User re-enters the username/password in the respective field.

UC4.7.1 The system returns to a ready state

4. Non-Functional Requirements

4.1 Efficiency

- SMS and Email sent to the user must be delivered in real-time.
- If the user who has tested positive for COVID - 19 is not responding within 24-48 hours. SMS and Email must be sent to the local authority to contact the user who tested positive for COVID 19 infection.

4.2 Voluntariness and compliance

- NHS Contact tracer will send the user who is positive for COVID 19 infection an Email and SMS of instructions on how to share details of his/her proximity contacts. The user should respond immediately and voluntarily to share accurate information with Contact Sharing Service.

4.3 Security and confidentiality

- The user's information and proximity contacts must have all their details protected and stored encrypted in a secured database for 14 days.

4.4 Usability

- The app must be easy and free to download
- Instructions to join and login must be simple to follow

4.5 Interoperability

- Both apple and android devices must be able to communicate with each other to screen proximity contacts and identify affected app users.
- The app can be used in England, Wales, Scotland, and Northern Ireland.

4.6 Legislation Requirement

- The user's and proximity contact details and information provided should be held in strict confidence. It should only be used in line with the Data Protection Act 2018 which is the UK's implementation of the General Data Protection Regulation (GDPR).

5. Pre – conditions

5.1 User authentication

The user will need to pre-register to the app to create a Unique ID/account. This information will be communicated to the main Track and Trace Service.

The user's consent will be asked to agree on the terms and conditions of using the app.

5.2 The user must request a test kit for COVID 19 infection.

The user will need to request a test kit to check if he/she has COVID - 19 infection and promptly sends the swab back to the laboratory for processing.

6. Post – conditions

6.1 Alert notification

NHS contact tracers will alert the affected person/s who were in close contact with the user who is positive for COVID 19 using SMS and Email service.

7. Extension Points

7.1 User with a positive result to share information of proximity close contacts on the TTS COVID app

UC 4.8 The Email and SMS service will send an Email and SMS containing instruction to share proximity contacts on the TTS COVID app.
UC 4.9 The user with a positive result will sign in using their username and password and clicks the contact sharing details button.

After a successful login, the secure website will prompt the user to complete the following mandatory – fill fields:

- Places/location recently visited
- Time of contact

- Date of contact
- Details of the person with whom they have had close and recent contacts:
  1. People of whom the user spent 15 minutes or more at less than 2m(6ft)
  2. Household members
  3. People whom the user has had a face-to-face conversation at less than 1m.
  4. Contact period must take place within 10 days, starting 48 hours before symptoms appeared.

The User enters data of the above fields and then selects the SAVE and SEND button.

The Contact Sharing Service will receive the above information.

7.2 Alerting affected app users

The proximity service will alert affected app users.

The device of the user with a positive result will communicate with other app users using Bluetooth Low Energy.

UC 4.10. The system will perform close contact screening of other app users based on risk analysis
UC 4.11. The system will perform a device-to-device communication using Bluetooth Low Energy.
UC 4.12 The system will send alert to potentially affected app users.

3.5   Use case 5 Specification: Daniel Brown – Inform user of COVID 19 contact

1. Brief Description

This use case describes the tasks that the NHS contact tracer would undergo to inform a user that has been in contact with a person who has tested positive for COVID-19 that they have indeed been in contact with the virus.

2. Actors

Email service: The service that is responsible for receiving email requests and sending them out accordingly.

NHS contact tracer: Responsible for bridging contact between the track and trace services and the user who tested positive for Covid – 19 infection.

SMS service: The service that is responsible for receiving SMS requests and sending them out accordingly.

3. Flow of Events

3.1 Basic Flow

3.1.1 When the NHS contact tracer is required to do the task, they will carry out the following procedure:

NHS contact tracer will create a message and prompt the email and SMS service to send out the message to the user in contact with COVID-19

If the infected user has provided their contacts email address on the system, then:

UC5.1 – the email system will send an email prompted by the NHS contact tracer to the user in contact with COVID-19.

If the infected user has provided their contacts phone number on the system, then:

UC5.2 - the SMS system will send a text prompted by the NHS contact tracer to the user in contact with COVID-19.

Use case ends

3.2 Alternative flows

3.2.1   Services unavailable

NHS contact tracer uses the E-Mail service to send an email to a person in contact when the service is unavailable due to connection or other issue:

UC5.3 – an error message is generated by the system to state that the E-Mail service is unavailable

NHS contact tracer uses the SMS service to send a text to a person in contact when the service is unavailable due to connection or other issue:

UC5.4 – an error message is generated by the system to state that the SMS service is unavailable

Use case ends

4. Non-Functional Requirements

Data provided by people for use in these systems must be protected under the data protection act 2018 and stored in a secured, encrypted database.
The app will only be available for use in England, Wales, Scotland, and Northern Ireland.
Link for the secure NHS contact sharing website will be provided in each SMS message and email.

5. Pre-Conditions

A user who has tested positive for COVID-19 or a proximity program showing who was nearby that person must have provided information and contact details of a person that they have been in contact with for this use case to be carried out.

6. Post -Conditions

A message, whether by SMS, email, or both, must be sent to the user in contact detailing what they will need to do. This includes the need to self-isolate for a fortnight, log into the test and trace website and to inform the recipient to get a COVID-19 test if they have symptoms.

4. System Models

4.1 Class Diagram
Our proposed approach to the system will use a client-server application. Therefore, we have a diagram for both the client and server. (The first diagram is the client, the second is the server).

Main (Client)

Used to launch the JavaFX window

+ start(stage : Stage) : void
+ main(args : string[]) : void

MainController

This class controls the mainApp.fxml file that is included in the program.

+ homeTab : Tab
+ loginTab : Tab
+ adminTab : Tab
+ loginButton : Button
+ registerButton : Button
+ passwordField : PasswordField
+ emailField : TextField
+ hyperLink : Hyperlink
+ covidInfoButton : Button
+ commonSymptomsButton : Button
+ testKitInfoButton : Button
+ settingsButton : Button
+ mainPane : TabPane
+ getEmailButton : Button
+ getSmsButton : Button
+ accountField : TextField
+ messageIdField : TextField
+ adminPasswordField : PasswordField
+ deleteAccountButton : Button
+ adminPasswordButton : Button
+ contactSharingButton : Button
+ updateAccountButton : Button
- loggedIn : boolean
- emailText : boolean
- passwordText : boolean
- admin : boolean
- loggedInAs : string
- client : Client

+ onEmailKeyPressed() : void
+ onPasswordKeyPressed() : void
- toggleLoginRegisterButtons(value : boolean) : void
+ onLoginButton() : void
+ onRegisterButton() : void
+ onEmailButton() : void
+ onSmsButton() : void
+ onCovidInfoButton() : void
+ onCommonSymptomsButton() : void
+ onTestKitInfoButton() : void
+ onShareContactButton() : void
+ onUpdateAccountButton() : void
+ onAdminUpdatePasswordButton() : void
+ onDeleteAccountButton() : void
+ onLoginLinkPress() : void
+ onHomeLinkPress() : void
- enableAfterLogin() : void
- enableAfterAdmin() : void
- showRegistrationWindow() : void
- showContactSharingWindow() : void
- showUpdateAccountWindow(email : string) : void
- showPopupWindow(title : string, text : string) : void

UpdateAccountWindowController

This class controls the updateAccountWindow.fxml file that is included in the program.

+ updateAccountButton : Button
+ helpButton : Button
+ addressTextField : TextField
+ passwordField : PasswordField
- client : Client
- email : string

+ setClient(client : Client) : void
+ setEmail(email : string) : void
+ onHelpButton() : void
+ onUpdateAccountButton() : void
+ showPopupWindow() : void

RegistrationWindowController

This class controls the registrationWindow.fxml file that is included in the program.

+ registerButton : Button
+ passwordTextField : PasswordField
+ emailTextField : TextField
+ addressTextField : TextField
+ phoneNumberField : TextField
+ nameTextField : TextField
+ registerHyperlink : Hyperlink
- client : Client
- emailText : boolean
- passwordText : boolean
- phoneNumberText : boolean
- addressText : boolean
- nameText : boolean

+ onEmailKeyPressed() : void
+ onPasswordKeyPressed() : void
+ onPhoneNumberKeyPressed() : void
+ onAddressKeyPressed() : void
+ onNameKeyPressed() : void
+ setFieldText(emailText : string, passwordText : string) : void
+ setClient(client : Client) : void
+ onRegisterLinkPress() : void
+ onRegisterButton() : void
- toggleRegisterButton(value : boolean) : void
- showPopupWindow(title : string, text : string) : void

ContactSharingController

This class controls the contactSharing Window.fxml file that is included in the program.

+ helpButton : Button
+ shareDetailsButton : Button
- contactDetailsField : TextArea
- client : Client

+ onHelpButton() : void
+ onShareDetailsButton() : void
+ setClient(client : Client) : void
- showPopupWindow(title : string, text : string) : void

PopupController

This class controls the popupWindow.fxml file that is included in the program.

+ messageText : TextArea
+ titleLabel : Label
+ okButton : Button

+ setMessageText(text : string) : void
+ setTitleLabel(title : string) : void

Client

Socket, PrintWriter and BufferedReader are from Java, allows for communication with the server.

- connected : boolean
- socket : Socket
- out : PrintWriter
- in : BufferedReader
- port : int
- fromUser : string
- hostName : string

+ run() : void
+ requestFromServer() : string
- initSocket() : void

Launches ->

<- Launches

Launches ->

Launches ->

<- Launches

1

1

1

1

1

24

pkg

Main (Server)
+ main(args : string[]) : void

Starts the server

SocketServer
- middleman : ServerSocket
- port : int
- client : Socket
- accountManager : AccountManager
- contactTracingService : ContactTracingService
+ run() : void
- registerAccount(params : string[]) : boolean
- deleteAccount(params : string[]) : boolean
- editAccount(params : string[]) : boolean
- shareContacts(params : string[]) : string
- loginAccount(params : string[]) : string
- getEmail(params : string[]) : string
- getSms(params : string[]) : string
- splitInput(input : string) : string[]

Reads in a command from the client in the form of a string and outputs a response in the form of a string

ContactTracer
+ UNAVAILABLE : int
+ WAITING : int
+ BUSY : int
- status : int
- id : int
- name : string
+ contactLabMember(name : string, address : string, labMember : LabMember) : void
+ setStatus(status : int) : void
+ getStatus() : int
+ getName() : string

The capitalised variables are static and callable from anywhere so that things like ContactTracer.BUSY can be used when setting status.

AccountManager
- registeredAccounts : List<Account>
- adminAccounts : List<AdminAccount>
- account_csv : string
- admin_csv : string
+ registerAccount(account : Account) : boolean
+ deleteAccount(email : string) : boolean
+ editAccountData(email : string, replacement : string[], admin : boolean) : string
+ saveWholeCsvFile(csvPath : string, admin : boolean) : void
+ accountExists(email : string) : boolean
+ clearCsvFile(csvPath : string) : void
+ writeAccountToCsvFile(account : Account, csvPath : string) : void
+ loadExistingAccountsFromDatabase() : void
+ loadAdminAccountsFromDatabase() : void
+ getCsvData(csvPath : string, admin : boolean) : List<string>

ContactTracingService
- contactTracers : List<ContactTracer>
- laboratories : List<Laboratory>
- labPath : string
- contactPath : string
+ contactLabMemberForTest(ipFreq(name : string, address : string)) : void
+ getAvailableContactTracer() : ContactTracer
+ contactPersonInContact(contact : string) : void
+ getAvailableLaboratory() : void
+ loadContactTracer() : void
+ loadLabs() : void
+ getCsvData(csvPath : string) : List<string>

SmsService
- csvPath : string
- nextId : int
- fileData : List<string>
+ sendSms(sender : string, recipient : string, content : string) : void
+ saveSmsToCsv(sms : Message) : void
+ getLineCountOfCsv() : void
+ getSms(id : int) : string
+ getCsvData() : List<string>

CommunicationService
- emailService : EmailService
- smsService : SmsService
+ sendEmail(sender : string, recipient : string, subject : string, content : string, attachments : string[]) : void
+ sendSms(sender : string, recipient : string, content : string) : void
+ getEmail(id : int) : string
+ getSms(id : int) : string

EmailService
- csvPath : string
- nextId : int
- fileData : List<string>
+ sendEmail(sender : string, recipient : string, subject : string, content : string, attachments : string[]) : void
+ saveEmailToCsv(email : Email) : void
+ getLineCountOfCsv() : void
+ getEmail(id : int) : string
+ getCsvData() : List<string>

AdminAccount

Account
- email : string
- password : string
- phoneNumber : string
- address : string
- name : string
+ getEmail() : string
+ getPassword() : string
+ getPhoneNumber() : string
+ getAddress() : string
+ getName() : string
+ setPassword(password : string) : void
+ setAddress(address : string) : void

Message
- id : int
- sender : string
- recipient : string
- content : string
+ getId() : int
+ getSender() : string
+ getRecipient() : string
+ getContent() : string

Email
- subject : string
- attachments : string[]
+ getSubject() : string
+ getAttachments() : string[]

FilePath
+ filePath : string

Any class that reads in and saves to files uses this class. It contains a static variable that holds the path to the folder that contains all .csv files.

Laboratory
+ UNAVAILABLE : int
+ WAITING : int
+ BUSY : int
- status : int
- name : string
- labMembers : List<LabMember>
+ getAvailableLabMember() : LabMember
+ getStatus() : int
+ setStatus(status : int) : void

The capitalised variables are static and callable from anywhere so that things like Laboratory.BUSY can be used when setting status.

LabMember
+ UNAVAILABLE : int
+ WAITING : int
+ BUSY : int
- status : int
- name : string
+ prepTest(id(name : string, address : string)) : void
+ setStatus(status : int) : void
+ getStatus() : int
+ getName() : string

The capitalised variables are static and callable from anywhere so that things like LabMember.BUSY can be used when setting status.
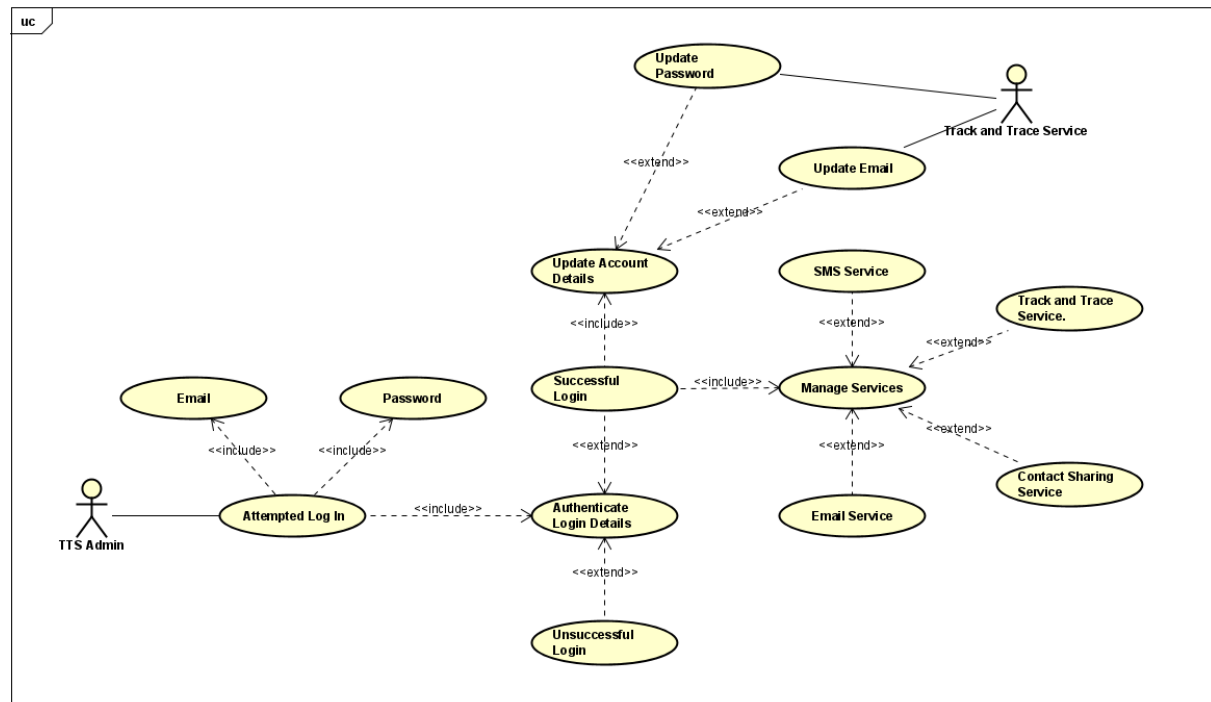
Uses

4.2 Use case views

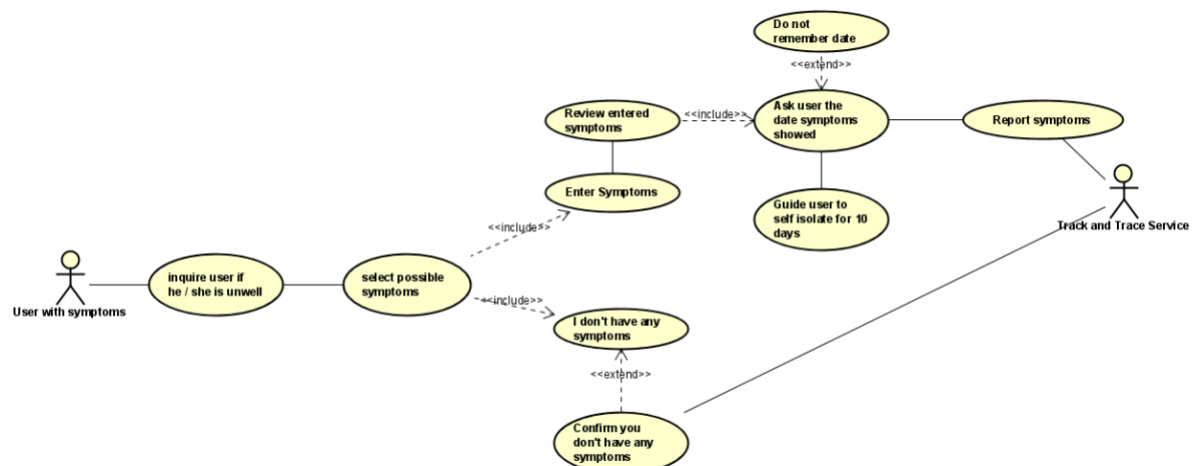## 4.2.1 System level use case model

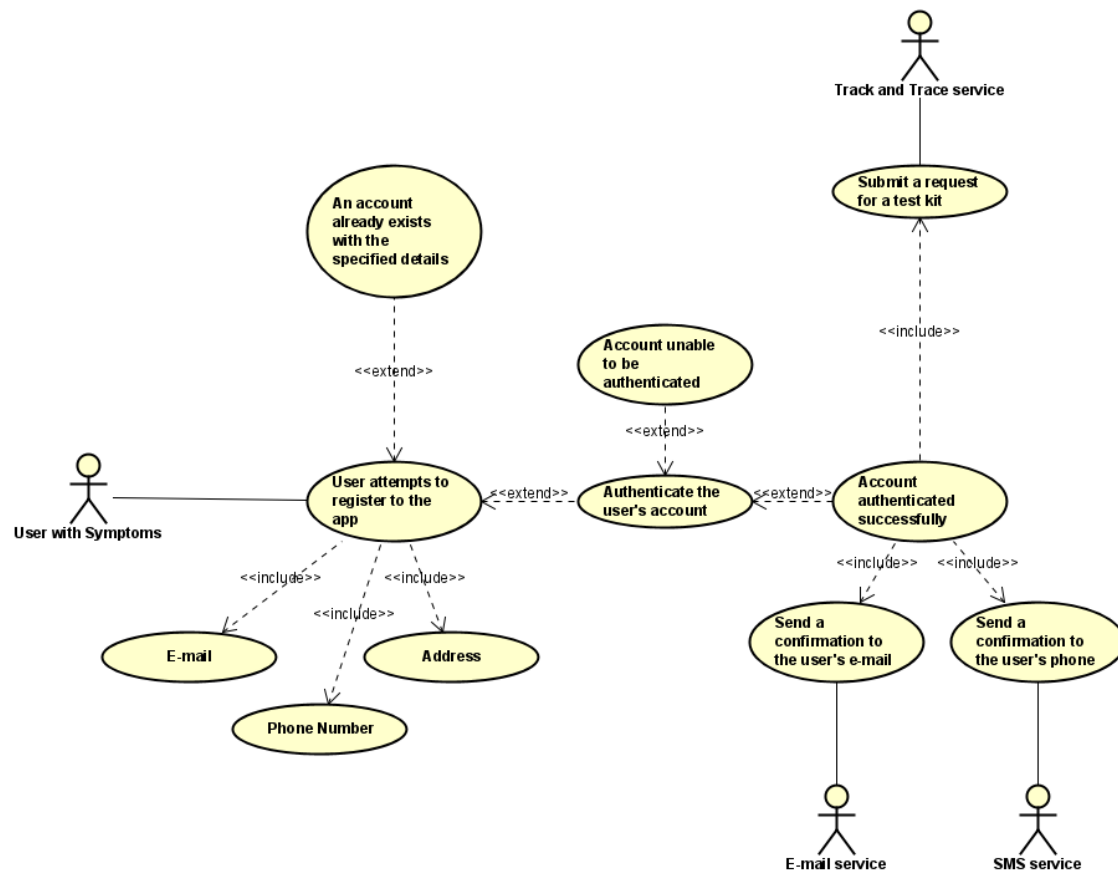## 4.2.2 Individual use case view model

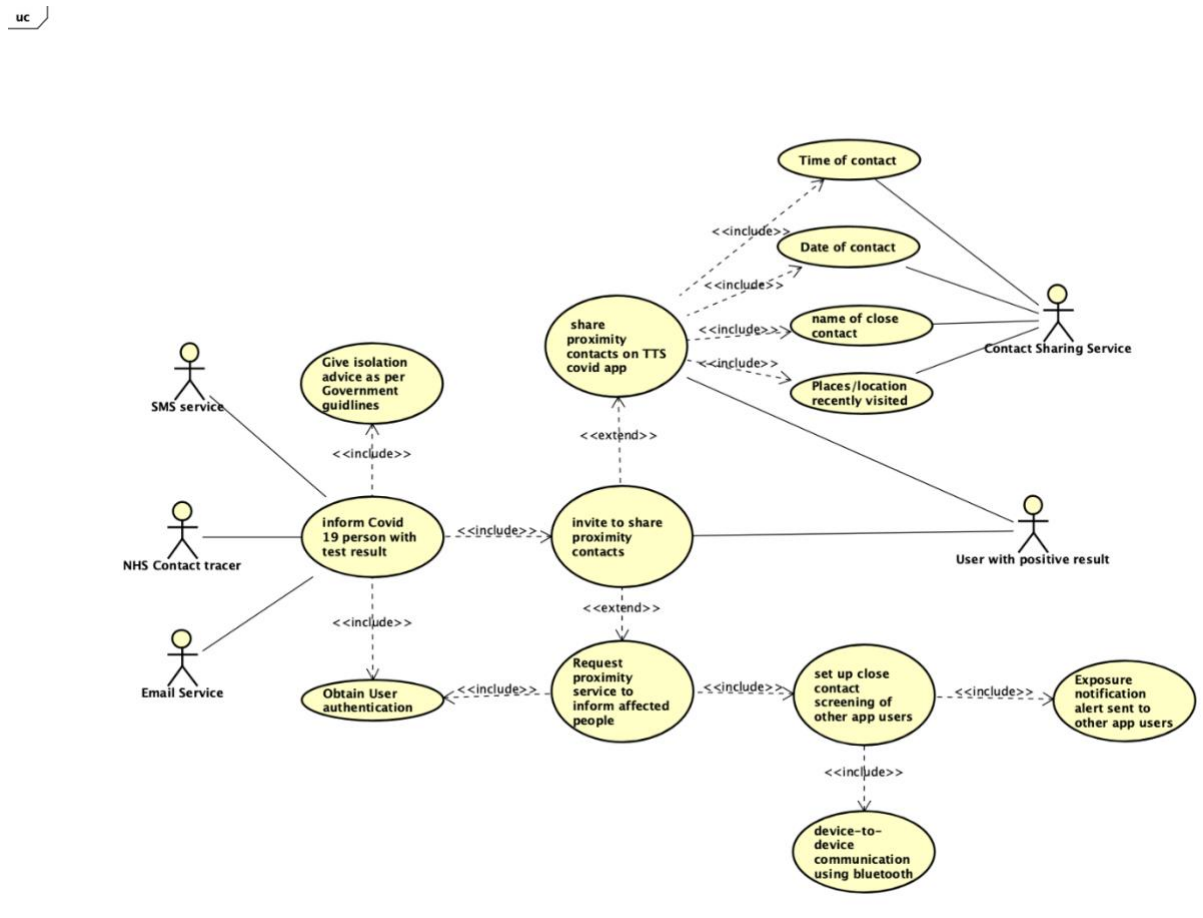### 4.2.2.1 Use case view model: Harry Bishop – Manage Account

4.2.2.2 Use case view model: Lean Bernardo – A person with possible COVID 19 report symptoms to the Track and Trace Service
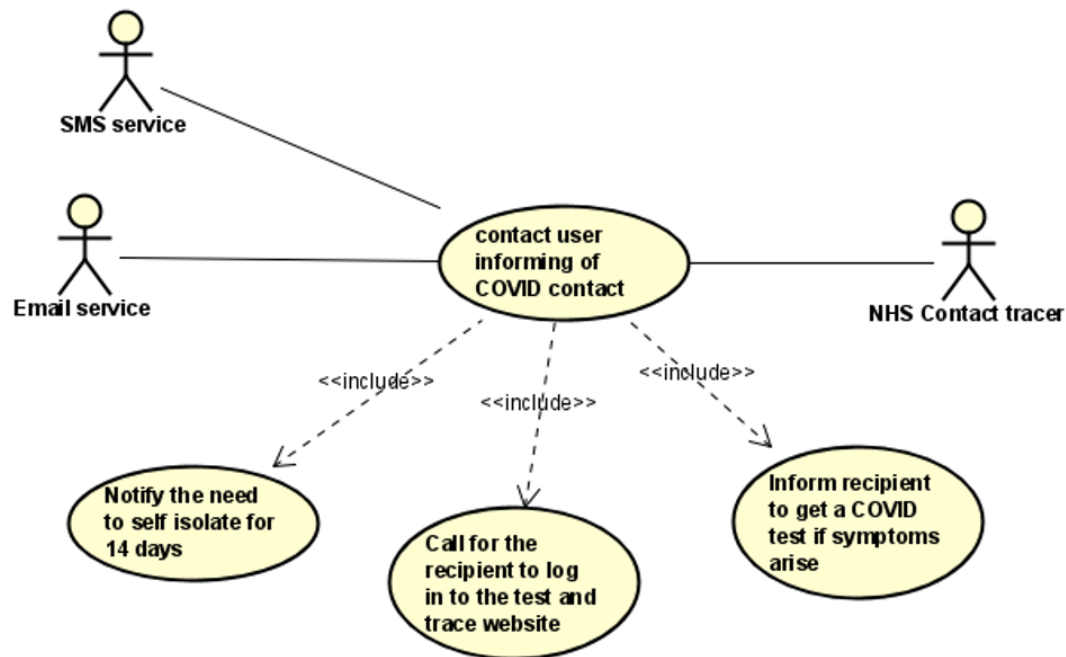
4.2.2.3 Use case view model: Luke Butterick – A user (with symptoms of COVID 19) Creates an account.

4.2.2.4 Use case view model: Vince Verdadero – NHS contact Tracer informs the user who tested positive for COVID – 19 infection.

4.2.2.5 Use case view model: Daniel Brown – Inform user of COVID 19-contact.

4.3 Sequence diagrams

4.3.1 Sequence Diagram, Harry Bishop – Admin Account
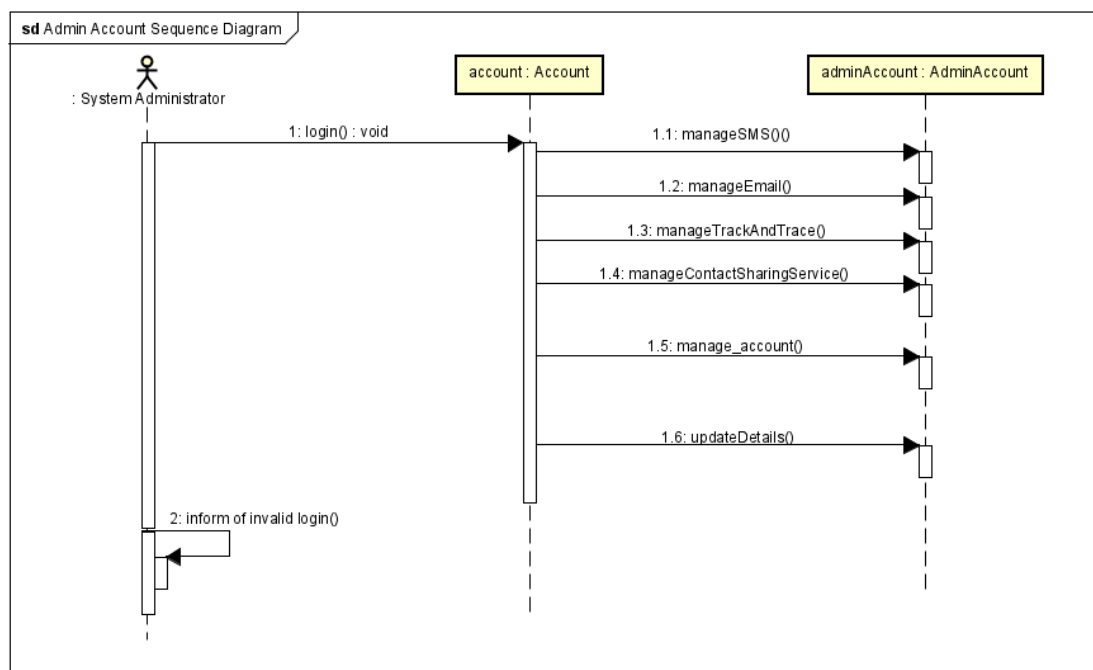
4.3.2 Sequence Diagram, Lean Bernardo – A person with possible COVID 19 reports symptoms to the Track and Trace Service
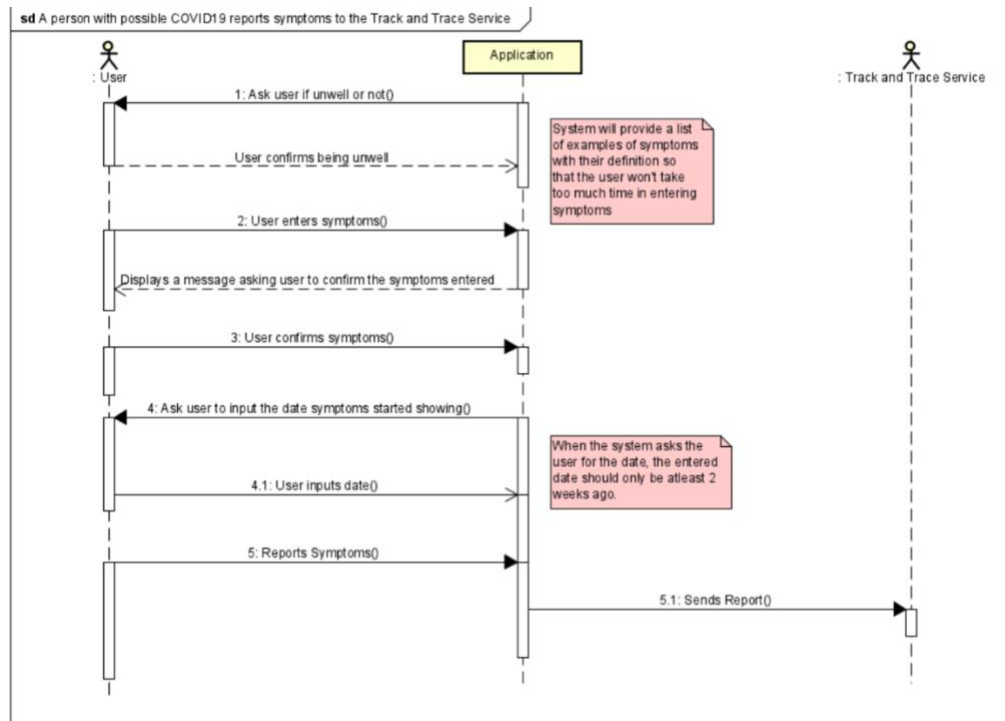
4.3.3 Sequence Diagram, Luke Butterick – User with COVID – 19 Symptoms Creates an Account
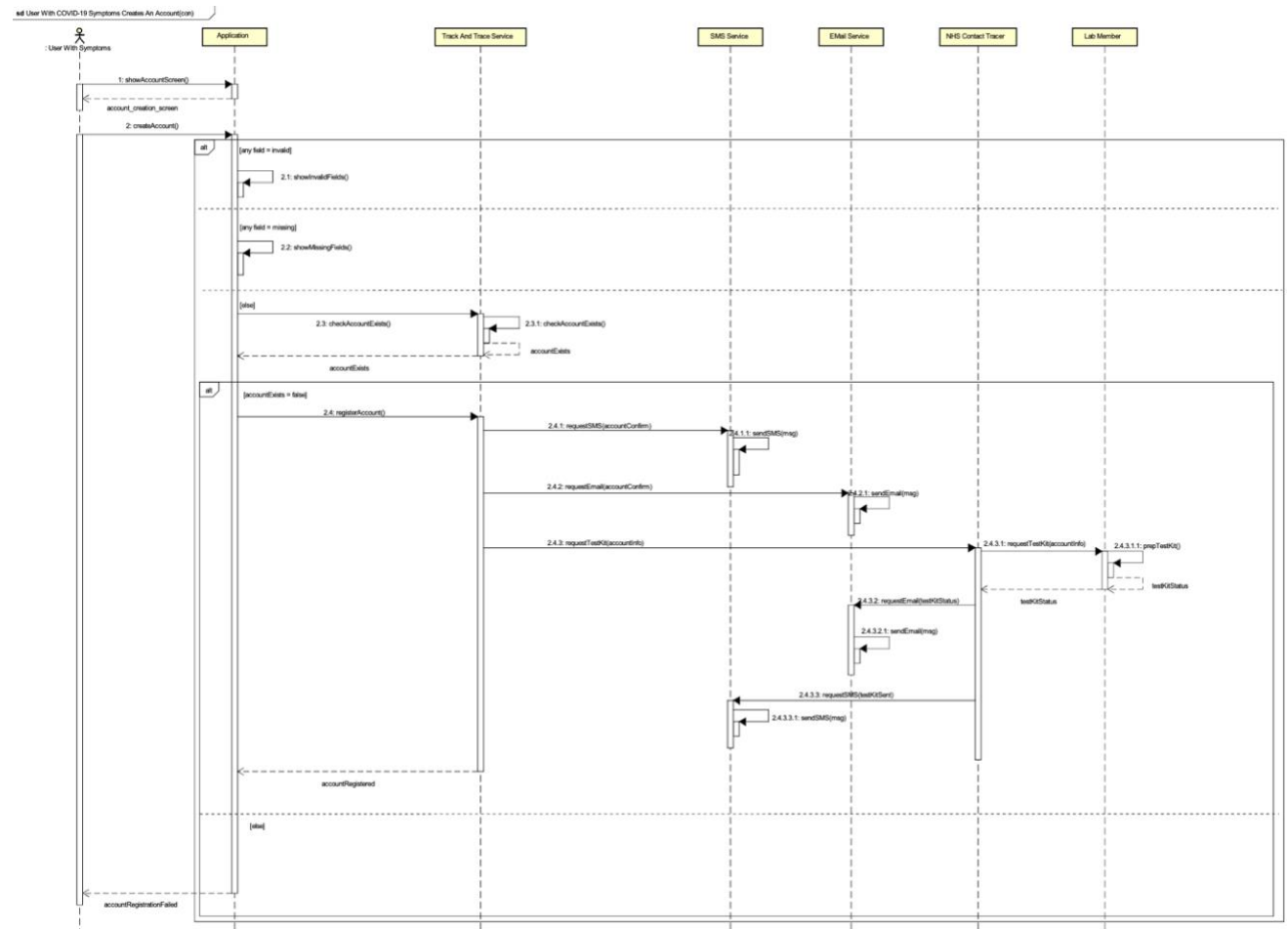
4.3.4 Sequence Diagram, Vince Verdadero – NHS Contact Tracer informs the user who tested positive for COVID – 19 infection
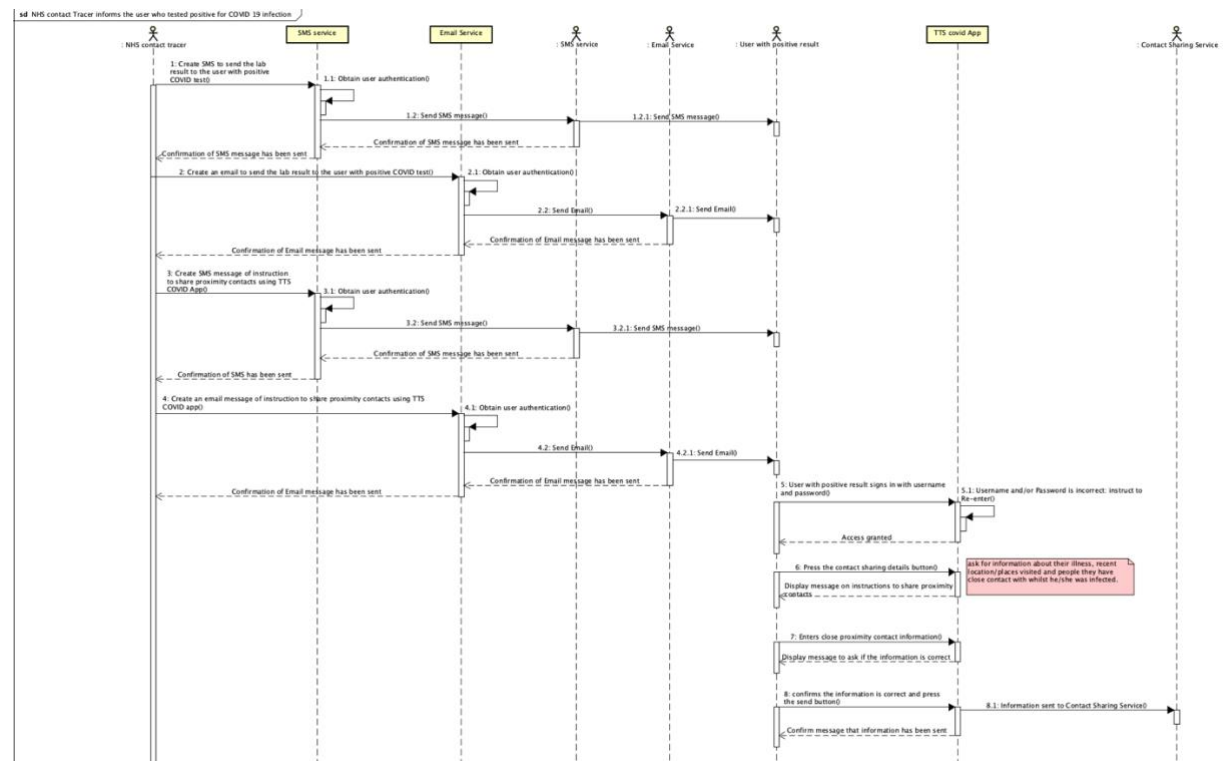
4.3.5 Sequence Diagram, Daniel Brown – Contact tracer informs user of covid contact.
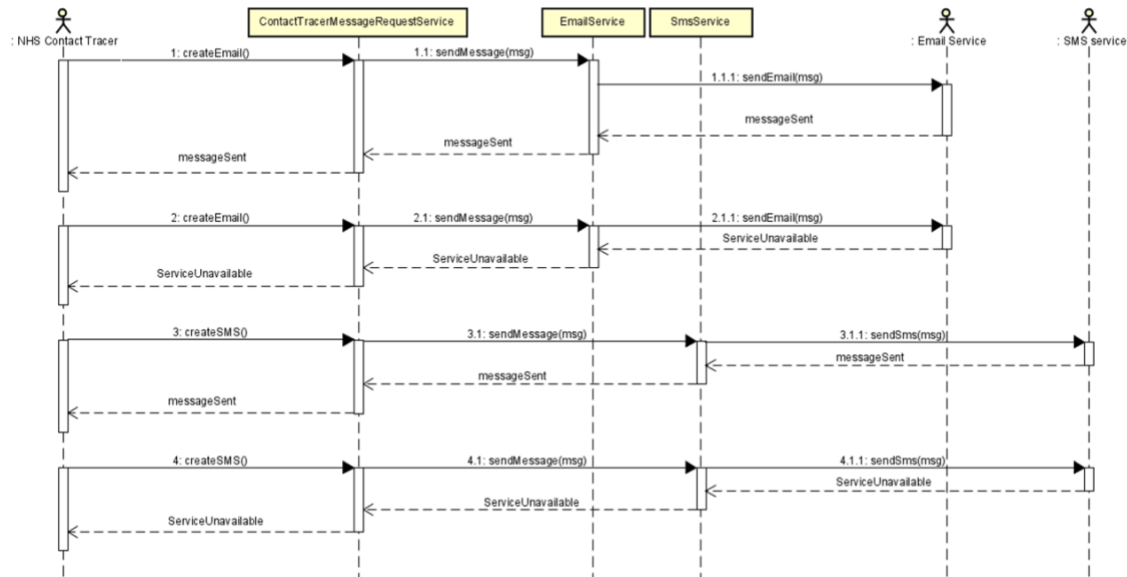
5. Non-Functional Requirements (NFRs)

5.1 Product Related Requirements

Security and confidentiality

- The user's information and proximity contacts should have all their details protected and should be stored encrypted in a secured database for 14 days.
- The NHS will have to approve the use of the TTS COVID 19 system.
- Malware threats and unauthorised access to the system's data must be avoided at all costs.

Usability Requirement

- The system must be accessible 24 hours a day, seven days a week, as it can be used at all hours of the day and night.

- The chances of information leakage or data loss must be limited, and the time it takes to reboot must be quick.
- The system must be user friendly to provide a user experience that is clear, navigate and easy to understand.

Performance Requirement

- It must be reasonably fast - the response time or process time of the system must be fast so that users can maximise work efficiency
- It must be robust so the probability of data corrupting must be low or not at all
- The system must be able to account for the fact that users might be using different email services or mobile network providers.

Portability and compatibility

- The app must run on the following operating systems - The phone will need Bluetooth Low Energy (BLE) 4 or above and Operating System software Android Marshmallow, or iOS 13.5 or above.

5.2 Process Related Requirements

Operational Requirement

- The system can only be used by the staff as well as other approved personnel, so users must be confirmed.

Developmental Requirement

- Standards such as those defined by the ISO and IEEE must be included in the development process.

5.3 External Related Requirements

Legislative Requirements

- The user's and proximity contact details and information provided should be held in strict confidence. It should only be used in line with the Data Protection Act 2018 which is the UK's implementation of the General Data Protection Regulation (GDPR).

Ethical Requirements

- Users must agree to a set of guidelines and regulations.

<u>Interoperability</u>
- The app can be used in England, Wales, Scotland, and Northern Ireland.
- Both apple and android devices must be able to communicate with each other to screen proximity contacts and identify affected app users.
- Requires interoperability with NHS Contact Tracers - contact tracers are responsible for contacting the lab and requesting the preparation of a test kit and providing them with the correct information.

6. Glossary

<u>Use case diagram</u>

A use case diagram is a type of behavioural diagram identified by and derived from a use-case analysis in the Unified Modelling Language (UML). Its aim is to deliver a graphical representation of a system's functionality. As stick figures, actors in the process, who may be human or other systems, are represented. Each class of interaction is defined as an ellipse by name. Use cases can be described textually, with a focus on the event flow between actor and system.

<u>Class diagram</u>

A Class diagram represents a static view of applications. Also, it is not only used for visualising different aspect of a system but can also help describe and construct the code of the model application. The class is represented by a rectangle, which is separated into three parts. The name appears at the top, the attributes appear in the centre, and the operations appear at the bottom.

<u>Sequence diagram</u>

A sequence diagram is a type of UML diagram that interacts and illustrates the order in which messages are sent.

<u>Actors</u>

In UML diagrams, an actor is a behavioural classifier that determines the purpose of an external entity which interrelates with the topic.

User authentication

Both human-to-computer communications enable the user to register and log in that are protected by the security process.

Non-Functional Requirement

A review of the properties and constraints of the system which are required. It's not easy, yet it has to do with the system's compatibility, usability, efficiency, etc.

Pre and post conditions

Which are statements that are true before and after a processing activity has been enacted or a product produced.