# Software Engineering and Professional Practice and Building Useable Systems Coursework

*<SafeTravel>*

**Group 50**


**Members:**

2035179, vxf979@student.bham.ac.uk
1906868, wxj868@student.bham.ac.uk
2026739, wxc939@student.bham.ac.uk
1903981, hxp881@student.bham.ac.uk
2049919, tjs919@student.bham.ac.uk
2068412, gxk912@student.bham.ac.uk
2048464, yxf965@student.bham.ac.uk
2037641, mkk941@student.bham.ac.uk
2078371, wjc971@student.bham.ac.uk

# TABLE OF CONTENTS

# A1. Introduction

Travelling between places safely has proven to be difficult during this COVID-19 era. The mobile application which we have designed would help people travel between places in a way with minimum infection risk. By implementing various tracking methods and algorithms, the app would help people travel safely and hence impact the rate of spread of the virus. The application can be broken down into three core components.

### 1.1 Colour coded map

The colour coded map visualises levels of danger of COVID-19 infection. A green to red gradient is used to signify the level of danger, with green representing low danger and red representing high danger. There are four different levels of map view which will change automatically depending on how much the user zooms in or out. The more the user zooms, the more accurate information and colour codes get for each area. Further information including trend rates of infection, confirmed cases and vital information regarding COVID-19 regulations and restrictions is available when a segment of the map is selected.

An algorithm taking into consideration the amount of infections during a certain amount of days and density of population at the given place and time would be used to calculate the danger level. The density of population for the areas in the map will be monitored with the use of Bluetooth Low Energy beacons and QR codes. Additionally, we will use WiFi routers to provide the users with an internet connection when the user does not have mobile data. The beacons and WiFi routers will be attached to traffic lights around the country in order to record footfall throughout the entire area. This information will be displayed and updated regularly in the app, indicating how busy an area is. Unique QR codes will be accessible at every public building and public transport, recording the number of people at the premise when scanned by each user. Users will be able to disclose their departure in the app. In these ways, we will create our density of population information to display in our maps and overlay with colour codes that are specific down to the street and building level.

### 1.2 Declaration of COVID-19 Positive Case

Allowing users to declare oneself COVID-19 positive on the app is another essential function of the app. A unique code would be provided to each COVID-19 positive user with assistance from the NHS. Entering the code will allow the user to declare a positive case and thus the app would provide the user with detailed instructions regarding self-isolation and other information regarding the virus. The application would also collect the location history data of an infected user, assisting in notifying each user in contact with the infected user in the past 7 days about the risk of infection.

### 1.3 Suggestion of Safest Route to Specified Locations

The final core function would be calculating the safest routes to the given locations. The data used for calculating danger levels will be obtained by analysing COVID-19 case history and detailed density of population at a given time to provide the user with the safest route between points. The density data would be collected by using the previously mentioned tracking methods. The app will also take into account the mode of transport, providing different routes for different modes of transport (e.g. on foot, by car and by public transport). The paths will be displayed on the map without any colour codes from the starting address to the destination. Upon selecting a path and starting, the system will navigate the user to his destination, prompting to scan the available QR codes at the checkpoints throughout and at the destination.

## Assumptions

There's undoubtedly a variety of assumptions that we have taken into consideration while designing the above features:

- A code system verifying COVID-19 declarations would be implemented with assistance from the NHS to deter false declarations.
- Majority of users will use the application with Bluetooth and an active Internet connection.
- All users agree to their private information such as location history and case history being collected.
- All COVID-19 related information such as local cases and restrictions will be collected from government bodies.
- Bluetooth beacons will be fitted to traffic lights within our scope.
- Unique QR codes will be displayed in each public premise for user scanning.
- Google Maps API will be implemented to display maps and assist in the data collection by the beacons.

Scope of the system involves the entire United Kingdom.

# A2. Functional and non-functional requirements

## Functional Requirements

### 1. Agreement

1.1 The system must present the user with a privacy policy/user agreement when he opens the app for the first time.
1.2 The system must obtain the user's permission for location services in the form of a pop-up.
1.3 The system must obtain the user's permission for push notifications in the form of a pop-up.

### 2. Colour coded map

2.1 The system must provide the user with a colour coded interactive map of a combination of the density of users and infection rates with different levels of details.
a. The system should have four different levels of map view: Block/street level, collection of multiple blocks level (the outward code), city level and national level.
b. The street level should include buildings on the street and will show detail about the current occupancy in each public building in the form of colour codes.
c. The collection of multiple blocks level should include a wider look on multiple streets and assign colour codes to every collection of streets.
d. The city level should show two or three cities and the major roads and motorways between them with colour codes assigned to each area/district.
e. The national level should have the entire country with colour codes being assigned to each city or county.
f. The colour coded map should be updated with the latest data every time the user views the map.
g. The system should allow users to search locations by postcode, providing the nearest level of view available.
2.2 The map must be overlaid with colour codes for each level of the map and is more generalised the further out the map gets.
a. The colour codes should use a gradation from light green to dark red to signify the levels of danger in an area.
b. In the navigation map the colour codes are automatically toggled off.
2.3 The map should be interactive with users being able to select different segments of the map and view advanced details of that area specifically.
a. The map should provide details about confirmed cases and trend rates of infection at every level of the map view, separating each view into distinct segments, based on where the colour code lines fall on the view.
b. The information displayed on the map must be taken from the NHS COVID-19 figures and gathered density information from our server’s database.
c. The map should provide the user with information regarding COVID-19 regulations and restrictions of the selected segment of the map.

d. Information of regulations and restrictions must be taken from official government sources.

## 3. Finding safest routes

3.1 The user must be allowed to choose a starting point.
a. The user should be allowed to choose his location as his starting point (if his location services are enabled on his phone)
b. The user should be allowed to select his starting point by typing the address.
c. The system should provide address auto-complete options for the user once there is user input.
3.2 The user must have the ability to select his destination point.
a. The user should be able to click a point on the map for selecting his desired destination.
b. The user should be able to select his destination by typing the address.
c. The system should provide address auto-complete options for the user once there is user input.
3.3 The user must be allowed to select his preferred way of reaching his destination (e.g. by public transport, car or on foot).
3.4 The user should be able to input his maximum extra time desired to travel.
3.5 - The user should be shown at least 2 paths and at most 4 paths that will take him from his starting point to his selected destination.
a. For each path, the user should be able to see the estimated time it takes to reach the destination.
b. For each path, the user must be informed about how dangerous the path is, as follows: use numbers from 0 to 100 to illustrate the danger level. If the number is between 0 and 30, display that number with a red colour to emphasize that the path is very dangerous. If the number is between 31 and 60, display that number with a yellow colour. If the number is between 61 and 100, display that number with a green colour.
c. Each path should be highlighted on the map.
3.6 The user should be allowed to navigate to his destination and his location should be updated on the map in real time with an active Internet connection and location services turned on.
3.7 If the user's location services are off / is not connected to the Internet , the user should be presented with the instructions of how to get to his desired destination, highlighting the main checkpoints with red pins on the map. (e.g. bus station).
3.8 The user should be informed when he arrives at the selected destination and then given the option to go back to the main screen if the device has an active Internet connection.

## 4. Declaration of positive case

4.1 The system must allow the user to declare themselves COVID-19 positive on the app.
4.2 The user must scan a QR code/enter an alphanumeric code provided by the NHS to declare him/herself COVID-19 positive on the app.
4.3 The system must provide a button for the user to declare themselves COVID-19 positive.
4.4 The system must have the means to scan the provided NHS QR code.

4.5 The system should provide relevant information regarding self-isolation when a positive case is registered.
4.6 The system must notify the users in-contact with the COVID-19 positive user within a period of 7 days of the danger of infection and recommend them to take a test.
4.7 The system should provide information regarding the symptoms and recommended actions in case the illness gets worse.
4.8 The system must provide a button to declare oneself as COVID-19 negative 10 days after a registering of a positive case.

## 5. User guidance

5.1 The app should provide the user with information concerning the main functions of the app and how it works and how it can be best utilised.

a. The app should show the user what information will be taken from the user and how it will be used in every function of the app, in order to have transparency around data use.

b. The app should provide the user with examples of how the app functions will help with dealing with the COVID-19 pandemic.

c. The app should tell the user what the main goal of each functionality is.

## 6. Data

6.1 The system should be able to read QR codes.
6.2 The system should receive the unique beacon information when the user is within a radius of 3 meters from a beacon.
6.3 The system must send both the beacon information and the user information to the server.

a. If there's no connected internet, the system should connect to the installed wifi routers.

6.4 The server must calculate the number of users based on the user information provided by the app.

a. The system should count the number of users street by street.
b. The system should decrement the count for that beacon by one for each user identifier that is detected at another beacon.

6.5 The server must keep a record of the user's location history through recording which beacons they passed while they travelled.

# Non-Functional Requirements

## 1. Efficiency (Performance)

1.1 The system should respond within a reasonable amount of time to ensure that the user does not believe the system has crashed.
1.2 The system should show the latest colour coded map within 10 seconds.
1.3 The system must update the colour coded map every 10 minutes.
1.4 The system should authenticate the user's declaration code and declare the user COVID-19 positive or inform the user the code was incorrect within 10 seconds.
1.5 The system should provide the user with the safest route to their destination within 30 seconds.
1.6 The system should notify users in-contact with a COVID-19 positive person in 30 minutes after declaration.
1.7 The system should react to the user's input (zoom in and out, moving around) in the dynamic map in 1 second.
1.8 The system should display the correct level and detail of the map in 1 second.

## 2. Efficiency (Space)

2.1 The servers must have sufficient storage to support location data of 70 million users.
2.2 The system must be able to record and save the location history of the past 7 days of each user in the server's database.
2.3 The system must be able to record and save COVID-19 declarations of each user in the local storage of the device and server's database.

## 3. Usability

3.1 The system should be easy to use and navigate for all types and ages of users.
3.2 The system must have a simple and consistent graphical interface which displays the functions in a concise way.
3.3 The system must provide a button for on-screen help which explains the currently selected function and legends for any map in use.
3.4 The system must take no longer than 5 minutes to be understood how to navigate for less experienced users.

## 4. Interoperability

4.1 The system must correctly receive data from government databases for case numbers, NHS case codes and regulations concerning COVID-19.
4.2 The system must correctly receive Bluetooth beacon data from the user device.
4.3 The system must correctly receive data from the server database.

## 5. Scalability

5.1 The system should be constructed such that it can be scaled to accommodate an increased number of users.

a. The system could utilise multiple identical servers, each of which can provide every service of the software.
b. The system should be designed that additional databases and servers can be easily added in order to expand the system.

## 6. Reliability, Availability and Maintainability

6.1 The system must be available at least 99% of the time 24/7 if not in maintenance.
6.2 The system should back up user data in a database every hour to prevent loss of data in case of a system malfunction.
6.3 The system should notify users about any software updates upon entering an older version of the app.
6.4 The system should be available to the user on older versions unless the version is no longer compatible.
6.5 The system must carry out software maintenance/updates at times of lowest application usage.
6.6 The system must notify users at least 24 hours beforehand about software downtimes due to maintenance.

## 7. Portability and compatibility

7.1 The software shall run on every mobile device with operating systems of at least Android 6.0 or iOS 10.
7.2 The user must own a mobile device with functional GPS and Bluetooth modules.
7.3 The software should not conflict with any other applications and processes.

a. The system should send a detailed report to the developing team with user permission in any case of malfunction.

7.4 The software should be as resource-efficient as possible, minimising RAM and CPU usage and keeping only vital processes running in the background if the app is not being directly used by the user.

## 8. Legislative (Privacy)

8.1 The system must provide a high-level of security at all stages to protect the privacy of the user's data.
8.2 The system must ensure that COVID-19 positive-case registration is secured.
8.3 The system must ensure that all user history data is private, secured and only accessible to the user.

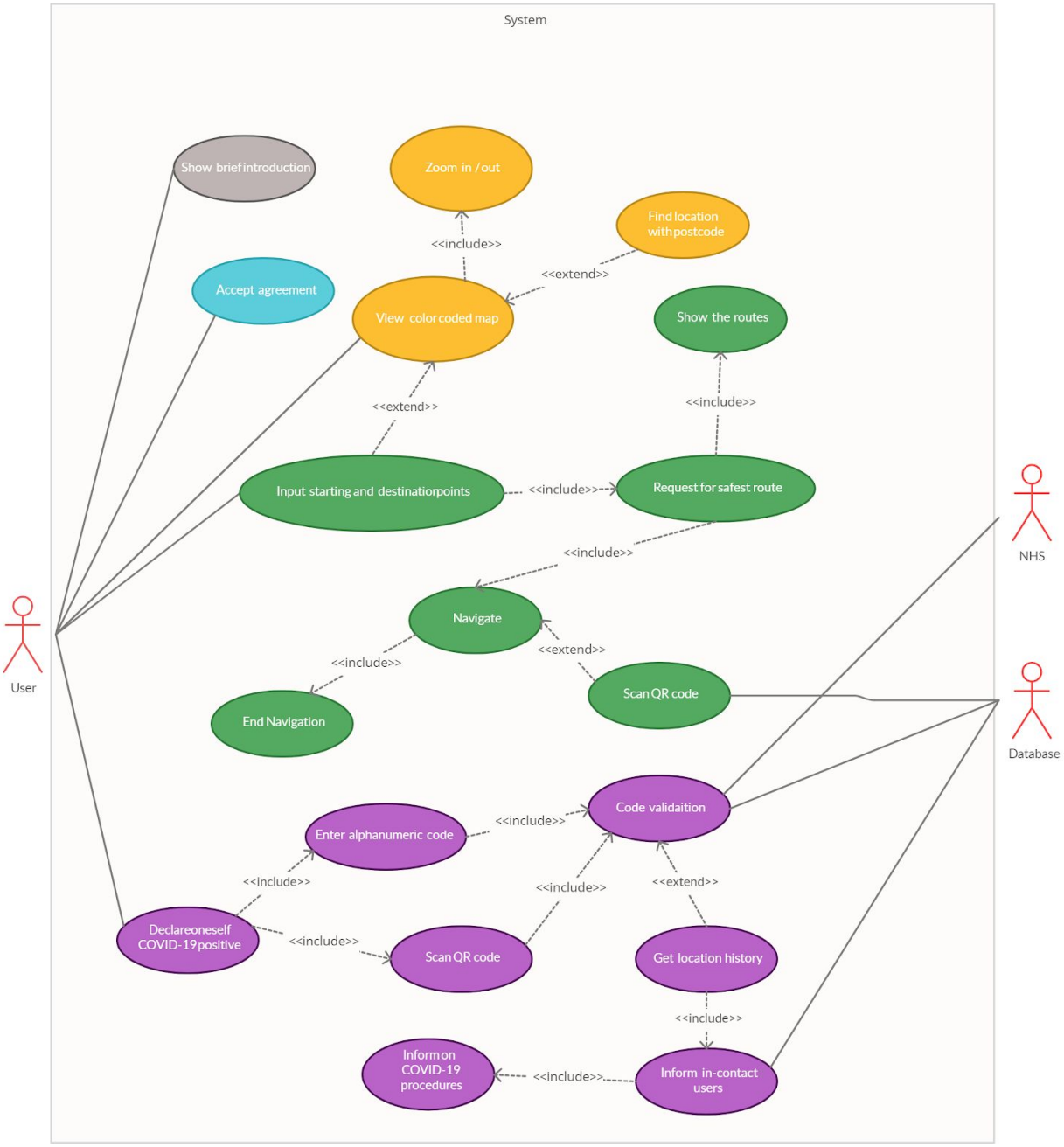
8.4 The system must keep a record of the user's location history up to the last two weeks in a private, secure database and discard any records later than this point.

## 9. Ethical

9.1 The system must correctly store and protect customer's personal information to ensure that it is properly secure and is not lost or tampered with.
9.2 The system must correctly notify the user's danger of infection in case of in-contact with a COVID-19 positive case user in an anonymous capacity.
9.3 The system must receive user permission before tracking the location of the user.

## B1. Use Case Diagram

The diagram presented below depicts the use cases and the relationships between them. The actors are shown outside of the system boundary, which is highlighted by the rectangle which surrounds the diagram. Two use cases have been selected for further analysis, using pre-conditions, post-conditions, flow of events and possible scenarios to describe them. Additionally, different colors have been used for the use case diagram in order to make it easier to distinguish between the different functionalities.  The diagram leaves out relationships of common functionalities that are implicit in order to preserve readability such as periodic updates while including those relationships for use cases that are manually started so are more transparent to the user


System
Show brief introduction
Zoom in / out
Find location with postcode
<<include>>
<<extend>>
Accept agreement
View color coded map
Show the routes
<<extend>>
<<include>>
Input starting and destinatiorpoints
<<include>>
Request for safest route
<<include>>
NHS
Navigate
<<include>>
<<extend>>
User
End Navigation
Scan QR code
Database
Code validaition
Enter alphanumeric code
<<include>>
<<include>>
<<include>>
<<extend>>
Declareoneself COVID-19 positive
<<include>>
Scan QR code
Get location history
<<include>>
Inform on COVID-19 procedures
<<include>>
Inform in-contact users

# B2. Documented Use Cases

| Use Case: safest routes COVID-19 Case | |
|---|---|
| Actors | User, Server, Bluetooth Beacons |
| Preconditions | 1. The user's device has Bluetooth turned on.<br>2. The system has the data related to the density of population (from the database).<br>3. All WiFi routers and beacons have enough battery power to perform their tasks.<br>4. The user's device has an active internet connection at the start of the use case. |
| Flow of Events | 1. The user selects the "Find safest route" option from the main menu.<br>2. The system prompts the user to select the starting and destination points.<br>a. If the user selects the starting point as his current location, the system gets his current location and sets it as the starting point.<br>b. The user manually types the starting and/or destination points.<br>c. If the user's starting or destination point is not valid, then the system will ask the user to provide a valid address.<br>3. The user selects his preferences regarding the means of transportation and the maximum amount of time they are willing to spend on the trip.<br>4. The system calculates the safest route from the starting point to the destination point.<br>5. The system shows the user up to 4 paths, including the safest one (according to his selected preferences). For each path, the user will be able to see the estimated time and the safety score of that path.<br>6. The user selects one of the presented paths.<br>7. The user starts to navigate to his destination.<br>a. While the user has access to wifi or mobile data, the system will guide the user to his destination by providing live directions regarding his chosen route<br>b. If the user can't access wifi or mobile data, then the system will show only the critical information regarding his journey (starting and destination points, check points, whole route highlighted on the map). |

| | |
|---|---|
| | c. While the user is navigating, he may choose to scan QR codes along the way.<br>8. The system informs the user of arrival when he/she reaches their destination.<br>9. The system gives the user the option to scan the QR code located at the user's destination.<br>a. If there is no QR code available at the destination point, the user can choose to skip this option and the system will return to the main screen. The information is sent to the server and the database gets updated correspondingly.<br>b. If the user scans the QR code, the information is sent to the server and the database gets updated correspondingly. |
| Postconditions | 1. The user's visiting record is sent to the server and the database is updated with the new information.<br>2. The user has successfully arrived at his destination.<br>3. The system has now returned to the main screen. |

(*'Checkpoints'* mentioned in the scenarios below are the key areas: such as bus stops, train stations, shops)

| Use Case: Declare COVID-19 Case | |
|---|---|
| Actors | User, NHS database, Servers |
| Preconditions | 1. The user has agreed with the terms and conditions.<br>2. The user's device has an active internet connection.<br>3. The user has not declared himself COVID-19 positive in the past 14 days. |
| Flow of Events | 1. The use case starts when the user selects the "Declare COVID-19 Positive" function from the main menu.<br>2. The system requests the user to select to declare the user positive.<br>3. The system requests the user to enter the NHS provided test result code, which can be either typed (alphanumeric code) or scanned (QR code).<br>a. If the code is invalid, the user is prompted to enter the code again.<br>b. If the code is valid, the use case proceeds to the next step.<br>4. The system shows a confirmation message that the positive case was declared.<br>5. The system shows a message with appropriate COVID-19 self-isolation and illness information.<br>6. The system returns the user to the main menu. |
| Postconditions | 1. The system has changed the status of the user to COVID-19 positive if the declaration was successful.<br>2. The server has matched the location history of the COVID-19 positive user to every user in the server database in contact with the COVID-19 positive user in the past 7 days.<br>3. The system has notified all users in contact with the COVID-19 positive user in the past 7 days.<br>4. The system allows the user to declare oneself as COVID-19 negative at least 10 days after a self-declaration of a positive case. |

# B3. Two scenarios for each documented use case

### Scenario 1 (for “safest route”):

User Joe has both mobile data and WiFi turned on. He chooses “Find safest route” from the main screen of the app. He is then required to choose a starting point, destination point and his maximum extra time spent to travel as well as his mode of transport. Joe decides to choose University Station as his starting point, the city centre as his destination, public transport as his mode of transport and 1hr maximum added travelling time.

He is presented with four “safe routes” each varying on the estimated time to take and the safety score all of which is within the specified 1hr maximum extra travel time.
The calculated safe routes are:

1. Travel time: 1hr, Safety Score 70
2. Travel time: 20min, Safety Score 20
3. Travel time: 40min, Safety Score 80
4. Travel time: 30min, Safety Score 35.

Joe decides to take the absolute safest route and chooses option 3 and clicks on the option. The route is then highlighted on the map. After clicking go, he is presented with a set of instructions to navigate while reflecting his exact location on the map concurrently. Along the route, Joe scans QR codes located at checkpoints through the integrated QR code scanner on the app. After successfully arriving at his destination, he scans the QR code and the app returns to the main menu.

### Scenario 2 (for “safest route”):

User Vadim has both mobile data and WiFi turned on. He chooses “Find safest route” from the main screen of the app. He is then required to choose a starting point, destination point and his maximum time wanted to travel as well as his mode of transport. Vadim decides to choose his current location as his starting point and his friend’s house located in Selly Oak as his destination, “on foot” as his mode of transport and 50 min maximum extra travelling time.

He is presented with four “safe routes” each varying on the estimated time to take and the safety score all of which is within the specified 50 min maximum added travel time.
The calculated safe routes are:

1. Travel time: 50min, Safety Score 80
2. Travel time: 20min, Safety Score 10
3. Travel time: 40min, Safety Score 50
4. Travel time: 30min, Safety Score 35

Vadim selects option 3 and is presented with a highlighted route on the map. After which, Vadim loses connection to the internet via mobile data and WiFi. He is then prompted through the app that he has lost connection to the internet. Vadim is then presented with a list of directions and checkpoints to navigate to his desired location. However, his phone automatically connects to the WiFi routers which are placed along his way (on every traffic light) .Vadim goes through the route scanning QR codes located at checkpoints along the way to his destination. Upon his arrival at the destination, he chooses to not scan any QR code and the app returns to the main menu.

### Scenario 3 (for “declaration”):

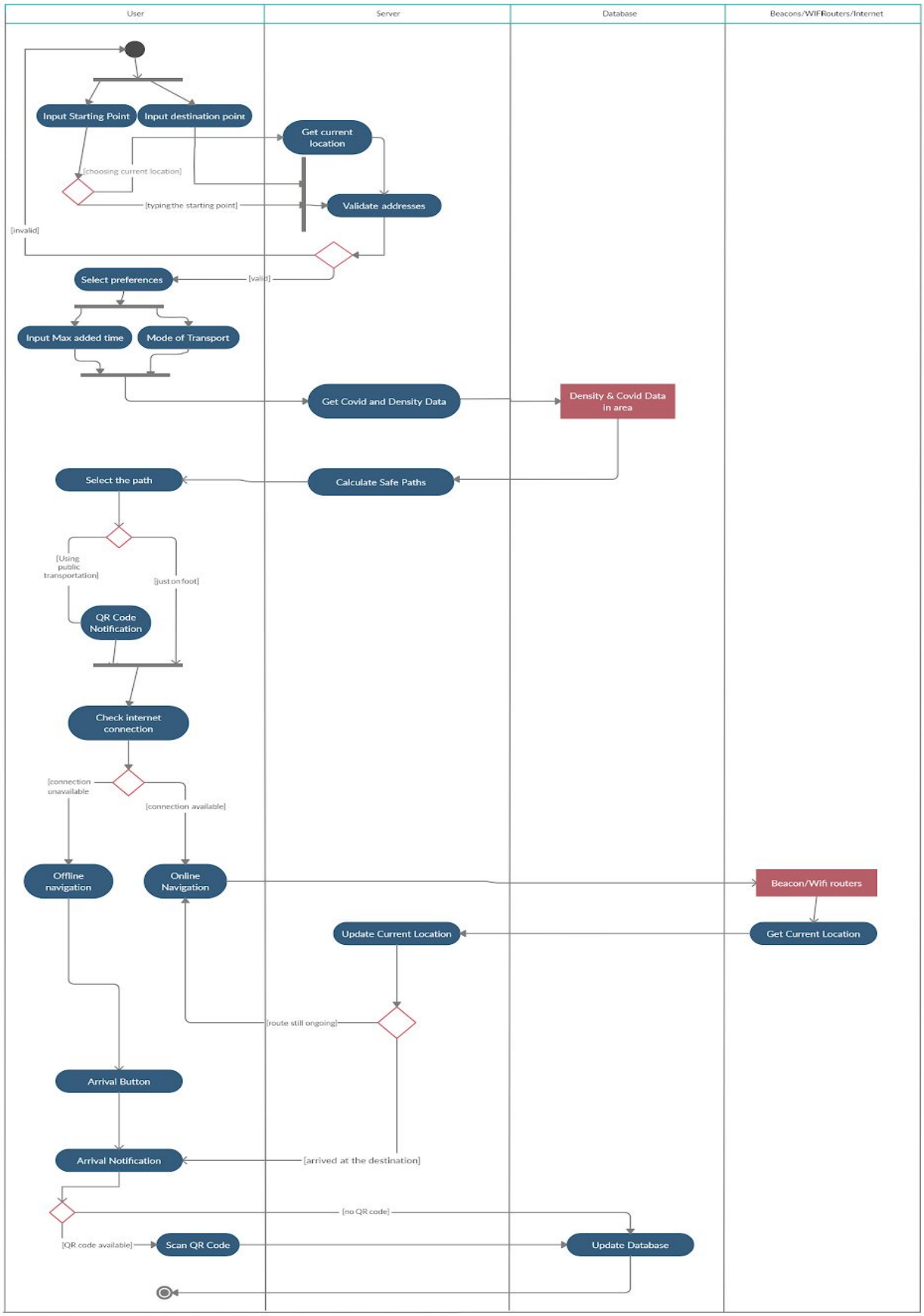
User Joe chooses to Declare COVID-19 Case from the main menu of the app and is requested to choose to declare himself as positive. He chooses to declare himself positive and the system prompts him to enter a NHS provided code. He enters the NHS provided alphanumeric code incorrectly and the system fails to verify the code. The system informs him the code is incorrect and redirects him to the code input screen. Joe then scans the QR code and the system checks with NHS databases if the code is valid which it is and the declaration is successful. The system then shows him a message confirming that he has declared himself COVID-19 positive and shows him information regarding COVID-19 and self-isolation. Meanwhile, users Marian, Vicentiu and Eusebiu are notified that they have been in contact with a person who has been found positive and are made aware of the potential danger of them being infected as well. The is then sent to the main menu. After 10 days Joe has no more symptoms and declares himself as COVID-19 negative.

### Scenario 4 (for “declaration”):

User Tifanny chooses to Declare COVID-19 Case from the main menu of the app and is requested to choose to declare herself as positive. She chooses to declare herself positive and the system prompts her to enter a NHS provided code. She enters the NHS provided alphanumeric code incorrectly. The system checks with NHS databases if the code is valid which it is not and the declaration is unsuccessful. The system redirects Tiffany to the code input screen. She chooses to scan the QR code. The system returns that the QR code is invalid and returns Tiffany to the scan screen. Tiffany exits the Declaring Positive Case function.

## B4. Activity Diagram

The following activity diagram details the order in which the events are going to happen in the first documented use case, namely “safest route”, which is one of the main functionalities of the app.


User
Server
Database
Beacons/WIFRouters/Internet
Input Starting Point
Input destination point
Get current location
[choosing current location]
[typing the starting point]
Validate addresses
[invalid]
[valid]
Select preferences
Input Max added time
Mode of Transport
Get Covid and Density Data
Density & Covid Data in area
Select the path
Calculate Safe Paths
[Using public transportation]
[just on foot]
QR Code Notification
Check internet connection
[connection unavailable]
[connection available]
Offline navigation
Online Navigation
Beacon/Wifi routers
Update Current Location
Get Current Location
[route still ongoing]
Arrival Button
Arrival Notification
[arrived at the destination]
[no QR code]
[QR code available]
Scan QR Code
Update Database

# B5. I. Noun-verb analysis and CRC

## Noun/Verb Analysis

| Potential Classes (Nouns) | |
|---|---|
| Agreement (attribute) | Navigation view (class) |
| Bluetooth beacon (component) | Positive case declaration (class) |
| Colour codes (class) | QRCodeScanner (class) |
| Danger level (attribute) | Safest route (class) |
| Further information (class) | Safety score (attribute) |
| Government database (component) | Server database (component) |
| InContact User (class) | Track and Trace (attribute) |
| Introduction view (class) | User (class) |
| Location QR code (component) | WIFI Router (component) |
| Map (class) | server connection (class) |
| NHS database (component) | |

| Potential Operations (Verbs/Methods) | |
|---|---|
| agreeToConditions() | matchUsersWithLocationHistory() |
| autoCompleteAddress() | navigateUser() |
| calcDangerLevel() | notifyInContactUsers() |
| calcInContactUsers(user) | removeOutdatedLocationHistory() |
| calcSafeRoutes() | saveNavOffline() |
| checkBluetoothActivation() | scanForBeacon() |
| checkWiFiActivation() | scanLocationQRCode() |
| chooseRoute() | scanNHSQRCode() |
| countUsersAtQRCode() | selectDestination() |
| createUserIdentifier() | selectModeOfTransport() |
| declareUserNegative() | selectStartLocation() |
| declareUserPositive() | sendBeaconInfo() - Beacon operation |
| deleteUser() | sendUserIdentifier() |
| enterAlphanumericNHSCode() | showColourCodedMap() |
| enterMaximumExtraTravelTime() | showColourCodes() |
| exitToMainMenu() | showFurtherInformation() |
| findUserLocation() | showIntroduction() |
| getBluetoothLocationHistory() | showKeyFunctions() |
| getConfirmedCases() | getMap() |
| getDensityOfUsers() | showSafestRoutes() |
| getQRLocationHistory() | showSafetyScore() |
| getRateOfInfection() | updateMap() |
| getRestrictions() | validateNHSCode() |
| mapZoom() | encryptUserData() |

| | |
|---|---|
| useQRScanner() | getBeaconLocationHistory() |
| getDestination() | getStartingLocation() |
| endNavigation() | startNavi() |
| getMaxExtraTravelTime() | getModeOfTransport() |
| createMap() | getColourCodes() |
| calcColourCodes() | retrieveConfirmedCases() |
| getFurtherInformation() | |

## CRC

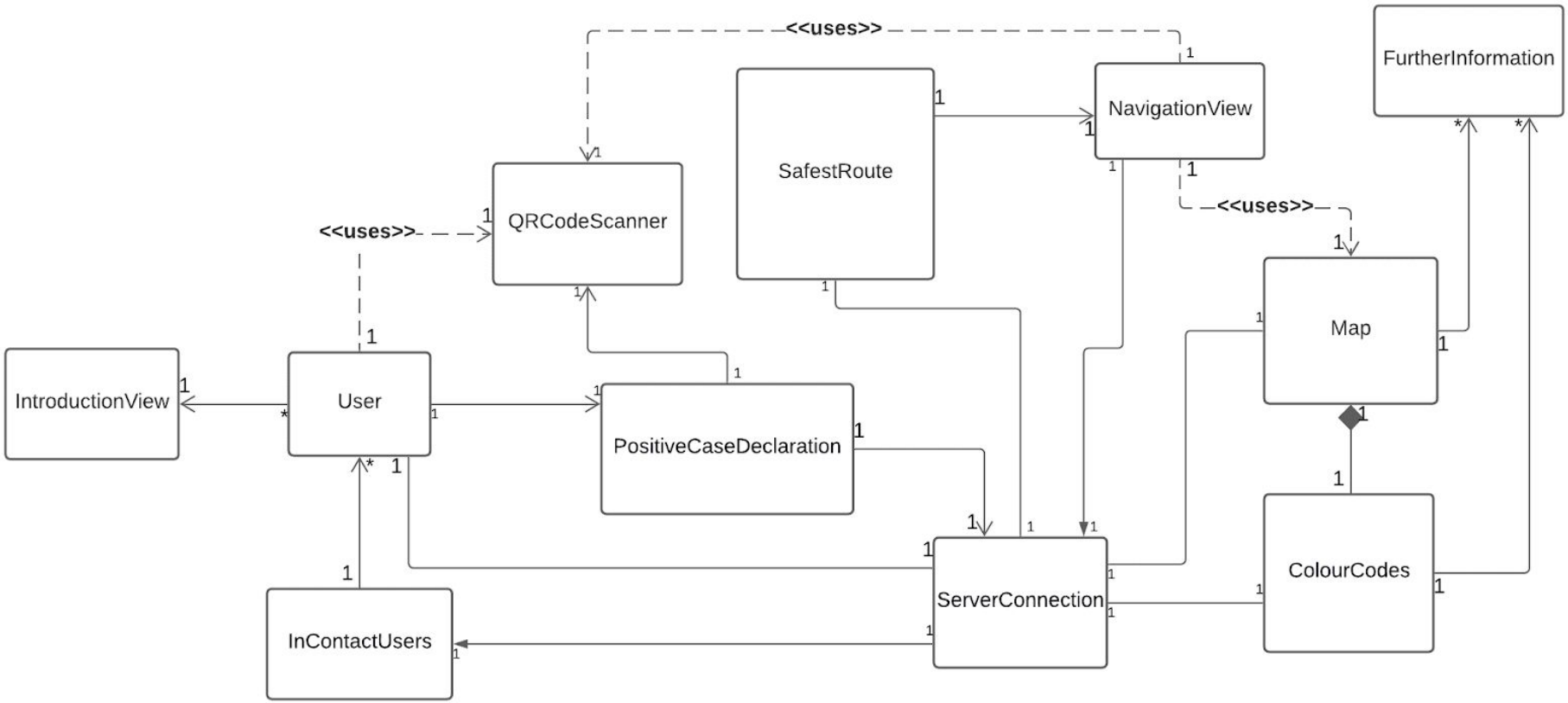
| Class | Collaborators |
|---|---|
| **ColourCodes**<br>● Provides a colour overlay to the map signifying different danger levels<br>● Shows colour codes from up to two weeks ago | Map<br>FurtherInformation<br>ServerConnection |
| **FurtherInformation**<br>● Provides information about restrictions and regulations for the chosen area<br>● Provides information about the number of confirmed cases and trend rates of infection | NHS database<br>GOVUKDatabase |
| **InContactUser**<br>● Maintain data on the positive-case user.<br>● Should enable the system to match location data to a recently declared positive case user from Location History<br>● Should be able to notify the InContactUser of the danger of infection. | User<br>ServerConnection |

| | |
|---|---|
| **IntroductionView**<br>● Ensures all users agree to the terms and conditions before proceeding to the app.<br>● Maintains data relating to whether the user agreed for introductory walkthrough.<br>● Displays key information about functionalities the app contains if the user wished to. | User |
| **Map**<br>● Provides a map view of the UK<br>● Allows the user to search for a location using a postcode<br>● Shows the next or previous levels (up to 4) of the map when the user zooms in and out. | Google API<br>Colour codes<br>FurtherInformation<br>User |
| **NavigationView**<br>● Navigate the chosen route on the map<br>● Provides QRCodeScanner for users to scan checkpoints | QRCodeScanner<br>SafestRoute<br>Map<br>ServerConnection |
| **PositiveCaseDeclaration**<br>● Maintain data relating to their case and time of declaration.<br>● Maintain and provide data on self-isolation and other advice regarding COVID-19. Should be able to update the information from the NHS.<br>● Should be able to accept and validate an alphanumeric/QR code and send the used code to the server to prevent it from being reused<br>● Should change User's case status data. | User<br>ServerConnection |

| | |
|---|---|
| **QRCodeScanner**<br>● Provides the user with a method to scan a QR code to make a COVID-19 case declaration inside the app and record user location with location QR codes. | User<br>PositiveCaseDeclaration |
| **SafestRoute**<br>● Calculates and shows various paths from the starting area to the destination using danger levels and mode of transport to show safety scores and travel times for every route | Map<br>NavigationView |
| **ServerConnection**<br>● Provides and stores user location history in the past 7 days.<br>● Maintains the data about declared-positive users and beacon/location QR code data.<br>● Provides and maintains an up to date count of all users at beacon locations and location QR codes | Positive Case Declaration<br>Safest Route<br>Map<br>User |
| **User**<br>● Maintain and provide access to user's data, including ID, account and case status.<br>● Receive notifications from the server if they have been in contact with a positive user.<br>● Receive information from Bluetooth beacons and send this information with a unique identifier to the server. | PositiveCaseDeclaration<br>ServerConnection<br>QRCodeScanner |

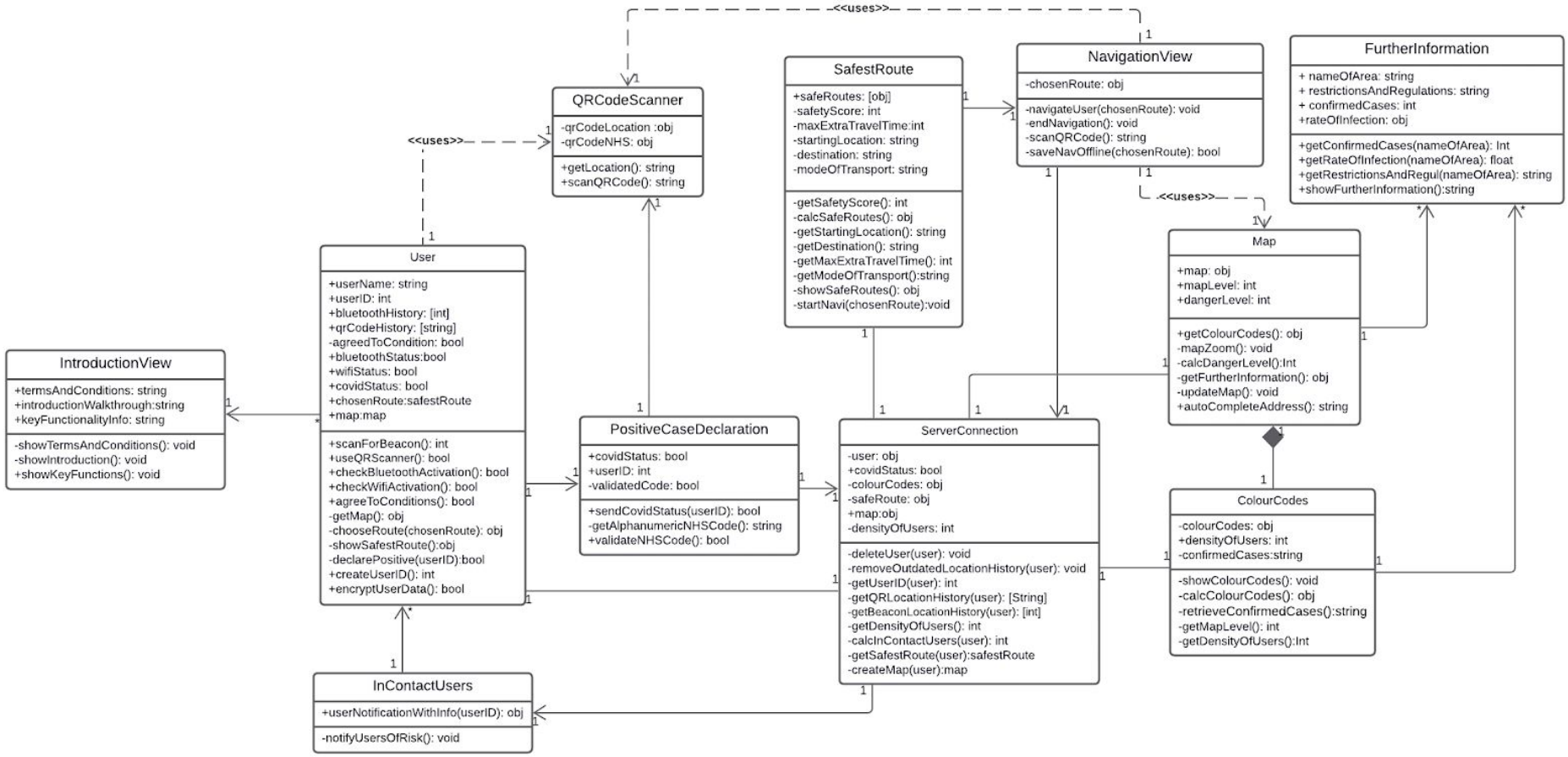
## B5. II. First-Cut Class Diagram

The first-cut class diagram presented below shows how the classes will be linked together. The diagram has been made by using the CRC cards, as well as the noun-verb analysis.


<<uses>>
FurtherInformation
NavigationView
SafestRoute
QRCodeScanner
<<uses>>
<<uses>>
Map
IntroductionView
User
PositiveCaseDeclaration
ColourCodes
ServerConnection
InContactUsers

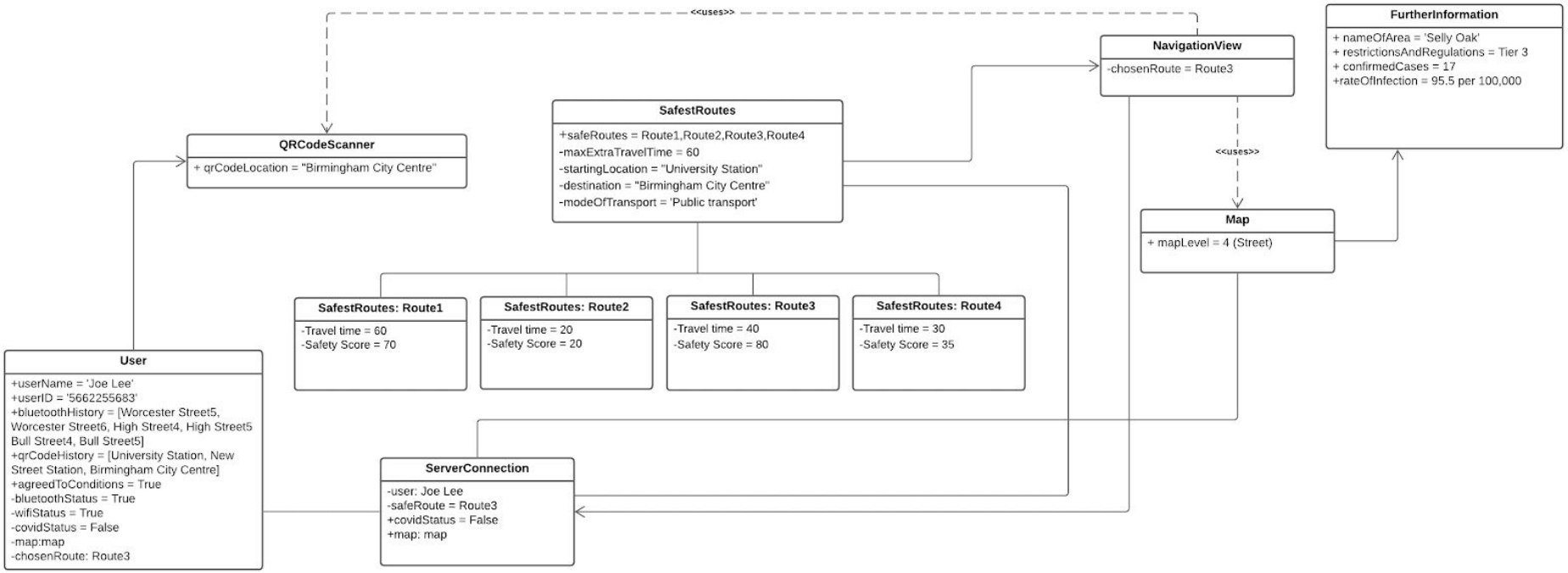
## B5. III. Full-class diagram

The next page presents the full class diagram. Not only does it offer information about what each class will contain, but it also shows the structure of the design. It is structured as a facade design pattern


QRCodeScanner
-qrCodeLocation :obj
-qrCodeNHS: obj
+getLocation(): string
+scanQRCode(): string
SafestRoute
+safeRoutes: [obj]
-safetyScore: int
-maxExtraTravelTime:int
-startingLocation: string
-destination: string
-modeOfTransport: string
-getSafetyScore(): int
-calcSafeRoutes(): obj
-getStartingLocation(): string
-getDestination(): string
-getMaxExtraTravelTime(): int
-getModeOfTransport():string
-showSafeRoutes(): obj
-startNavi(chosenRoute):void
NavigationView
-chosenRoute: obj
-navigateUser(chosenRoute): void
-endNavigation(): void
-scanQRCode(): string
-saveNavOffline(chosenRoute): bool
FurtherInformation
+ nameOfArea: string
+ restrictionsAndRegulations: string
+ confirmedCases: int
+rateOfInfection: obj
+getConfirmedCases(nameOfArea): Int
+getRateOfInfection(nameOfArea): float
+getRestrictionsAndRegul(nameOfArea): string
+showFurtherInformation():string
Map
+map: obj
+mapLevel: int
+dangerLevel: int
+getColourCodes(): obj
-mapZoom(): void
-calcDangerLevel():Int
-getFurtherInformation(): obj
-updateMap(): void
+autoCompleteAddress(): string
User
+userName: string
+userID: int
+bluetoothHistory: [int]
+qrCodeHistory: [string]
-agreedToCondition: bool
+bluetoothStatus:bool
+wifiStatus: bool
+covidStatus: bool
+chosenRoute:safestRoute
+map:map
+scanForBeacon(): int
+useQRScanner(): bool
+checkBluetoothActivation(): bool
+checkWifiActivation(): bool
+agreeToConditions(): bool
-getMap(): obj
-chooseRoute(chosenRoute): obj
-showSafestRoute():obj
-declarePositive(userID):bool
+createUserID(): int
+encryptUserData(): bool
IntroductionView
+termsAndConditions: string
+introductionWalkthrough:string
+keyFunctionalityInfo: string
-showTermsAndConditions(): void
-showIntroduction(): void
+showKeyFunctions(): void
PositiveCaseDeclaration
+covidStatus: bool
+userID: int
-validatedCode: bool
+sendCovidStatus(userID): bool
-getAlphanumericNHSCode(): string
+validateNHSCode(): bool
ServerConnection
-user: obj
+covidStatus: bool
-colourCodes: obj
-safeRoute: obj
+map:obj
-densityOfUsers: int
-deleteUser(user): void
-removeOutdatedLocationHistory(user): void
-getUserID(user): int
-getQRLocationHistory(user): [String]
-getBeaconLocationHistory(user): [int]
-getDensityOfUsers(): int
-calcInContactUsers(user): int
-getSafestRoute(user):safestRoute
-createMap(user):map
ColourCodes
-colourCodes: obj
+densityOfUsers: int
-confirmedCases:string
-showColourCodes(): void
-calcColourCodes(): obj
-retrieveConfirmedCases():string
-getMapLevel(): int
-getDensityOfUsers():Int
InContactUsers
+userNotificationWithInfo(userID): obj
-notifyUsersOfRisk(): void
<<uses>>
<<uses>>
<<uses>>

## B6. Object Diagram

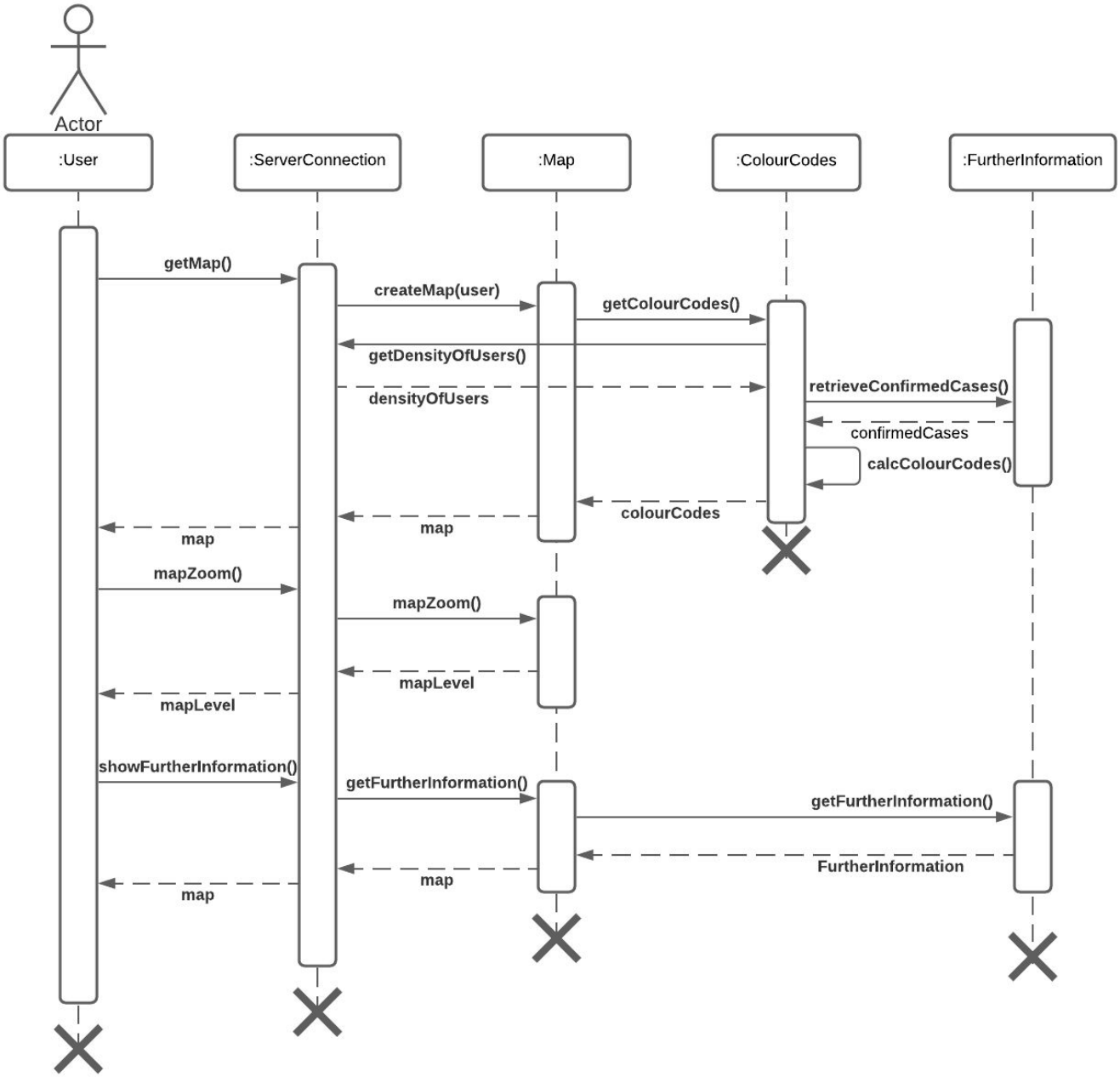
This diagram is based on the first scenario of the first use case, namely about User Joe in the ‘safest route’ use case. This is at the endpoint when Joe reaches his destination of Birmingham City Centre and declares his journey over by scanning the QR code at the destination.


<<uses>>
NavigationView
-chosenRoute = Route3
FurtherInformation
+ nameOfArea = 'Selly Oak'
+ restrictionsAndRegulations = Tier 3
+ confirmedCases = 17
+rateOfInfection = 95.5 per 100,000
QRCodeScanner
+ qrCodeLocation = "Birmingham City Centre"
SafestRoutes
+safeRoutes = Route1,Route2,Route3,Route4
-maxExtraTravelTime = 60
-startingLocation = "University Station"
-destination = "Birmingham City Centre"
-modeOfTransport = 'Public transport'
<<uses>>
Map
+ mapLevel = 4 (Street)
SafestRoutes: Route1
-Travel time = 60
-Safety Score = 70
SafestRoutes: Route2
-Travel time = 20
-Safety Score = 20
SafestRoutes: Route3
-Travel time = 40
-Safety Score = 80
SafestRoutes: Route4
-Travel time = 30
-Safety Score = 35
User
+userName = 'Joe Lee'
+userID = '5662255683'
+bluetoothHistory = [Worcester Street5, Worcester Street6, High Street4, High Street5 Bull Street4, Bull Street5]
+qrCodeHistory = [University Station, New Street Station, Birmingham City Centre]
+agreedToConditions = True
-bluetoothStatus = True
-wifiStatus = True
-covidStatus = False
-map:map
-chosenRoute: Route3
ServerConnection
-user: Joe Lee
-safeRoute = Route3
+covidStatus = False
+map: map

# B7. Sequence Diagrams

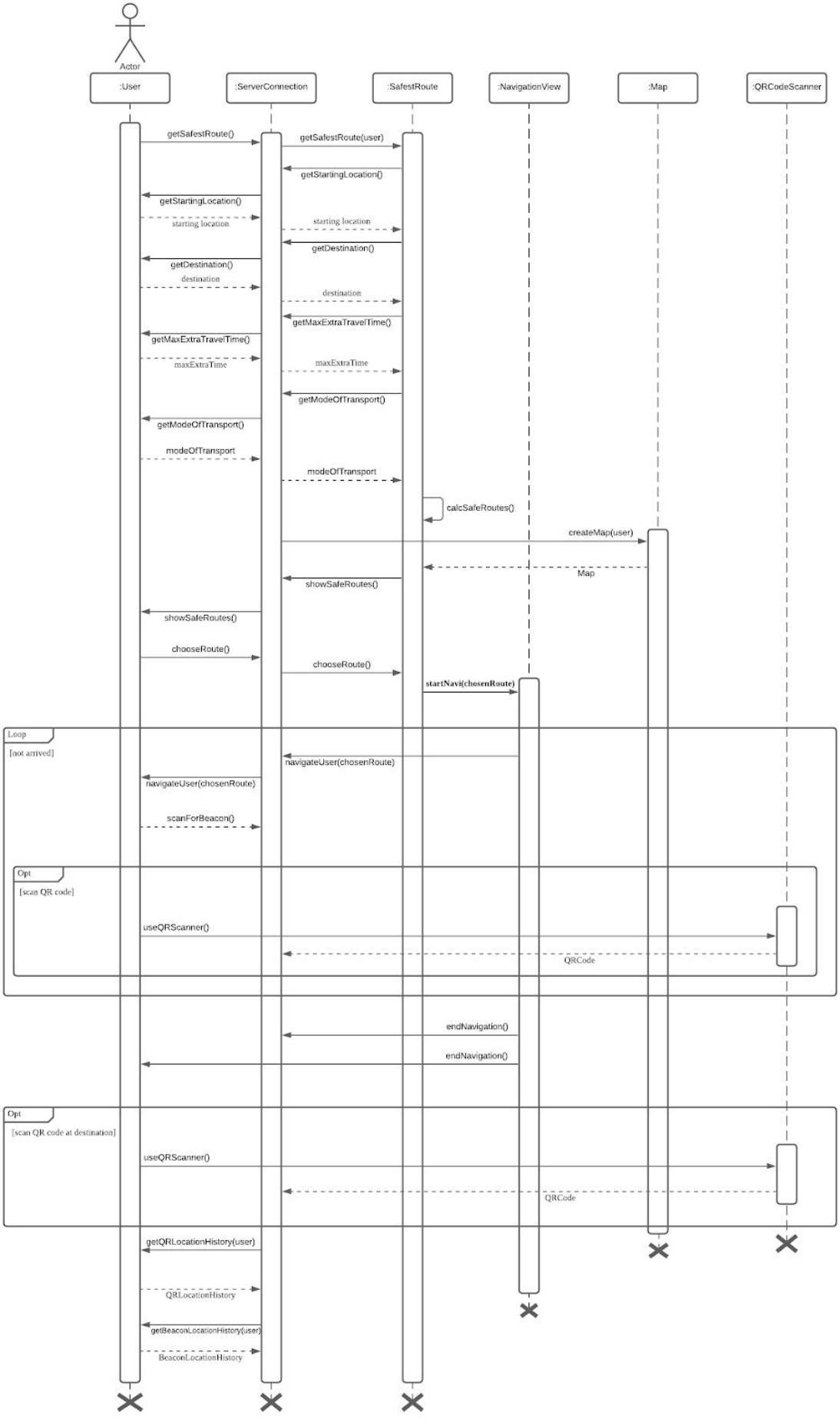
## Sequence Diagram (View Colour Coded Map):

This is a sequence diagram for a scenario where the user views the colour coded map and looks for further information about a selected area. Our diagram only displays the specific actions that the user makes according to the scenario, meaning that while this diagram shows one possible interaction with our system, it is by no means meant to be exhaustive so doesn't cover the whole scope of this feature. The interaction starts when the user enters the app and by default is shown the colour coded map. The user then zooms in on the map and is shown different levels of the map according to the zoom level applied. They then select a location, the map retrieves further information such as total cases and local restrictions and regulations of the selected location and shows it to the user. The user requests a map from the ServerConnection class, which then creates a new map using the Map class. Because our class diagram was made using the facade design pattern, any interaction between the user and other classes passes through the ServerConnection class, which is why any function call from the user such as mapZoom() or showFurtherInformation() is repeated by the ServerConnection class which finally communicates with that specific class, in this case, the Map class. The reverse applies as well, meaning that any return values from other classes have to go through the ServerConnection class to reach the user. Also, when we create new objects in the sequence diagram, we use the corresponding functions such as createMap() to mean the same thing as writing <<create>>. The user is able to use the function mapZoom() because the user class has a map object from earlier when it was created by the ServerConnection class therefore has access to the Map class methods and attributes.


Actor
:User
:ServerConnection
:Map
:ColourCodes
:FurtherInformation
getMap()
createMap(user)
getColourCodes()
getDensityOfUsers()
densityOfUsers
retrieveConfirmedCases()
confirmedCases
calcColourCodes()
colourCodes
map
map
mapZoom()
mapZoom()
mapLevel
mapLevel
showFurtherInformation()
getFurtherInformation()
getFurtherInformation()
FurtherInformation
map
map

## Sequence Diagram(View Safest Route):

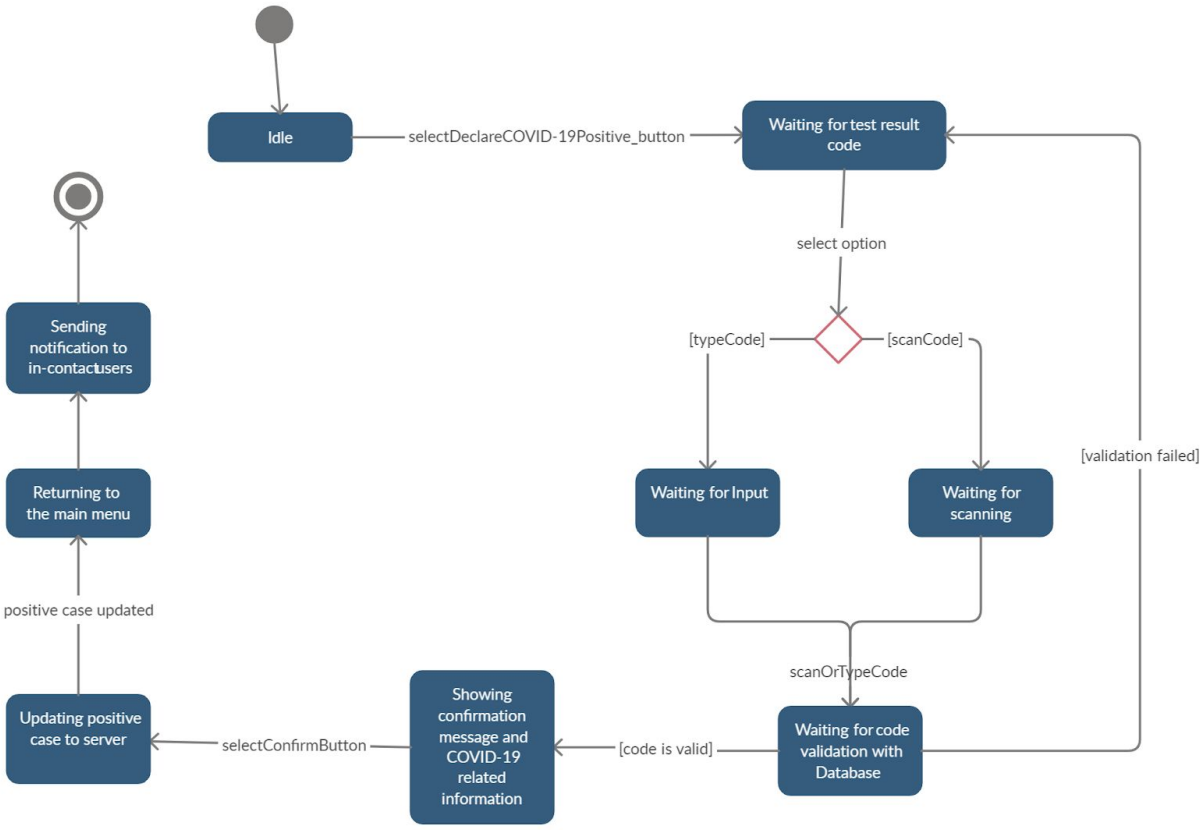
This is a sequence diagram for a scenario where the user views the safest route from their starting location to their destination in the selected time frame. The interaction starts when the user enters the app and chooses to generate a safe route. The user then inputs their starting location, in this scenario their current location, their destination, mode of transport and the maximum extra time they can take to travel safely. The user then chooses their route and starts travelling, being guided by the navigation view until they reach their destination. During their journey, they can scan QR codes available on public transport if this is their chosen mode of transport. Finally, the user scans the QR code available at the destination. Similarly to the sequence diagram above, we create a safest route object in the ServerConnection class when the user sends the function call getSafestRoute(), meaning that we are able to access methods and attributes of the SafestRoute class from the user and ServerConnection classes. Furthermore, the same methods are used from the ServerConnection class to the SafestRoute class to show the communication between both classes in first creating safe routes for the user to choose from and then again when the user chooses a route and returns to the SafestRoute class to start the navigation view. The same is done again when the ServerConnection uses the function createMap(user) in order that attributes and methods of the Map class can be used in the ServerConnection class. When we create new objects in the sequence diagram, we use the corresponding functions such as createMap() to mean the same thing as writing <<create>>. At the end of the diagram, the ServerConnection class retrieves the bluetooth beacon location history and the QR code location history from the user. Because our class diagram was made using the facade design pattern, any interaction between the user and other classes passes through the ServerConnection class, just as the previous sequence diagram. Our diagram only displays the specific actions that the user makes according to the scenario, meaning that while this diagram shows one possible interaction with our system, it is by no means meant to be exhaustive so doesn’t cover the whole scope of this feature.


Actor
:User
:ServerConnection
:SafestRoute
:NavigationView
:Map
:QRCodeScanner
getSafestRoute()
getSafestRoute(user)
getStartingLocation()
getStartingLocation()
starting location
starting location
getDestination()
getDestination()
destination
destination
getMaxExtraTravelTime()
getMaxExtraTravelTime()
maxExtraTime
maxExtraTime
getModeOfTransport()
getModeOfTransport()
modeOfTransport
modeOfTransport
calcSafeRoutes()
createMap(user)
Map
showSafeRoutes()
showSafeRoutes()
chooseRoute()
chooseRoute()
startNavi(chosenRoute)
Loop
[not arrived]
navigateUser(chosenRoute)
navigateUser(chosenRoute)
scanForBeacon()
Opt
[scan QR code]
useQRScanner()
QRCode
endNavigation()
endNavigation()
Opt
[scan QR code at destination]
useQRScanner()
QRCode
getQRLocationHistory(user)
QRLocationHistory
getBeaconLocationHistory(user)
BeaconLocationHistory

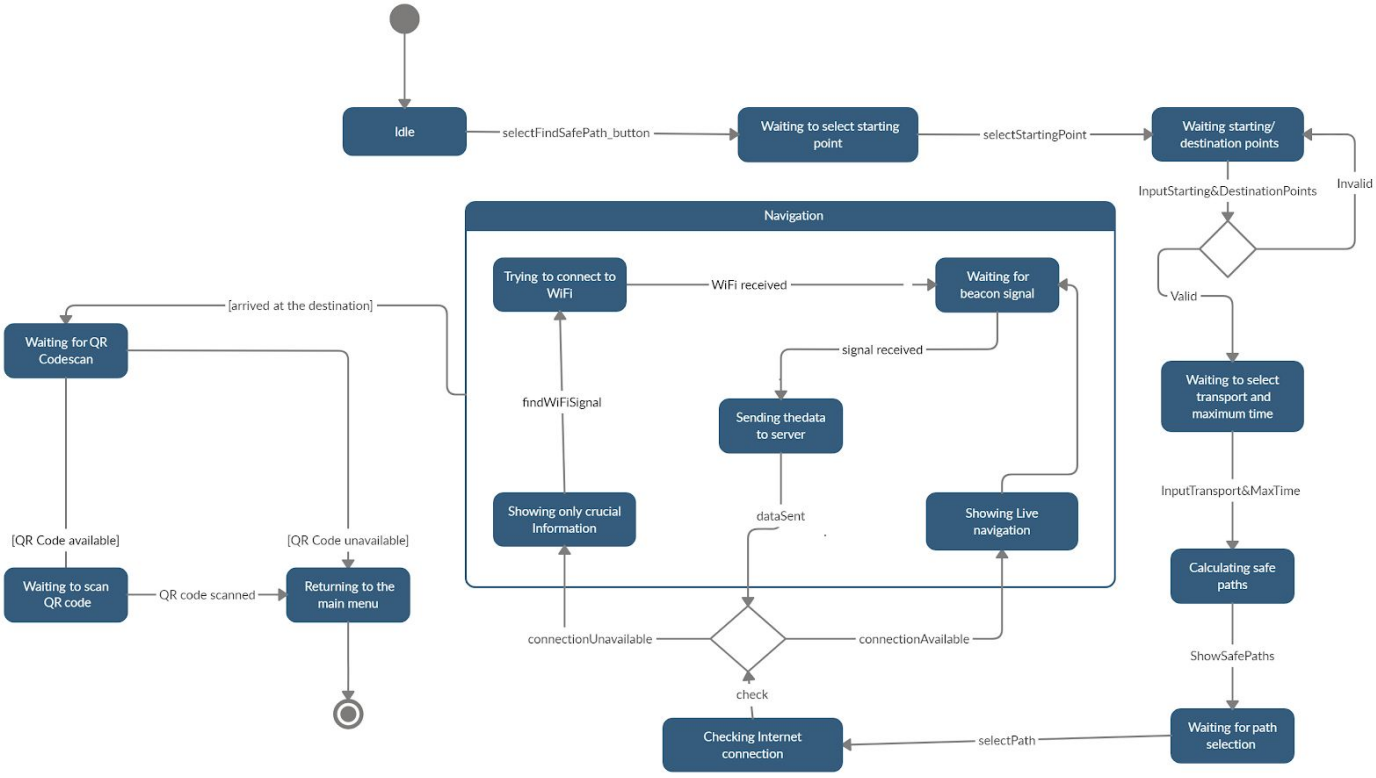
# B8. State Diagrams

Below, two state diagrams are provided for two main features of the app, namely “declaration” (which allows the user to declare a positive Covid-19 case) and “safest routes” (which the user can use to find multiple safe routes from their desired starting point to the imputed location).

## Declaration

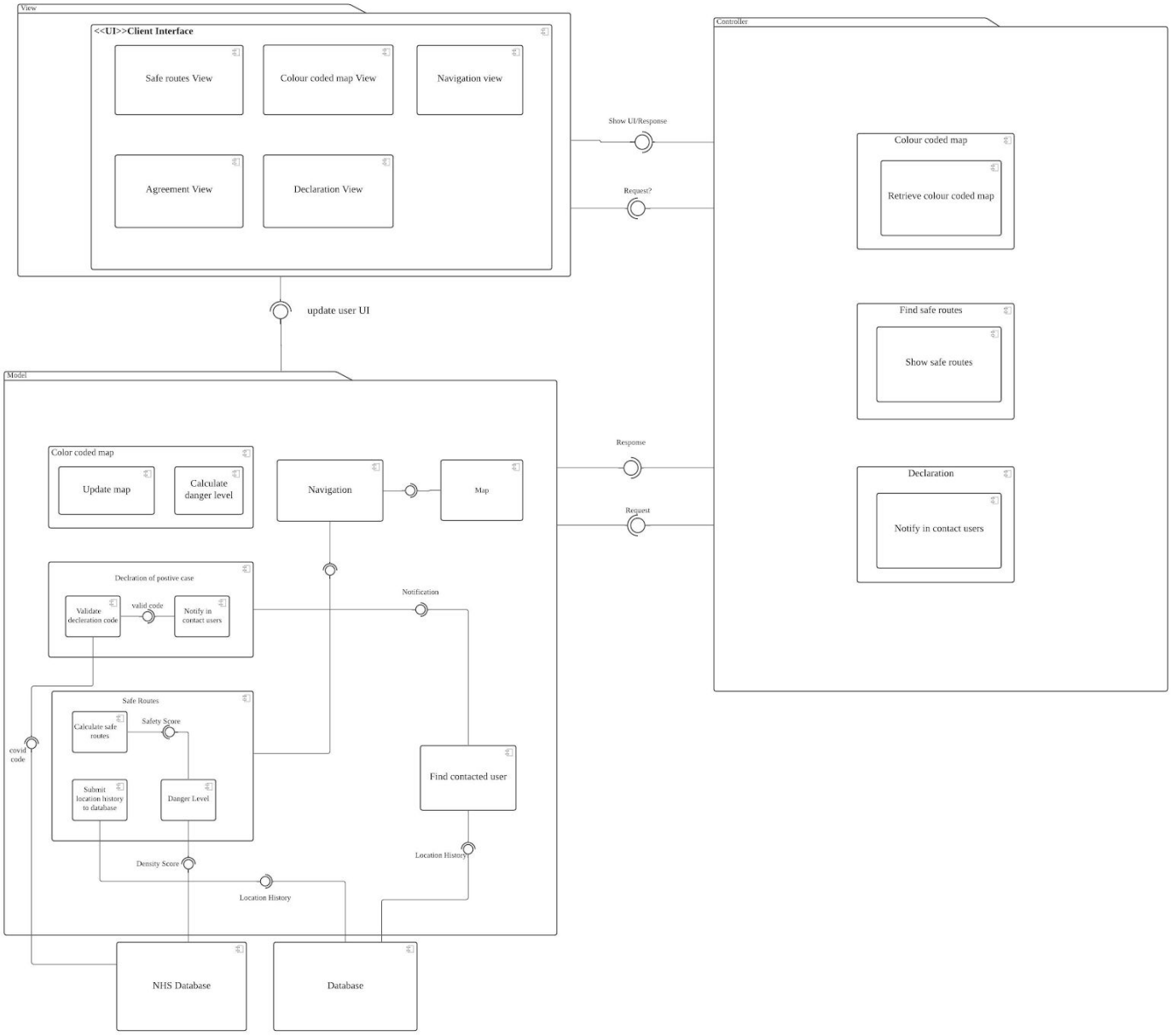


## Safest Routes

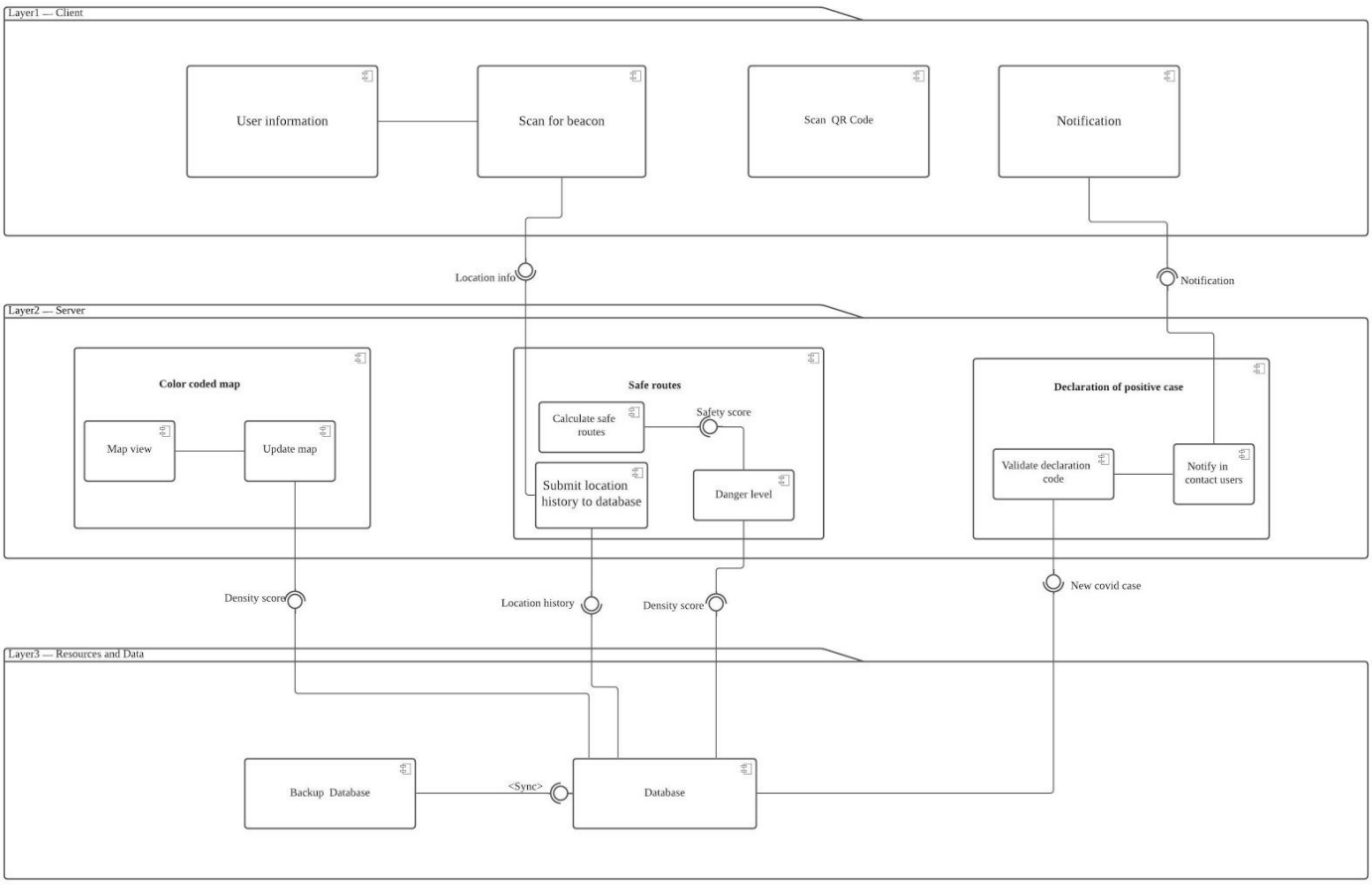


# C1. Component Diagrams

This diagram models our system's architecture after the MVC architecture (Model-View-Controller)

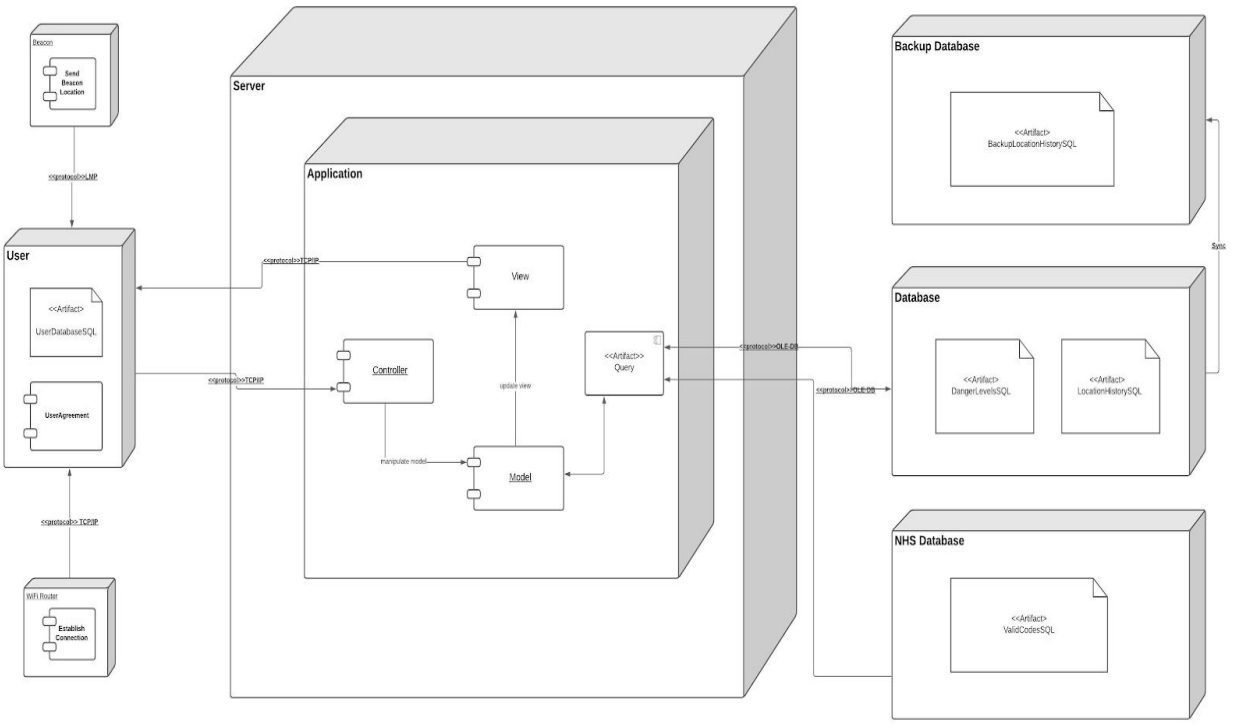


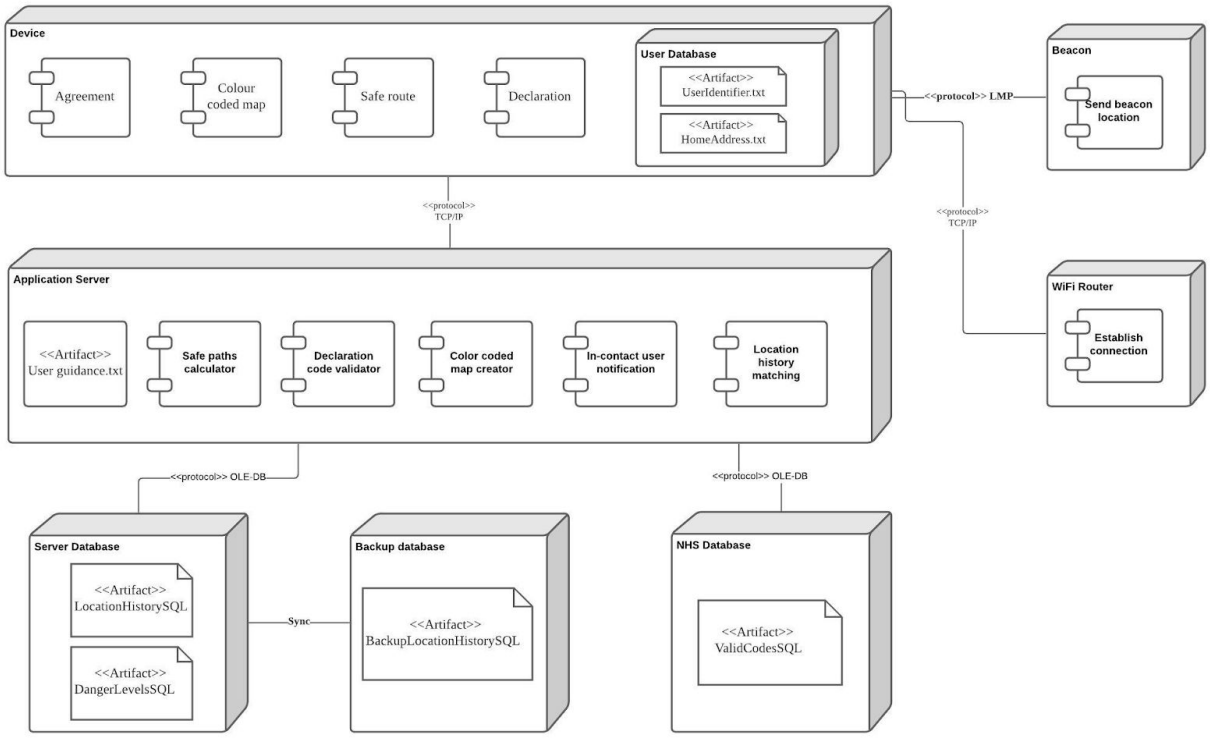
The following is the component diagram depicting the 3-tier architecture style:



## C2. Deployment Diagrams

The first deployment diagram corresponds to the MVC architecture style, while the second one presents the 3-tier architecture style.


Beacon
Send Beacon Location
<<protocol>>LMP
User
<<Artifact>>
UserDatabaseSQL
UserAgreement
<<protocol>> TCP/IP
WiFi Router
Establish Connection
Server
Application
<<protocol>>TCP/IP
View
Controller
update view
manipulate model
Model
<<Artifact>>
Query
<<protocol>>OLE-DB
Backup Database
<<Artifact>>
BackupLocationHistorySQL
Sync
Database
<<Artifact>>
DangerLevelsSQL
<<Artifact>>
LocationHistorySQL
NHS Database
<<Artifact>>
ValidCodesSQL


Device
Agreement
Colour coded map
Safe route
Declaration
User Database
<<Artifact>>
UserIdentifier.txt
<<Artifact>>
HomeAddress.txt
<<protocol>> LMP
Beacon
Send beacon location
<<protocol>>
TCP/IP
<<protocol>>
TCP/IP
Application Server
<<Artifact>>
User guidance.txt
Safe paths calculator
Declaration code validator
Color coded map creator
In-contact user notification
Location history matching
WiFi Router
Establish connection
<<protocol>> OLE-DB
<<protocol>> OLE-DB
Server Database
<<Artifact>>
LocationHistorySQL
<<Artifact>>
DangerLevelsSQL
Sync
Backup database
<<Artifact>>
BackupLocationHistorySQL
NHS Database
<<Artifact>>
ValidCodesSQL

## C3. Architecture Tradeoff Analysis:

Choosing the right architecture style is crucial for the development of any type of application. Not only does it dictate how the components will interact with each other, but it will also influence factors such as performance, security or reliability. Out of all the various possible design architectures, we decided to focus on the 3 tier and the Model-View-Controller (MVC) architectures. While the 3-tier architecture divides the application into presentation layer (UI), Business Logic layer (BLL) and Data Access Layer (DAL), the MVC architecture structures the app into 3 interconnected parts: Model, View and Controller.

Key advantages of the 3-tier architecture include extensibility and security. Regarding the disadvantages, performance degradation stands out most to us. Having the ability to expand our application is not crucial to the project as this application is relevant during the Covid-19 period and will not be very useful afterwards. Security-wise, this architecture boasts a very secure way to protect user data. However, we have an inbuilt way of randomising user IDs to protect user data.

Performance of our application is of utmost importance, at the rate at which people are getting infected, accuracy and speed are crucial to the success of this application. Unfortunately, this architecture comes with a huge performance hit as it has to always go through the BLL to get access to the DAL and this is going to slow down the updates given to the application on new positive cases.

One of the greatest advantages of choosing the MVC architecture is that it will allow us to develop the application much faster. Having multiple levels properly laid out in the application will allow for significantly faster debugging, updating the application will be easier and the developers will have an easier time working together. Everybody is suffering during this pandemic, so it is of the utmost importance that we deliver the system as fast and as efficiently as possible. However, this design can be difficult to grasp as it can get rather complex. Fully understanding such a complicated design will certainly take some time.

All in all, we have chosen the MVC architecture over 3-tier as it best suits our needs for the application. To reiterate, the performance of the application is a high priority and the 3-tier architecture has major performance issues compared to the MVC architecture. Multiple pros of the MVC like development speed and ease of updates found us leaning towards the MVC as our final choice.

# D. Software Testing

## Introduction:

The system is an app designed to assist people in the current COVID-19 pandemic. Its core functions are mainly broken down into :

- Allowing users to view a colour coded map which signifies COVID-19 danger level in various areas.
- Providing users with the safest routes for travelling and providing navigation through said routes.
- Warning users who are at risk of infection with the aid of user COVID-19 case declarations

## Objectives:

- Check whether the app functionality is working as expected without any error or bugs in a real-world environment.
- Ensure the app conforms to the functional and non-functional requirements.
- Critical criteria should be identified and fit specific requirements:
    - Response time of the system:
        - Response time for viewing and interacting with the map.
        - Response time for acquiring the safest routes.
        - Response time for declaring oneself COVID-19 positive/negative.
        - Response time when 100 simultaneous users are declaring and the system is verifying their COVID-19 positive case.
    - System works correctly and in a reasonable amount of time on every mobile device with operating systems from Android 6.0 or iOS 10.
    - Correct and accurate records of the location history of the user's past 7 days in the server's database.

## Test Items:

The systems to be tested include the app frontend and the database system.
These systems should be tested in all versions of Android/iOS fulfilling the minimum requirement.

## Features To Be Tested:

Features to be tested include the following:

- As a user, view the colour coded map.
- As a user, view the colour coded map on different levels of zoom.
- As a user, view local information of a location selected on the colour coded map.
- As a user, obtain at least two routes to a selected destination when requesting for the safest routes.
- As a user, obtain navigation to a selected destination through the selected route.
- As a user, obtain instructions with main checkpoints to travel to a selected destination when there is no active Internet connection.

- As a user, declare oneself COVID-19 positive after testing positive.
- As a user, declare oneself COVID-19 negative after the self-isolation period.
- The system filters out all users in-contact with a COVID-19 positive user in the past 7 days by matching location history data and notifies them.

The following non-functional requirements will also be tested:

- As a user, receive response from each function of the app within the timeframes listed below :
    - The colour coded map is shown to the user within 10 seconds.
    - The safest routes to a requested destination are shown to the user within 30 seconds.
    - The colour coded map is updated every 10 minutes.
    - Authentication of COVID-19 declaration code is completed within 10 seconds.
- As an in-contact user, receive a notification warning about risk of infection within 30 minutes after a declaration.
- As a user, run the application on a mobile device with all operating system versions equal to or higher than Android 6.0 or iOS 10.

## Features Not To Be Tested:

The COVID-19 code verification will not be tested. We do not have valid data available as it is yet to be implemented.
The system will not be tested on Android / iOS versions below Android 6.0 / iOS 10 and other operating systems such as Windows Mobile.
Security aspects of data transfer and storage will not be tested as standard pre-existing encryption methods will be used.

## Testing Strategy:

The quality team will create a test set for each tester using white-box and black box testing methods. The tester will execute the tests in TestLog and mark each case as Pass / Fail. The tester should leave notes on actual results and any other relevant details when possible.

When tests are marked as Fail, bug reports will automatically be created and assigned to a developer. The developer makes the change and returns it back to the responsible tester. The test manager reviews the test report in Testlog for final approval.

## Pass Fail Criteria:

('*Actual results*', '*Pass/Fail*' and '*Test comments*' columns to be filled during the tests)

| Test Case ID | Test description | Test steps | Test Data | Expected result | Actual result | Pass/ Fail | Test comments |
|---|---|---|---|---|---|---|---|
| FT-01 | Verify the default | - Open the app | | - System should by default be showing a | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | function of the app: colour coded map of UK | - Check if the colour codes on the map are correct. | | colour coded map of the UK with the correct colour codes. | | | |
| FT-11 | Zoom in on the map to city level | - Open the app<br>- Zoom-in on the colour coded map until the dynamic map changes to city view | Target city: Birmingham, UK | - System should show a colour coded map with the selected city zoomed in | | | |
| FT-12 | Zoom in on the map to the postcode level | - Open the app<br>- Zoom-in on the colour coded map until the dynamic map changes to city view then to postcode view | Target postcode area: B1 | - System should show a colour coded map with the selected area zoomed in | | | |
| FT-13 | Zoom out on the map to the country level | - Open the app<br>- Zoom-in until the dynamic map changes to city level then zoom-out back to country level | Target: UK | - System should show a colour coded map with the selected wide area | | | |
| FT-21 | Get local information of a selected postcode area | - Open the app<br>- Zoom-in on the desired area.<br>- Press on the area | Target location: B2 | - System should show the number of cases and local restrictions of the selected area in a pop-up. | | | |
| FT-22 | Get local information of a selected city in the UK | - Open the app<br>- Press on the city on the map. | Target: London | - System should show the number of cases and local restrictions information of the selected in a pop-up. | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| FT-31 | Obtain the safest route for a valid destination from a typed starting location | - Open the app<br>- Select the traveling function<br>- Type in the starting address<br>- Type in the destination address<br>- Select maximum travel time and preferred mode of transport | - Starting address: 61 Albion St, Birmingham<br>- Destination address: University of Birmingham<br>- Maximum time: 2h<br>- Mode of transport: Bus | - System should show the map without the colour codes and generate at least two routes from the starting position to the destination, highlighting them on the map. | | | |
| FT-32 | Obtain the safest route for a valid destination from current location | - Open the app.<br>- Select the travelling function.<br>- Select “Use current location”<br>- Type in the destination address.<br>- Select maximum travel time and preferred mode of transport. | - Starting address: Current address of the tester<br>- Destination address: 34 Waterworks Rd, Birmingham<br>- Maximum time: 1.5h<br>- Mode of transport: Bus | - System should show the map without the colour codes and generate at least two routes from the current location to the destination, highlighting them on the map. | | | |
| FT-33 | Obtain the safest route for an invalid destination from current location | - Open the app.<br>- Select the travelling function.<br>- Select “Use current location”<br>- Type in the invalid destination address. | - Starting address: Current address of the tester<br>- Destination address: 25 abc | - System should not generate the safest routes, instead generating a pop-up message with “Invalid address” | | | |
| FT-41 | Obtain the safest route by public transport for a valid destination | - Open the app.<br>- Select the travelling function. | - Starting address: Current address of the tester | - System should show the map without the colour codes and generate at least two routes by valid public | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | from current location | - Select “Use current location”<br>- Type in the destination address.<br>- Select maximum travel time and to travel by public transport. | - Destination address: 34 Waterworks Rd, Birmingham<br>- Maximum time: 1.5h<br>- Mode of transport: Public transport | transport from the current location to the destination, highlighting them on the map. | | | |
| FT-42 | Obtain the safest route by bus for a valid destination from current location | - Open the app.<br>- Select the travelling function.<br>- Select “Use current location”<br>- Type in the destination address.<br>- Select maximum travel time and to travel by bus. | - Starting address: Current address of the tester<br>- Destination address: 34 Waterworks Rd, Birmingham<br>- Maximum time: 1.5h<br>- Mode of transport: Public transport | - System should show the map without the colour codes and generate at least two routes from the current location to the destination, highlighting them on the map. | | | |
| FT-43 | Obtain the safest route by foot for a valid destination from current location | - Open the app.<br>- Select the travelling function.<br>- Select “Use current location”<br>- Type in the destination address.<br>- Select maximum travel time and to travel on foot. | - Starting address: Current address of the tester<br>- Destination address: 34 Waterworks Rd, Birmingham<br>- Maximum time: 1.5h<br>- Mode of transport: Public transport | - System should show the map without the colour codes and generate at least two routes accessible on foot from the current location to the destination, highlighting them on the map. | | | |
| FT-51 | Obtain navigation to a destination through the selected route with an active | - Open the app.<br>- Select the travelling function. | - Starting address: Current address<br>- Destination address: 34 Waterworks | - System should provide interactive navigation to the destination until the user selects “Stop Navigation” or the | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | internet connection. | - Select “Use current location”<br>- Type in the destination address.<br>- Select maximum travel time and preferred mode of transport.<br>- Select one of the provided routes.<br>- Select “Navigation” | Rd, Birmingham<br>- Maximum time: 2h<br>- Mode of transport: Bus<br>- Selected route:1st | user has arrived at the destination. | | | |
| FT-52 | Obtain navigation to a destination through the selected route without an active internet connection | - Open the app.<br>- Select the travelling function.<br>- Select “Use current location”<br>- Type in the destination address.<br>- Select maximum travel time and preferred mode of transport.<br>- Select one of the provided routes.<br>- Select “Navigation” | - Starting address: Current address of the tester<br>- Destination address: 34 Waterworks Rd, Birmingham<br>- Maximum time: 2h<br>- Mode of transport: Bus<br>- Selected route: 1st | - System should provide main instructions to travel to the destination, highlighting main checkpoints on the route until the user selects “Stop Navigation” or the user arrives at the destination. | | | |
| FT-61 | Declare the user’s positive COVID-19 case with a valid testing alphanumeric code | - Open the app.<br>- Select the declaring positive case function.<br>- Select “Declare oneself COVID-19 positive” | Valid test code | - System should accept the valid code and mark the user as COVID-19 positive, showing information on the virus and self-isolation and starting the countdown (10 days) until the user can declare oneself | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | - Type in the valid test code | | as COVID-19 negative.<br>- System should also update the case in the database. | | | |
| FT-62 | Declare the user's positive COVID-19 case with a valid testing QR code | - Open the app.<br>- Select the declaring positive case function.<br>- Select "Declare oneself COVID-19 positive"<br>- Scan the valid test QR code. | Valid test QR code | - System should accept the valid code and mark the user as COVID-19 positive, showing information on the virus and self-isolation and starting the countdown (10 days) until the user can declare oneself as COVID-19 negative.<br>- System should also update the case in the database. | | | |
| FT-63 | Try to declare the user's positive COVID-19 case with an invalid testing alphanumeric code | - Open the app.<br>- Select the declaring positive case function.<br>- Select "Declare oneself COVID-19 positive"<br>- Type in the invalid test code | Invalid test code | - System should not declare the user COVID-19 positive, instead returning a pop-up message informing the user that the code is invalid and returning the user to the declaration screen. | | | |
| FT-64 | Declare the user's positive COVID-19 case with an invalid testing QR code. | - Open the app.<br>- Select the declaring positive case function.<br>- Select "Declare oneself COVID-19 positive"<br>- Scan the invalid testing QR code. | Invalid QR Code | - System should not declare the user COVID-19 positive, instead returning a pop-up message informing the user that the code is invalid and returning the user to the declaration screen. | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| FT-81 | Declare oneself as COVID-19 negative after the self-isolation period ends. | - Open the app<br>- Select the declaring positive case function.<br>- Select “Declare oneself as COVID-19 negative” | | - System should mark the user as COVID-19 negative and reset the declaration function, allowing the user to declare oneself positive again. | | | |
| FT-83 | Declare oneself as COVID-19 negative before the self-isolation period ends. | - Open the app<br>- Select the declaring positive case function. | | - System should not allow the user to declare himself negative, rather showing the remaining time until the self-isolation period is over. | | | |
| FT-91 | Notify users in-contact with a COVID-19 positive person in the last 7 days. | - Add 20 test users and location history data with 10 of the test users being in-contact with each other in the past 7 days while the other users have zero contact with the other users.<br>- One of the in-contact test users declares themselves COVID-19 positive. | Testing QR code | - System should filter out the users in contact with the positive case test user in the last 7 days by matching the location history of the users.<br>- System should then send a notification of the danger of infection to the in-contact users. | | | |
| NFT-01 | Show the colour coded map to the user within 10 seconds. | - Open the app<br>- Record the time required for the colour coded map to show. | | - The map should finish loading on the user’s mobile device within 10 seconds. | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| NFT-02 | Show the safest routes to a destination to the user within 30 seconds. | - Open the app<br>- Select the traveling function<br>- Type in the starting address<br>- Type in the destination address<br>- Select maximum travel time and preferred mode of transport<br>- Record the time required for the app to show the safest routes. | - Starting address: Current address of the tester<br>- Destination address: 34 Waterworks Rd, Birmingham<br>- Maximum time: 1.5h<br>- Mode of transport: Public transport | - The app should display at least two routes within 30 seconds. | | | |
| NFT-03 | Update the colour coded map every 10 minutes. | - Open the app<br>- View the colour coded map<br>- Add test data to the database to alter the danger level and colour codes.<br>- View the map again after 10 minutes. | | - The app should display a different and updated map with the altered information after 10 minutes. | | | |
| NFT-04 | Authenticate COVID-19 declaration within 10 seconds. | - Open the app<br>- Select “Declare COVID-19 Case” function.<br>- Enter a test code.<br>- Record the time required for the app to respond to the test code. | Testing code | - The app should authenticate the COVID-19 declaration code and show the authentication result within 10 seconds. | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| NFT-10 | As an in-contact user, receive a notification warning about risk of infection within 30 minutes after a declaration | - Open the app<br>- Add test COVID-19 positive user data to the tester's location history data.<br>- Record the time required for the tester to receive the notification. | | - The app should notify the in-contact user about the risk of infection within 30 minutes after the COVID-19 declaration data is sent to the server. | | | |
| NFT-20 | As a user, run the application on a mobile device with all operating system versions equal to or higher than Android 6.0 or iOS 10. | - Open the app with a device with an operating system equal to or higher than the minimum requirements.<br>- View the colour coded map and view various areas for local information.<br>- Request for safest routes to a destination.<br>- Make a COVID-19 case declaration.<br>- View the user guidance in each function. | | - The app should function normally and not crash when any of the functions are carried out on a device satisfying the minimum operating system requirements. | | | |

## **Exit Criteria:**

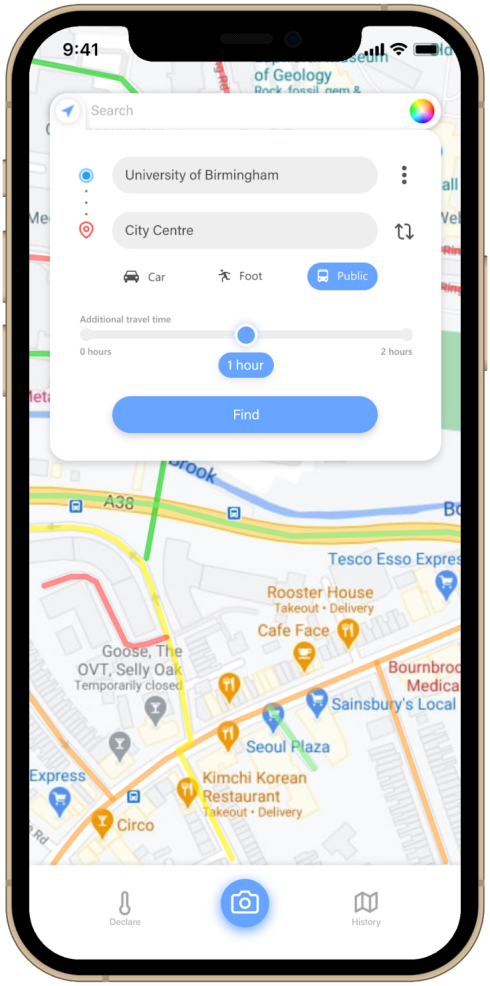
95% of all test cases should pass with no failed critical cases.

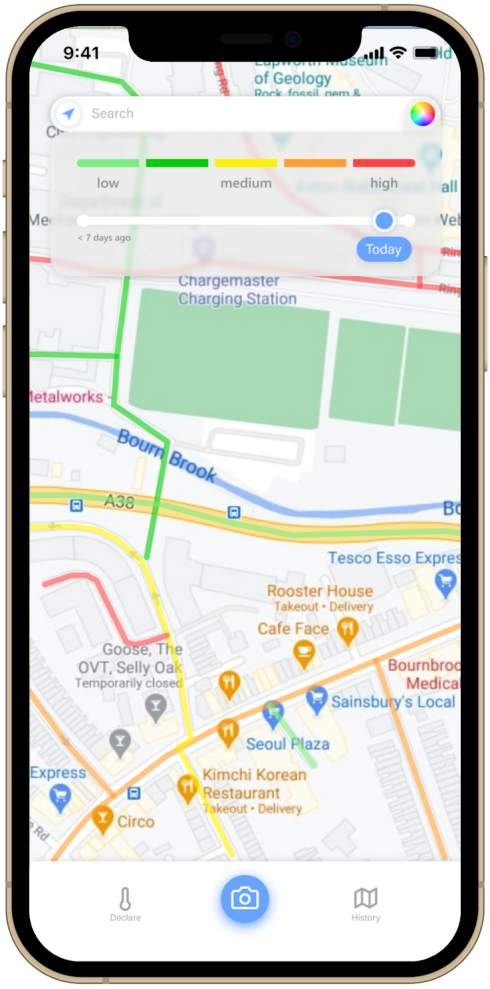
# E. Usability and Prototyping

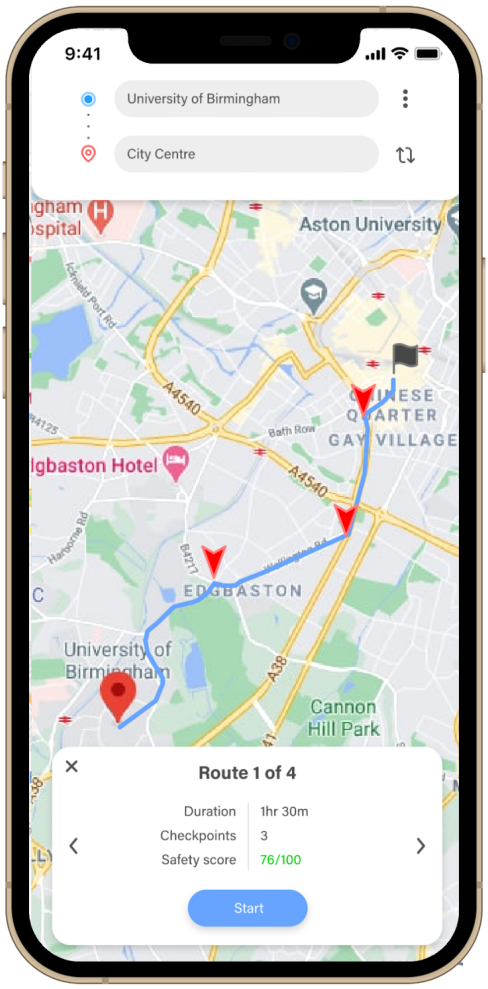
## E1. Interactive prototypes

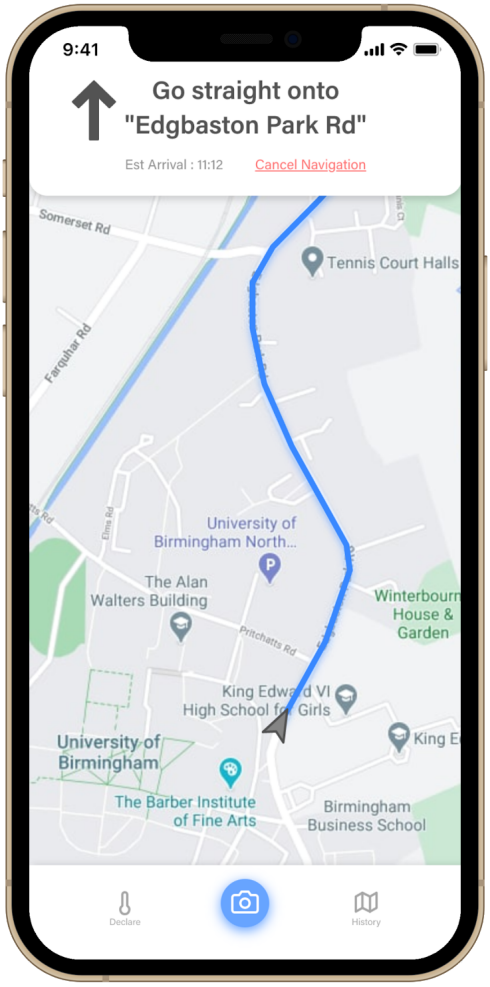
The following represent 5 prototypes, each of them showing a main screen:

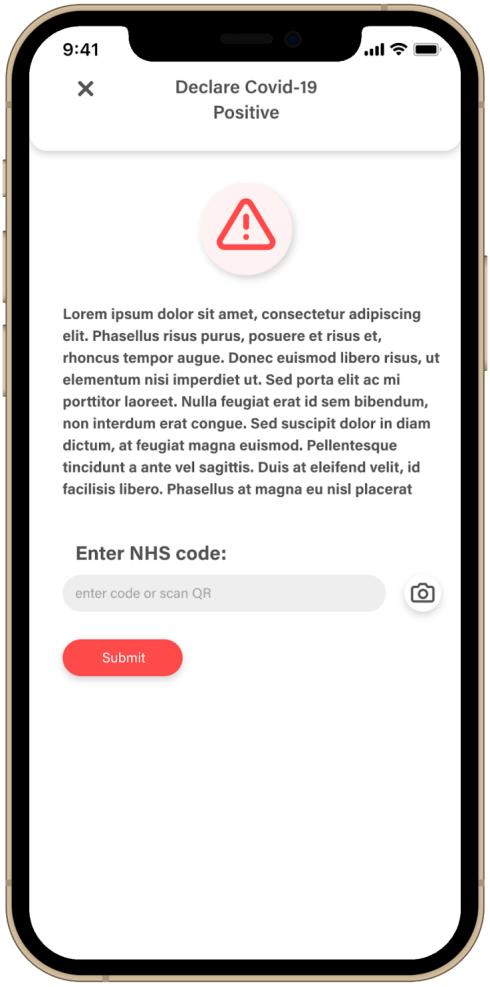
a) First prototype image: color coded map view
b) Second prototype image: safe routes view 1
c) Third prototype image: safe routes view 2
d) Fourth prototype image: navigation view
e) Fifth prototype image: declaration of positive case view




9:41
Search
low
medium
high
< 7 days ago
Today
Chargemaster
Charging Station
Bourn Brook
A38
Tesco Esso Expres
Rooster House
Takeout • Delivery
Cafe Face
Goose, The
OVT, Selly Oak
Temporarily closed
Sainsbury's Local
Seoul Plaza
Express
Kimchi Korean
Restaurant
Takeout • Delivery
Circo
Declare
History


9:41
University of Birmingham
City Centre
Aston University
A4540
Bath Row
EDGBASTON
University of
Birmingham
A38
Cannon
Hill Park
Route 1 of 4
Duration
1hr 30m
Checkpoints
3
Safety score
76/100
Start




## E2. Video recording

(The video is attached on Canvas)

# F. Ethics and Professional Practice : Self-Appraisal

The challenge was to create an application to provide a societal value and support during the COVID-19 period. Our group has come up with the above application concept in order to fulfil the requirements. In our attempt, we have been actively considering and following the main principles of the ACM code of ethics.

In the design of our software, we strive to contribute to society by minimizing the risk of COVID-19 infection for every single person in the general public, consistent with the aims of Principle 1.1. The application will be completely public and accessible by every member of society, following Principle 1.4 of the ACM code. We are aware that the implementation of our application would collect highly sensitive data which is abusable in the wrong hands. Hence, we aim to maintain a high level of security to prevent the abuse of the data collected which could potentially cause harm to all users, exemplifying important aspects of Principle 1.2 and Principle 2.9. Instead of a custom encryption method discussed during initial stages, standard cryptographic algorithms will be implemented to protect the privacy of the data collected, in compliance with Principle 2.6. All data collected will only be used for the purpose of this software to prevent any breach of privacy, showing commitment to Principle 1.6. We will also comply with the DPA and UK-GDPR in our handling of data.

In the case of an actual implementation of the software, we aim to achieve and maintain high quality in the process and product of our work, complying with Principle 2.1. The team would actively seek professional review to identify crucial issues and constantly improve the software, a key aspect of Principle 2.4. We would also ensure that all operations of our application comply with both domestic and international laws, in regard to Principle 2.3.

During creation, implementation and maintenance of the system, in accordance to Principle 4.1 and 4.2, the code of ethics will be followed throughout all the project and any breaches of the code by any member of the team will not be tolerated and reasonable actions will be taken against the transgressor. Furthermore, we will strive to ensure that the public good remains the central concern during the whole process, in regards to Principle 3.1.

After the implementation of the system, we will monitor the system and its integration in society, modifying the app to meet the changing needs of the public or retiring the system in a safe and responsible way if it becomes redundant after the COVID-19 pandemic is over, in respect to Principles 3.6 and 3.7.

In our work, we, as a team, strive to follow all key principles in the ACM code of ethics wherever applicable and includes but is not limited to the principles stated above.