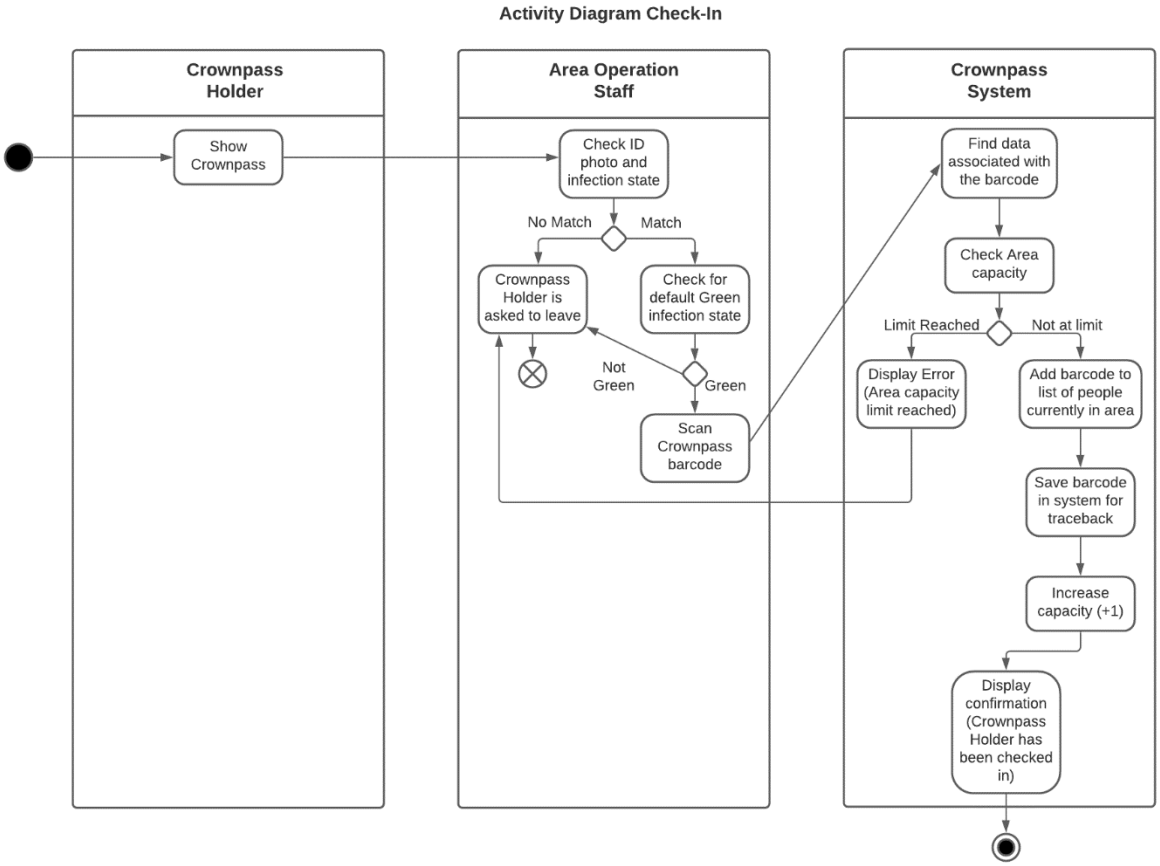
Subsystem B – Area Owner and Operation Staff

1b) Activity Diagram



1b) Activity List

Area Operation Staff – Check-in

Activity List:

Use Case: Check-in

Actors: Crownpass Holder wanting to enter an area and Area Operation Staff who will allow them to enter the premises

Entry Conditions:

1) Crownpass Holder must be a registered user with information in database

2) Crownpass Holder must have their Crownpass either showing on their mobile device or printed out on a sheet of paper to pass to the Operation Staff on entry

3) There is no evacuation in progress in the Area

Exit Conditions:

1) Crownpass ID is added to area database including time of entry for traceback

2) Capacity of an area is increased by 1

3) Operation Staff receives a confirmation of a successful check in and let the Crownpass Holder enter the premises

Events:

1) Crownpass Holder shows their Crownpass on their mobile phone or a printed version to the Operation Staff

2) Operation Staff manually verifies the Holder against ID photo shown on the Crownpass:

- If photo does not match the Crownpass Holder is told to exit the premises as only a valid holder will be able to be checked-in

- If photo matches the Operation staff will look for a default infection state “Green” on the Crownpass

3) Operation staff verifies the infection state of Crownpass Holder:

- If the infection state is not “Green” the Crownpass Holder is told to exit the premises

- If infection state is “Green” the Operation Staff will scan the Crownpass ID using their mobile phone

4) System finds data associated with the Crownpass ID

5) System checks the area capacity (compares the number of holders present in the area to the maximum capacity of an area):

- If the capacity is reached the system will display an error on the mobile phone of the Operation Staff (“Area Capacity Limit Reached”) and the Crownpass Holder will be told to exit the premises

- If the capacity is not reached the Crownpass ID will be added to area database (current holders in area) alongside the entry time which will be used for traceback

6) Capacity is increased (+1)

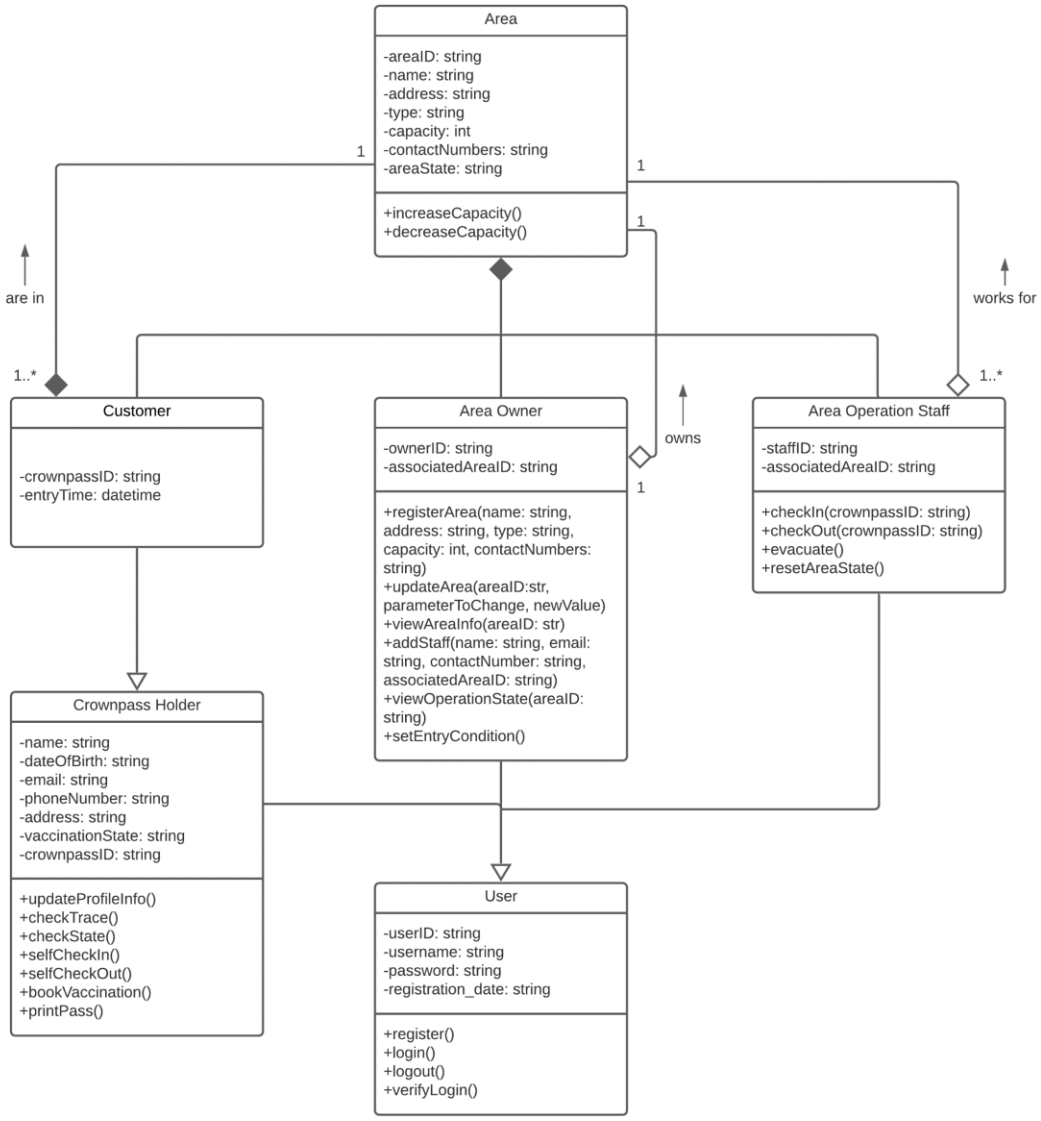
7) Confirmation of a successful check in is displayed on Operation Staff mobile

8) Operation Staff lets the Crownpass Holder enter the premises

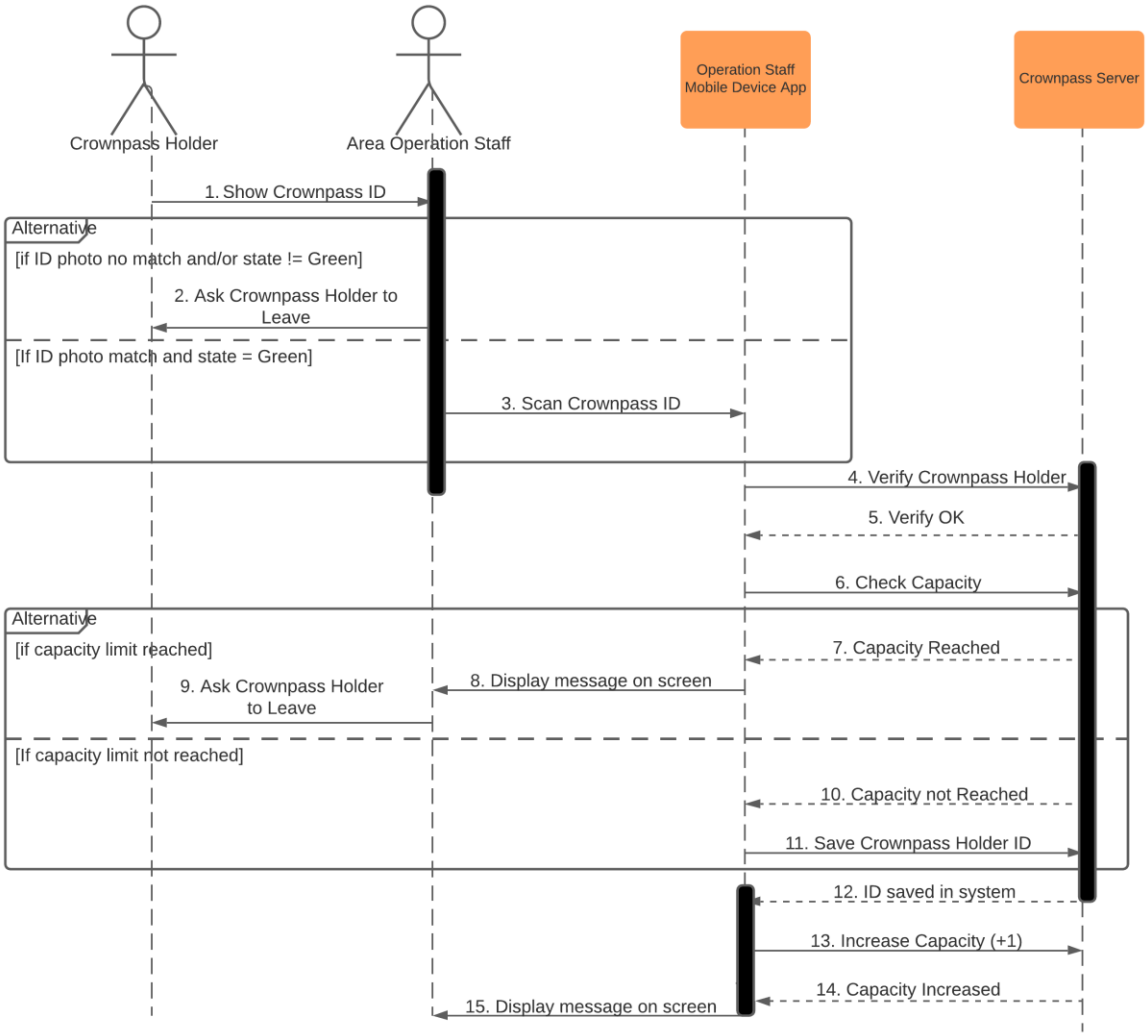
Special Requirements:

The response time for check-in a Crownpass holder into a controlled area should be no more than 3 seconds.

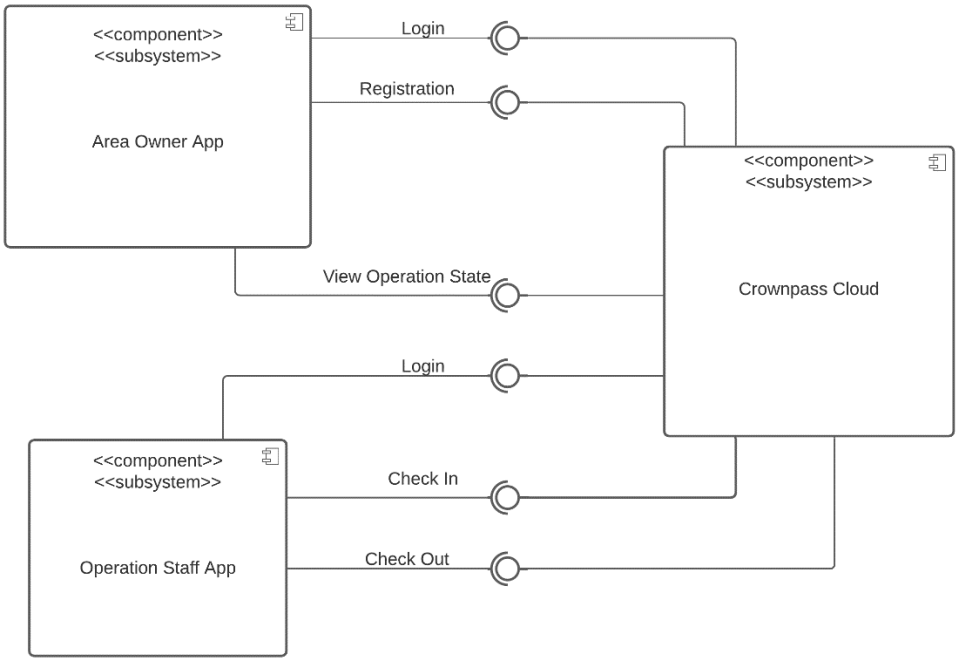
1c) Class Diagram



1d) Sequence Diagram



2a)



2a) Textual Documentation

<<interface>> Owner Login
+ownerLogin(email: str, password:str) +ownerRegister(email:str, Password:str)

<<interface>> Staff Login
+staffLogin(email: str, password: str)

<<interface>> Registration
+registerArea(name: str, address: str, type: str, capacity: int, contact_no: str) +updateRegistrationData() +viewRegistrationData()

<<interface>> Check In
+checkIn(crownpassId)

<<interface>> View Operation State
+viewOperationState()

<<interface>> Check Out
+checkOut(crownpassID)

Subsystem B – Textual documentation of Components and Connectors

Service Name	**Service Provided**
Login Manager	Microservice which allows users to be able to send requests relating to Area owner and Operation Staff accounts present on the Cloud
Authentication Manager	Microservice running on the cloud which is able to manage accounts when it comes to creating them, verifying users, updating account information and interacting with user data which is saved in the User Database
Registration Manager	This is a microservice which allows for the Area owner to be able to manage registration queries such as registering a new area, updating the registration data, and viewing the data. Input is saved inside of the area database which can be queried by the Area Manager
Area Manager	This microservice is used to query the Area database to be able to see current capacity, set an entry condition and see a list of crownpass holders currently on premises of the area
Area Operations Manager	This microservice deals with operations performed within the area such as checking in and out of crownpass holders with valid crownpass ID's, evacuating of the area if a holder who's state turned to red is present and resetting the area state after evacuation procedure

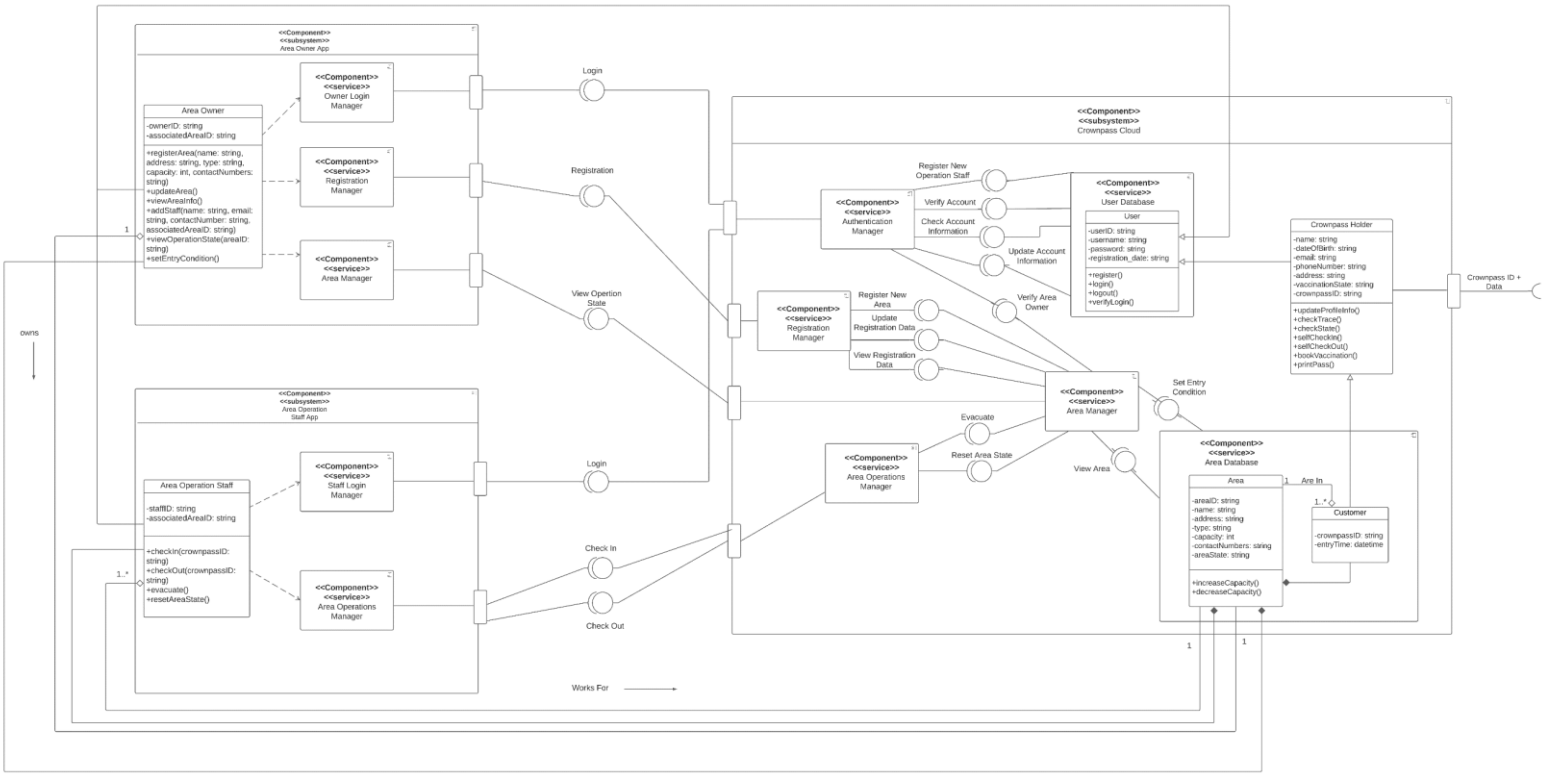
Database Name	**Service Provided**
User Database	Database which stores information about the Area owner and all of the area's operation staff including emails, passwords, contact numbers, their own crownpass details and vaccination states. This database is used with the authentication manager for logins to be able to verify users before they can enter their mobile apps (area owner, operation staff)
Area Database	This separate database contains Crownpass ID's of people who have entered a registered area, their entry time and exit time which can be used for queries on traceback. This database can be used for the evacuation function because data about a specific holder can be pulled through the use of the saved Crownpass ID and if their state changes, name can be

	looked up to then notify the operation staff who will begin the evacuation sequence

Connector Name	**Connector Description**
Login	Provided by the Authentication Manager on Cloud and is required by the Login manager within the app. This connector is used to associate an account present in the user database based on the information inputted (email, password) to log in view and update information associated with an account
Verify Account	Provided by the authentication manager on cloud for Login Manager on the app and is used to verify whether correct information has been put into the app (email, password) to allow the user to log in if the data matches the credentials in database
Check Account Information	Provided by the authentication manager on cloud for Login Manager on the app and is used to allow a registered used to view their account information
Update Account Information	Provided by the authentication manager on cloud for Login Manager on the app and is used to allow a registered in the database user to edit their account information if needed
Verify Area Owner	This is a special verification provided by authentication manager to be able to verify an Area Owner before they can view confidential information about state of an Area they own and manage it
Check In	Provided by the Area Operations Manager on both the Cloud and the Operation Staff App to allow them to check in a Crownpass holder by scanning their Crownpass ID which will initialise the check in sequence. If successful, the Crownpass holder will be let into the area and the time of the entry as well as their ID will be saved inside of the Area Database for traceback
Check Out	Provided by the Area Operations Manager on both the Cloud and the Operation Staff App to allow them to check out a Crownpass holder by scanning their Crownpass ID as the holder exits the premise. This can also be triggered by the evacuation sequence which will make users on the list of current holders in area all be checked out automatically and the time of exit will be saved alongside their ID for traceback
Register New Area	Provided by the Registration Manager on the Cloud and the Area Owner app and allows the

	Area Owner to register a new Area inside of the system and saves the information inside of the Area Database
Update Registration Information	Provided by the Registration Manager on the Cloud and the Area Owner app and allows the Area Owner to update registration data such as capacity of an area or contact details. Any changed information is saved inside of the Area Database
View Registration Information	Provided by the Registration Manager on the Cloud and the Area Owner app and allows the Area Owner to view Area registration information which can be queried from the Area Database through the Area Manager
Register New Operation Staff	Provided by the authentication manager on the cloud and allows the Area Owner to set up new staff accounts either using information about users (crownpass holders) from the user database or creating completely new profiles for them
Set Entry Condition	Provided by the registration manager on the cloud for the Area Manager and allows for the Area Owner to be able to set and edit the entry condition (infection state) of crownpass holders that will be able to enter Area Premises
Evacuate Area	Provided by the Area Operations Manager on the cloud for the Area manager and allows the Area Operation Staff to initialise the evacuation sequence. If a state of a crownpass holder currently in area (provided from Area Database) changes to red, all Operation Staff will be notified and must get all the people out of the area. This sequence blocks the check-in on the app and makes all crownpass holders check out from the Area database list
Reset Area State	Provided by the Area Operations Manager for the Area manager and allows the Operation staff to reset the area state upon completed evacuation sequence
View Area	Provided by the Area Manager for Area Database and allows the Area Owner to view information regarding the area

2b) Component Diagram



3a) Unit Test Plan

Subsystem B – Unit Test plan

Selected Interface:

<<interface>> Registration
+registerArea(name: str, address: str, type: str, capacity: int, contact_no: str) +updateRegistrationData(areaID, parameterToChange, newValue) +viewRegistrationData(areaID)

Test Case	Pre-Conditions	Methods and Parameters	Expected Output
New Area Registered	Area does not exist in the system	registerArea(area01) area01 = area(“Willow”, “36 Vanders Street, London, E17DG”, “restaurant”, 47, “07412589385”)	“Area has been registered successfully” Area is given a random areaID which will be used to query information about it from the Area Database and also

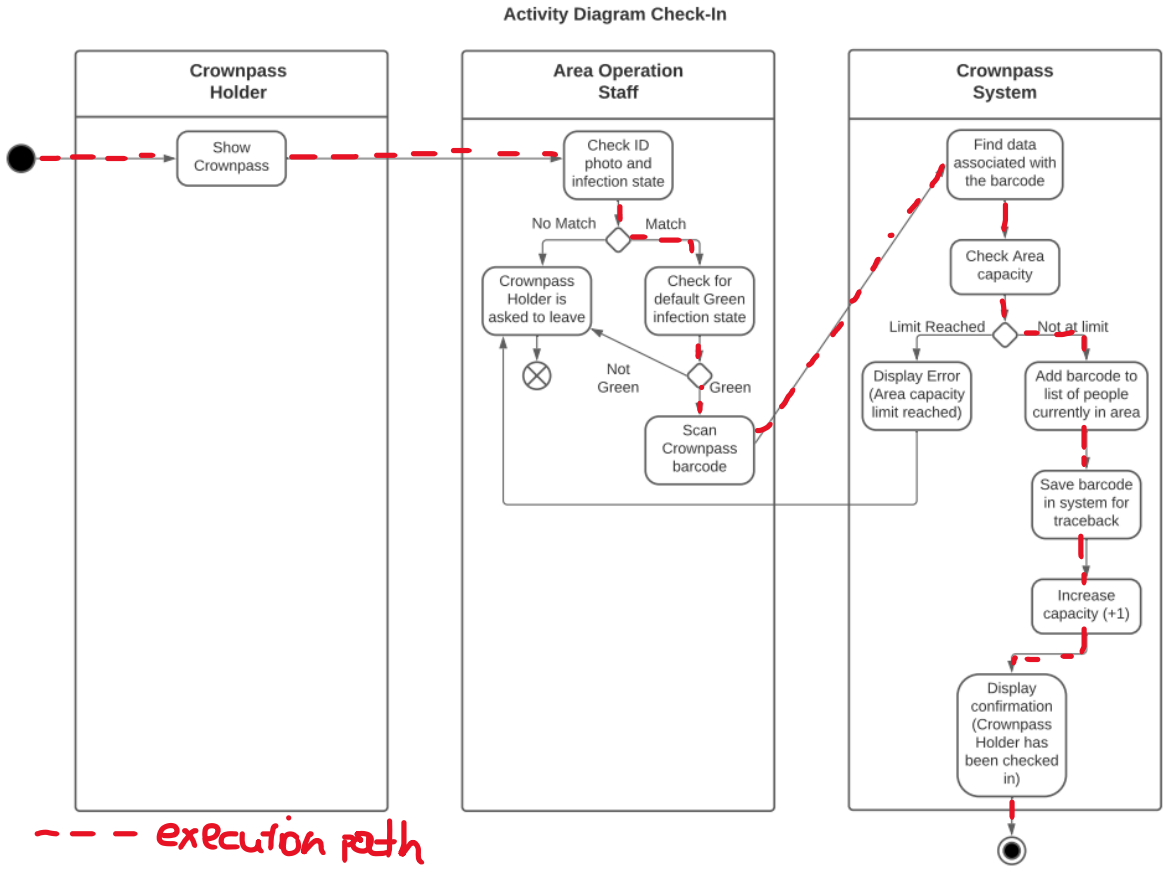
			will be used to update any registration data Returns: 01,Willow, 36 Vanders Street, London, E17DG, restaurant, 47, 07412589385
Area Registration unsuccessful because Area already exists	Area has already been registered and can be queried from Area Database	registerArea(area02) area02 = area("Willow", "36 Vanders Street, London, E17DG", "restaurant", 47, "07412589385")	Error: "Area could not be registered because it already exists in the database"
Area Registration Unsuccessful because capacity is null	Area Registration form filled in except the capacity which cannot be NULL	registerArea(area01) area01 = area("Willow", "36 Vanders Street, London, E17DG", "restaurant", NULL, "07412589385")	Error: "Area capacity cannot be set to NULL"
Update successful	Area successfully registered in the system	updateRegistrationData (areaID, parameterToChange, newValue) areaID = 01 parameterToChange = capacity newValue = 59	"Area information has been successfully updated" Area information is updated with new capacity value and data is saved in Area database Returns (updated data): 01,Willow, 36 Vanders Street, London, E17DG, restaurant, 59, 07412589385
Update not successful because input data is invalid	Area successfully registered in the system	updateRegistrationData (areaID, parameterToChange, newValue) areaID = 01 parameterToChange = type newValue = 12345	"Area information could not be updated because parameter type does not take in numbers"

Update unsuccessful because parameterToChange doesn't exist	Area successfully registered in the system	updateRegistrationData (areaID, parameterToChange, newValue) areaID = 01 parameterToChange = wallColour newValue = "blue"	"Area information could not be updated because"wallColour" does not exist"
Registration data viewed successfully	Area exists in the system Area owner logged in as they are the only ones that will be able to query data regarding their area	viewRegistrationData(areaID) areaID = 01	Returns: 01,Willow, 36 Vanders Street, London, E17DG, restaurant, 59, 07412589385

3b) System test Plan

Subsystem B – System test Plan(3b)

Use Case: Check – in (Area Operation Staff)



Scenario 1: successful check-in (Crownpass holder verified and area capacity not reached)

Crownpass Holder	Operation Staff	System
1. Show Crownpass ID	2. Check ID photo (holder verified)	
	3. Check Holder infection state (default Green)	
	4. Scan Crownpass ID	
		5. Find data associated with barcode (successful)
		6. Check area capacity (not at limit)
		7. Add Holder ID to a list of people currently in area
		8. Save Holder ID for traceback
		9. Increase area capacity (+1)
		10. Send confirmation that user has been checked in
	11. View notification	

Test Data

- Input
- Manual check by Operation staff: ID photo, infection state (provided by Crownpass Holder app)
- Function input: crownpass ID
- Stored Info
- On Operation Staff Phone:
- Crownpass ID: no data stored
- On Cloud:
- Crownpass ID: data stored for traceback as well as in area database (holders_currently_in_area)
- Output:
- Check-in Confirmation (on Operation staff mobile that check in has been successful)

Test Process

1. Test context:

a. Crownpass Holder: logged into their app, showing a Crownpass ID on their phone

b. Operation Staff: logged into their mobile app ready to scan the ID

2. Crownpass Holder shows the Crownpass ID to Operation Staff

a. Check if the ID photo matches the Crownpass Holder

b. Manual check successful (photo matches)

3. Operation Staff checks the infection state on Crownpass

a. Operation staff looks for default “Green” infection state

b. expected output: Infection state = “Green” (valid)

4. Operation staff scans the Crownpass ID

5. User Database is queried to find matching credentials associated with the ID

a) Check if ID is valid

b) expected output: ID = valid

6. System queries the Area Database to check whether the capacity is at limit

a. Check if capacity at limit

b. expected output: capacity not at limit

7. Crownpass ID is added to Area Database including Entry time (time of Crownpass ID being scanned for check in)

8. A request is sent to the Area Database to increase the capacity limit (+1)

9. Confirmation is sent to the Operation Staff App (“Successful check-in)

Scenario 2: unsuccessful check in (crownpass holder ID photo does not match)

Crownpass Holder	Operation Staff	System
1. Show Crownpass ID	2. Check ID photo (holder not verified)	
	3. Ask crownpass holder to leave the area	

Test Data

- Input
- Manual check by Operation staff: ID photo, infection state (provided by Crownpass Holder app)
- Function input: none
- Stored Info
- On Operation Staff Phone:
  - Crownpass ID: no data stored (crownpass ID not scanned)
- On Cloud:
  - Crownpass ID: none (crownpass ID not scanned)
- Output:
- None (crownpass ID was not scanned as manual verification failed)

Test Process

1. Test context:

a. Crownpass Holder: logged into their app, showing a Crownpass ID on their phone

b. Operation Staff: logged into their mobile app ready to scan the ID

2. Crownpass Holder shows the Crownpass ID to Operation Staff

a. Check if the ID photo matches the Crownpass Holder

b. Manual check unsuccessful (photo does not match)

3. Operation staff tells the Crownpass Holder that their photo ID does not match their looks and that they are not allowed to be let in without a valid Crownpass ID

Scenario 3: unsuccessful check in (crownpass holder infection state does not match)

Crownpass Holder	Operation Staff	System
1. Show Crownpass ID	2. Check ID photo (holder verified)	
	3. Check infection state (not Green)	
	4. Ask crownpass holder to leave the area	

Test Data

- Input
- Manual check by Operation staff: ID photo, infection state (provided by Crownpass Holder app)
- Function input: none
- Stored Info
- On Operation Staff Phone:
  - Crownpass ID: no data stored (crownpass ID not scanned)
- On Cloud:
  - Crownpass ID: none (crownpass ID not scanned)
- Output:
- None (crownpass ID was not scanned as manual verification failed)

Test Process

1. Test context:

a. Crownpass Holder: logged into their app, showing a Crownpass ID on their phone

b. Operation Staff: logged into their mobile app ready to scan the ID

2. Crownpass Holder shows the Crownpass ID to Operation Staff

a. Check if the ID photo matches the Crownpass Holder

b. Manual check successful (photo matches)

3. Operation Staff checks the infection state on Crownpass

a. Operation staff looks for default “Green” infection state

b. Infection state != Green

4. Operation Staff tells the Crownpass Holder that their infection state does not match the entry condition

Scenario 4: unsuccessful check in (area capacity limit reached)

Crownpass Holder	Operation Staff	System

1. Show Crownpass ID	2. Check ID photo (holder verified)	
	3. Check Holder infection state (default Green)	
	4. Scan Crownpass ID	
		5. Find data associated with barcode (successful)
		6. Check area capacity (at limit)
		7. Send an error message to Operation Staff mobile (“Capacity Limit Reached. Unable to check-in.”)
	8. View Notification Message	
	9. Ask Crownpass Holder to leave the Area	

Test Data

- Input
- Manual check by Operation staff: ID photo, infection state (provided by Crownpass Holder app)
- Function input: crownpass ID
- Stored Info
- On Operation Staff Phone:
  - Crownpass ID: no data stored
- On Cloud:
  - Crownpass ID: data stored for traceback as well as in area database (holders_currently_in_area)
- Output:
- Check-in Error message (“Capacity Limit has been reached”)

Test Process

1. Test context:

a. Crownpass Holder: logged into their app, showing a Crownpass ID on their phone

b. Operation Staff: logged into their mobile app ready to scan the ID

2. Crownpass Holder shows the Crownpass ID to Operation Staff

a. Check if the ID photo matches the Crownpass Holder

b. Manual check successful (photo matches)

3. Operation Staff checks the infection state on Crownpass

a. Operation staff looks for default “Green” infection state

b. expected output: Infection state = “Green” (valid)

4. Operation staff scans the Crownpass ID

5. User Database is queried to find matching credentials associated with the ID

a) Check if ID is valid

b) expected output: ID = valid

6. System queries the Area Database to check whether the capacity is at limit

a. Check if capacity at limit

b. expected output: capacity not at limit

c. actual output: capacity limit reached

7. System sends an error message to the Operation Staff that capacity limit has been reached and that check in has been unsuccessful

If there are any problems with picture quality, everything is uploaded onto GitHub.