

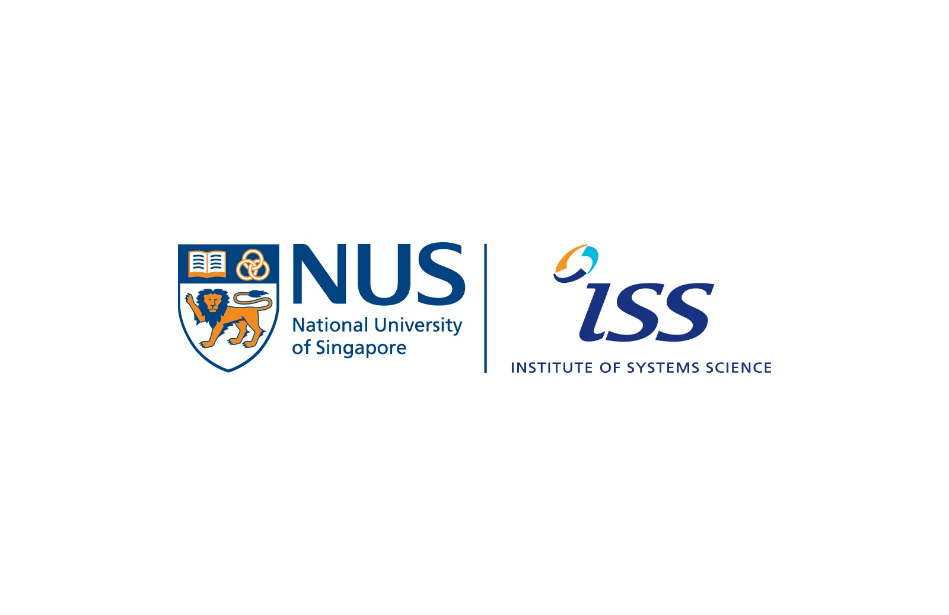
ISS Project –

**Intelligent Mask detection and alert generation Platform**

Hong Xiaohui (S9476943D)

Anita Koo Shi Qi (S9444480B)

Sanjeven Ramakrishnan (S9139938E)



National University of Singapore | Institute of Systems Science

https://github.com/xiaohuihong/IRS-PM-2021-01-16-IS03PT-GRP-3Musketeers-CoviDetector.git

**Table of Content**

[1. Business Case 3](#_Toc87613562)

[2. System Model 3](#_Toc87613563)

[3. System Development & Implementation 4](#_Toc87613564)

[3.1. System Architecture 4](#_Toc87613565)

[3.1.1. Overview 4](#_Toc87613566)

[3.2. Data Sources 4](#_Toc87613567)

[3.2.1. Labelled RGB images 4](#_Toc87613568)

[3.3. Reasoning System 5](#_Toc87613569)

[3.3.1. Overview 5](#_Toc87613570)

[3.4. Cognitive System 5](#_Toc87613571)

[3.4.1. Overview 5](#_Toc87613572)

[3.4.2. Alert system 5](#_Toc87613573)

[3.5. Web Application 6](#_Toc87613574)

[4. Challenge and Conclusion 6](#_Toc87613575)

[4.1. challenges 6](#_Toc87613576)

[4.1.1. Balancing work and project 6](#_Toc87613577)

[4.1.2. First time using Flask framework 6](#_Toc87613578)

[4.2. Future Improvements 7](#_Toc87613579)

[4.2.1. Alert system 7](#_Toc87613580)

[4.2.2. Real-time API for web scrape 7](#_Toc87613581)

[4.2.3. Chatbot icon to be shown at the bottom right of the webpage 7](#_Toc87613582)

[4.3. Conclusion 7](#_Toc87613583)

[APPENDIX OF REPORT A - Project Proposal 9](#_Toc87613584)

[APPENDIX OF REPORT B - Mapped System Functionalities against knowledge, techniques and skills of modular courses 15](#_Toc87613585)

[APPENDIX OF REPORT C - Installation and User Guide 17](#_Toc87613586)

[(Refer to separate document for Application & Deployment User Manual) 17](#_Toc87613587)

[Requirements 18](#_Toc87613588)

[Prerequisite 18](#_Toc87613589)

[Recommended browsers 18](#_Toc87613590)

[System Overview 18](#_Toc87613591)

[Deployment 18](#_Toc87613592)

[User interface 19](#_Toc87613593)

[Questionnaire 21](#_Toc87613594)

[FAQ Chatbot 28](#_Toc87613595)

[APPENDIX OF REPORT D - Individual Reports 29](#_Toc87613596)

[Individual Report by Hong Xiaohui (S9476943D) 30](#_Toc87613597)

[Individual Report by Anita Koo Shi Qi (S9444480B) 31](#_Toc87613598)

[Individual Report by Sanjeven Ramakrishnan (S9139938E) 32](#_Toc87613599)

**Abstract:** With the recent spike in community cases of covid-19, we must still remain vigilant although majority of our population are vaccinated. As Singapore slowly opens up its borders and embraces the new normal, we need to make arrangements to transit smoothly. Once such arrangement would be to redeploy staff from patrolling common areas to ensure people are wearing masks. This project proposes a solution to this problem by using a machine learning model to determine if the subject is wearing a mask and to provide an alert to authorities if there is a breach.

# Business Case

Recent news reports have showed that there has been a significant increase in local community covid-19 cases. This has led to covid-19 helplines being overwhelmed due to lack of operators and for resources to be used up at a significantly higher rate. There is also news of Singapore opening their borders via the vaccinated travel lines as Singapore moves towards an endemic state. As we move towards this ‘new normal’ we would need to streamline our processes for us to be sustainable as a country and to allocate resources efficiently. One such avenue is the jobs carried our by officials who patrol to ensure that the all patrons are wearing their masks at all times.

We propose that this can be handled using a web application that is able to provide real-time mask detection in public places and to provide a form of real-time alert to authorities to enforce the law when there is a breach.

# System Model

## Model Comparison

|  |  |  |  |
| --- | --- | --- | --- |
| Model Name | Accuracy | F1-Score | Recall |
| Knn with no hyperparameter tuning | 81.5% | 0.815 | 0.815 |
| Knn with dimension reduction | 83.9% | 0.839 | 0.839 |
| Knn with hyperparameter tuning | 80.9% | 0.809 | 0.809 |
| Knn with dimension reduction and hyperparameter tuning | 78.7% | 0.788 | 0.788 |
| CNN v0 | 93.98% | 0.9398 | 0.9395 |
| CNN v1 | 95.76 | 0.9576 | 0.9575 |
| CNN v2 | 96.43% | 0.9643 | 0.9642 |
| CNN v3 | 96.43% | 0.9643 | 0.9642 |
| CNN v4 | 96.36% | 0.9636 | 0.9636 |
| CNN v5 | 97.22% | 0.9722 | 0.9721 |
| CNN v6 | 98.01% | 0.9801 | 0.9801 |
| CNN v7 | 98.41% | 0.9841 | 0.9842 |
| CNN v8 | 99.40% | 0.9940 | 0.9941 |
| SVM |  |  |  |

# System Development & Implementation

## System Architecture

### Overview

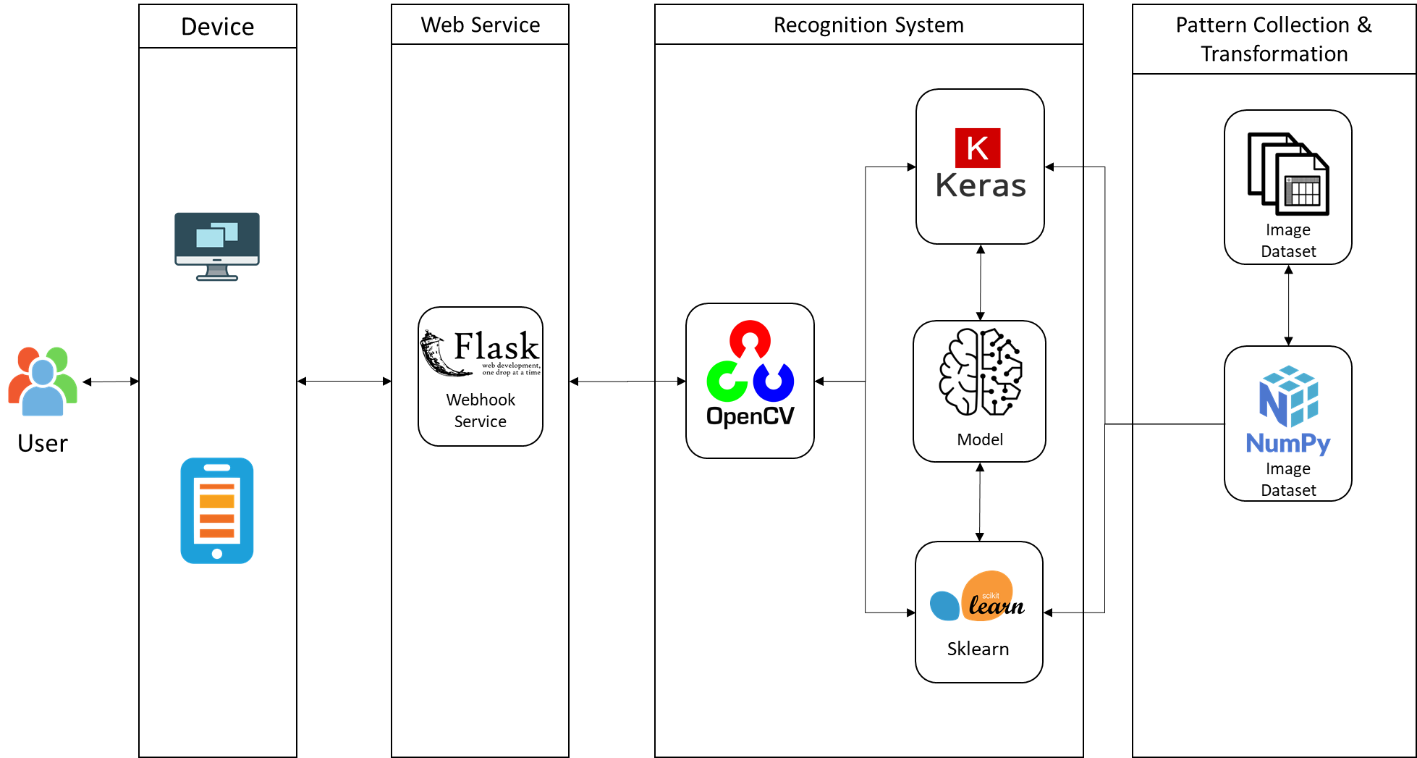


Figure 3 MaskDetective System - System Architecture

In order to build up a system which contains front-end and back-end and can integrate with machine learning libraries, as a result, Flask, a python-based web framework, is decided to use this project.

Our eco-system is constituted by two components, predictive model for mask detection. We choose to leverage scikit-learn which is a python library for predictive data analysis to build a model to predict if a user is wearing a mask based on their picture.

## Data Sources

### Labelled RGB images

We will be using labelled datasource which is available on Kaggle. This data source consists of 7553 RGB images in 2 folders. These images are labeled with\_mask and without\_mask.

A screenshot of a phone

Description automatically generated with low confidence

Figure 4 With mask data set

A screen shot of a person's face

Description automatically generated with low confidence

Figure 5 Without mask data set

The data source can be found here: https://www.kaggle.com/omkargurav/face-mask-dataset.

## Reasoning System

### Overview

## Cognitive System

### Overview

### Alert system

When a subject who has is not wearing a mask is detected, our system will automatically send a notification to the authorities to take action. For this proof of concept, we used telegram as the mode of communication. This is because telegram is widely used and can support various different types of devices (desktop, phones, tablets). The alert system will provide the user with the image and time it was recorded. In our proof of concept, this works for both images and video. For video, we implemented a 30 second timeout between frames to prevent notification spamming.

Graphical user interface, website

Description automatically generated

Figure 6 Telegram chat group

## Web Application

## Model Comparison

# Challenge and Conclusion

## challenges

### Balancing work and project

The main challenge for our team is finding time to have meet-ups for project discussion because we have different work schedules (e.g. on different client sites, doing overtime and etc) .

### First time using Flask framework

## Future Improvements

If we have a longer timeframe to work on this project, we would have work on these areas:

### Alert system

Other forms of alerts such as SMS can be implemented depending on the situation. For implementation in big shopping centres or open public spaces, we can implement zoning in our alerts so that the authorities will be able to go straight to the right area. We can also implement a database to help keep track of repeated offenders. For example, someone may have been caught and approached by staff. Once the staff has left, they proceed to remove their mask again. These people should be tracked as repeat offenders.

### Real-time API for web scrape

### Chatbot icon to be shown at the bottom right of the webpage

## Conclusion

Our team has learned quite a few things while working on this project.

Technical wise, we picked-up Django and web scraping. Also, combining a few technical skills we learnt in the course to form a new one.

In addition, while completing this project, it shows us the importance of knowledge elicitation. Without it, the main features of this application will not be feasible. We used the knowledge to build the model for the Covid-19 prediction and utilised similarity-based reasoning on the knowledge base for the chatbot response.

Overall, it was a truly enjoyable process. We get to learn from one another and apply what we learned onto a solution which resolves an everyday problem in the current Covid-19 situation.

# APPENDIX OF REPORT A - Project Proposal

**GRADUATE CERTIFICATE: Intelligent Reasoning Systems (IRS)**

**PRACTICE MODULE: Project Proposal**

|  |
| --- |
| **Date of proposal:**  1 May 2021 |
| **Project Title:**  ISS Project – Intelligent Covid Detection and Chatbot Platform |
| **Sponsor/Client:** *(Name, Address, Telephone No. and Contact Name)*  Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore  NATIONAL UNIVERSITY OF SINGAPORE (NUS)  Contact: Mr. GU ZHAN / Lecturer & Consultant  Telephone No.: 65-6516 8021  Email: [zhan.gu@nus.edu.sg](mailto:zhan.gu@nus.edu.sg) |
| **Background/Aims/Objectives:**  The proposed intelligent eco-system is aimed to integrate various advanced machine reasoning techniques to help screen vaccinated members of the public before they attend various events in Singapore, and to gather dynamic and heterogeneous information to build up an intelligent chatbot system to respond users’ questions about COVID-19. |
| **Requirements Overview:**   * Research, data collection and cleaning ability * Programming ability - python * System integration ability – Integration of Django, JQuery, Javascript, HTML, CSS and machine learning libraries |
| **Resource Requirements (please list Hardware, Software and any other resources)**  Hardware proposed for consideration:   * CPU   Software proposed for consideration:   * Pertained machine learning models, e.g. NLP, decision tree * machine learning tools, e.g. Python sklearn * Chat-bots, e.g. ChatterBot * Cognitive systems, e.g. nltk * Web crawler, e.g. beautifulsoup4 * Web framework, e.g. Django |
| **Number of Learner Interns required: (Please specify their tasks if possible)**  a team of three project members. Their completed tasks are specified in their individual reports. |
| **Methods and Standards:**   |  |  |  | | --- | --- | --- | | **Procedures** | **Objective** | **Key Activities** | | | **Requirement Gathering and Analysis** | The team should meet with ISS to scope the details of project and ensure the achievement of business objectives. | 1.        Gather & Analyze Requirements | | 2.        Define internal and External Design | | 3.        Prioritize & Consolidate Requirements | | 4.        Establish Functional Baseline | | **Technical Construction** |  | 1.        Setup Development Environment | | To develop the source code in accordance to the design. | 2.        Understand the System Context, Design | | 3.        Perform Coding | | 4.        Conduct Unit Testing | | **Integration Testing and acceptance testing** | To ensure interface compatibility and confirm that the integrated system hardware and system software meets requirements and is ready for acceptance testing. | 1.        Prepare System Test Specifications | | 2.        Prepare for Test Execution | | 3.        Conduct System Integration Testing | | 4.        Evaluate Testing | | 5.        Establish Product Baseline | |  | | **Acceptance Testing** | To obtain ISS user acceptance that the system meets the requirements. | 1.        Plan for Acceptance Testing | | 2.        Conduct Training for Acceptance Testing | | 3.        Prepare for Acceptance Test Execution | | 4.        ISS Evaluate Testing | | 5.        Obtain Customer Acceptance Sign-off | |  | | **Delivery** | To deploy the system into production (ISS standalone server) environment. | 1.        Software must be packed by following ISS’s standard | | 2.        Deployment guideline must be provided in ISS production (ISS standalone server) format | | 3.        Production (ISS standalone server) support and troubleshooting process must be defined. | |  | |

**Team Formation & Registration**

|  |
| --- |
| Team Name:  3Musketeers |
| Project Title (repeated):  ISS Project – Intelligent Covid Detection and Chatbot Platform |
| System Name (if decided):  CoviDetector |
|  |
| Team Member 1 Name:  Hong Xiaohui |
| Team Member 1 NRIC Number:  S9476943D |
| Team Member 1 Contact (Mobile/Email):  +65-97805666 / xiaohui.hong@ncs.com.sg |
|  |
| Team Member 2 Name:  Anita Koo Shi Qi |
| Team Member 2 NRIC Number:  S9444480B |
| Team Member 2 Contact (Mobile/Email):  +65-96216596 / anita.koo@ncs.com.sg |
|  |
| Team Member 3 Name:  Sanjeven Ramakrishnan |
| Team Member 3 NRIC Number:  S9139938E |
| Team Member 3 Contact (Mobile/Email):  +65-98153769 / sanjeven.ramakrishnan@ncs.com.sg |
|  |

|  |  |  |
| --- | --- | --- |
| **For ISS Use Only** | | |
| **Programme Name:** | **Project No:** | **Learner Batch:** |
| **Accepted/Rejected/KIV:** | | |
| **Learners Assigned:** | | |
| **Advisor Assigned:**  Contact: Mr. GU ZHAN / Lecturer & Consultant  Telephone No.: 65-6516 8021  Email: [zhan.gu@nus.edu.sg](mailto:zhan.gu@nus.edu.sg) | | |

# APPENDIX OF REPORT B - Mapped System Functionalities against knowledge, techniques and skills of modular courses

|  |  |
| --- | --- |
| Modular Courses | System Functionalities / Techniqe Applied |
| Machine Reasoning (MR) | * **Knowledge Elicitation and Extraction**:  Web crawling from websites   Manual extraction from Israeli MOH dataset   * **Business rule and Business process**   Restrict rules to allow the vaccinated user to process the business rules   * **Constraint Satisfaction**   Setup threshold for the similarity of knowledge chatbot response and auto-switch to the chatbot of content based filtering   * **Rule Based System**:   Decision tree to derive rules from COVID-19 symptoms and user profile   * **Knowledge Discovery & Reasoning**:   Knowledge Chatbot developed by Chatterbot based on FAQ worksheet   * **Knowledge Representation**:  Decision tree diagram for COVID-19 symptoms   Classification report and confusion matrix table for the training results  Data representation in django SQLite database |
| Reasoning System (RS) | * **Knowledge Discovery & Transparency**:   Decision trees and Decision rule learning, including One-Hot Encoding, on predicting covid based on symptoms  Similarity-based Reasoning, including Similarity & Distance Measures by Cosine Similarity, on chatbot response from covid knowledge base |
| Cognitive System (CGS) | * **Cognitive System**:   Chatbot developed by Nature Language Processing, tf-idf indexing and Similarity Measurements |

# APPENDIX OF REPORT C - Installation and User Guide

# (Refer to separate document for Application & Deployment User Manual)

# Requirements

## Prerequisite

* Computer with internet access
* Python 3.7 installed

## Recommended browsers

Our screening system supports the following browsers:

* Microsoft Edge version 90 and above
* Google Chrome version 90 and above
* Opera 75 and above

# System Overview

Our mask detection system is targeted at businesses who currently have employed safe distancing officers who make sure patrons wear their masks at all times. In our current situation in Singapore, wearing masks is mandatory and is enforced. Safe distancing officers are employed to patrol and make sure patrons have their masks on at all times. Our system would like to reduce human intervention so that majority of these safety officers can be re-deployed to work on other sectors which require more manpower. One of such sectors is covid-19 helplines which have a long waiting time due to lack of manpower.

Besides detection, our system will provide the authorities with an alert for them to act on.

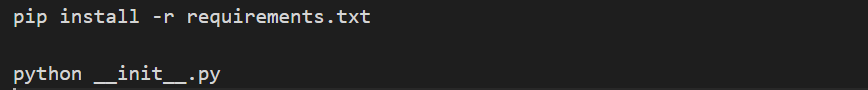
# Deployment

Our system is deployed to a windows server. In order to run our system locally, you would need python 3.6 installed along with the following packages. These packages can also be installed via the requirements files provided.

To get started, we can download the source code at <https://github.com/xiaohuihong/PRS-PM-2021-09-15-GRP-3Musketeers-DetectiveMask.git> or via the git clone command below:



Navigate to \app folder. Install the required packages and start up the server locally with the following commands:



To setup the alert notification system, join the chat group using the following link:

<https://t.me/joinchat/VtSxgK0B-i84ZWE9>

or by scanning the QR code below:

Qr code

Description automatically generated

# User interface

Once the server is running, use your preferred browser to visit the link localhost:8000. We would recommend using Google chrome. We have two main sections to our application, live detector and image detector.

Live detector

This is the page users will first see when they access the application. Here, they will also be able to start using the system. This can be done by clicking the start button to start the live detection.

Graphical user interface, timeline

Description automatically generated

*Image 1: Landing page/ live detector*

Image detector

In this page, users will be able to manually input a still image to test the model. Users can do this by firstly browsing to the image using the browse button and then to press detect mask to run the prediction.

Graphical user interface, application

Description automatically generated

*Image 2: Image detector*

# Alert System

This proof of concept is currently using telegram as the mode of communication. Once a face is detected without a mask foe either video or image detection, the notification will be automatically pushed to the telegram group. For video detection, there is a 30 second delay between frames to prevent notification spamming.

Graphical user interface, website

Description automatically generated

*Image 3: Chatbot replying to covid related questions*

# APPENDIX OF REPORT D - Individual Reports

### Individual Report by Hong Xiaohui (S9476943D)

**1). Personal contribution to the project**

As a team lead of this project, I am responsible for the following items:

a. Project topic and idea generation

b. System architecture design, development and implementation including:

i. Django web service backend and frontend setup, e.g. data schema, questionnaire workflow webpages and integration of predictive model, chatbot and backend codes.

ii. Training dataset and knowledge base selection, e.g. covid-19 symptoms dataset and FAQ worksheet preparation.

iii. Predictive model build for Covid-19 detection, e.g. decision tree and rules model

iv. FAQ chatbot build for both chatterbot and natural language processing

c. Project report generation

i. Section System Development & Implementation

ii. Project Proposal

iii. Mapped System Functionalities against knowledge, techniques and skills of modular courses

**2). What learnt is the most useful for you**

Throughout this project module, I have learnt quite a number of useful things:

a. Django web framework. In order to do this project, I had in-depth understanding on how to utilize different features of it such as multi-page forms, views and models.

b. Predictive model. Although at the end we chose the decision tree model, we still went through different types of predictive models we learnt from the course, Reasoning System. It helped enhance our understanding and our programming skill about how to use scikit-learn API to deploy these algorithms.

c. Natural Language Processing. While building the chatbot, I learnt more in tokenization, case lowering, punctuation removal and lemmatization.

d. Similarity Measurements. While developing the chatbot, two similarity measurements, Jaccard Similarity and Cosine Similarity, were implemented. I learnt how to use scikit-learn library and chatterbot library to apply these algorithms.

e. System Integration. The last thing I learnt from this project is how to integrate a web service framework with machine learning models to make a real product.

**3). How you can apply the knowledge and skills in other situations or your workplaces**

Since the Covid-19 pandemic happened from 2020, applications about it in different fields have sprung up rapidly. The experience of processing real Covid-19 data is valuable and precious for a data engineer like me in the career path or job market. In the future, if I need to process the similar type of dataset, this project is a quite good reference and I can also apply the same model to those datasets.

With the chatbot development knowledge that I have learnt, I would also apply the knowledge in helping my company to develop the similar chatbot in some projects. For example, as a data engineer, I am responsible for building data lake systems. I can develop a chatbot to answer customers’ questions about how to use our system.

### Individual Report by Anita Koo Shi Qi (S9444480B)

**1). Personal contribution to the project**

I have contributed to the followings:

* Add-on idea towards the main idea (covid detector)
* Code: Web scrape
* Project report generation
  + Abstract and Business Case
  + Section - System Model
  + Section - System Development & Implementation
  + Section - Challenge and Conclusion
* Video for System Design

**2). What learnt is the most useful for you**

The importance of knowledge elicitation. Without a valid knowledge base, an intelligent system is not possible to be created. As a data engineer, I had processed various typeof data. Often, I ponder “why is this data needed? It seems insignificant”. However, through the process of this project, my thoughts started to change. Instead, I started thinking scenarios where such data are useful , using different algorithms. With different scenarios of the data usage, one might be able to spot missing data that will be required to fulfil the scenarios thus expanding the knowledge base.

**3). How you can apply the knowledge and skills in other situations or your workplaces**

From the viewpoint of a data engineer and someone who interacts with the user, I would be able to suggest the type of information that the user might be interested in for their use cases and where to get those information.

Internally in my workplace and externally on client sites, the chatbot that was created in this project can be used for different aspects.

Externally, the system operation guide document we created for a project could act as the knowledge base for the chatbot. Our user could just type their queries (e.g how to stop a service) as an input and the chabot will output the instructions/commands to run.

Internally, this chatbot can be re-use as

* a FAQ chatbot for company’s welfare, events etc.
* a chatbot which provides solutions to common issues faced by the team on different projects. ( the knowledge base could be past jira tickets from the previous and current projects)
* a chatbot that recommends solution/system architecture base on the similarity of the requirements from past solution and system we implemented

### Individual Report by Sanjeven Ramakrishnan (S9139938E)

**1). Personal contribution to the project**

I contributed to the following:

* Business implementation plan and use case to address business needs in our current covid-19 situation
* Alert system implementation and integration with telegram
* Project report generation
* Compatibility testing
* Appendix D - User and installation guide
  + - Section - Installation
      * Requirements
      * System Overview
      * Deployment

**2). What learnt is the most useful for you**

In this project, I learned that systems integration is an important aspect in any project. To make the application operate and to satisfy our business needs, I had to learn how telegram bots worked and how we can use it to implement our alert system. This allowed me to learn more on the different ways such bots can be configured and their limitations I feel that this is an important skill to have as it brings utility to a business case.

**3). How you can apply the knowledge and skills in other situations or your workplaces**

In my daily work environment, I need to make sure that all the software’s used by our application is compatible. Sometimes there are more than one way to achieve the same results. I learned how to identify the best solution based on performance constraints to produce the best result.

Our proof of concept can be easily tailored to fit different use cases. In my experience as a consultant, this is very useful as it allows us to speed up development and deployment for different projects. Designing a product that can be easily tailored or repurposed is very useful in this sector.