Software Design Specifications

COVID-19 SOP Enforcement system using mask detection and facial recognition

Version: 1.4

Project Code	CS491
Supervisor	Mr. Shoaib Raza
Co Supervisor	Mr. Bahram Khan Baloch
Project Team	Kaamla Salman Rida Fatima Zoha Yousuf
Submission Date	14-12-2020

Document History

Version	Name of Person	Date	Description of change
1.0	Zoha Yousuf	26-11-2020	Document created Added Document Information
1.1	Rida Fatima	27-11-2020	Added chapter 1 and 2
1.2	Kaamla Salman	12-11-2020	Added chapter 3 and 4
1.3	Zoha Yousuf	24-11-2020	Added chapter 5
1.4	Rida Fatima	08-12-2020	Added References and Appendices

Distribution List

Name	Role
Mr. Shoaib Raza	Supervisor
Mr. Bahram Khan Baloch	Co Supervisor
Computer Science Department	

Document Sign-Off

Version	Sign-off Authority	Project Role	Signature	Sign-off Date
1.4	Mr. Shoaib Raza	Supervisor		
1.4	Mr. Bahram Baloch	Co-Supervisor		

Document Information

Category	Information
Customer	FAST-NU
Project	COVID-19 SOP Enforcement system using mask detection and facial recognition
Document	Software Design Specification
Document Version	1.0
Status	Draft
Author(s)	Kaamla Salman; Rida Fatima; Zoha Yousuf
Approver(s)	
Issue Date	
Document Location	Karachi, Pakistan
Distribution	Advisor Project Coordinator's Office (through Advisor)

Definition of Terms, Acronyms and Abbreviations

Term	Description
ASP	Active Server Pages
DD	Design Specification
NA	Not applicable
MTCNN	Multi-Task Cascaded Convolutional Neural Network
YOLO	You only look once
SMTP	Simple Mail Transfer Protocol
MongoDB	mongo database
OpenCV	Open-Source Computer Vision Library

Table of Contents

1 Introduction 5
1.1 Purpose of Document 5
1.2 Intended Audience 5
1.3 Document Convention 5
1.4 Project Overview 5
1.5 Scope 5
2 Design Considerations 6
2.1 Assumptions and Dependencies 6
2.2 Risks and Volatile Areas 6
3 System Architecture 7
3.1 System Level Architecture 7
3.2 Software Architecture 9
4 Design Strategy 10
5 Detailed System Design 11
5.1 Database Design 13
5.1.1 ER Diagram 13
5.1.2 Data Dictionary 14
5.1.2.1 Employee 14
5.1.2.2 Log 15
5.2 Application Design 16
5.2.1 Sequence Diagram 16
5.2.2 State Diagram 18
References 19
6 Appendices 20

1 Introduction

1.1 Purpose of Document

This document contains and tracks all the information regarding the COVID-19 SOP Enforcement System using facial recognition and mask detection. This document provides a detailed description of the system designed giving an understanding of the system architecture to the development team. This system uses the object-oriented design methodology.

1.2 Intended Audience

The audience intended is the IT team, developers, designers, and software testers. The users will also have access to this document.

1.3 Document Convention

The font style is Arial and the font size for this document is 11.

1.4 Project Overview

Violation of SOPs, specifically not wearing a mask and not practicing social distancing, increases the rate of transmission of COVID-19 or other contagious diseases. To prevent the transmission, Violators need to be identified, notified of their violation and a complete dashboard should be maintained for understanding the severity of the matter. It will perform mask and face detection. If a face is identified without a mask, it will run face recognition model to identify the violator by checking all the employee data in the database. A dashboard will be created which will show statistics and graphs regarding violations and generate reports accordingly. Crowd detection will simultaneously be performed detecting social distancing violations.

1.5 Scope

The language used is python. This system will assist authorities or employers in accurately identifying individuals who do not follow SOPs in a workplace or public area. It will recognize the violator if they are not wearing a mask and inform the authorities if there is a crowd gathered in close proximity. It also increases accountability by sending warnings via email. What would have been a strenuous job for a human is automatically made effortless. The transmission of COVID-19 can be lessened via the proper use of this system. It would create a sense of responsibility in an individual towards their fellow peers and country. Deep learning concepts will be majorly utilized for the implementation of the system

2 Design Considerations

While designing a product for SOP enforcement the system was kept simple and functional simultaneously.

2.1 Assumptions and Dependencies

While designing the system it is assumed that all the employees will have their respective email addresses and the organizations will have the images of all the employees. These images will be used to feed the facial recognition model.

2.2 Risks and Volatile Areas

ID	Description of Risk	Impact	Risk Response	Risk Level
1	Unavailability of data sources	Bad training of detection models	Create a dataset with masked faces	Medium
2	Dataset is biased against Pakistani faces	Inaccurate and racist predictions	Create a dataset strictly with local Pakistani faces	Low
3	Low scalability to multiple camera streams	Low coverage area	Multiprocessing, Multiple data pipeline	High
4	Does not work on real-time data due to bad performance	System will not work	Increase performance by optimization	High
5	Unavailability of hardware resources	Bad performance and scalability	Use cloud services and battery-operated cameras	Low
6	Unable to fetch embeddings from database due to connectivity issues	Facial recognition module will not work	Keep a local copy of the database	Low

Table 1: Risk Assesment Table

3 System Architecture

This section explains the conceptual view and functionalities of the system.

3.1 System Level Architecture

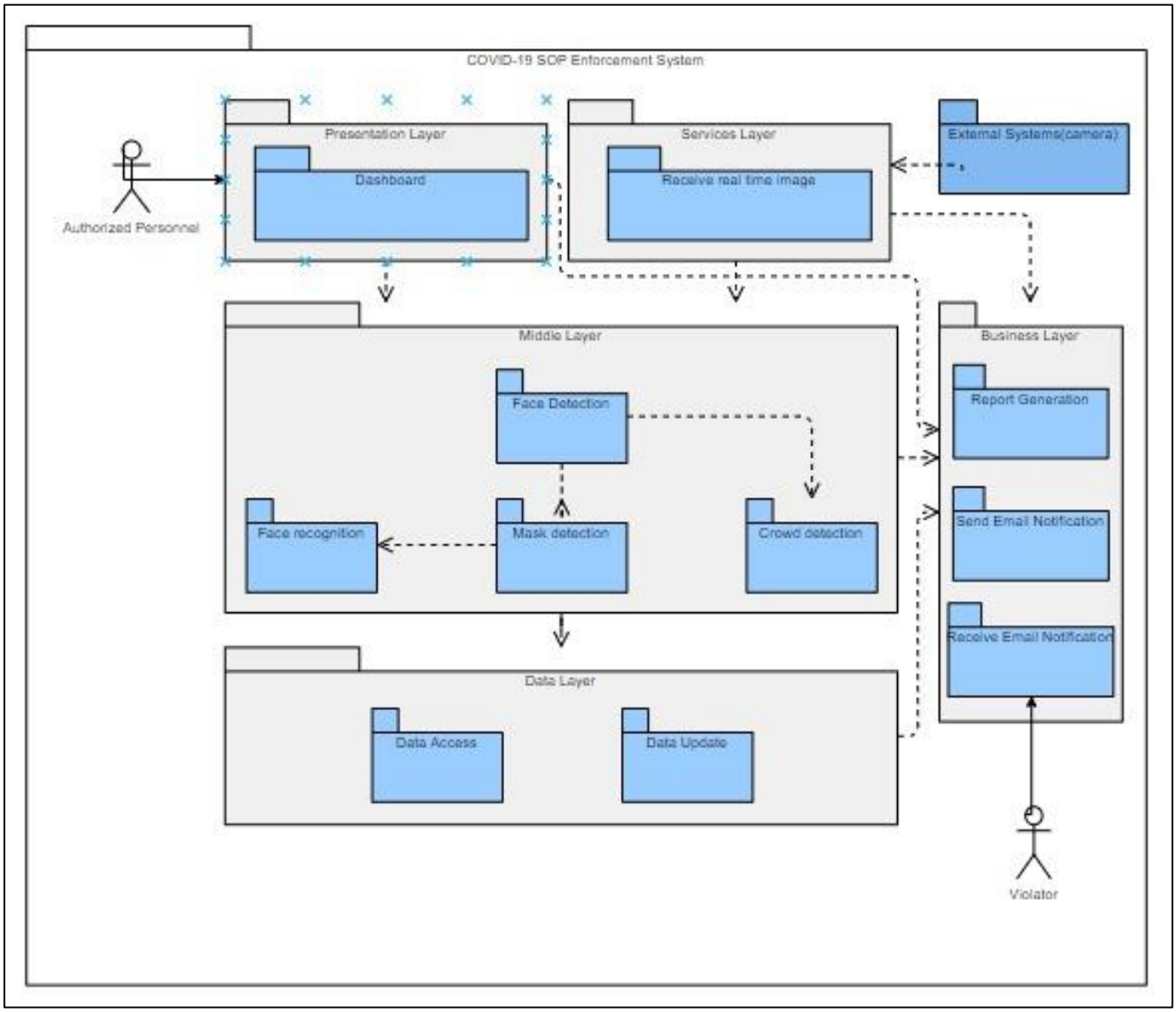

Figure 1: Package Diagram of COVID-19 SOP Enforcement System

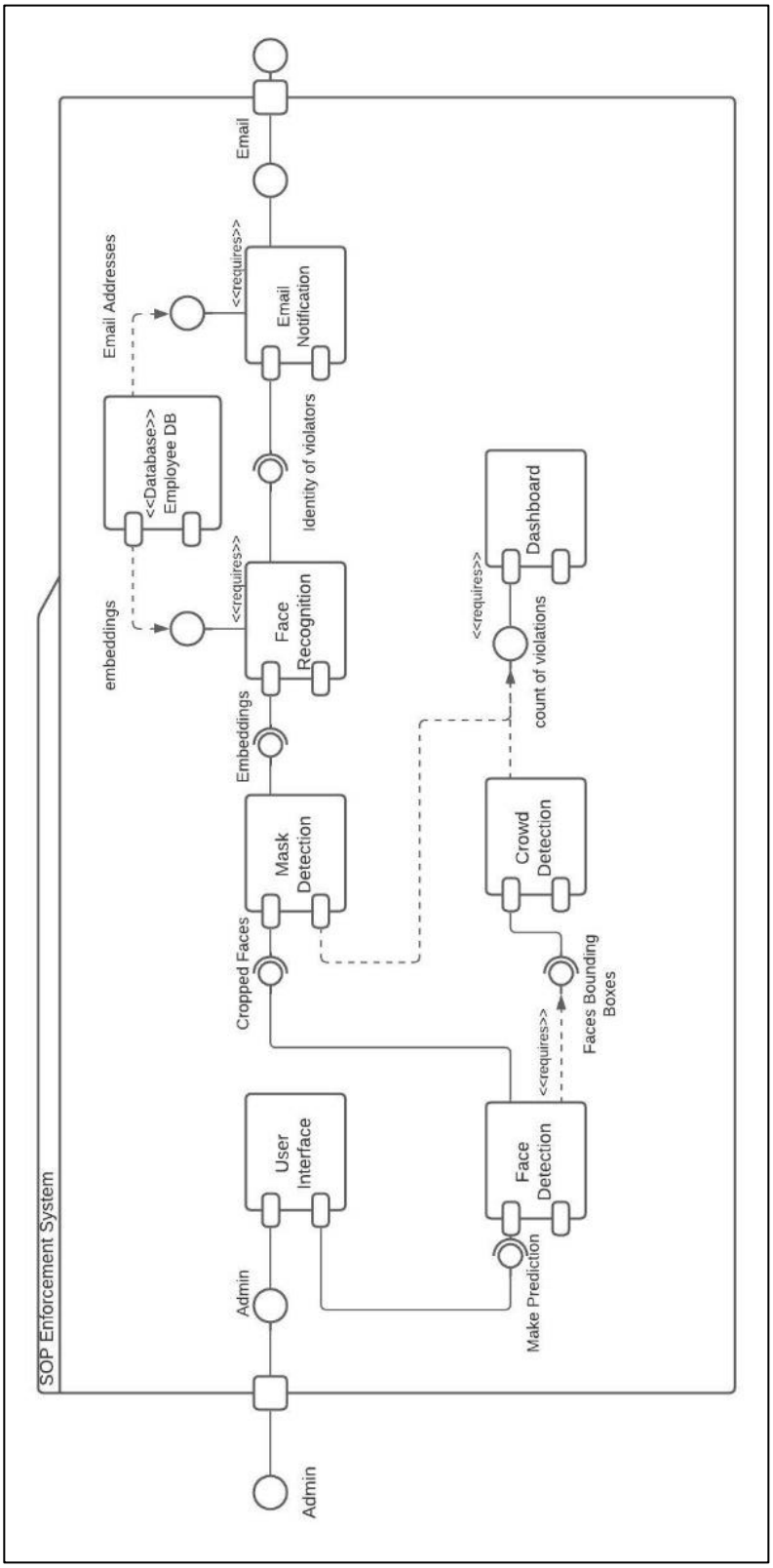

Figure 2: Component diagram of COVID-19 SOP Enforcement System

3.2 Software Architecture

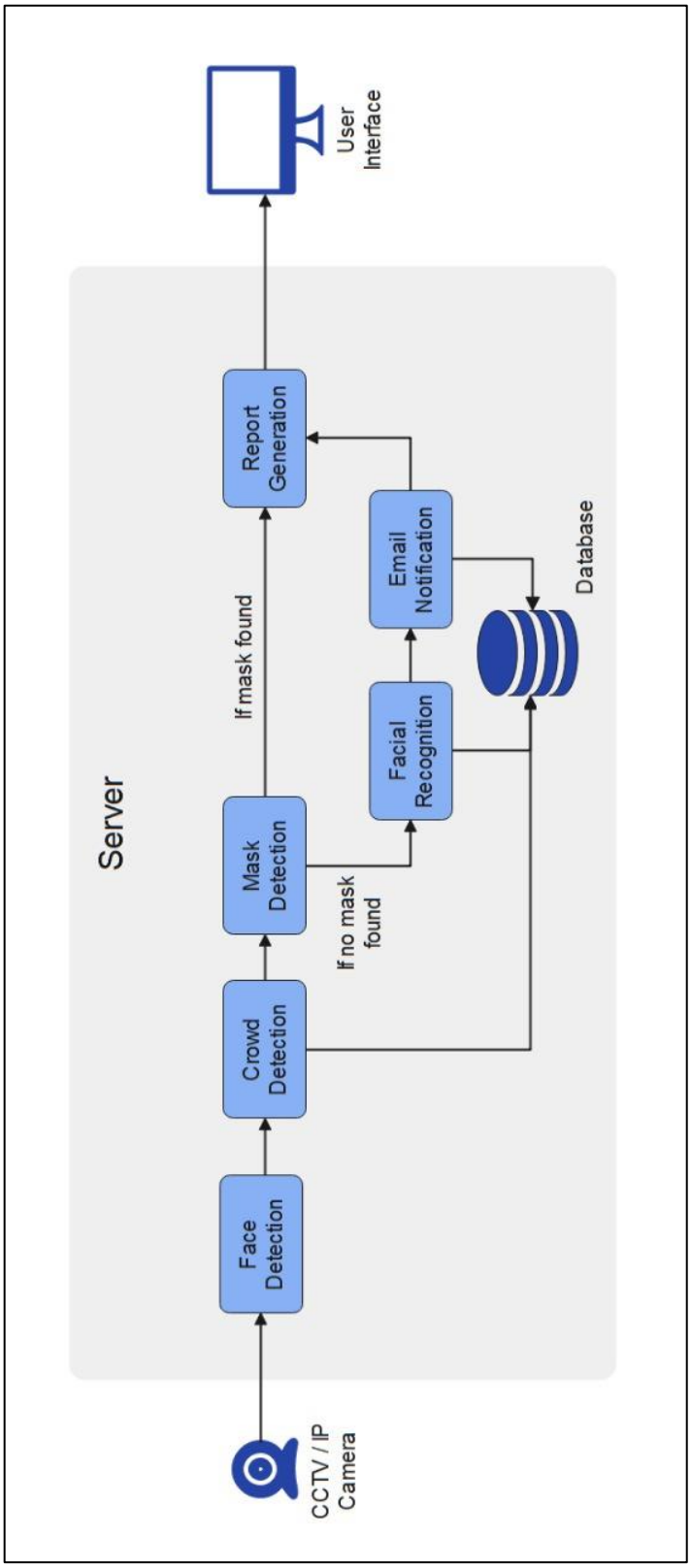

Figure 3: Software Architecture diagram of COVID-19 SOP Enforcement System

4 Design Strategy

Prior to the implementation of the system, research was carried out on the basis of study types, keywords. The key words used in the study are (“Face Detection” OR “Mask detection” OR “Face recognition”): The databases used were IEEE Xplore, ICCCS, arxiv, and hindwai. On the basis of the research the system majorly utilized deep learning concepts for the implementation of the system. The language used is python due to its numerous deep learning libraries which makes neural network implementation possible. For mask and face detection: transfer learning and YOLO algorithm with custom classes were used based on efficiency [1]. Another paper proposed a methodology for mask detection composing of two components: feature extraction using ResNet and classification based on decision making trees [2]. Facenet and MTCNN are deep convolutional network models that are used for face recognition [3,4] and are also utilized for this system. A database will contain the information and images regarding the employees or associates of that organization. The database is planned to be made using MongoDB [5]. Since the system is in its initial stages and does not require many entities, a NoSql database was chosen.

The dashboard created shows statistics and graphs regarding violations and generate reports accordingly. Crowd detection was performed by thresholding with face detection modules by using face coordinates. Sending emails automatically once a SOPs violator is identified was done using the Smtplib library in python.

Furthermore, for the user interface, the PyQt tool was used to combine all modules and incorporated them into a functional and easily navigable interface.

For future work, sickness detection can be added to detect facial cues of sickness and a Heat map can be generated for representing areas with a high density of people.

5 Detailed System Design

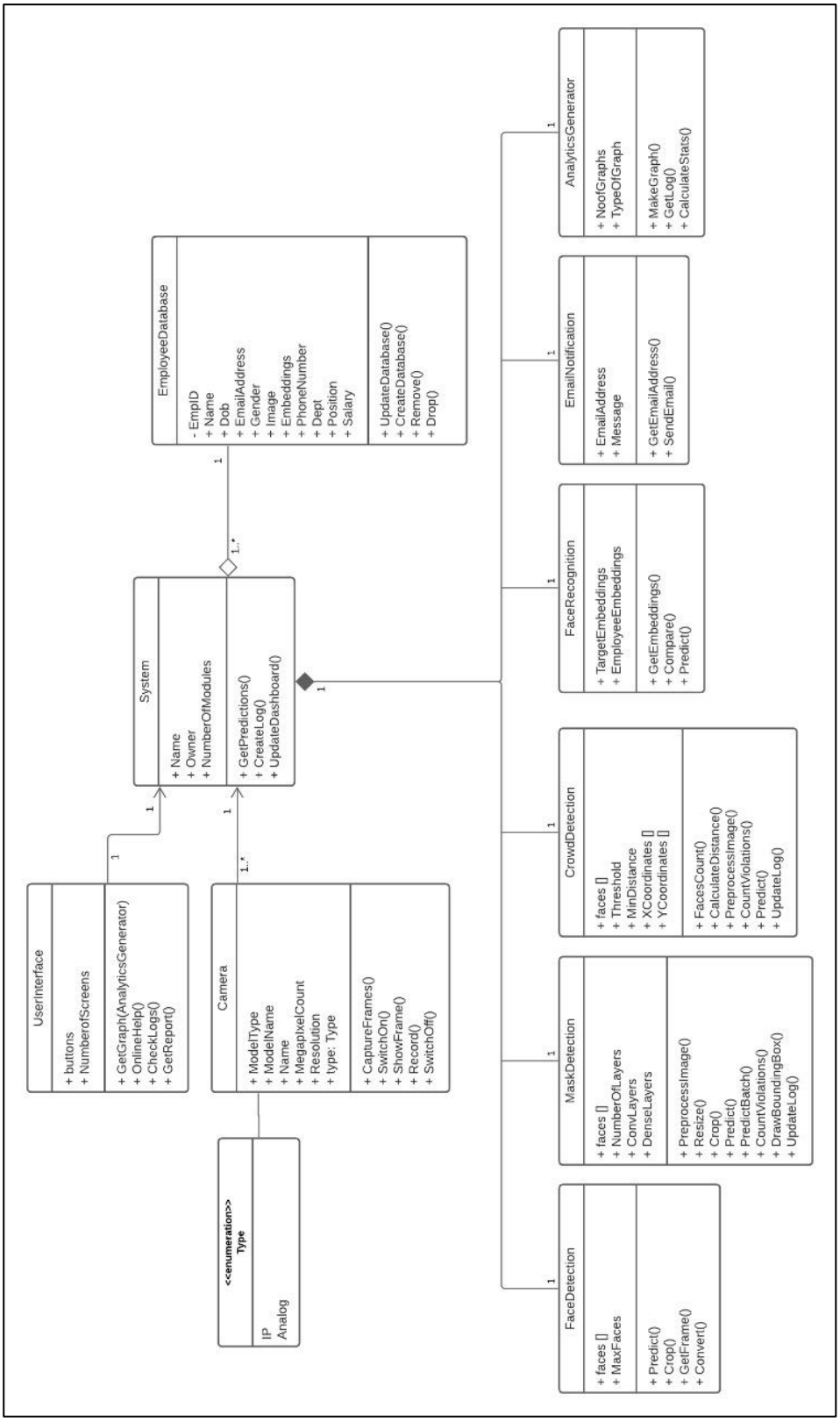

Figure 4: Class Diagram of COVID-19 SOP Enforcement System

User interface: A GUI interface to display the various outputs, violations and reports. It contains Buttons and number of screens which will be built using Flutter to perform 4 functions: GetGraph(), OnlineHelp(), CheckLogs(), and GetReport(). It has a one-to-one relation with the system.

Camera: An external device that will be used to capture real-time video frames. It contains ModelType, ModelName, Name, MegapixelCount, Resolution, and type to perform 5 functions: CaptureFrames(), SwitchOn(), ShowFrame(), Record(), and SwitchOff(). It has a many-to-one relation with the system.

System: The system composes of all the models. It contains Name, Owner, and NumberOfModules to perform 3 functions: GetPredictions(), CreateLog(), and UpdateDashboard(). The System has one-to-one relationship with all the modules.

EmployeeDatabase: It is used for managing the data in the entire system. It comprises of EmpID, Name, Gender, EmailAddress, PhoneNumber, deptName, Salary, Position, Address, Image, embeddings. Its functions are to UpdateDatabase(), CreateDatabase(), Remove(), and Drop(). It has a one-to-many relation with the system.

Face detection: It will be used to determine if there are any faces in the image. It will use Face landmarks to perform four functions: predict(), crop(), Getframe(), and convert().

Mask Detection: A module that will detect if an individual is wearing a mask or not. It will work for all types of masks ranging from surgical masks, N95 to cloth masks. It will take face embeddings, NumberoFLayers, ConvLayers, and DenseLayers to perform Preprocessimages(), Resize(), Crop(), Predict(), predictedbatch(), countviolations(), Drawboundingboxes(), UpdateLog().

Face recognition: After detecting the face in the frame, it will be compared to other faces in the database to see if there is a match and learn the person's identity. It will take targetEmbeddings and EmployeeEmbeddings to perform 3 functions: GetEmbeddings(), Compare(), and Predict().

Crowd detection: To ensure that people are maintaining social distance, the system will detect groups of people. It will take Face embeddings, threshold, MinDistance, XCoordinates, and YCoordinates to perform 6 functions: FacesCount(), CalculateDistance(), PreprocessImage(), CountViolations, Predict(), and UpdateLog()

Record generation: A report will be generated containing graphs and information regarding a violation of SOPs.

Email notification: An email will be sent to the person who violates the SOPs. It will take Employees email address and the message to perform 2 functions: GetEmailAddress() and SendEmail().

5.1 Database Design

The database design is explained via ER Diagram and details are provided through data dictionary.

5.1.1 ER Diagram

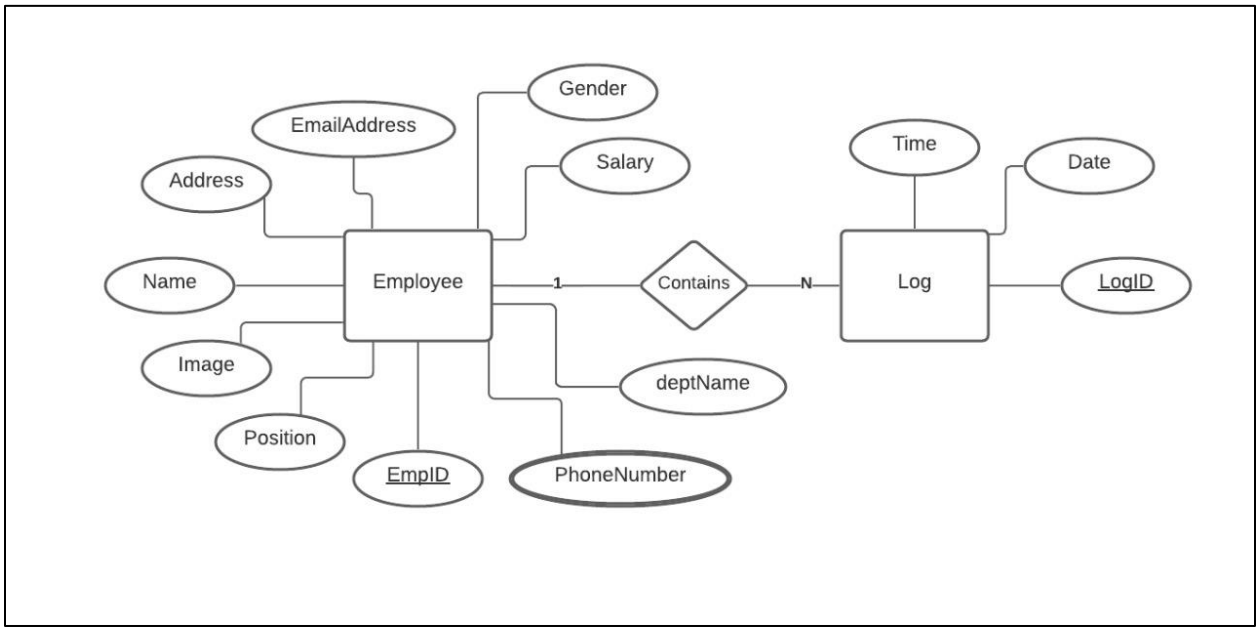

Figure 5: ER diagram of COVID-19 SOP Enforcement System

The ER diagram has 2 entities; Employee and Log. The employee contains Log. It has a one-to-many relationship with the log as one employee can have multiple violations. The employee entity has nine attributes: EmpID, Name, Gender, EmailAddress, PhoneNumber, deptName, Salary, Position, and Address. The EmpID is the unique identifier for the Employee entity. Whereas, the Log entity has 3 attributes LogID, time, and date. The LogID is the unique identifier for the log entity.

5.1.2 Data Dictionary

5.1.2.1 Employee

Employee	
Name	Employee
Alias	Violator
Where-used/how-used	Facial recognition, Email notification, Dashboard
Content description	Facial recognition = Employee's image; for recognizing the face of the violator. Email Notification = Email address; will be used to notify them about violation committed. Email notification Content = Name + department number + position + EmpID.

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
EmpID	The registration ID of the employee. Unique identifier	String	8	No	NA	PK
Name	Name of the employee	String	256 varchar	No	NA	
Gender	Gender of the employee	ENUM	3	No	NA	
EmailAddress	Email address of the employee	String	256 varchar	No	NA	
PhoneNumber	Contact number of the employee	int	11	Yes	00000000000	
deptName	Department designated to the employee	String	256 varchar	No	NA	
Salary	Monthly pay of the employee	Int	20	No	NA	
Position	Exact position of the employee	String	256 varchar	No	NA	
Address	Residence of the employee	String	256 varchar	No	NA	

Table 2: Data Dictionary of Employee entity

5.1.2.2 Log

Log	
Name	Log
Alias	Report
Where-used/how-used	It is used for dashboard. The dashboard will be using the report generation feature where all the logs will be calculated and displayed to the Authorized Personnel.
Content description	Dashboard = LogID + Time + Date

Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
LogID	The registration ID of the violation. Unique identifier	String	6	No	NA	PK
Time	Time of the violation	timestamp	10	No	NA	
Date	Date of the violation	Date	10	No	NA	

Table 3: Data dictionary of Log entity

5.2 Application Design

5.2.1 Sequence Diagram

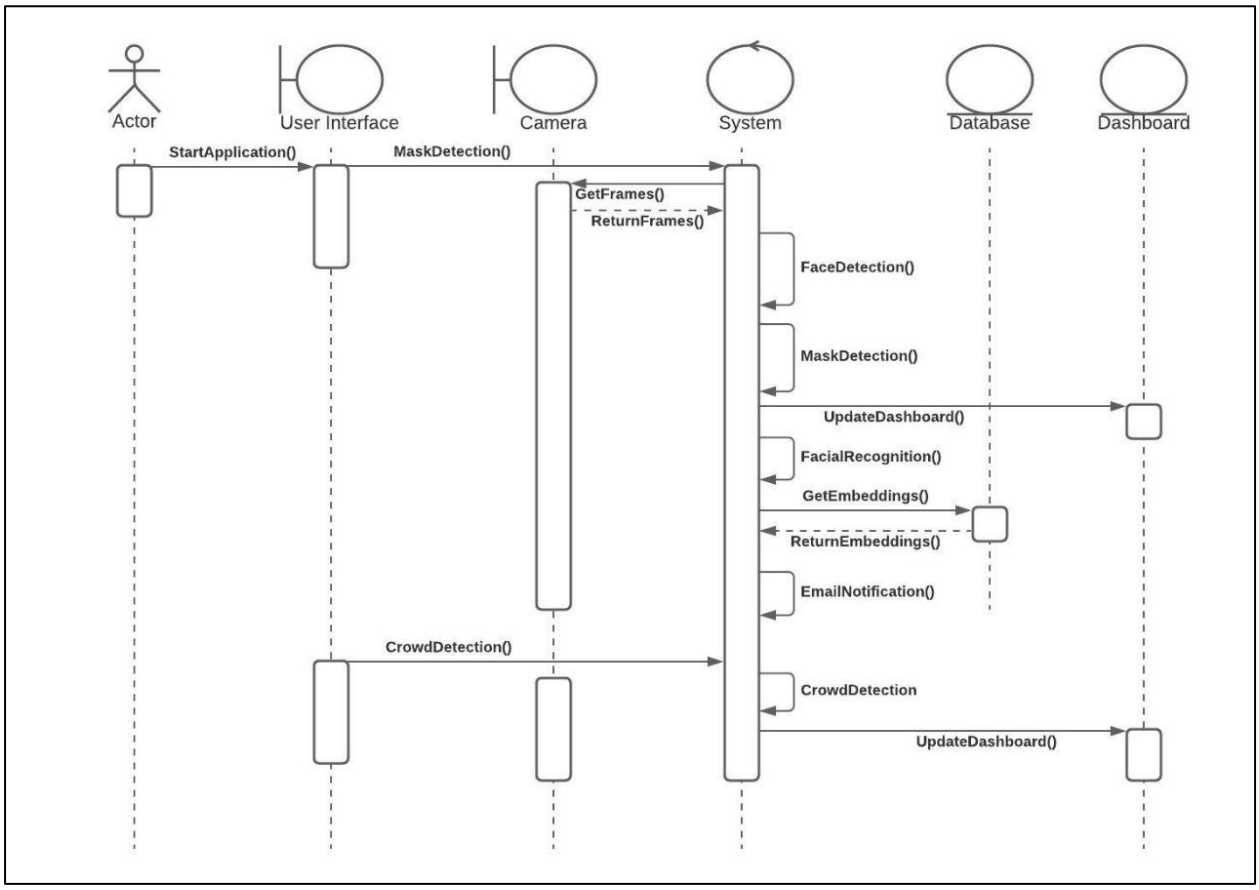

Figure 6: Sequence diagram of COVID-19 SOP Enforcement System

All the functionalities of the SOP Enforcement System are incorporated into one module. This module deals with every functionality of the system; mask detection, face detection, facial recognition, crowd detection, report generation, and email notification.

Firstly, this module takes frames as input from cameras. It then applies face detection on the frames and detects all the facial landmarks. These landmarks are then used for detecting masks and simultaneously calculates social distancing by applying crowd detection model. In case a violation, if a mask is detected, the system will perform facial recognition by taking the embeddings from the database. Once the violator is identified, the system will then send an email notification to both, the authorized personnel and the violator. Likewise, if a violation of social distancing is detected, the authorities are immediately notified. The module will also update the dashboard for report generation.

5.2.2 State Diagram

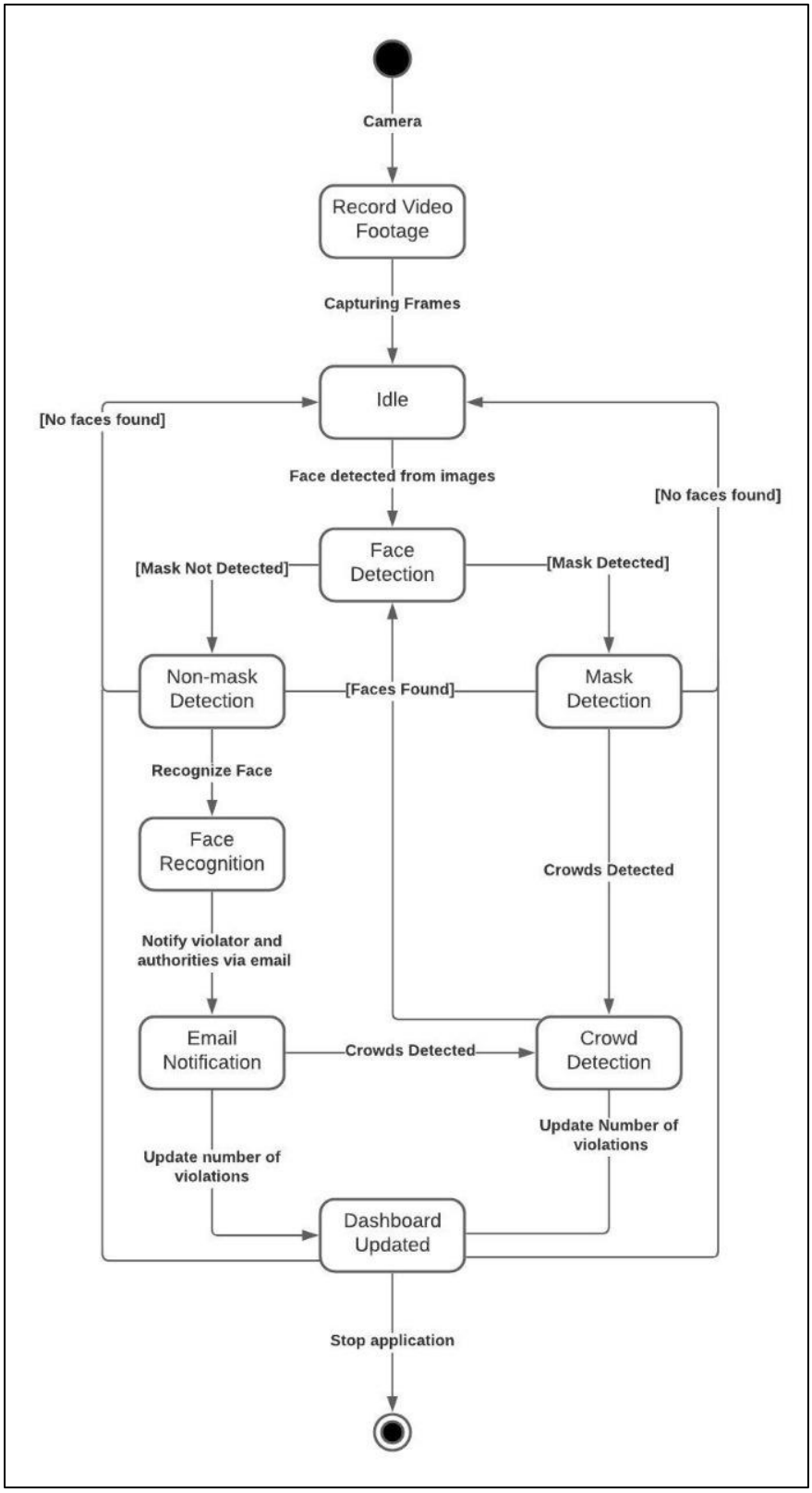

Figure 7: State diagram of COVID-19 SOP Enforcement System

When the camera is switched on, the system becomes active. In this state the camera records video footage and it captures frames from the video. Now that the environment is continuously being scanned, the system detects human faces in the frames. Moving on to the next state, mask detection is performed. If a mask is detected, it proceeds to a state where crowd detection is activated in which the social distancing among people is calculated. If the system detects a crowd violating the social distance, authorized personnel will be notified. During any state, if a person is identified not wearing a mask, the facial recognition state is activated and the violator is notified via email. Upon any sort of violation, the system transitions to a state where the Dashboard updates the number of violations. If faces in any of the frames is not detected, the system remains in the face detection state on a new frame as these frames are captured in real-time. When the system is turned off by the users all these states are deactivated.

References

[1] I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, "A brief introduction to OpenCV," 2012 Proceedings of the 35th International Convention MIPRO, Opatija, 2012, pp. 1725-1730.

[2] M. Loey , G. Manogaran , M. H. . N. Taha and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," Measurement, 2020.

[3] H. Ku and W. Dong, "Face Recognition Based on MTCNN and Convolutional Neural Network," Frontiers in Signal Processing*,* Issacpub, Chengdu, China, 2020, pp. 37-52.

[4] F. Schroff, D. Kalenichenko, J. Philbin, "Facenet: A unified embedding for face recognition and clustering," Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, Boston, USA 2015, pp. 815-823

[5] Y. Gu, S. Shen, J. Wang and J. Kim, "Application of NoSQL database MongoDB," *2015 IEEE International Conference on Consumer Electronics - Taiwan*, Taipei, 2015, pp. 158-159, doi: 10.1109/ICCE-TW.2015.7216831.

6 Appendices

- Important SOPs
  http://covid.gov.pk/guideline

- Mask detection
  https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/

- Facial Landmark detection
  https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/