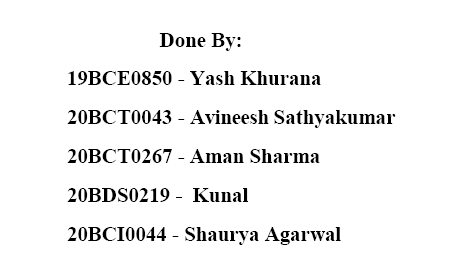
**Software Requirements Specification**

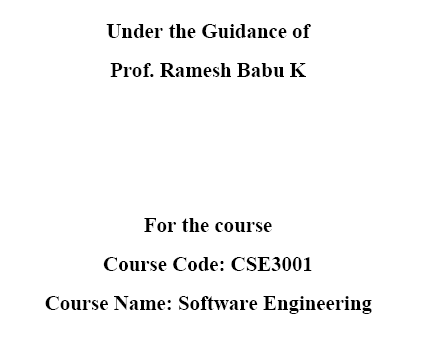
**for**

**Crowd Control Monitor**

### Version 1.0 approved

**Prepared by:**

****

****

***Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.***

## Table of Contents

**Table of Contents Revision History**

1. [Introduction 1](#_TOC_250008)
   1. [Purpose 1](#_TOC_250007)
   2. [Document Conventions 1](#_TOC_250006)
   3. [Intended Audience and Reading Suggestions 1](#_TOC_250005)
   4. [Product Scope 2](#_TOC_250004)
   5. [References 2](#_TOC_250003)
2. [Overall Description 2](#_TOC_250002)
   1. [Product Perspective 2](#_TOC_250001)
   2. [Product Functions 3](#_TOC_250000)
   3. User Classes and Characteristics 3
   4. Operating Environment 3
   5. Design and Implementation Constraints 3
   6. User Documentation 3
   7. Assumptions and Dependencies 3
3. External Interface Requirements 4

|  |  |
| --- | --- |
| 3.1 User Interfaces | 4 |
| 3.2 Hardware Interfaces | 4 |
| 3.3 Software Interfaces | 4 |
| 3.4 Communications Interfaces | 3 |

1. **System Features 4**
   1. System Feature 1 4
   2. System Feature 2 (and so on) 4
2. **Other Nonfunctional Requirements 4**
   1. Performance Requirements 4
   2. Safety Requirements 5
   3. Security Requirements 5
   4. Software Quality Attributes 5
   5. Business Rules 5
3. **Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Version 1 | 06/10/21 | Initial Documentation | 1.0 |

## Introduction

### Purpose

Manually counting the number of people in a location is very tedious. It also involves adding more people like watchmen/manpower to the area Hence, there is a need to fulfill the above role without engaging man in it.

The purpose of this document is to solve the above problem by counting people in a store or mall in real-time and alerting the authorities if the maximum limit is reached.

### Document Conventions

This document uses the following conventions.

|  |  |
| --- | --- |
| **Term** | **Intended Meaning** |
| ER | Entity Relationship |
| DFD | Data Flow Diagram |

### Intended Audience and Reading Suggestions

This document is intended for the following stakeholders.

1. Developers
2. Project Manager
3. IT Department (End Users)
4. Staff Manager (End Users)
5. Security (End Users)
6. Testers

The sequence for reading the document is serial, and specific elements can be seen as shown in the Table of Contents.

### Product Scope

We plan to tackle the above-mentioned problem using Computer Vision. The CCTV Camera feed will be retrieved in real-time and will be passed through our model. It will detect the number of people entering and leaving an area and the data will be stored.

This concept has multiple benefits over manual counting and is yet to be implemented in malls and stores. We plan to build this solution for local business stores and malls to give them more control over their customer counts. The use case is counting people in real-time and sending an alert if the limit is reached.

The camera feed can be directly retrieved from the CCTV cameras already installed in the stores. Hence, this solution has a negligible hardware cost. This project can be scaled for larger regions and areas too. The real-time stream can also be optimized for better performance using threading. Schedulers and timers also make the work of shopkeepers easy. This acts as a measure towards footfall analysis and in a way to tackle COVID-19.

### References

We have used the following resources for inspiration and resources.

1. [https://www.researchgate.net/publication/228437591\_Real-time\_people\_counting\_system\_ using\_a\_single\_video\_camera](https://www.researchgate.net/publication/228437591_Real-time_people_counting_system_using_a_single_video_camera)
2. <https://arxiv.org/pdf/1804.04339.pdf>
3. <https://docs.opencv.org/2.4.13.7/>

## Overall Description

### Product Perspective

With the rise of big data management and need for data sets for various analytical procedures, the number of entities at a given place at a given time, can be tough to calculate manually. This product utilizes, the CCTV Camera footage to do the job for us. Calculating the number of people/entities crossing a given area, can be really helpful in various different scenarios, for example-: Calculating number of customers to a mall or a shop all through the day/week, or gathering information of endangered species of animals in the migration season, calculation of number of people attending a gathering like, concerts, sermons, presidential addresses etc.

### Product Functions

The major functions of this project are as follows.

* + 1. **Real-Time people counter:** counts the number of people present in an enclosed area by taking data from the CCTV.
    2. **Real-Time alert:** sends an alert about the count to the user.
    3. **Threading to handle lag in real-time stream:** It helps in simultaneously processing and outputting the result, to prevent delays in the stream.
    4. **Scheduler and Timer:** Another feature of our project is to automatically schedule when to start and run the software.
    5. **Log and analysis:** This logs all data at the end of the day and stores it in the database.

### Operating Environment

The operating environments for our project are listed below

### Design and Implementation Constraints

This product will be implemented using the programming language Python.

We are using the following technologies, tools and programming standards for our product: (with versions)

* schedule==0.6.0
* numpy==1.19.2
* argparse==1.4.0
* imutils==0.5.3
* dlib==19.18.0
* opencv-python==4.2.0.32 7.
* scipy==1.4.1

The product can have following hardware limitations:

1. Camera quality can be bad.
2. Lower versions of GPU can hamper speed and accuracy.

## System Features

### Real Time People Counter

* + 1. Description and Priority

This is the main feature of our project. It counts the number of people present in an enclosed area by taking data from the CCTV real time stream. This enables a shop to get to know how many people are present in their shop.

This feature has the highest priority as the main functionality of the product depends on it.

* + 1. Functional Requirements

*REQ 1:*

Input: Input from CCTV camera.

Output: Count of no. of people present in the shop. Process: SSD and DCNN.

### Real Time Alert

* + 1. Description and Priority

This feature will send an alert to the user (shopkeeper or staff) in real-time. If the total number of people (say 10 or 30) is exceeded in a store/building, we simply alert the staff so that they can take necessary action to maintain social distancing. Maximum value of the total number of people can be set by the staff. This is very useful considering the COVID-19 scenario.

* + 1. Functional Requirements

*REQ 2:*

Input: Count from 3.1.

Output: Alert message according to the count to the provided email.

Process: When the total customer count exceeds a particular threshold, the alert will be automatically sent.

### Threading

* + 1. Description and Priority

A common problem faced by image processing devices is a noticeable lag or delay in the real time stream. If your system is not capable of simultaneously processing and

outputting the result, you might see a delay in the stream. This is where threading comes into action.

Threading removes OpenCV's internal buffer (which basically stores the new frames yet to be processed until your system processes the old frames) and thus reduces the lag/increases fps. It is most suitable for a solid performance on complex real-time applications.

* + 1. Functional Requirements

*REQ 3:*

Input: Clip of the CCTV footage with lag.

Output’s footage without lag.

Process: With the help of OpenCV Multithreading function.

### Scheduler and Timer

* + 1. Description and Priority

Another feature of our project is to automatically schedule when to start and run the software. The user can configure it to run at the desired regular intervals as per his/her need, be it at any time of the day, or daily from Monday to Friday. The user can also enable a timer to run the software for a specific time, say 5 hours.

This is extremely useful in a business scenario, where shopkeepers are busy running their businesses and can simply schedule the software once for the whole week and month, by choosing their desired time intervals (say 9 to 5).

* + 1. Functional Requirements

*REQ 4*

*:*

Input: Opening and closing time of the shop

Output: The system automatically starts and stops the counting.

Process: The shopkeeper will enter the opening, closing time of the shop and through the help of this function, the application will automatically shut off during the remaining time

### Log and Analysis

* + 1. Description and Priority

This logs all data at the end of the day and stores it in the database. After that, the user can see and analyze the data and understand his/her crowd distribution at different parts of the day and take necessary measures. This will be a data analysis tool.

* + 1. Functional Requirements

*REQ 5:*

Input: CCTV footage of the entire day.

Output: Storage of the log (total IN and total OUT), footage and count of people inside at a certain time.

Process: Writing the CCTV footage and the log with timestamps to the backup drive

## Other Nonfunctional Requirements

### Performance Requirements

In image processing, since you are dealing with a large number of operations per second, it is mandatory that your code is not only providing the correct solution, but that it is also providing it in the fastest manner. So in this chapter, you will learn: To measure the performance of your code. Some tips to improve the performance of your code. You will see these functions: cv.getTickCount, cv.getTickFrequency, etc. Apart from OpenCV, Python also provides a module time which is helpful in measuring the time of execution. Another module profile helps to get a detailed report on the code, like how much time each function in the code took, how many times the function was called, etc. But, if you are using IPython, all these features are integrated in an user-friendly manner.

Sometimes you may need to compare the performance of two similar operations. IPython gives you a magic command time it tajes to perform this. It runs the code several times to get more accurate results.

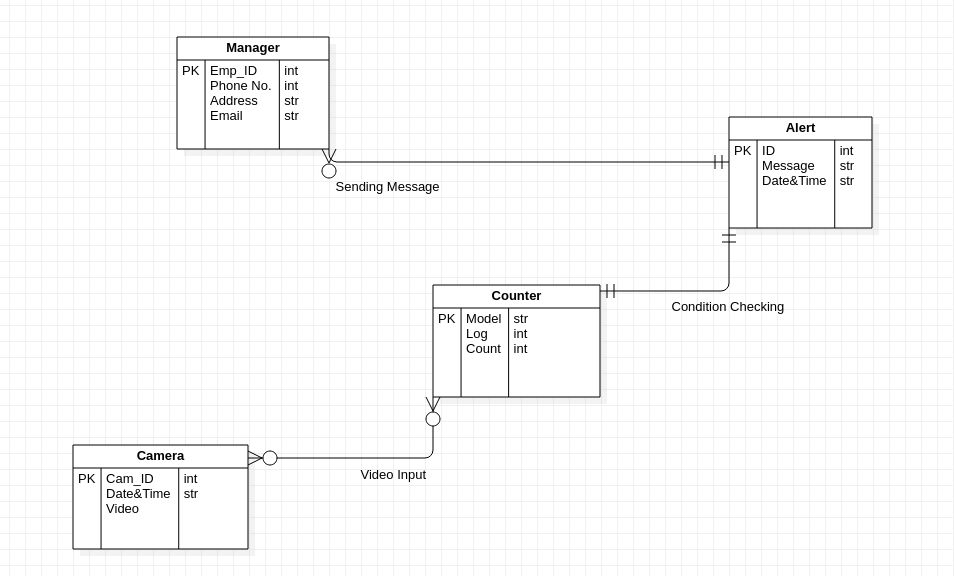
To measure the performance of our application we will compare the final value of the no. of customers acquired from our system to the values given by traditional methods. A good model will be indicated if the final values of both the methods are similar then.

### Business Rules

Apart from our functional requirements and features, a non-functional requirement that can be implemented in our application is storing and sending collected data to the associated companies. This means that the number of customers entering a specific shop(franchise of a bigger company, for instance KFC) will be tracked and saved along with the footage obtained from the CCTV cameras. Such data can be highly insightful for the company when conducting surveys and

predicting the number of customers for the future. An example of this can be, KFC wanting to know which franchise (between two located at different places in a city) has more customers during the summer season.

# ER DIAGRAM:



**Data Dictionary:**

#### Count : Integer Movement : Boolean Video Input : AVI/MP4

Log : Date & time, Integer Threshold : Integer

Alert : String

IF (yes) : Boolean

Image : Array

Message : String