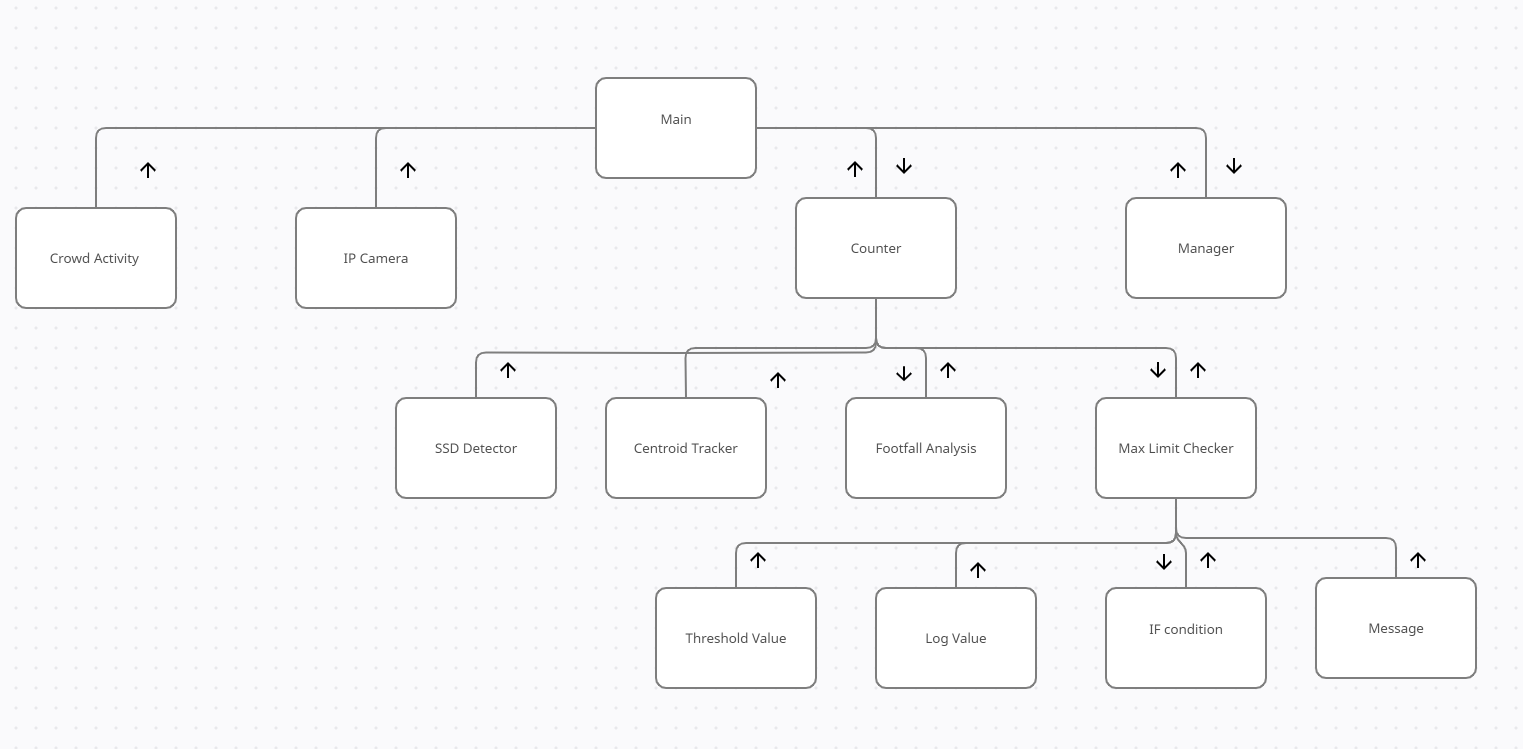
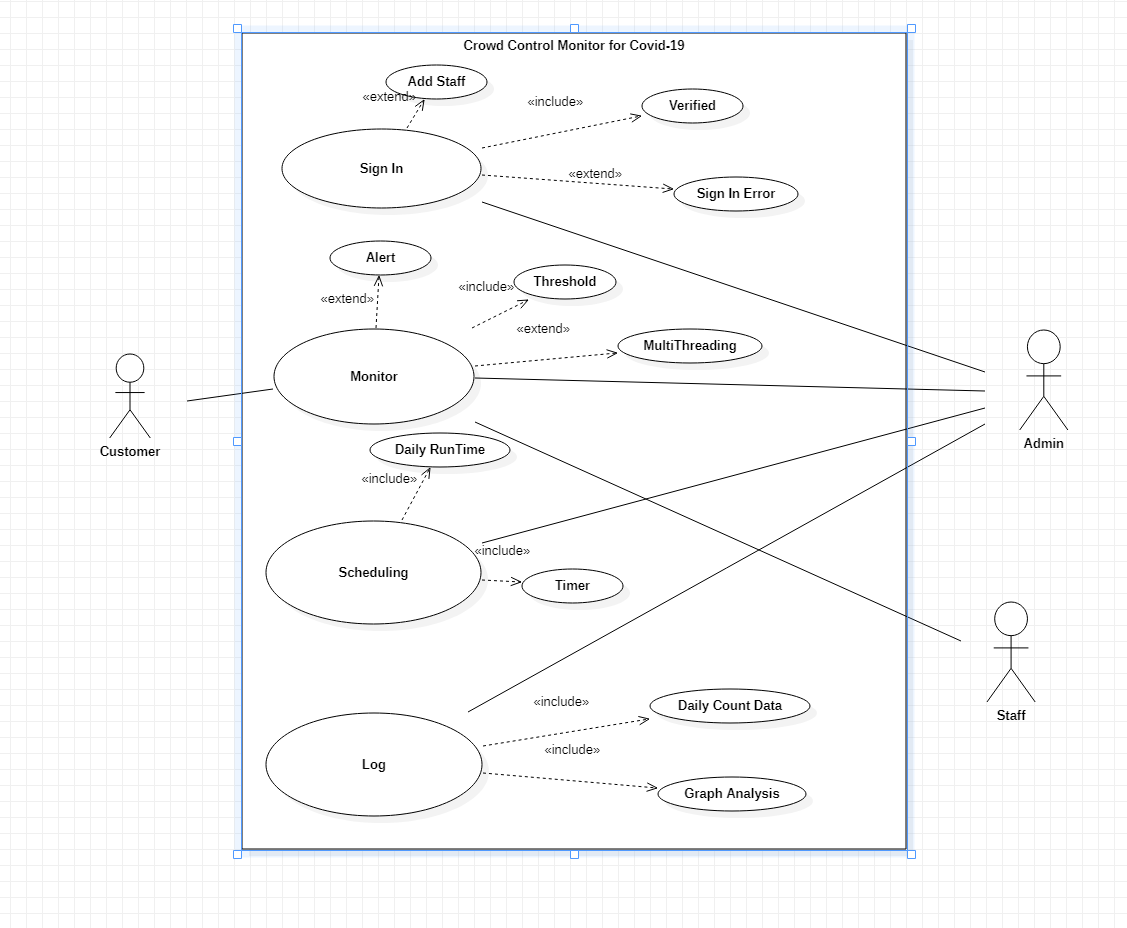
# Structure Chart:



**Use Case Diagram:**



**Use Case Description for all the use cases:**

|  |  |
| --- | --- |
| Use Case ID: | UCID001 |
| Use Case Name: | Sign In |

|  |  |
| --- | --- |
| Actor: | Admin |
| Description: | Signs in user and creates an account if not already there. |
| Preconditions: | Stable network connection |
| Postconditions: | The user is signed in to the application |
| Priority: | medium |
| Frequency of Use: | nil |
| Normal Course of Events: | 1. User opens application 2. User enters username 3. User enters password 4. User clicks on Sign In button |
| Alternative Courses: | 1) Show error if not signed in |
| Exceptions: | nil |
| Includes: | Verified |
| Special Requirements: | nil |
| Assumptions: | nil |
| Notes and Issues: | nil |

|  |  |
| --- | --- |
| Use Case ID: | UCID002 |
| Use Case Name: | Monitor |

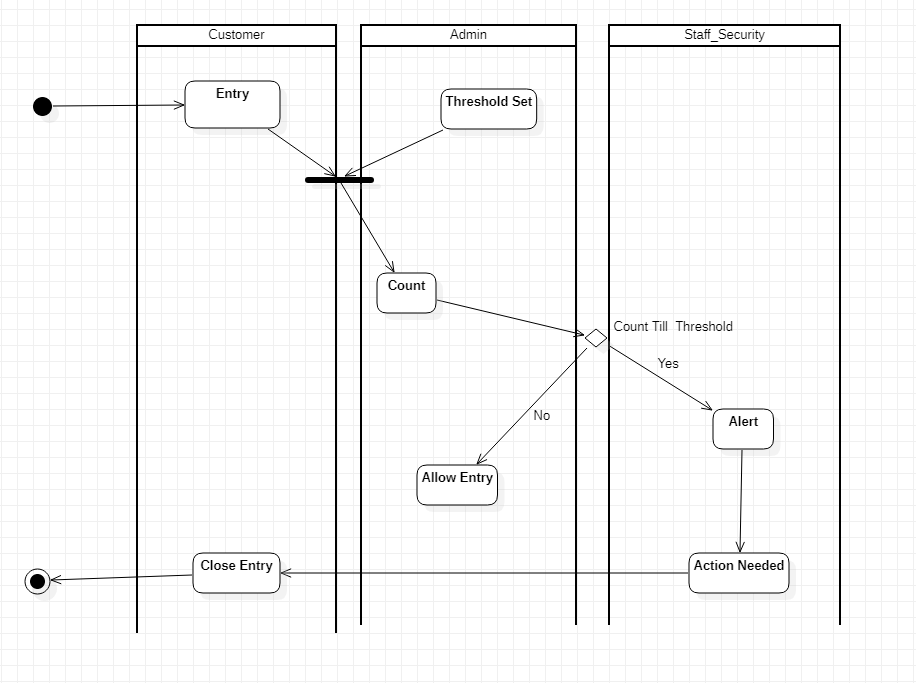
|  |  |
| --- | --- |
| Actor: | Customer, Admin, Staff |
| Description: | Counts the number of people present under the surveillance of  camera. |
| Preconditions: | The user must be signed in |
| Postconditions: | The number of people in the crowd counted |
| Priority: | High |
| Frequency of Use: | nil |
| Normal Course of Events: | 1. Camera surveillance fetched 2. The number of people counted 3. Count check against a threshold value 4. Display count 5. Send an alert if the count exceeds the threshold |
| Alternative Courses: | 1) Show error if not signed in |
| Exceptions: | nil |
| Includes: | Threshold |
| Special Requirements: | nil |
| Assumptions: | nil |
| Notes and Issues: | nil |

|  |  |
| --- | --- |
| Use Case ID: | UCID003 |
| Use Case Name: | Scheduling |

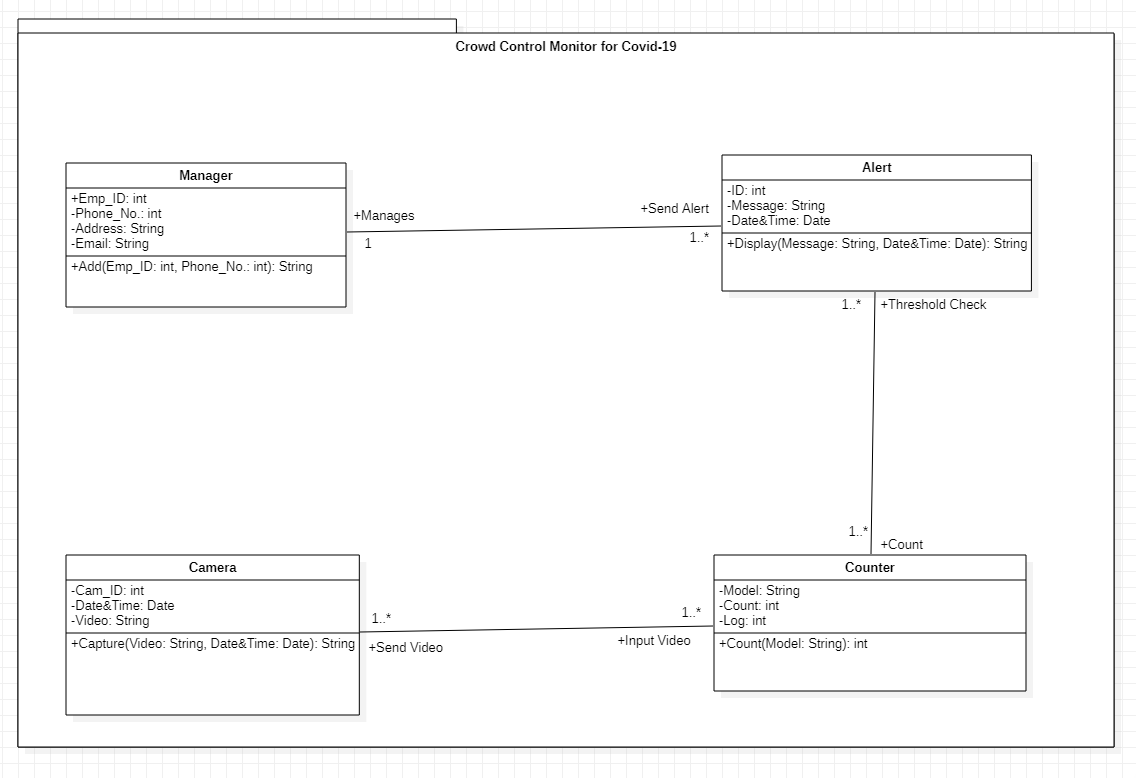
|  |  |
| --- | --- |
| Actor: | Admin |
| Description: | Schedules runtimes for the crowd control surveillance and sets timers |
| Preconditions: | The user must be signed in |
| Postconditions: | Timer scheduled |
| Priority: | low |
| Frequency of Use: | nil |
| Normal Course of Events: | 1. Go to the scheduling page 2. Choose starting time 3. Choose ending time 4. Set scheduler 5. Choose to stop time 6. Set timer |
| Alternative Courses: | 1. Show error if not signed in 2. Show error is cant set timer |
| Exceptions: | nil |
| Includes: | Timer, Daily run time |
| Special Requirements: | nil |
| Assumptions: | nil |
| Notes and Issues: | nil |

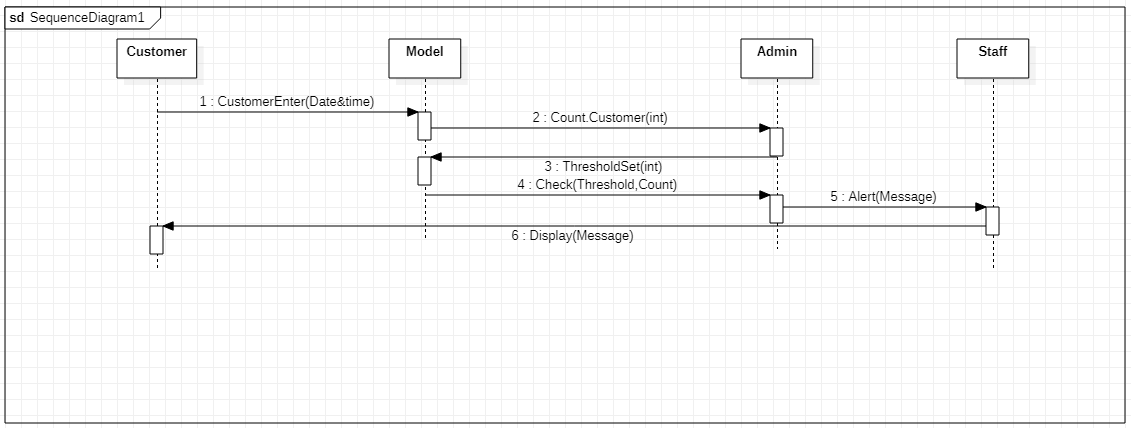
|  |  |
| --- | --- |
| Use Case ID: | UCID004 |
| Use Case Name: | Log |

|  |  |
| --- | --- |
| Actor: | Admin |
| Description: | Show Log of count and visualize it using charts |
| Preconditions: | 1. The user must be signed in 2. Log data must be available |
| Postconditions: | Log displayed and visualized |
| Priority: | medium |
| Frequency of Use: | nil |
| Normal Course of Events: | 1. Go to log page 2. Choose a day to view data from 3. Data displayed in charts 4. Tap on view data to see raw data |
| Alternative Courses: | 1. Show error if not signed in 2. Show error if data not available 3. Show raw data if visualization cant be done |
| Exceptions: | nil |
| Includes: | Graph analysis, daily count data |
| Special Requirements: | nil |
| Assumptions: | nil |
| Notes and Issues: | nil |

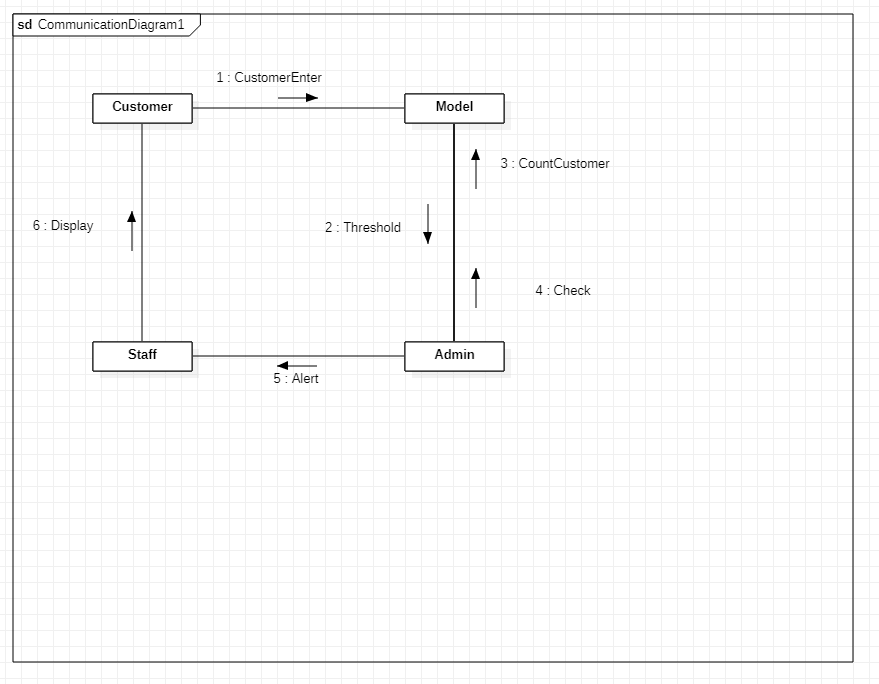
**Activity Diagram:**

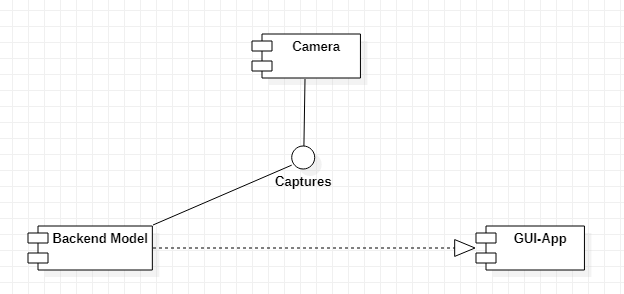
**Class Diagram:**



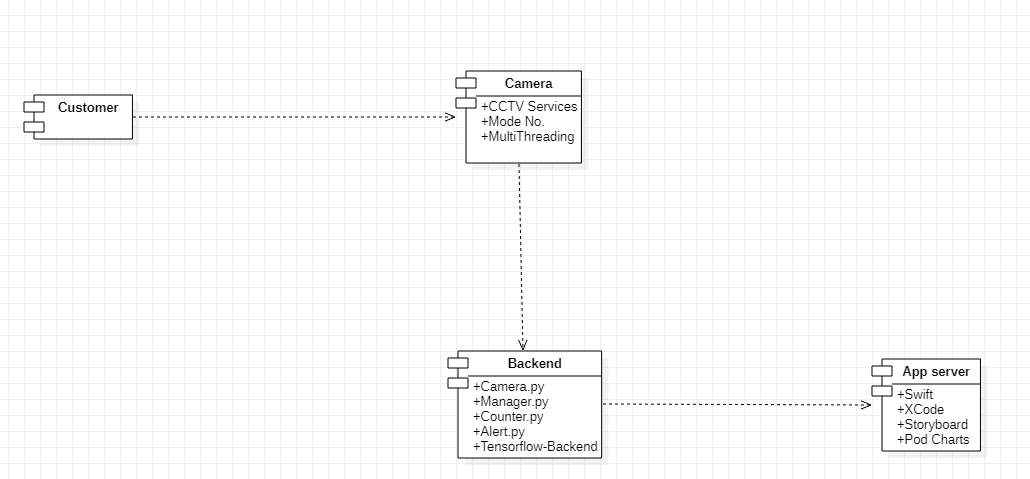
**Sequence Diagram:**

**Collaboration / Communication Diagram:**



**Component Diagram:**

**Deployment Diagram:**



**Generate the code as per your developed language.:**

1. Alert.py

**Code:**

#!/usr/bin/python

#-\*- coding: utf-8 -\*-

class Alert:

def init (self): self.ID = None self.Message = None

self.Date&Time = None self.Manages = None self.Count = []

def Display(self, Message, Date&Time): pass

1. Camera.py

**Code:**

#!/usr/bin/python

#-\*- coding: utf-8 -\*-

class Camera:

def init (self): self.Cam\_ID = None self.Date&Time = None self.Video = None self.Input Video = []

def Capture(self, Video, Date&Time): pass

1. Counter.py

**Code:**

#!/usr/bin/python

#-\*- coding: utf-8 -\*-

class Counter:

def init (self): self.Model = None self.Count = None self.Log = None self.Threshold Check = [] self.Send Video = []

def Count(self, Model): pass

1. Manager.py

**Code:**

#!/usr/bin/python

#-\*- coding: utf-8 -\*-

class Manager:

def init (self): self.Emp\_ID = None self.Phone\_No. = None self.Address = None self.Email = None self.Send Alert = []

def Add(self, Emp\_ID, Phone\_No.): pass