

**SCHOOL OF COMPUTER SCIENCE ENGINEERING (SCOPE)**

Lab Digital Assignment - II

# Done By: 19BCE0850 - Yash Khurana

**20BCT0043 - Avineesh Sathyakumar 20BCT0267 - Aman Sharma 20BDS0219 - Kunal**

**20BCI0044 - Shaurya Agarwal**

**Under the Guidance of Prof. Ramesh Babu K**

**For the course Course Code: CSE3001**

**Course Name: Software Engineering**

### What is UML?

The Unified Modeling Language (UML) is a language used in the field of software engineering that represent the components of the Object-Oriented Programming concepts. It is the general way to define the whole software architecture or structure

In Object-Oriented Programming, we solve and interact with complex algorithms by considering themselves as objects or entities. These objects can be anything. It can be the bank or a bank manager too. The object can be a vehicle, animal, machine, etc. The thing is how we interact and manipulate them that they can perform tasks and they should.

The tasks can be interacting with other objects, transferring data from one object to another, manipulating other objects, etc. The single software could have hundreds or even thousands of objects. So, UML provides us a way to represent and track those objects in a diagram to become a blueprint of our software architecture.

### What are UML Diagrams?

A UML diagram shows the unified visual presentation of the UML (Unified Modeling Language) system intending to let developers or business owners understand, analyze, and undertake the structure and behaviors of their system.

So far, the UML diagram has become one of the most common business process modeling tools, which is also highly significant to the development of object-oriented software.

### Advantages of UML Diagrams

UML diagrams have many benefits for both software developers and businesspeople, and the most key advantages are:

Problem-Solving - Enterprises can improve their product quality and reduce cost especially for complex systems in large scale. Some other real-life problems including physical distribution or security can be solved;

Improve Productivity - By using the UML diagram, everyone in the team is on the same page and lots of time are saved down the line;

Easy to Understand - Since different roles are interested in different aspects of the system, the UML diagram offers non-professional developers, for example, stakeholders, designers, or business researchers, a clear and expressive presentation of requirements, functions and processes of their system.

Usages of UML Diagrams

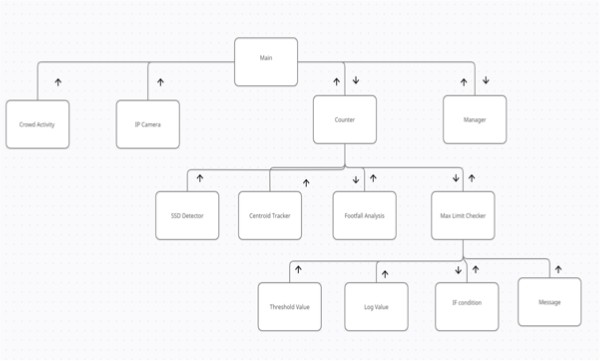
The diagram can be used in many different fields including software engineering or business processes to strengthen efficiency.

Draft the System- In this case, the UML diagram is used by the development team to discuss the outlines and structure the overall system. This may include the forward design and the backward design for different activities, roles, actors, and so on;

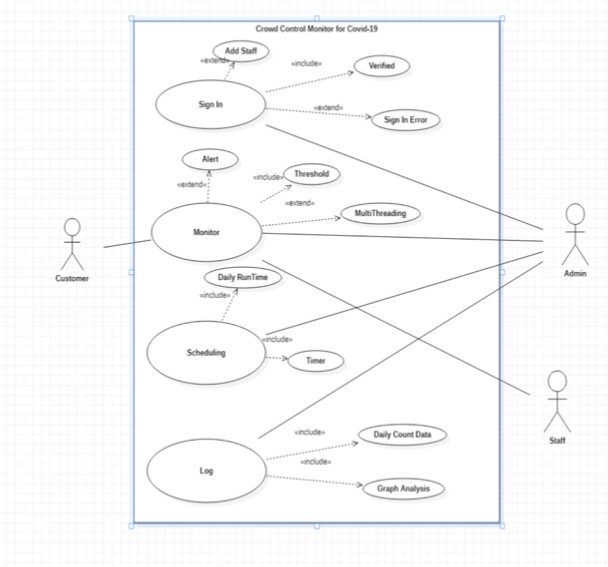
Visualize Programming Language - Different types of UML diagrams in a certain system can be translated into code directly to save time for software or related application development;

Business Analysis - In reality, the UML diagram can also be used to analyze the business sales pathway in order to improve customer service;

# Structure Chart



**Use Case Diagram**



## Description of our UML Diagram (Architecture)

The main interaction happens between the application and the admin/staff.

The user or the customers that are examined by the admin via CCTV footage or the video feeds. The frontend shows the basic layout of the app and what it consists of such as logs and keeps track of the no. of people coming and going i.e count.

The application layer consists of Python Script, SSD Detector, and Centroid Tracker which is based on the concept RCNN/ Convolutional Neural Network. The Database layer used is Firestore Database with Firebase Server and we have implemented

authentication via Firebase Authentication System. This UML Diagram perfectly represents our software architecture and how will it be implemented.

## Why are we using this Architecture?

1. It enables you to refresh the innovation heap of one level, without affecting different spaces of the application.
2. It takes into account distinctive advancement groups to each work on their specialized topics. The present engineers are bound to have profound competency around there, such as coding the front finish of an application, rather than dealing with the full stack.

12

1. You can increase the application and out. A different back-end level, for instance, permits you to send to an assortment of information bases as opposed to being secured in one specific innovation. It additionally permits you to increase by adding different web workers.
2. It adds unwavering quality and more freedom to the basic workers or administrations. 5.It gives simplicity of support of the code base, overseeing show code and business rationale independently, with the goal that a change to business rationale, for instance, doesn't affect the show layer.