**User Requirements (Functional)**

**1. Image Capture and Quality**

1.1 The system should capture clear and detailed images.  
1.2 The system shall operate effectively in a variety of lighting conditions (bright, dim, etc.) and capture clear and detailed images.

**2. Object and Person Recognition**

2.1 The system shall accurately identify specific objects and persons.  
2.2 The system shall support incremental learning to recognize new objects over time.  
2.3 The system shall recognize and identify objects in real time.  
2.4 The system shall validate object recognition performance using academic principles of machine learning and computer vision.  
2.5 The system shall use YOLO as the preferred model for object detection.

**3. People counting**

3.1 The system shall count the number of people entering a defined area.  
3.2 The system shall allow evaluation using test videos of known crowd sizes (validated through manual inspection).

**4. Alerts and notifications**

4.1 The system shall allow users to configure alerts for specific events or detections.  
4.2 The system shall send alerts via multiple channels (email, SMS, in-app notification).

**5. Data Management**

5.1 The system shall integrate with existing data systems (where applicable).  
5.2 The system shall provide easy access to historical and real-time data.  
5.3 The system shall transmit data to a central server using the MQTT protocol.

**6. User Interface and Accessibility**

6.1 The system shall have a simple user interface for setup and monitoring.  
6.2 The interface shall clearly display detected objects and alert history.  
6.3 A web-based dashboard shall present relevant insights and metrics.

**7. Remote access**

7.1 The system shall allow remote access to the live video feed and system settings.

**8. Labeling and Training Data**

8.1 The system shall use labelling tools (e.g., Label Img) to annotate images and train recognition models.

User Requirements (Prioritized)

1. **High Importance:**
   1. Take Good Pictures:
      * Capture clear, detailed pictures.
      * Work well in different lighting (bright, dim, etc.).
   2. Recognize Objects:
      * Accurately identify specific things (products, people).
      * Learn to recognize new things over time.
      * Make sure the camera is good at recognizing simple objects.
      * Find and name objects in the picture in real-time.
   3. Send Alerts:
      * Let users set up alerts for certain objects or events.
      * Send alerts in different ways (email, text, app).
   4. Be Easy to Use:
      * Have a simple interface for setting up and watching the camera.
      * Show detected objects and alerts clearly.
      * Use a dashboard to display the information to the user
   5. Allow Remote Access:
      * Let users see the camera feed and data from anywhere.
      * Allow users to change camera settings remotely.
   6. Use the Right Model:
      * Validate the performance of the camera for simple object recognition. This will include some academic content in the field of machine learning and computer vision.
      * Consider YOLO for recognizing objects.
2. **Medium Importance:**
   1. Manage Data:
      * Work with existing data systems (if any are used).
      * Make it easy to find and use the data.
      * Send data to a central computer (server) using MQTT for later use.
   2. Do Specific Tasks:
      * Smart Alarm: See if it's possible to use the camera as a smart alarm.
      * People Counter: Count how many people go into a place.
      * Water Monitoring: Watch for water being dumped into rivers.
      * Sewer Monitoring: Watch a sewer valve remotely.
      * Monitor Water Companies: to monitor the times and durations that they discharge any untreated sewage into the river network.
   3. Label Data:
      * Use tools like LabelImg to label the images and draw boxes around the objects of interest.