ORTA DOĞU TEKNİK ÜNİVERSİTESİ
KUZEY KIBRIS KAMPUSU

CNG 491
Senior Project and Seminar: Design

Final Report

Project Title:
Intelligent Security System for Abandoned Luggage

Fall 2017

Team Members:

Damla Gül Altunay – 2015881
Naciye Karademir – 2016095

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

This section includes the information about Intelligent Surveillance System for Abandoned Luggage (SSAL). Following parts are divided into three. In the first part, what kind of Surveillance System is going to be designed and its usage is explained. After that, scope and importance of the system is given. Finally, details regarding the development process of the project is outlined.

1.1 What kind of Surveillance System is SSAL?

SSAL is an intelligent surveillance system that combines Computer Vision and Machine Learning techniques in its implementation. The main purpose is to find the abandoned luggage/s in the scope of the indoor environment of the airport, and track its owner/s by checking previous frames.

1.2 Why is this project important?

Due to security concerns in the last decades, implementing an intelligent video surveillance system in order to help the human actor in monitoring public places has become a necessity. Unattended object detection is one of the most important and critical tasks in such systems. Most of the systems requires a security personnel to track all dangerous situations and checking illegal items of people. Besides that, they also need to keep track of abandoned objects and its owner to prevent an unexpected accident such as the bombing of a public space. It is not the most sufficient way to control public safety. SSAL will take this job from security stuff and provide a better secure environment for people in airports.

1.3 How will the process be?

Software implementation will consist of two parts. In the first part, by using computer vision algorithms, static objects in each frame of the camera will be detected. Detected objects will be sent to the second part of the program which is recognizer. In this part, it will be decided whether or not the object is luggage with the help of trained model of luggage dataset. If luggage, it will check short-term memory to find its owner, otherwise luggage will be considered as abandoned object and the alarm triggers for personnel.

2. Background, Aims and Organization

With rapidly increasing deployment of surveillance cameras in public areas, it is required to design a reliable system for keeping citizens safe. As Dahivelkar et al. [1] stated in their paper published in 2017, as a result of terrorism and global security, people are feared to take public transportation with the attacks in their mind.

The detection of suspicious (dangerous) items is one of the most important applications. These items can be grouped into two main classes, dynamic suspicious behaviors (e.g., a person attempting to attack others) and static dangerous objects (e.g., luggage or bomb abandoned in public places) [2]. The scope of this project falls into the latter category and the environment is airport as a public place.

Our aim is to design a system that detects abandoned luggage which is left in public places such as airports, and then define these objects if it becomes a threat. These threat issue will be held by doing event recognition so that we can find the owner of the luggage and evaluate its threat. This will be the main feature to distinguish this system from existing systems.

The system should be able to detect luggage and also its owner from the coming frame so that if the luggage is abandoned, we can find its owner. Abandoned object is defined as an entity which is absolutely static in the scene for more than a time period T and the owner is not present around it. To sum up, main objectives are:

- Fast detection of baggage that has been abandoned,
- Fast identification of the individual who left the baggage, and
- Fast determination of owner after moving away from the luggage

This project is initiated as a Senior Project in Middle East Technical University and carried by two students and two advisors. Project organization is handled with division of labor. Naciye is working on Computer Vision part which is the first part for the flow of the program. Damla is working on Machine Learning which is the second part. The flow of the program which is the solution for the formulated problem above will be explained in detail in the following parts.

3. Literature Review

3.1 Prior Concepts

There are lots of researches and projects about this kind of systems. Although techniques that is used can vary from one to another, generally solutions are based on only computer vision algorithms. Each have its own advantages and limitation on parameters of accuracy and computation.

There are mainly two types of methods in the literature. The first type is based on tracking [10-12]. The second type of methods use background model to detect suspected region [14-16]. The tracking-based methods encounter the problems of merging, splitting, entering, leaving, occlusion, and correspondence [13]. These problems are not easy to solve in many cases. Especially in crowded situations, it is hard to track all the objects. On the contrary, methods based on background subtraction, using an appropriate threshold procedure on the difference between each image of the sequenc and a model image of the background can provide the best compromise between performance and reliability.

In addition to that, only using background subtraction method for detection requires a dedicated size of object. In other words object should be in a defined shape, size, not necessarily to be exact but should satisfy a given threshold. It is almost impossible to consider any luggage in one type, there are lots of colors, shapes and sizes. For example, we call a musical instrument bag as luggage as well as we do for shoulder bag. In these studies, an instrument bag which is abandoned in a public space do not have a risk to become a threat. But, in 2001, a Jerusalem café was attacked by a suicide bomber carrying a bomb built into a guitar case [17]. So, it is possible for these type of luggage to be risky. Hence, this reduces the reliability of the whole system.

Besides that, machine learning is a very new topic in the field of Computer Vision. When looked at the references that we give in this report, many of them are published in last 1 or 2 years.

There are two baseline for us to solve these problem. First one's paper was written in 2017 by Liao et al., and other one was written by Sidyakin et al. Both papers are about real-time abandoned object detection. In Sidyakin's research, abandoned luggage is detected using Convolutional Neural Networks (CNNs), they designed their own architecture for CNN. But the precision and recall results are not satisfying to consider as a robust system. In addition to that they are not interested in finding owner of the object. In Liao's paper, the main objective is to recognize and analyze events which are related with abandoned/left objects. They did not specify their objects as luggage like we do, and also their events are consisted of “Taken by owner”, “Moved by un-owner” and “Theft” cases which are different from ours. So, in this paper, main issue is not to detect abandoned objects and their threat, it is about identifying the future of the abandoned objects with the given in cases. We are planning to use these paper while associating luggage and its owner.

3.2 Innovations

One of our contributions will be the usage of the system. It is going to be used as a real-time application unlike in other projects (they are working on video datasets, at least their results are belong those). According to our literature review, there is not any research for this kind of abandoned luggage detection applications with deep learning algorithms.

In addition to that, instead of using only Faster R-CNN and looking for every possible object if they are luggage, we decrease the region of interest with “background modeling”. This will help us for the elimination of the false positive bounding boxes which came as an output from Faster R-CNN.

4. Problem Solution:

Besides existing researches, our contribution will be

- Output of the system will not only detect the abandoned object but also define the object as luggage with deep learning object detection algorithms and then find its owner, evaluate threat potentials and give alarms for it if it poses a danger.
- At the end of this project, we are going to show the usability of ‘deep neural network’ for these kind of applications by using Faster R-CNN object detection algorithm.

While detecting static objects in each frame, we are using image processing algorithms, these are done by taking as a baseline of algorithm implementations of OpenCV modules but in more complex way (more explanation will be in Computer Vision sub-section). Detected static objects are sent to machine learning part of the program in which objects are detected as luggage or non-luggage (any other static object in the frame) by using trained model of created luggage dataset.

If it is a luggage, it looks for its owner back from the archive video frames, which is done with short-term and long-term foreground modelling, and tracks them. By checking time and distance with its owner, it is decided whether luggage is abandoned or not after leaving as a static object in the environment.

Below is the flow of the program that currently we are working on:

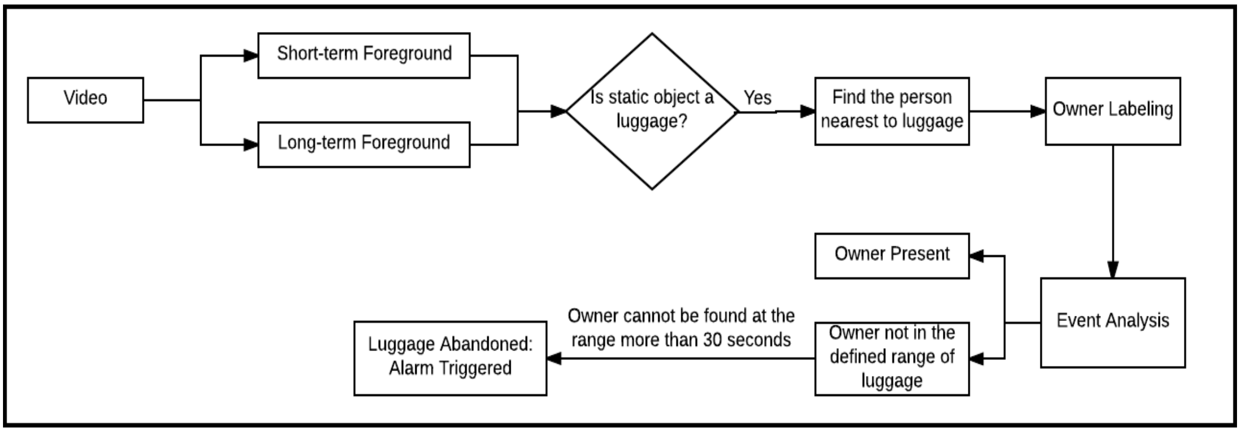


Figure 1: Flowchart of the program

Every person in the frame is detected using Faster R-CNN, and then by using short-term foreground modeling, we start to check every frame if there is a static object that has not appeared on the previous frames. If there is a new static object/s, we check these are luggage or not by using Faster R-CNN. At the same time we define the nearest person to the object as its owner by using same trained model which is created with both luggage and person dataset. Now, we have a luggage and owner of it. After that we start to analyze the event. If the owner is not in the area which is bordered with the defined distance range around the luggage, then we start to count time (30 seconds). After 30 seconds, if the person is not still in the frame or not around the luggage, then we trigger the alarm and show the luggage with a bounding box and name it as “abandoned luggage”.

4.1 Computer Vision (Background Modeling)

Since the system will be used for real-time video surveillance, we use computer vision techniques and this is one of the main parts. These techniques are used to make our system faster, more reliable and more robust.

In SSAL, time for image processing is very important because we are working on real- time videos. Using only machine learning without pre-processing in the frames leads to time-consuming calculations that cause delays that we don’t want to have in our system. According to Liao (2017), using computer vision techniques in the first step of their methodologies made their system faster and more robust for luggage detection. We are following their steps.

Machine Learning techniques are not slow but without reducing the number of objects that are candidate for abandoned luggage, the system will try to identify all of the objects in a frame. This process takes time and slows the system down. So, to reduce the number of objects, we are applying for following technique.

Since looking all objects in the video to decide whether it is abandoned luggage or not takes a lot of time, we aimed to find less number of objects for recognizer (where we decide if it is a luggage or not). We detect static objects in the video frames, and send the recognizer. To detect static objects, a dual background modeling is applied for the system. Dual background modeling is specifically used to detect static objects in videos. Flow chart for dual background modeling is below.

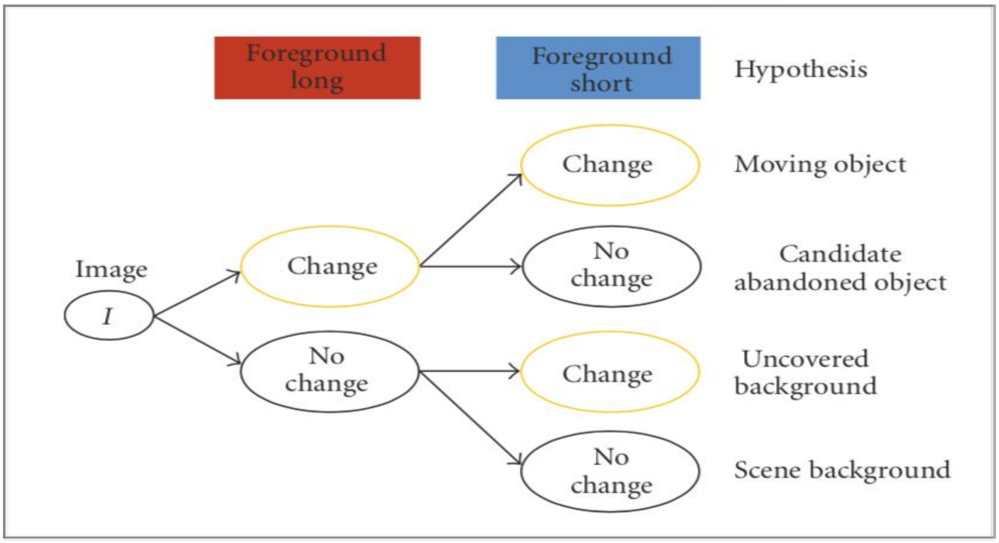
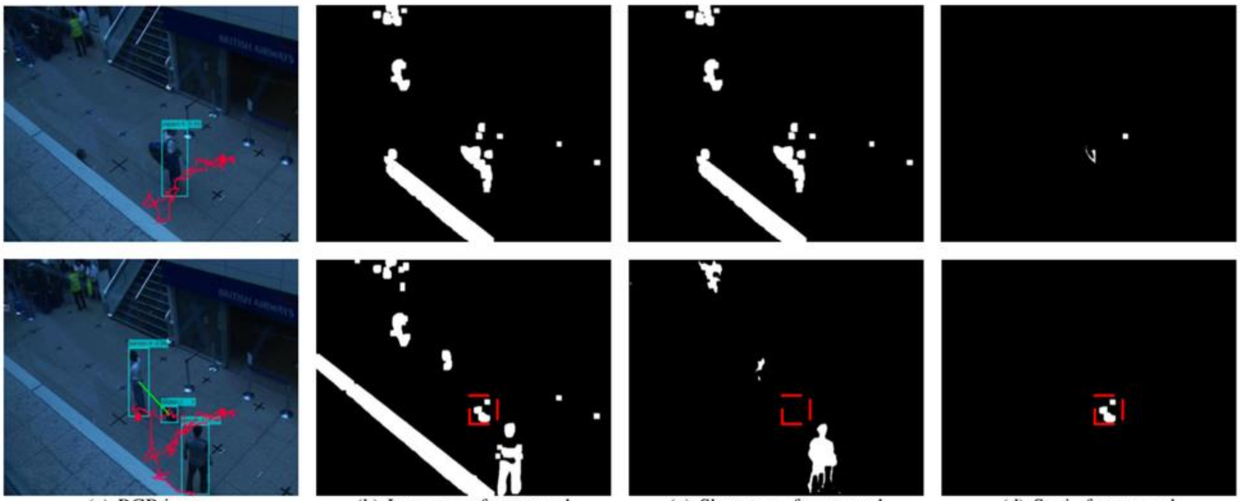


Figure 2: Hypotheses on long- and short-term foregrounds. [5]

So dual background usage basically is reducing the number of objects that should be checked by the recognizer. By this method, we are able to detect static objects first and then look at it if it is an abandoned luggage or not. To illustrate, dual background modeling steps are:

- Apply generic background modeling where we only can differentiate between static background and moving foreground.
- Differentiate between moving objects and static background in the frame.
- Keep two backgrounds in the system which are long-term and short-term backgrounds to compare background models later on. The main difference between long-term and short-term backgrounds is that system updates short-term background more than a long-term background.
- Liao (2017) states that they updated long-term background every 40 seconds and short-term background every 2 seconds. Then they compare long-term and short-term backgrounds to find which moving object in the long-term background is static now in the short-term background. We are also following their research with different parameters for the system.

- Send detected objects to the recognizer where we will understand if the object is a luggage or not by using the machine learning techniques. Liao (2017) argues that in their original work this method reduced the number of objects that proposed for abandoned luggage significantly. Their work is below.



(a) RGB image (b) Long-term foreground (c) Short-term foreground (d) Static foreground

Figure 3: An example of static foreground detection in PETS 2007 dataset. The time point in the second row is 270 frames after the one in the first row. (a) shows the person/object detection (bounding box) and owner labeling (green line). The red lines are the tracking traces of detected person. (b) and (c) are the foregrounds which are extracted from the long-/short-term model respectively. (d) shows the static foreground. The place of detected bag of interest is indicated by red bounding boxes. [4]

To use this method, Porikli (2007) says that results are better if the camera is static. We need to assume that camera that is used is static. If abandoned luggage belongs to the original background, luggage will not be detected. So, luggage should have brought to the scene after we initialize first long-term background.

The sample output from our real-time test is given below.

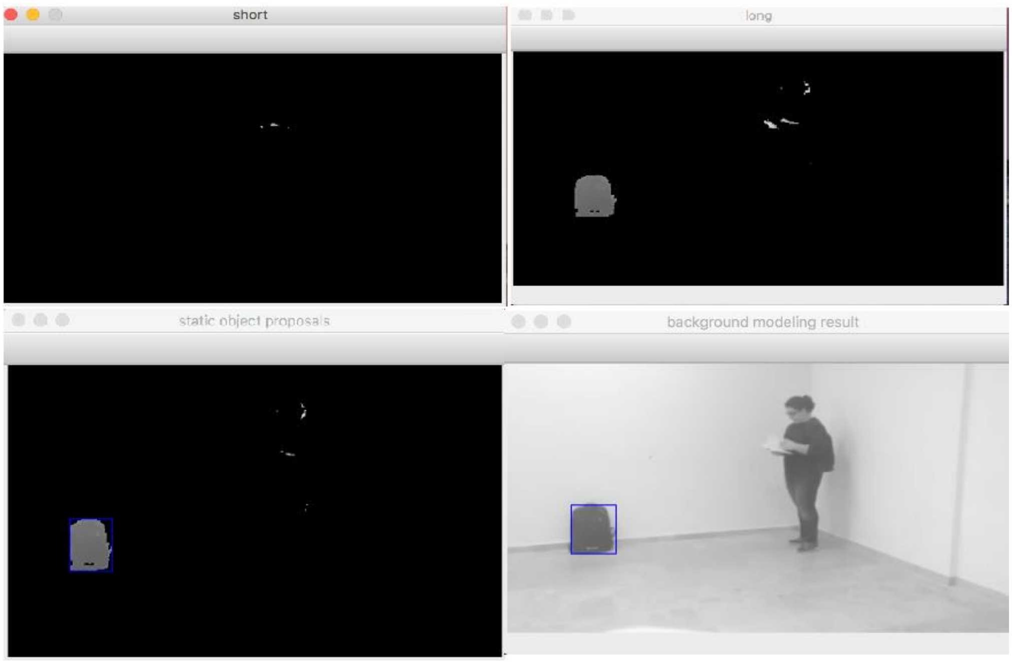


Figure 4:Short-term and Long-term Background Modelling Test Result

Before making background modeling we apply gamma correction to the current frame so that it reduces the effect of light changing in the scene. Then we use Gaussian Mixture Modeling (GMM) to decide if a pixel belongs to moving foreground or static background. After construction our first background models then we keep it as long-term background and the short-term background. The difference is we update them in different periods.

- For background subtraction, we need to apply GMM.
- Make a pixel zero if it belongs to the static background and one if it is belonging to the moving foreground. To understand the difference, we apply GMM that we are using suitable parameter for the SSAL.
- After construction our first background models then we keep it as long-term background.
- Keep the same background as the short-term background.
- Update long-term every 40 second and short-term every 2 seconds.
- Then compare two backgrounds continuously. To compare, if a pixel belongs to long-term background say it is LB, and if it belongs to short-term background say it is SB.
- Comparison (LB: Long-term Background, SB: Short-term Background):
    - LB(x, y) = 1, SB(x, y) = 1 → moving foreground.
    - LB(x, y) = 1, SB(x, y) = 0 → possible abandoned object.
    - LB(x, y) = 0, SB(x, y) = 1 → background pixel that was occluded before.
    - LB(x, y) = 0, SB(x, y) = 0 → static background.
- When a static object found, we are sending bounding box results of the object to the recognizer where we check if it is a luggage or not.

Figure 5: Static Object Background Modelling Test Result

4.2 Machine Learning

The modern history of object recognition goes along with the development of Convolutional Neural Networks (ConvNets), which was all started here in 2012 when AlexNet won the ILSVRC 2012 by a large margin. AlexNet bases on the decades-old LeNet, combined with data augmentation, ReLU, dropout, and GPU implementation. It proved the effectiveness of ConvNets and started a new era for computer vision.

4.2.1 Algorithms & Decision Process

Since 2012, many methods/algorithms have been published, and almost in every new algorithm, the results in terms of speed, accuracy and the computational cost are getting improved. For this project, we started to compare two algorithms: You Only Look Once (YOLO) [6] and Faster Region-based Convolutional Neural Networks (Faster R-CNN) [18] for training our luggage dataset. The aim was to find out the best algorithm for our application, after getting the results for both algorithms, we decided which detection algorithm is going to be used.

The results from both algorithms is shown below as a table with a small test image set.

	Faster R-CNN		YOLO	
ImageNo + description	Detection Speed in seconds	Accuracy	Detection Speed in seconds	Accuracy
Image1 – one luggage	1.14	99%	0.36	68%
Image2 –one luggage	1.30	99%	0.38	47%
Image3 – more than 2 luggage	1.36	89 – 99 %	0.32	42%
Image4 – one luggage	1.17	98%	0.37	61%
Image5 - two luggage	1.35	99% + 93%	0.43	79% + 7% + 1% (one false alarm)
Image6 - two luggage	1.12	99% + 95% + 96% (one false alarm)	0.36	36% + 11%
Image7 – one luggage	0.96	99%	0.32	31% + 56 % (one false alarm)

Table 1: Detection from image and accuracy & speed results

As shown in Table 1, although detection speed in YOLO is higher than Faster R-CNN's, accuracy results are better in Faster R-CNN. So, if the detection time is more important than the accuracy, then it is good to use YOLO. However, in our system, we applied a pre-processing operation on the image and reduce the region of interest for detection part. For this project, accuracy is the vital part since if we cannot detect the static object as luggage, then the following of the program will not have a meaning. Hence, we decided to use Faster R-CNN.

4.2.2 How Faster R-CNN works

Faster R-CNN has a two stage network, that is, it consists of two neural networks: Region Proposal Neural Network (RPN) for generating region proposals and the other network to use these proposals in order to detect/recognize the objects.

Working principle of Faster R-CNN is given in Figure 6 as stated in the paper of Ren et. al. in 2016 [18].

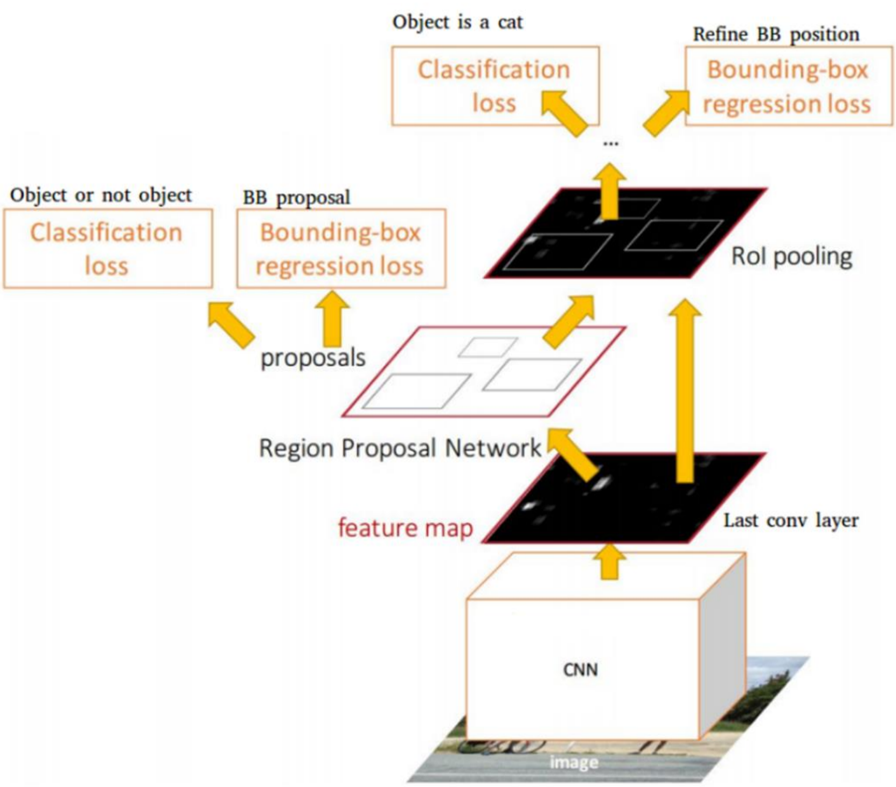


Figure 6: Faster R-CNN model [18]

Region Proposal Network is used to find out the locations where an object can be possibly, by using anchors. An anchor is a box. Illustration of anchors are given in Figure 7. In the default configuration of Faster R-CNN, there are 9 anchors, which are in different size and ratios, at a position of an image.

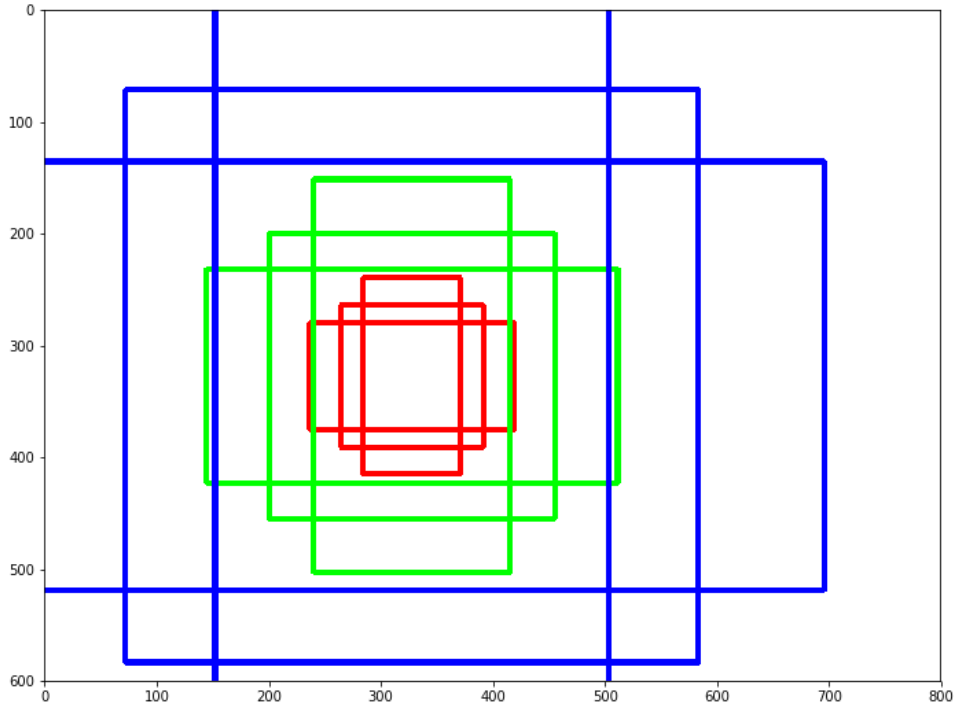


Figure 7: Anchors and location in the image

While training process, these bounding boxes are compared with the ground truth boxes in the dataset. Then, RPN calculates the intersection of unit (IoU) between anchors and boxes. If the IoU value is greater than a defined threshold, then we can say that there might be an object in that anchor/s, in paper these anchors are labeled as "Foreground". On the other hand, if IoU value is smaller than the threshold, then anchors are called as "Background".

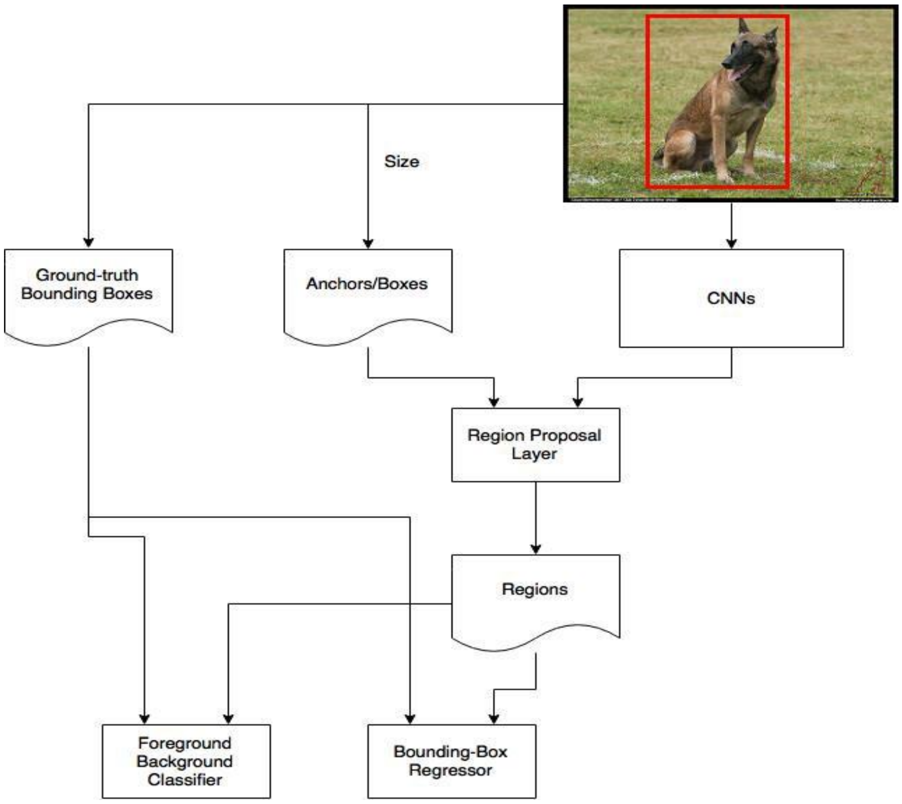


Figure 8: Working principle of Region Proposal Network in Training [19]

To summarize, every position in the feature map has 9 anchors, and every anchor has two possible labels (background, foreground). If we make the depth of the feature map as 18 (9 anchors x 2 labels), we will make every anchor have a vector with two values (called as logit) representing 'foreground' and 'background'. If we feed the logit into a softmax /logistic regression activation function, it will predict the labels. Now the training data is complete with features and labels.

4.2.2.1 ROI Pooling

After RPN, we get proposed regions with different sizes. Different sized regions means different sized CNN feature maps. Region of Interest Pooling can simplify the problem of 'working with different sized CNN maps' by reducing the feature maps into the same size. Unlike Max-Pooling layer which has a fix size, ROI Pooling splits the input feature map into a fixed number (let's say k) of roughly equal regions, and then apply Max-Pooling on every region. Therefore the output of ROI Pooling is always k regardless the size of input. With the fixed ROI Pooling outputs as inputs,

4.2.2.2 Training

The paper mentioned two ways: alternatively train the RPN, and the final classifier and regression; train them at the same time jointly. According to original paper, the latter is 1.5 times faster with similar accuracy. We trained our model with Tensorflow Object Detection API, in there, the latter approach is used.

4.3 Owner Labeling and Event Recognition

After finding out left luggage or in other words the luggage which is not moving by anyone, the person who is the closest to the luggage is defined as owner of it.

For now, we defined the events to check if the luggage is certainly unattended. It is one of the possible case that owner wants a break put his bag on a table and then go to somewhere else far away from luggage for a while, and come back. This case satisfies the rules for abandonment, but the bag is not abandoned. We aim to find out a solution for this cases.

As shown in Figure 9, a luggage item is unattended when the owner is further from the luggage. The border of yellow part is called as “warning zone” [7], where the luggage is neither attended to nor left unattended. After moving away from “red zone” which is almost 3 meters from the object, then alarm is waiting for 30 seconds to trigger.

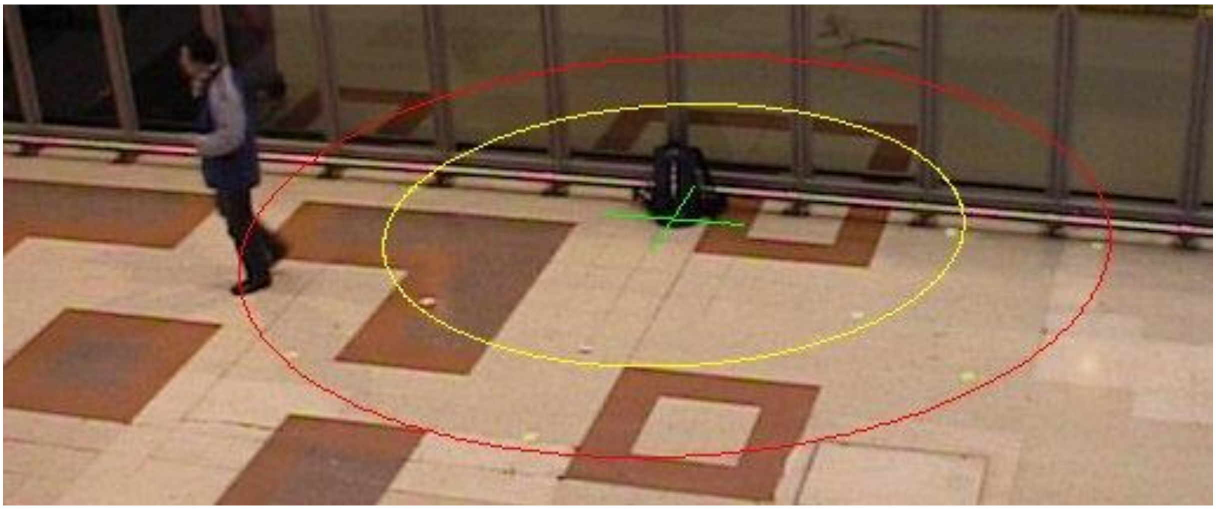


Figure 9: Distance around luggage [7]

At that time, like Liao et al. [4] did their paper by using the approach proposed by Xiao et al. in 2016 [8]. The method is to extract deep features for person and called as “Person Re-identification”. To find the person in the camera footage after leaving the red zone, we use his/her features and every time the person in the scene we show it by bounding box with the same color of bounding box of abandoned luggage.

Results from our real-time test cases are shown below:

Figure 10: Person and luggage detection in the footage

As shown in Figure 10, our system does not detect luggage until it becomes an static object. However, it always search for the person in the frame.

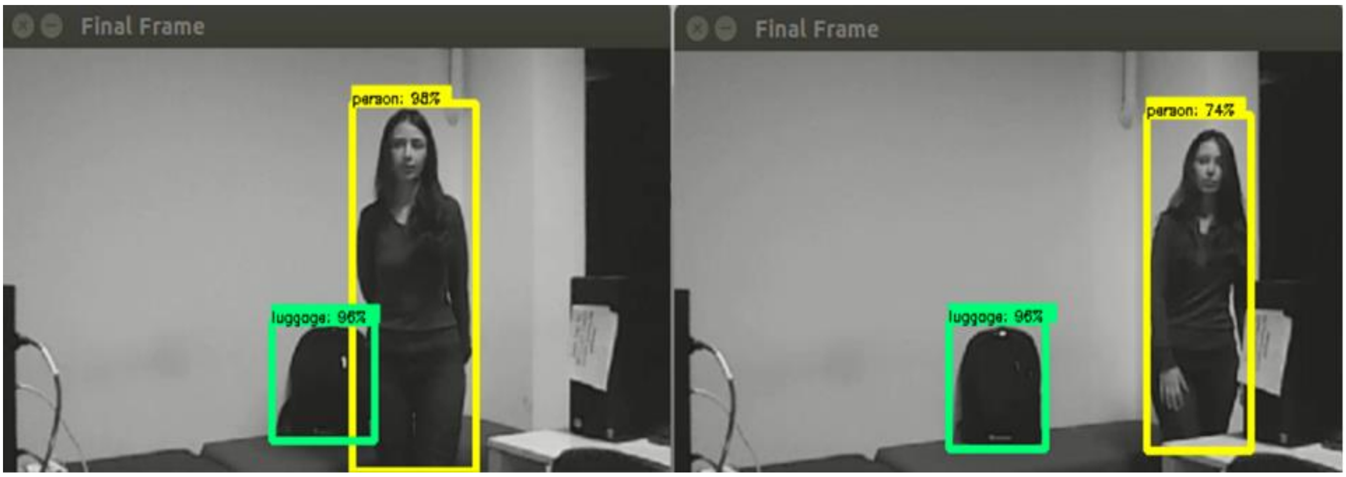


Figure 11: Owner Labeling case

For our next job, we would like to show this case (Figure 11). After luggage is defined as static object, we can still track the person around it. But, for the abandoned situation, we should consider the distance which is our next goal.

5. REQUIREMENTS

5.1 User Requirements

<u>Manage User Use Cases:</u>

Add Admin: An administrator must be defined when the system is deployed. First admin should provide the information of username, password, name, surname, department, phone number, Social Security Number (SSN), date of birth, and email address. After defining the first admin for the system, this person is responsible for adding other admins to the system if there are any.

Delete Admin: This will allow first admin to delete any other administrator. Deleting a admin will remove all information and permissions. User will not have any access to the system. This operation can be handled by using desktop application.

> **Rule:** This can be only done by the first admin who is the specified in the beginning, when the system is set up.

Add Personnel: This user case will allow administrators to add security personnel who is responsible for controlling and checking the system outputs such as warning and alarm states.

> **Rule:** While adding operation, admin should provide the information of personnel name, surname, telephone number, email, date of birth and SSN. Also number of personnel has a maximum limit which is determined by security office. After that limit, it should not be allowed to add more personnel.

Delete Personnel: This user case will allow administrators to delete security personnel from the system in case of being fired and resigning. In this situation, all information which belong to personnel will be deleted.

Edit Personnel: This will allow to administrators to update any information (such as name, surname, phone number etc.) of personnel. In this way, if there is any typing error or changed information, system can handle with this use case.

> **Rule:** This edit has to be done by administrators. Other users will not be allowed to edit process in the system.

Manage Camera Use Cases:

Activate Camera: Administrators or personnel can activate any camera from the desktop application. S/he can click "Camera" button from the home page and in the coming window, the cameras which will be activated is marked (ticked up).

Deactivate Camera: Administrators or personnel can deactivate any camera from the desktop application. S/he can click "Camera" button from the home page and in the coming window, the mark for the cameras which will be deactivated is removed.

Add Camera: Administrators can add a new camera to the system. It is done by entering IP address and the name of the camera in "Add Camera" panel after clicking the "Camera" button from the home page.

> **Rule:** This operation can be done only by administrators, security personnel are not allowed to do this operation.

Delete Camera: Administrators can remove information of a camera from the system. It is done by entering IP address and the name of the camera in the "Delete Camera" panel after clicking the "Camera" button from the home page.

> **Rule:** This operation can be done only by administrators, security personnel are not allowed to do this operation.

Manage View Use Cases:

Show Event Log: In the bottom of the home page, there will be 'Event Log' table. The table is to show the found abandoned luggage information, that is,

- Its 'ID' (assigned by the system in a sequential order for each event),
- 'Event Log' to display which event occurs in the system such as 'Abandoned Luggage', 'Owner of Abandoned Luggage,
- 'Time' to show when the event takes place
- 'Description' to indicate on which camera event took place.

Display Camera View Options: View option is to make admin able to manage camera viewing options. Admin can choose to view only one camera as well as s/h can view multiple cameras. This operation can be done from the home page. Admin/s can mark boxes to view cameras on the homepage screen or remove marks from boxes to remove the view of chosen camera.

> **Rule:** Only administrators are allowed to do this operations.

5.2 System Requirements Specification

5.2.1 Functional Requirements

Desktop Application	
DA -01	Administrator of the system should log in for management operations, otherwise there is no need to login for security personnel.
DA -02	Administrator should be able to add security personnel to the system, also s/he can edit profile of personnel and delete information from the system.
DA -03	Administrator can also add another admin to the system and do the same operation as can be done for security personnel.
DA -04	Administrator can connect a new camera to the system by entering its IP addresses.
DA -05	Administrator can remove any camera from the system by entering its IP address & name.
DA -06	Administrator can activate a camera which is already connected but not activated.
DA -07	Administrator can deactivate a camera which is connected and in use.
DA -08	Administrator can control which camera will be displayed on screen by selecting them on left-side menu.
DA -09	Desktop Application will keep track of the event log, which is created for detected luggage and its owners in order to show time and description.
DA -10	In Event log, detected abandoned luggage and its owner will have the same ID to associate each other.
DA -11	Every data in event log should be added to the database.
DA -12	After detecting abandoned luggage, it will be surrounded by a bounding box which refers that it could be a danger, and after waiting some time, if owner still does not arrive, then it will give warning signal.
DA -13	Desktop Application will include a “Help” section which gives detail information about the usage of the system.

Table 2: Functional Requirements for Desktop Application

Camera	
C -01	Camera should be connected to the Internet.
C -02	Minimum FPS should be 20, and pixel resolution should be 640*480 for the detection.

Table 3: Functional Requirements for Camera

5.2.2 Non-Functional Requirements

In the following sessions, non-functional requirements of the system are divided into usability, reliability, robustness, performance, supportability, and security.

i) Usability

The system should be easy to learn for desktop application users: Admin and security personnel. There will be a help page and complete user documentation which will explain how to achieve common tasks so that help system will explain all functions. Error messages should give specific instructions for recovery.

ii) Reliability

The reliability of the system essentially depends on the software tools (OpenCV, Tensorflow etc.) and hardware tools (camera) used for the system development. In order to handle this issue, system will always check for dependencies (tools, libraries) and functionality of camera. If there is any problem, then it will directly report the situation to the administrator of the system.

iii) Robustness

Problems in data delivery can occur anywhere with the communication between data sources, in our case it is the camera, and the surveillance system. We take robustness as not only that the system continues to run in the wake of data failures or delays, but also that it continues to provide meaningful information to its users. Our main priority should be this in order to handle the problem. Since if the users aware of what happened, then it could find a solution. Besides that, the system is going to check every error and failure to find a solution for them and if the problems cannot be handled then it will be reported to the administrator.

iv) Performance

Image data transfer through Internet connection and live streaming makes performance measures critical. For desired performance, image capturing, transferred data size, speed of connection, response time, processing speed must be considered. System should work in real-time which means that data should not be faced any serious delays. Because of this, IP cameras that is going to be used in this system requires high values for FPS and minimum 640x480 resolution pixel to recognize objects from each frame.

The database should have sufficient memory for incoming data from camera. Also, there will be another record for event log, in which detected abandoned luggage and its owner images, ID, time and description will be stored.

v) Security

All data for camera and event log is going to be stored in the database. To access this service, user should have an administrator username and password so that s/he can login to the system. In this way, we can protect our data.

5.3 Context Diagram & Architectural Model

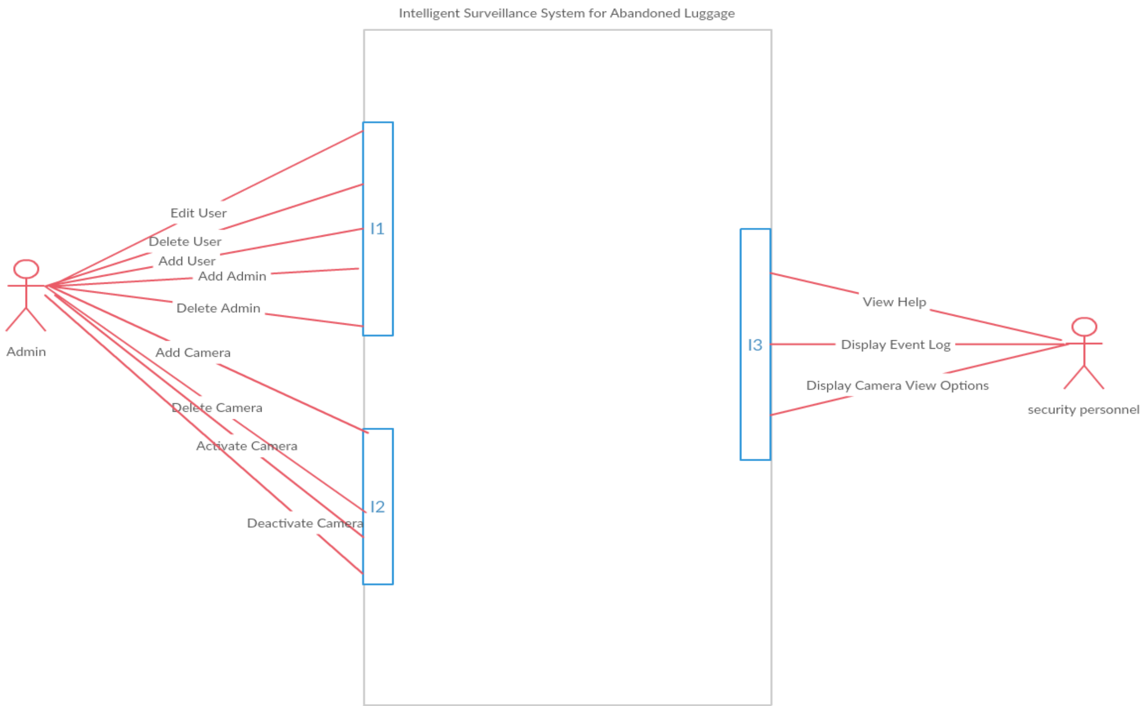


Figure 12: Context Diagram

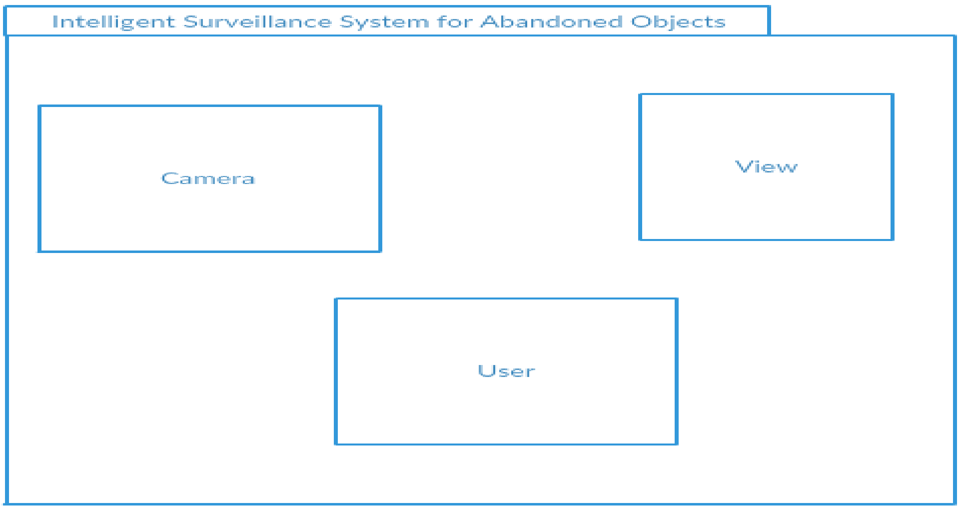


Figure 13: Architectural diagram

5.4 Use Case Diagram

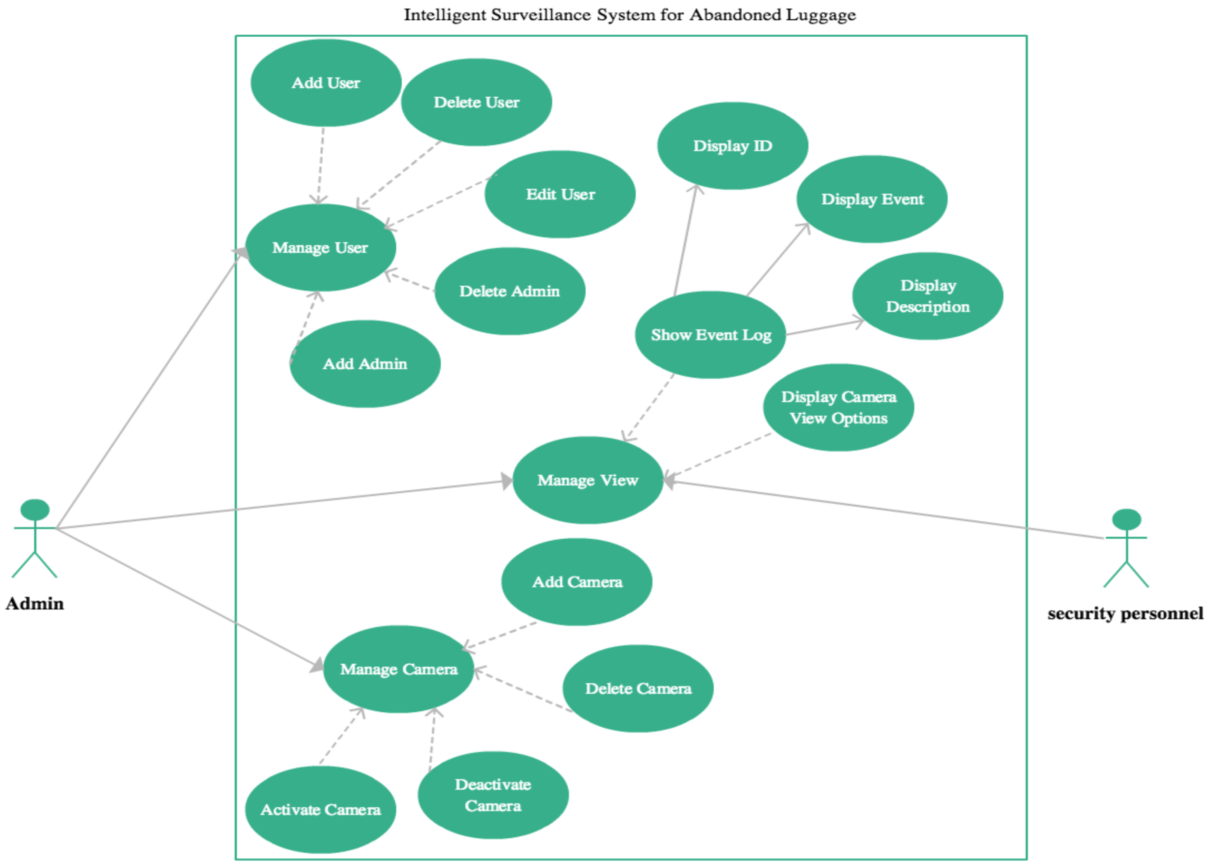


Figure 14: Use Case Diagram

5.5 Process Model

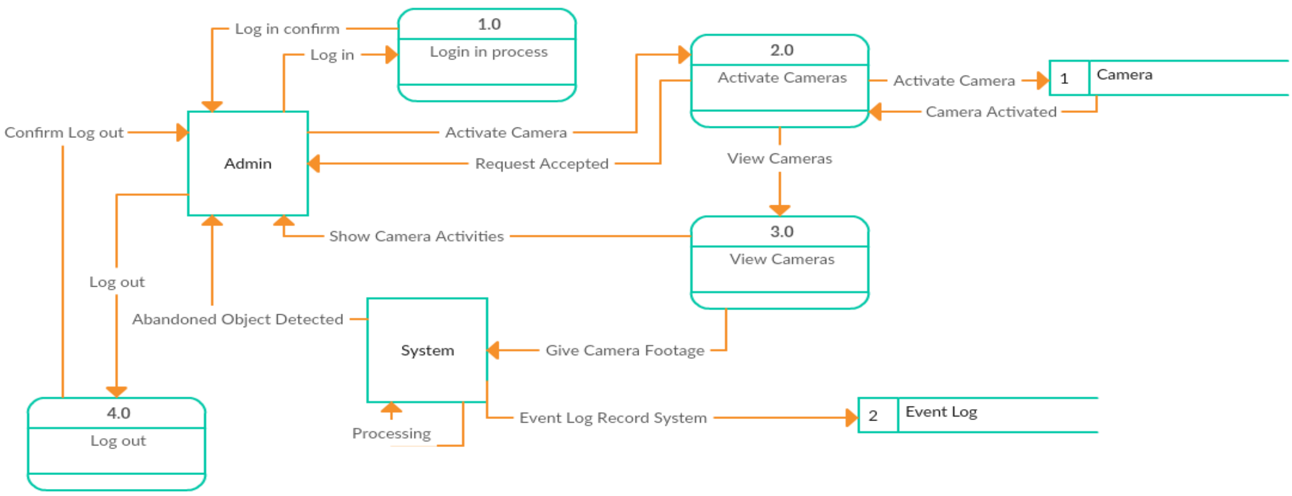


Figure 15: Process Model

5.6 Sequence Model

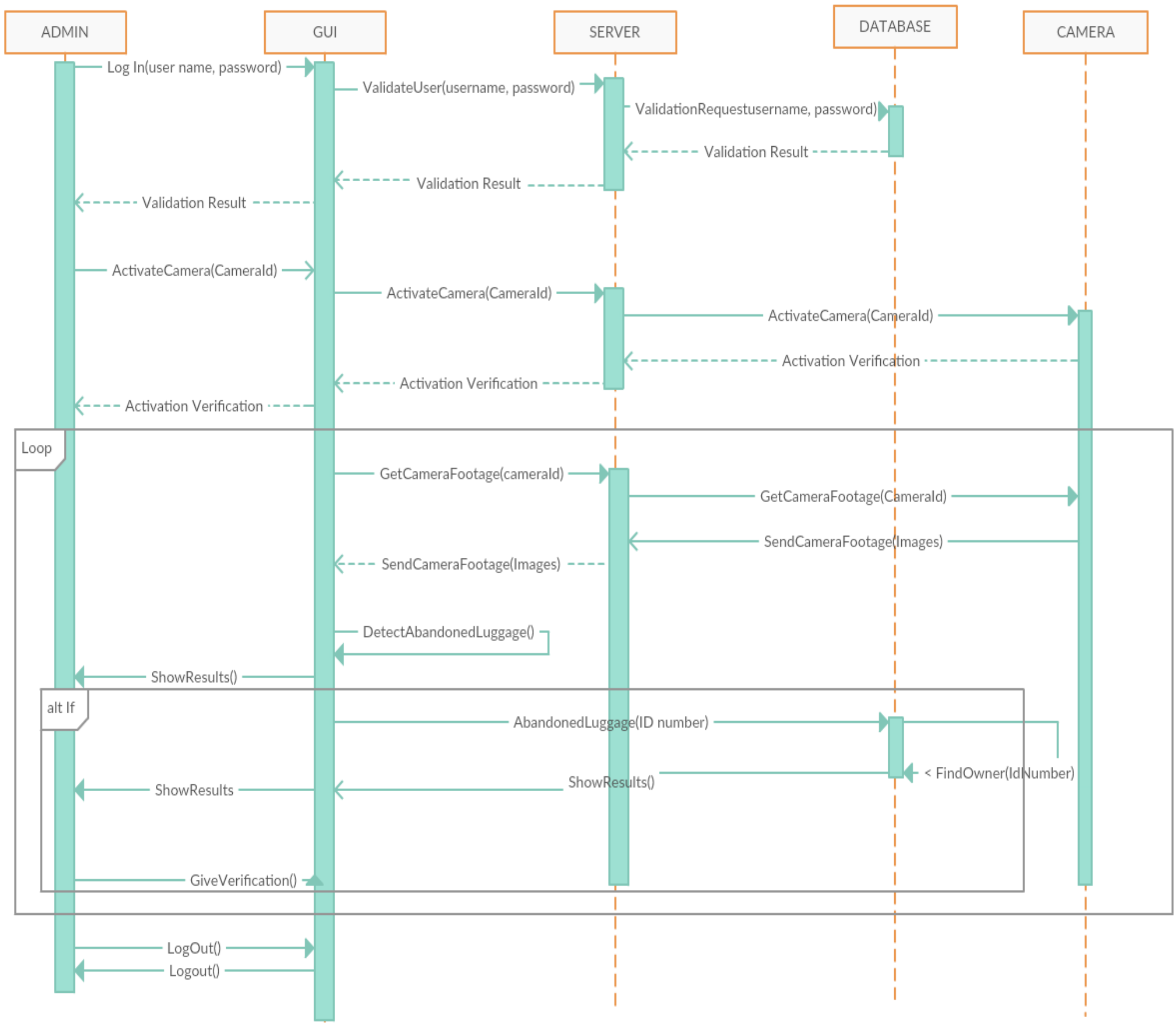


Figure 16: High-Level Sequence diagram

5.7 Graphical User Interface

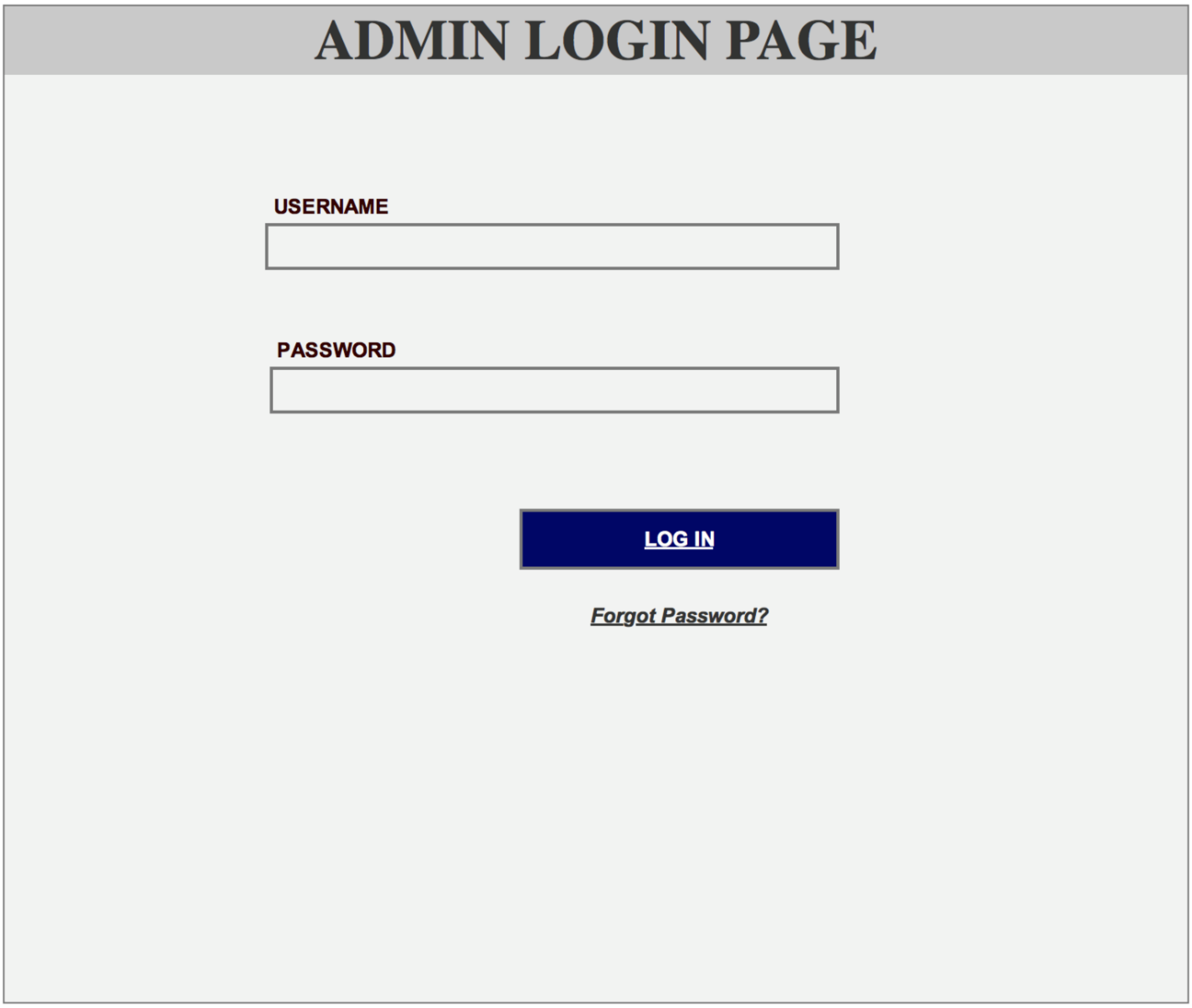


Figure 17: Login Page for Admin

GUI design for our surveillance system desktop application is designed to have multiple users: Administrator/s and Security Personnel. In order to do some operations on adding, editing personnel, admin etc., one of the administrators should log into the system. The page above is planned to use in the first step of our application. Every admin will have a username and password for the system. If s/he tries to enter the system without a valid username and/or password, system will show an error message to the administrator. If the admin forget the password and click the “Forgot Password?” system will send an email to the registered address for the new password. After administrator logs into the application, home page will be displayed as the figure below.

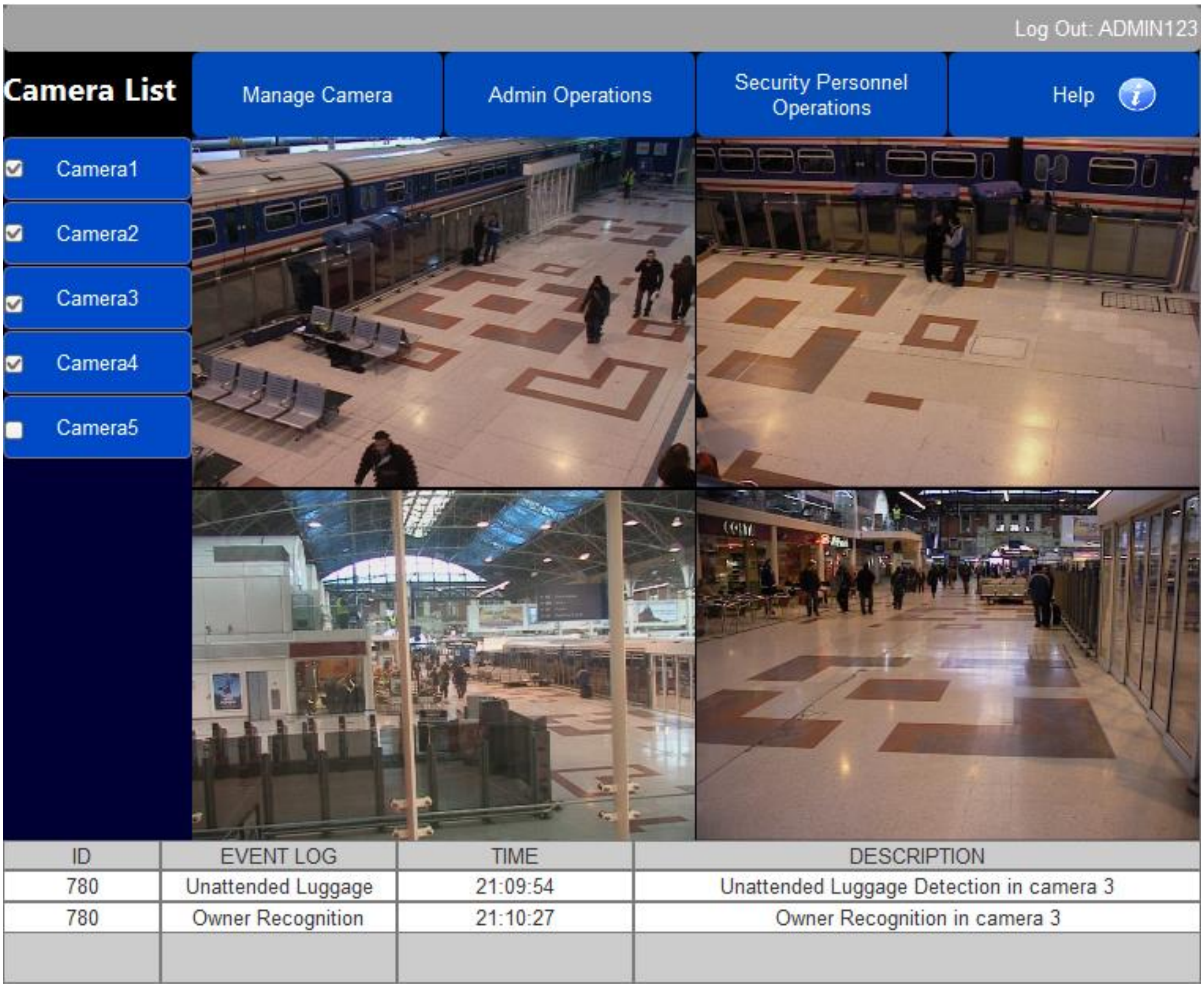


Figure 18: Home Page of the Application [1]

In the left side, admin will be able to see the camera list to arrange which cameras will be displayed in the home page. By marking the boxes for the camera, the screen will be added. If the mark is removed, then the screen for that camera will not be displayed any more.

In the below of the page, 'Event Log' flow will be displayed as a table. After detecting any abandoned luggage and its owner, an ID will be assigned for them (which should be same to associate), and Event time, description (to indicate on which camera the event is occurred).

On the top side, there will be links to direct the administrators. These are "Manage Camera", "Admin Operations", "Security Personnel Operation" and "Help". In the following parts, all design of all these links and their function is explained.

5.7.1 Manage Camera

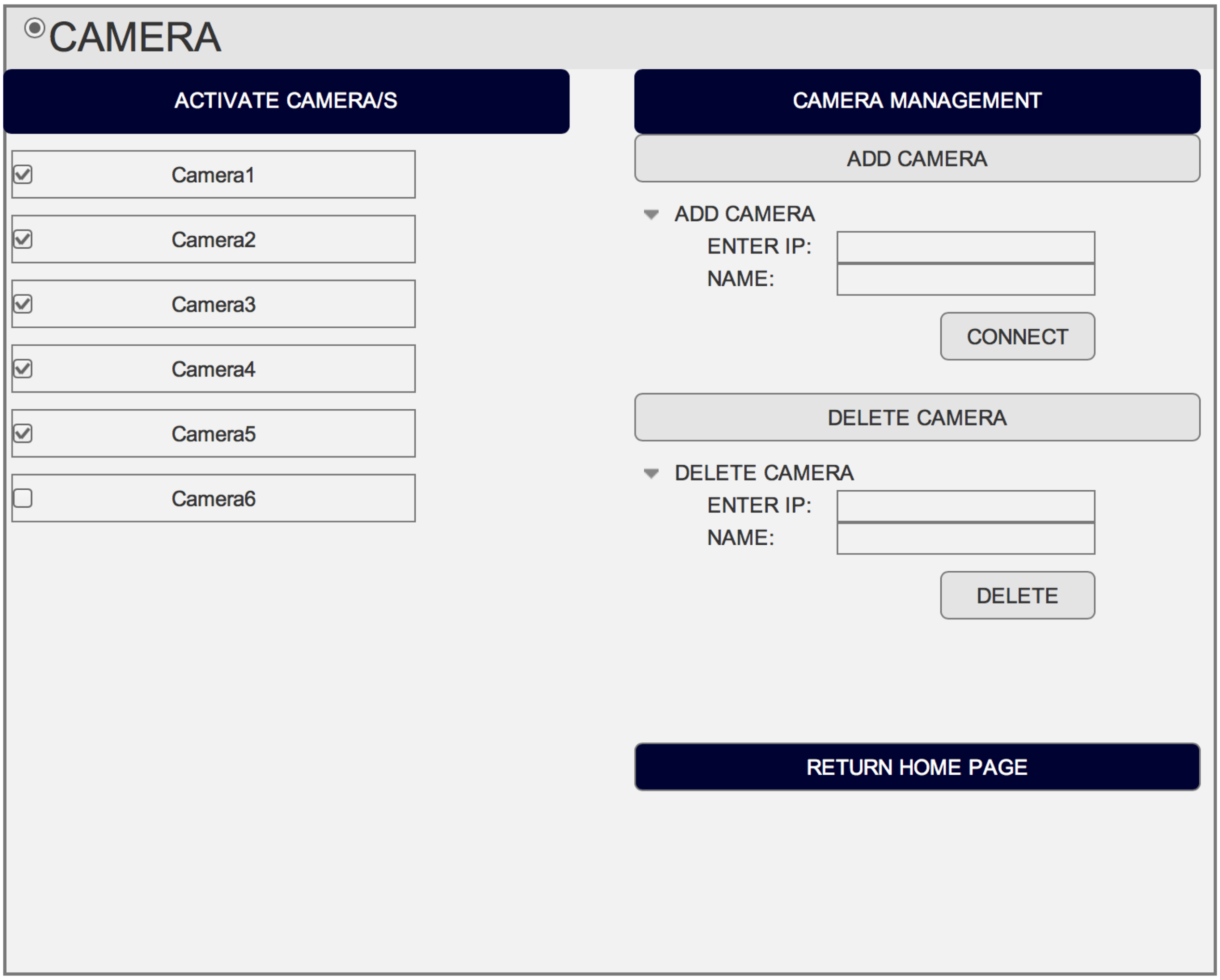


Figure 19: After clicking "Camera" link from Main Menu

In the 'Manage Camera' page, the administrators will be able to activate or deactivate any camera that the system has. In the same page, camera can be added by entering the IP address and name of the camera. Furthermore, delete camera option will be available on this page so that admin also can remove any camera from the system. Admin can return to the home page by clicking to the 'Return Home Page' button which is located the below of the page.

5.7.2 Admin Operations

The property of this page is that it is only accessible for first assigned administrator. In other words, other administrators have not right to access this page. This is designed for adding a new administrator to the system or deleting from the system. 'Admin Operations' page is shown in Figure 20.

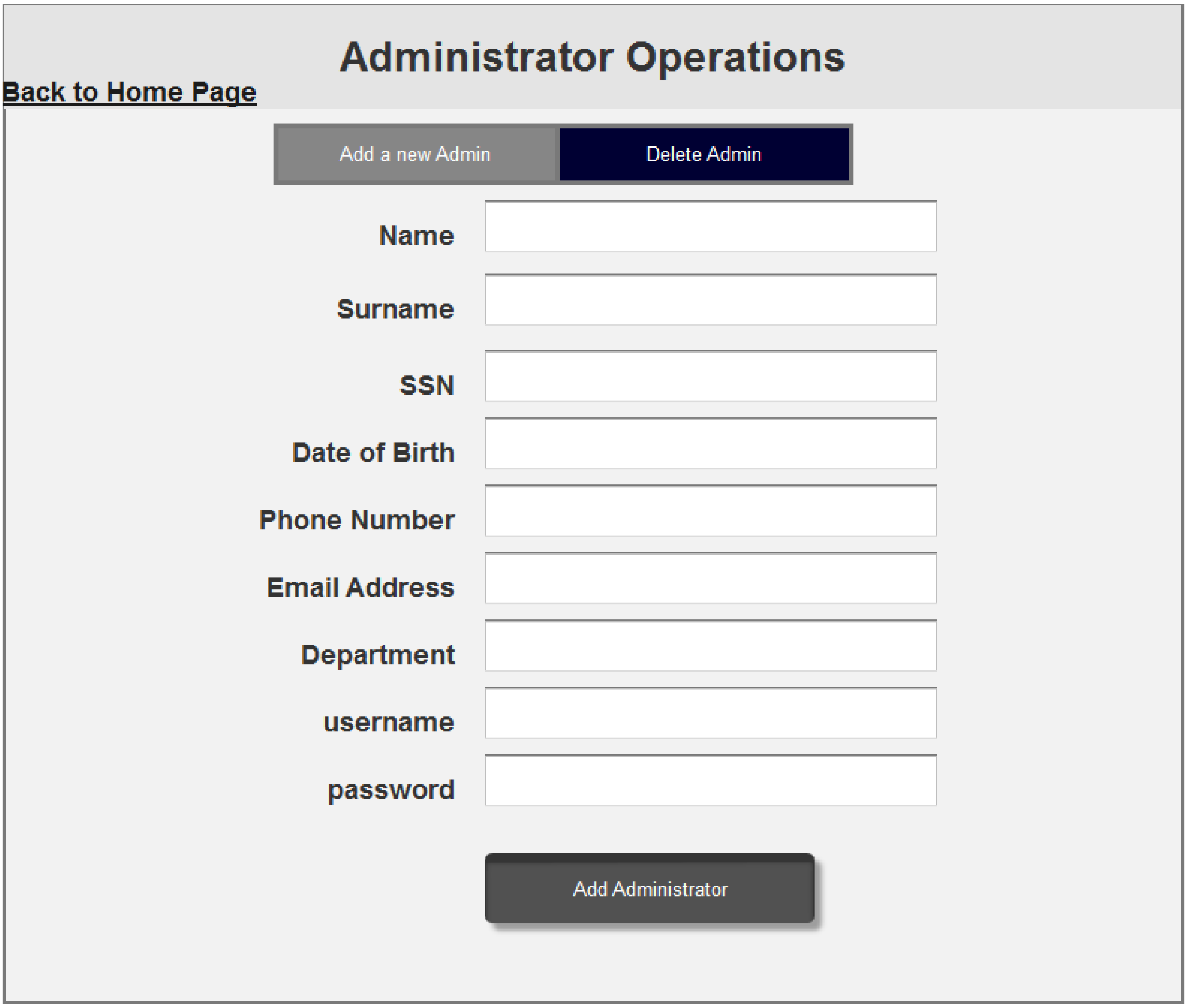


Figure 20: Administrator Operations Page

In this page, admin will have two options: Add a new Admin and Delete Admin. For the default option, 'Add a new Admin' page will be displayed. Here, all information should be provided in order to add the new person, otherwise system will give error messages and will not be able to add.

After clicking 'Delete Admin' button, following page will be shown (Figure 21).

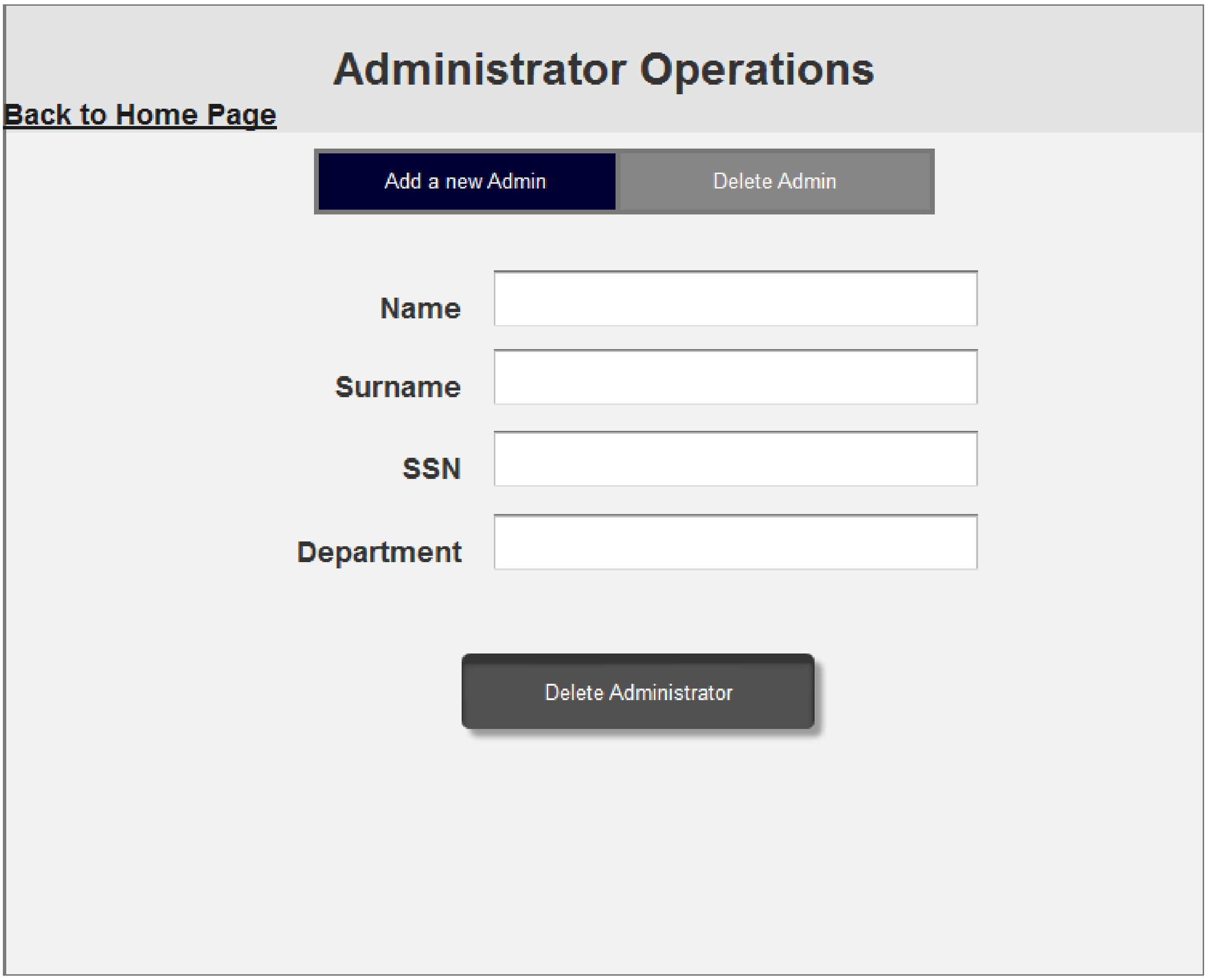


Figure 21: 'Delete Admin' page

To delete an admin, name, surname, SSN (social security number) and department of the person should be entered. If the information is found in the database, deleted operation can be handled successfully, otherwise it will give an error message.

5.7.3 Security Personnel Operations

Unlike in previous described page, the operations in "Security Personnel Operations" can be handled by all administrators. As a default, "Add a new Personnel" page will be displayed (Figure 22).

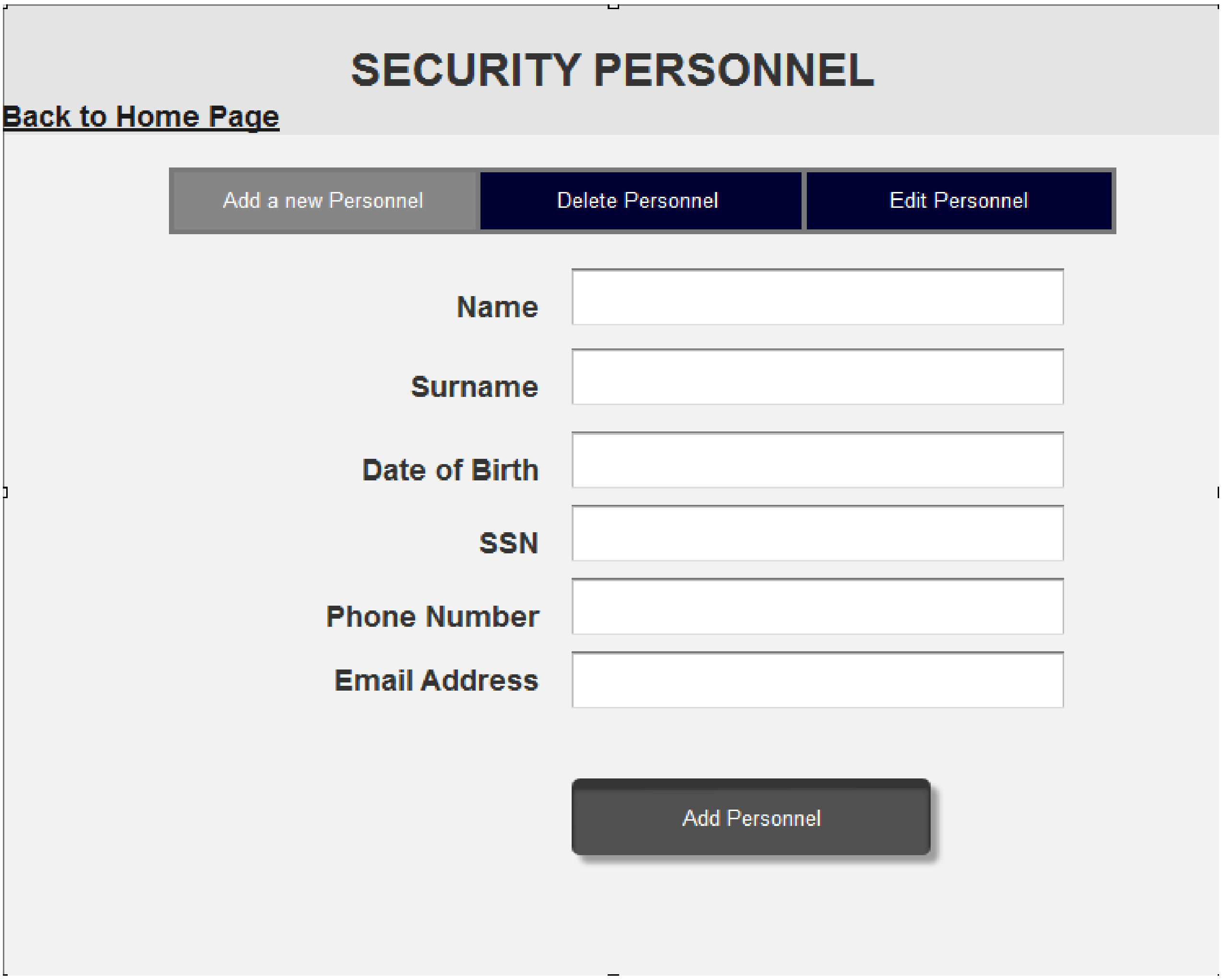


Figure 22: "Add a new Personnel" Page

As shown above, all information (name, surname, date of birth, SSN, phone number and email address) should be provided by the admin. If not, addition operation cannot be done.

In order to delete a personnel from the system, admin should click "Delete Personnel" from the top menu. The page will look like in Figure 23.

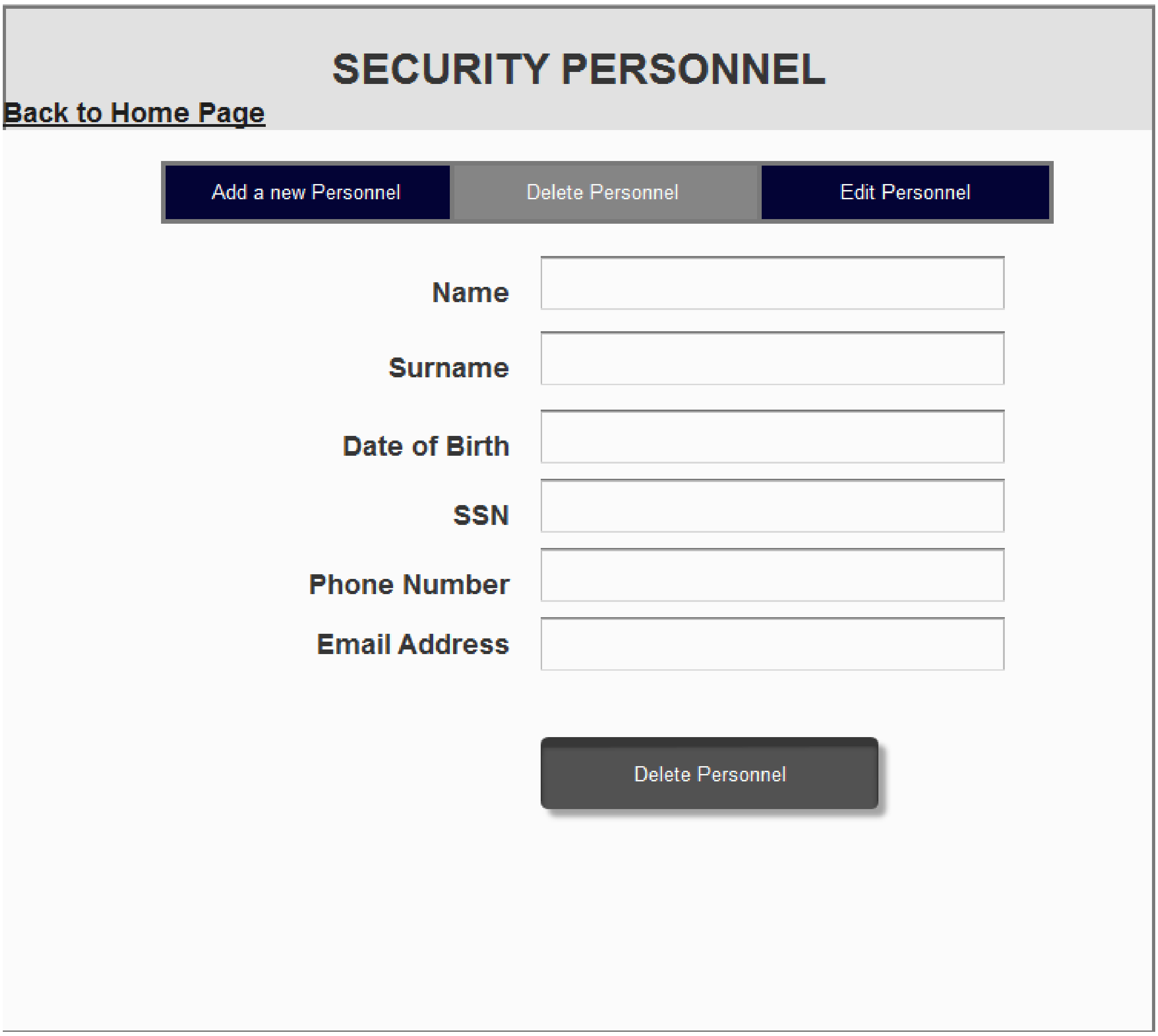


Figure 23: 'Delete Personnel' page

To delete the right person from the system, all information, that are entered while registering the person, should be entered so that administrator is prevented to delete another person.

"Edit Personnel" page is shown in Figure 24.

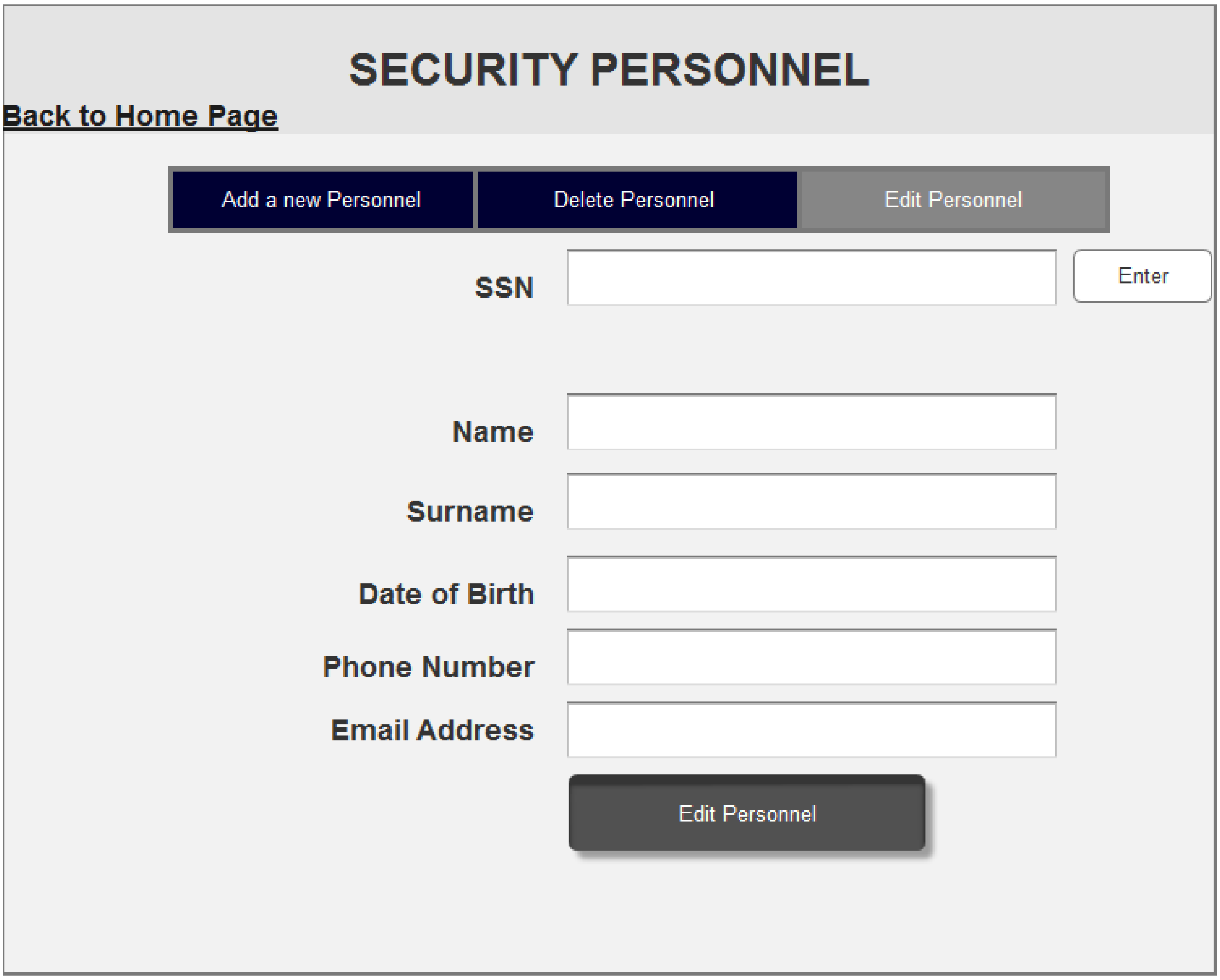


Figure 24: "Edit Personnel" page

Before entering data which would like to be changed, admin should enter the SSN of the personnel so that all information which belongs to that person will be displayed in the text fields but it will be allowed to change them. The purpose of entering SSN before other information is to check previous information of the person.

5.8 Special Requirements & Assumptions

We included a reasonable amount of assumptions to the SSAL so that our system could be more useful in a variety of surveillance systems. All assumptions are stated below.

1) Assumptions on Abandoned Object: There is various definition of what abandoned object is. Tian (2008) [9] gives the definition of an abandoned object as to be a stationary object that has not been on the scene before. We also assume that it is left by someone with a bad intention. Also, PETS dataset assumption is that an object has not its owner in 3 meters for more than 30 seconds. Based on these papers we assume that abandoned objects are left by its owner more than 30 seconds and its owner is not in the 3 meter and we will look if that object is a luggage or not.

2) What is Luggage: Dictionaries define luggage as suitcases, containers and other bags which hold personnel belongings while traveler is in transit. Since SSAL planned to be used in airports we assume that luggage is every bag can be threatened if it is left intentionally by its owner while traveling.

3) Indoor Environment: To make SSAL more useful another assumption is that it is not used for an outdoor environment. Indoor and outdoor systems both have usually similar features but outdoor systems should be able to compatible every kind of weather and lightning. This is why we assume that the environment is considered for only indoors.

4) Use of Static Camera: We assumed that camera that we add to the system should always remain static. This assumption relates to background modeling in our system. Porikli (2007) indicates background modeling is stronger under static cameras. The reason is that we are trying to find static objects and if the camera moves it is harder to detect static objects from it and usually gives wrong results.

5) First Frame of Camera: As we stated above we will use dual background modeling in the implementation. Since dual background modeling compares two frames to decide if there is an abandoned object, we assume that in the first frame that we got from camera shouldn't contain any abandoned object. Porikli (2007) says that an abandoned object should not be the part of the original area of scene. Because of this reason we assumed that the first frame is the original scene and nothing is abandoned.

5.9 Test Requirements & Test Cases

Use Case: Manage User

Actor: Administrator (First assigned administrator)

Security Personnel

Description: First administrator of the system is assigned when the system is deployed. If company or security office want to add more administrator, then for this operation first administrator will be responsible. In addition to that, administrator/s are in charge of security personnel information. To illustrate, they can add a new security personnel to the system, or they can delete information of security personnel who is fired or resign. Besides, admins can edit security personnel information in case of misspelling or changed data.

Precondition:

- The information of the first administrator **such as username, password etc. should be provided, while system is set up.**
- Database should be available for the application so that operations can be handled.

Primary Scenario:

Actor Intentions	System Responsibilities
1. For the first login procedure into the system, there will be a default username and password for first assigned administrator.	**2.** System will check these default username and password from the database.
	3. Home page will be displayed in desktop application, in which tabs and buttons are clearly shown to direct the user.
4. Administrator selects 'Security Personnel' in which s/he can do operations about security personnel.	**5.** There will be a sub-menu in the incoming page which consists of "Add a new Personnel", "Delete Personnel", and "Edit Personnel".
6. Admin selects "Add a new Personnel" from the sub-menu.	**7.** In this page, there will be a form to fill up. This form will include the parts which name, surname, social security number, date of birth, telephone number, and email.
8. Admin provides the required information to add the personnel to the system.	9. System will check if all required information is provided.
	10. If they are provided, then it will add the information to the database.

	11. If not, a message (near of each box) will be pop up to warn the admin. Admin will follow the steps given in 8.
12. Admin selects “Delete Personnel” from the sub-menu.	13. Navigation menu which is designed for “Delete Personnel”, is displayed.
	14. In order to delete the personnel, every information should be provided.
15. After filling the form, admin clicks “Delete Personnel” button.	16. Personnel will be deleted from the database.
17. Admin selects “Edit Personnel” from the sub-menu.	18. Navigation menu which is designed for “Delete Personnel”, is displayed.
19. Admin enters the SSN of the personnel whose information will be edited.	20. After checking SSN from the database, all information will be listed in the below of the page.
21. Admin enters the information to be changed, and then click “Edit Personnel”.	22. If all data is provided, then data is updated in the database.
23. Admin clicks the “Back to the Home Page” from the screen to go back to the home page.	
24. Admin clicks “Admin Operations” from the menu.	
25. Admin clicks “Delete Admin”	26. In order to delete an administrator from the database, name, surname, and SSN and department information should be provided by the admin. This data will displayed as a form in the page.
27. Admin fills the form, and clicks “Delete Admin”.	28. All information are checked. If there is not any missing data, delete the admin from the database.
29.. Admin clicks “Add Admin”	30. In this page, there will be a form to fill up. This form will include the parts which name, surname, social security number, date of birth, telephone number, email, department, username and password.
31. Admin enters the required information.	32. If there is not any missing data, then add the admin to the database.

Variants:

9a. If admin do not enter a data for adding another administrator, then s/he will be informed to enter these data.

16a. If the entered information is not found in the database, it will warn the admin that some information are wrong.

> 16a1. Admin will enter the information in the same page and again click "Delete Personnel".

20a. If SSN is not found in the database, it will send a message to the screen that indicates admin entered a wrong SSN.

22a. If there is not any change in the data of the personnel, then a message will be displayed as "There is not any change in the data!"

25a. If entered name, surname, SSN and department of the admin is not found in the database, it will display a message as "There is not any administrator in the system with these information!"

32a. If admin do not enter a data for adding another administrator, then s/he will be informed to enter these data.

Test Requirements:

1. Validate that administrator can access to the system.
   - 1.1 Login to the system with the username and password.
     - 1.1.1 Check the validity of username from database.
       - 1.1.1.1 Validate date if the username is wrong, inform the user about it.
       - 1.1.1.2 Validate date after warning notification, user will be directed to login page to enter the information again.
     - 1.1.2 Check the validity of password from database.
       - 1.1.2.1 Validate date if the password is wrong, inform the user about it.
       - 1.1.2.2 Validate date after warning notification, user will be directed to login page to enter the information again.
   - 1.2 Verify that administrator cannot access to the system without a valid username and password.

2. Validate that user can see the navigation menu of "Admin Operations".
   2.1 Validate that admin will see the "Admin Operations" in the main menu after successful login operation.
   2.2 Check that navigation menu for "Admin Operations" is shown properly.
      2.2.1 Check that menu consists of three options as buttons in the top of the page.
         2.2.1.1 Check that "Add Admin" options is displayed in the button options.
            2.2.1.1.1 Check that "Add Admin" option works properly.
            2.2.1.1.2 Validate that user will be directed to the "Add Admin" page.
         2.2.1.2 Check that "Delete Admin" options is displayed in the button options.
            2.2.1.2.1 Check that "Delete Admin" option works properly.
            2.2.1.2.2 Validate that user will be directed to the "Delete Admin" page.
   2.3 Check that navigation menu for "Security Personnel Operations" is shown properly.
      2.3.1 Check that menu consists of three options as buttons in the top of the page.
         2.3.1.1 Check that "Add a new Security Personnel" options is displayed in the button options.
            2.3.1.1.1 Check that "Add a new Security Personnel" option works properly.
            2.3.1.1.2 Validate that user will be directed to the "Add a new Security Personnel" page.
         2.3.1.2 Check that "Delete Personnel" options is displayed in the button options.
            2.3.1.2.1 Check that "Delete Personnel" option works properly.
            2.3.1.2.2 Validate that user will be directed to the "Delete Personnel" page
         2.3.1.3 Check that "Edit Personnel" options is displayed in the button options.
            2.3.1.3.1 Check that "Edit Personnel" option works properly.
            2.3.1.3.2 Validate that user will be directed to the "Edit Personnel" page.

Use case: Manage camera

Actor: Administrator

Security personnel

Description: The system will allow users to manage all cameras in itself. Users will be able to add or remove a camera from the system according to their requirements. Also, they will be able to activate or deactivate cameras. These will make users manage cameras in the system for all kind of security reasons as well as their reasons.

Precondition:

Add camera and delete camera will be only done by the administrator. Username and password are needed.

The database should be available so that we can check if the user is administrator or not.

User already logged into the system.

Primary Scenario:

Actor Intensions	System Responsibilities
	1. The home page is shown to the user.
2. The user wants to manage camera so s/he selected camera section in the GUI.	3. Camera page is shown to the users, which will allow manage all cameras. GUI will have add and delete camera sections.
4. The user wants to add a camera to the system. So, s/he enter camera IP address and a camera name.	5. The system requires username and password since only admin is allowed to add and delete camera.
6. Admin enters username and password so that system can check.	7. Database checked and admin allowed to add a camera.
	8. Database checked for add camera if there is no camera with same IP address and name camera added.
	9. The system will give a message saying the camera is added to the system and connection is done.
10. After camera added user wants to check camera so it will be activated.	11. This will activate the camera and show the camera footage on the main screen.
12. After camera footage is added to the main screen now user wants to deactivate the camera. User unchecked the camera.	13. After camera, unchecked System will stop to get data from that camera but it will continue to keep camera information in the database.
	14. It will continue to show camera page with the unchecked camera.
15. After deactivating the camera user now wants to delete camera.	
16. User entered IP address and name of the camera that should be deleted and pressed delete.	17. The system will ask for username and password so that it can delete camera.

Actor Intensions	System Responsibilities
18. User entered the username and the password and submitted to the system.	19. System checked the database and confirm that this is the administrator.
	20. Then, the system will check camera database to delete existed camera.
	21. After camera deleted system gives the message and returns to the camera page
22. The user now added, activated, deactivated and deleted the camera.	
23. The user wants to return to the main page.	24. The main page is now shown to the user and system exited from mange camera page.

7a. Database checked and the user is not the admin, it will give an error saying that only administrator is allowed to do these operations and turn to camera page.
8a. In database there is a camera with same IP address or name, so add camera will give an error to user which states camera cannot be added since it is already in the system. Then it will return to camera page.
11a. The camera couldn't be activated, so the system will give an error to the user and return to the camera page.
19a. The administrator is not confirmed so give an error message to the user and returns to the camera page

Test Requirements

Test Requirements

1. Validate that main GUI page is always available.
   - 1.1. Validate that main page available.
   - 1.2. Validate that from main page camera page can be selected.
     - 1.2.1. Validate that camera page can be clicked
     - 1.2.2. Validate that when camera page clicked system directed to the camera page.
   - 1.3. Validate that user always can return to the main page.
     - 1.3.1. Validate that main page is available from everywhere in the GUI
2. Validate that camera page is always available for the user.
   - 2.1. Validate that any user can go to the camera page.
     - 2.1.1. Validate that admin can go to the camera page.
     - 2.1.2. Validate user can go to the camera page.
   - 2.2. Validate that in the camera page all of the components available.

2.2.1. Validate that user can enter information to add a camera.
- 2.2.1.1. Validate that page has IP section to add the camera.
- 2.2.1.2. Validate that page has name section to add the camera.

2.2.2. Validate that delete camera sections are available.
- 2.2.2.1. Validate that page has IP section to delete camera.
- 2.2.2.2. Validate that page has name section to delete camera.

2.2.3. Validate that user can activate or deactivate cameras.
- 2.2.3.1. Validate that camera section is available on the page.
- 2.2.3.2. Validate that user can deactivate each camera.
- 2.2.3.3. Validate that user can activate each camera.

3. Validate that system has the database connection.
   - 3.1. Validate that system can access the database.
     - 3.1.1. Validate that system can access camera table.
     - 3.1.2. Validate that system can get the camera's all information from the database.
4. Validate that administrator can be confirmed for add/delete camera operations.
   - 4.1. Validate that from database system can access user table.
     - 4.1.1. Validate that system can get users' name and password to give permission to add and delete in the system.
   - 4.2. Validate that administrator can enter the username into the system.
   - 4.3. Validate that user can enter password to the system.
   - 4.4. Validate that if it is not administrator system can return to the camera page.
5. Validate that administrator can add or delete camera.
   - 5.1. Validate that system can add a camera into the database.
   - 5.2. Validate that added camera can be activated and deactivated.
     - 5.2.1. Validate that when camera added it is added to the activate/deactivate section.
   - 5.3. Validate that system can delete a camera from database.
     - 5.3.1. Validate that camera can be deleted from system.
     - 5.3.2. Validate that camera is deleted from activate and deactivate section.
6. Validate that any user can activate or deactivate the camera.
   - 6.1. Validate that activate and deactivate section works properly.
   - 6.2. Validate that activation can activate a camera.
     - 6.2.1. Validate that camera can be activated by its IP number.
     - 6.2.2. Validate that camera can be deactivated later also.
   - 6.3. Validate that deactivation section works properly.
     - 6.3.1. Validate that camera can cut the connection.
     - 6.3.2. Validate that activation can be done later on

Use case: Manage View
Actor: Administrator
Security personnel
Description: The system will allow users to manage views of cameras in it. Users will be able to add or remove a camera from the system view on the main page according to their requirements. Also, they will be able to see events occurred as well as event type which can be either abandoned luggage found or owner found. Also, users will be able to select which cameras they want to see. Selected cameras will be shown on the screen and if a camera is not selected it will continue to record and analyze events but it will not be on the GUI screen. Since in this section system doesn't have much user interaction, it will be mainly about what system should show or do.

Precondition:
- Database should be available so that system can record all events.

Primary Scenario:

Actor Intensions	System Responsibilities
1. The user on the main page.	2. The system has all view options on the main page.
	3. The system shows all cameras which are to be shown.
	Below of the page, all event log information is also shown.
5. The user sees all the camera and event logs. Security personnel waits for an event occur.	6. When an event occurred system will create an event and adds to the event log section of the GUI. Also, it will show in the camera section.
	7. If an event occurs it will be either abandoned luggage detection or owner detection of the abandoned luggage which detected earlier.
	8. The system will create one of the events and connect to the database and adds that event to the database with its all information.
9. The user sees all events occurred, now s/he wants to specifically choose which cameras to see.	
10. User deselects the cameras from view so that system stops showing them.	11. System remove deselected cameras from the user view of main.
	12. Keeps recording from cameras and records events occurred.

Actor Intensions	System Responsibilities
	13. While recording the events it also adds them to the event log.
	14. If event time exceeded 1 day, it will be deleted from event log look in the GUI, but it will be kept in the database till it is deleted by the administrator.

8a. The system couldn't connect to the database so it will give error and for a brief time, it will keep in the program till the connection is made or program is closed.

Test Requirements

1. Validate that system have a connection with database.
   1.1. Validate that system can access the database.
      1.1.1. Validate that system can add an event.
         1.1.1.1. Validate the system can create an event we id number, description of the event, time of the event.
         1.1.1.2. Validate that if owner is found, system can ad this to the event also.
      1.1.2. Validate that system can delete event.
         1.1.2.1. Validate that system have a access
      1.1.3. Validate that system can differentiate between events.
         1.1.3.1. Validate that system can store event as abandoned luggage founded.
         1.1.3.2. Validate that system can store event as owner found.
         1.1.3.3. Validate that system
2. Validate that cameras can be selected/deselected.
   2.1. Validate that a camera selection can be done in the main page.
      2.1.1. Validate that in selection area only activated cameras is displayed.
      2.1.2. Validate that the selected camera is activated.
   2.2. Validate that a camera deselection can be done in the main page.
      2.2.1. Validate that camera selection doesn't cut communication between camera and the system itself.
3. Validate that event log is functioning properly.
   3.1. Validate that events can be shown.
   3.2. Validate that event time is at most 24 house ago. Older events is not shown.
   3.3. Validate that they can be clicked.
      3.3.1. Validate that when they are clicked detected luggage or owner can be displayed to the user.

6. Project Management

6.1. Project Estimation

6.1.1. Inputs

Inputs	Complexity	Multiplier	Functional Points
Add Admin	Medium	4	1
Delete Admin	Low	3	1
Add Personnel	Medium	4	1
Delete Personnel	Low	3	1
Edit Admin/Personnel	Low	3	1
Add Camera	Medium	4	1
Activate/Deactivate Camera	Low	3	1

Table 4: Estimation – Inputs Table

6.1.2 Outputs

Output	Complexity	Multiplier	Functional Points
Display	Low	4	1
Luggage Detection	High	7	1
Owner Detection	High	7	1
Event Analysis	High	7	1

Table 5: Estimation – Outputs Table

6.1.3 Inquiries

Inquiries	Complexity	Multiplier	Functional Points
Add event/Camera	Low	3	1
Search for Camera	Medium	4	1
Search Event	Medium	4	1

Table 6: Estimation – Inquiries Table

6.1.4 Logical Internal files

Output	Complexity	Multiplier	Functional Points
User/Camera Table	Medium	10	2
Event table	Medium	10	1

Table 7: Estimation – Logical Internal Files Table

→Function Point Estimation

	Function	Points	
Program Characteristics	**Low Complexity**	**Medium Complexity**	**High Complexity**
Number of Inputs	4x3	3x4	0x6
Number of Outputs	1x4	0x5	3x7
Inquiries	1x3	2x4	0x6
Logical Internal Files	0x7	3x10	0x15
External Interface Files	0x5	0x7	0x10

Table 8: Estimation – Function Point Estimation Table

Function Point Estimation	
Unadjusted total of function points	90
Influence Multiplier	1.07
Adjust total of function point	96.3

Calculation of Influence Multiplier:

	Rating		
1- data communications	2	9- complex processing	4
2- Distributed Functions	1	10- reusability	4
3- Performance	6	11- installation ease	4
4- Heavily used configurations	1	12- operational ease	5
5- Transaction Rate	3	13- Multiple sites	0
6- online data entry	3	14- Facilitate change	3
7- end user efficiency	5	TDI	42
8- online updates	0		

Calculations:

IM/ VAF = TDI * 0.01 + 0.65 = 42 * 0.01 +0.65 = 1.07 <= Influence Multiplier

ATFP = UTFP * IM = 90 * 1.07 = 96.3

Since we train our model in Python, Python for the Language Unit Size (LUS). Since there is no value for python in the table of Programming Languages and Levels that is proposed by Jones (1998) [21]. We looked for similar languages with Python. These languages are Perl (value=20) and Java (value=55) in respect to their syntax and support for other application. We used Java's minimum value (minus 1 standard deviation which is given in the table of Jones) for our base on python.org [20]. Which concludes us with 40 =>: most common value:

Line of Codes (LOC) = LUS * ATFP = 96.3 * 40 = 3852

COCOMO (Constructive Cost Model)

KDSI = Thousand delivered source instructions (KLOC)

KDSI = ATFP * Language unit size /1000 = 96.3 * 40/ 1000 = 3.852

Since the LOC is only applicable on Basic COCOMO, we took Basic COCOMO's development mode as Semi-Detached. Hence, the following values of a, b and c were used:

a= 3.0 b = 1.12 c = 0.35

PM (Person-Month) = a * (KDSI) ^b

PM = 3.0 * (1.926) ^1.12

PM = 13.59

TDEV (Total Development Time) = 2.5* (PM) ^0.35

TDEV = 2.5 * (13.59) ^ (0.35)

TDEV = 6.23 months

6.2 Project Tasks and Durations

TASK	DESCRIPTION	DURATION
T1	Installation of Ubuntu 16.04 Virtual Machine on Windows and Mac (team members' laptops)	0.5 hours
T2	Installation of OpenCV & Tensorflow libraries to team members' laptops	1.5 hours
T3	Setting up cuDNN and CUDA Development Toolkit on Lab Computer	2 hours
T4	Installation of OpenCV & Tensorflow libraries to Lab PC	1 hour
T5	Creating a new luggage dataset (15000 images) from ImageNET	3 hours
T6	Extracting the images with the background	2 hours
T7	Annotation of 2500 images in the dataset	5 hours
T8	Installing Darknet framework and adjusting annotation file (xml) to train in YOLO algorithm	0.5 hours
T9	First training with small part of dataset (300 images)	16 hours
T10	Second training with all annotated images (2500 images)	45 hours
T11	Writing a luggage detection program with created model in training by using Tensorflow	3 hours
T12	Writing a background subtraction program	0.5
T13	Writing human detection program by using Support Vector Machine & Histogram of Gradient Descents algorithms for further purposes	5 hours
T14	Wrote a Background Subtraction program with Gaussian Mixture Model	1 hour
T15	Create short-term background subtraction using Zivkovic parameters for Mixture of Gaussian	1 hour
T16	Crate long-term background subtraction using Zivkovic parameters for long-term Mixture of Gaussian	0.5 hour
T17	Extracting moving foreground from each backgrounds	5 hours
T18	Compare proposals from background extracted static object proposals	3 hours
T19	Testing background modeling with videos and webcam	1 hour

T20	Creating dataset with the images that have not one-colored background (in previous sprint we didn't separate the images with/without background.)	4 hours
T21	Annotation of new dataset	2.5 hours
T22	Preparing xml files to train in Faster R-CNN	1 hour
T23	Trained Faster R-CNN with 300 images (for the first training attempt) Tensorflow Object Detection API (ResNet model)	12 hours
T24	Trained Faster R-CNN with 2500 images (for the second training attempt) Tensorflow Object Detection API (inception model)	15 hours
T25	Write a program to test Faster R-CNN	0.5 hours
T26	Write a program with threading to test Faster R-CNN	1 hour
T27	Literature review about comparison of Faster R-CNN and YOLO	4 hours
T28	Combining background modelling and detection part	3 hours
T29	Testing YOLO and Faster R-CNN with different images, videos and webcam (real-time)	3 hours
T30	Training SSD with Object Detection with Tensorflow API	16 hours
T31	Writing program for optimizing bounding boxes for Faster R-CNN	1 hour
T32	Creating person dataset and annotating of each images	5 hours
T33	Detection program with person and luggage dataset trained model	1 hour
T34	Background modeling implementation with Gamma filter	1 hour
T35	Installing Tensorflow (gpu) and OpenCV on another computer in RZ-11 lab.	5 hours
T36	Threading on main program for real-time video tests	8 hours
T37	Writing program for distance between luggage and person	14 hours

Table 9: Tasks & Durations Table

6.3. List of Milestones

With reference to Tasks and Durations table, mile stone tasks are:

- M1: Finishing setting up for both computers.
- M2: Dataset Creation
- M3: Training Dataset
- M4: Luggage Detection
- M5: Static Object Detection
- M6: Combining Recognition and Background Modeling
- M7: Person and luggage detection

6.4 Acitivity Diagram

Task 1: Set-up computers (MS)
Task 2: Dataset Creation (MS)
Task 3: First Training with Dataset (MS)
Task 4: Short-term Background modeling
Task 5: High precision luggage detection (MS)
Task 6: Long-term background
Task 7: Complete program for static object detection (MS)
Task 8: 2nd training
Task 9: Combination of background modeling and recognition (MS)
Task 10: Human-luggage detection (MS)
Task 11: Optimization of Program
Task 12: Owner labeling

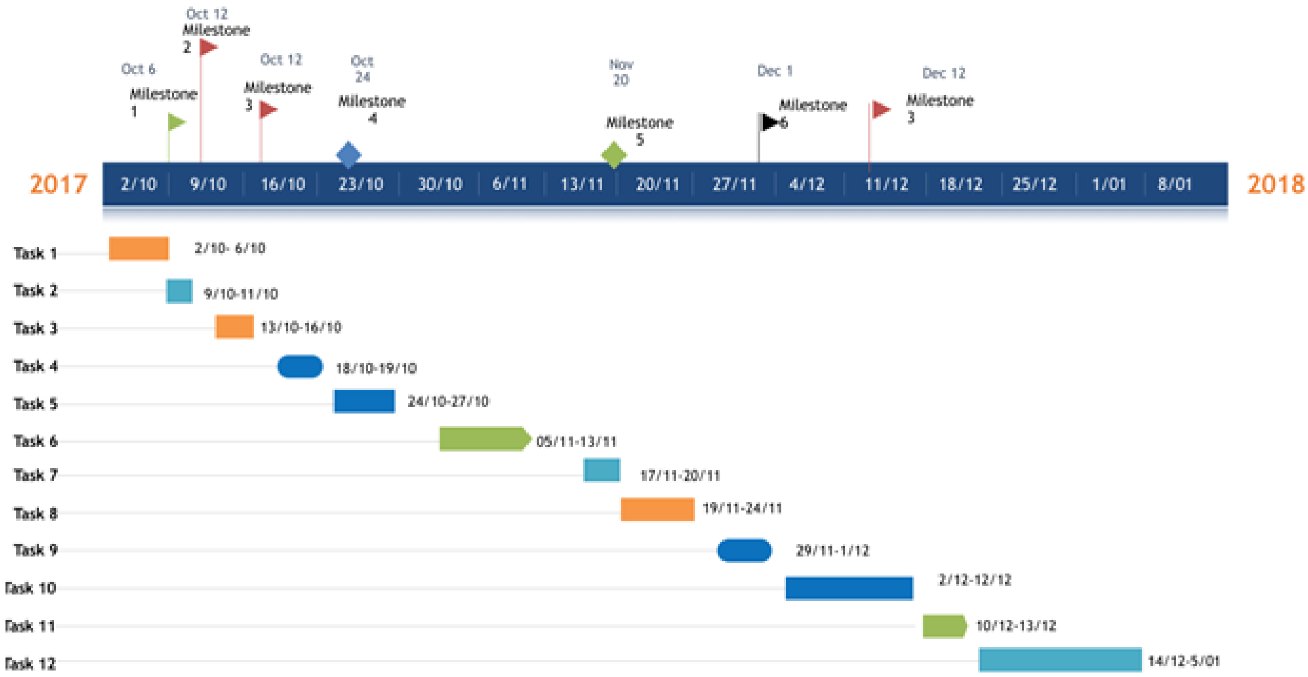


Figure 25: Acitivity Diagram

6.5. Gantt Chart for Milestones

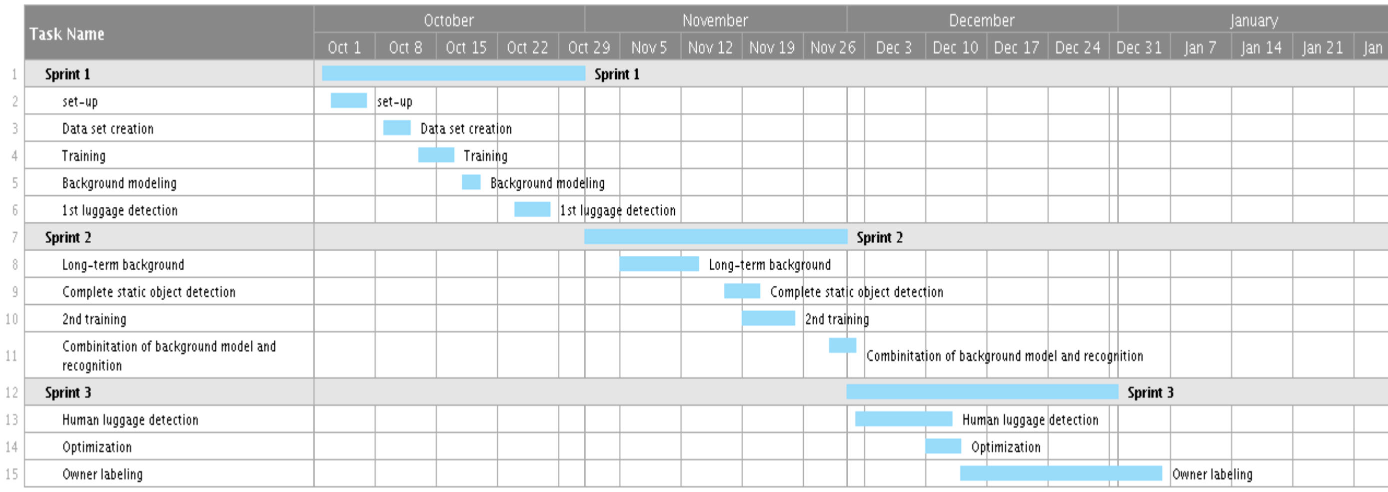


Figure 26: Gannt Chart for Milestones

6.6. Task Allocation

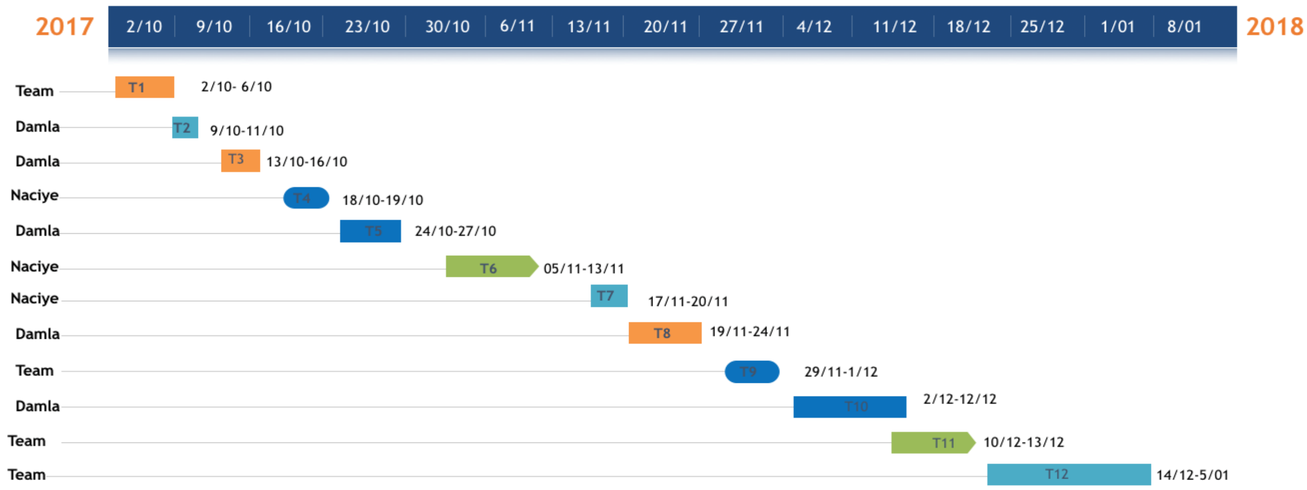


Figure 27: Task Allocation

6.7 Scrum Details

6.7.1 Sprint 1

6.7.1.1 Sprint Backlog

Sprint Backlog		
Sprint Term	Description	Effort in Hours
SPRINT 1 (01.10.17 – 01.11.17)	Installation of Ubuntu 16.04 Virtual Machine on Windows and Mac (team members' laptops)	0.5 hours
	Installation of OpenCV & Tensorflow libraries to team members' laptops	1.5 hours
	Setting up cuDNN and CUDA Development Toolkit on Lab Computer	2 hours
	Installation of OpenCV & Tensorflow libraries to Lab PC	1 hour
	Creating a new luggage dataset (15000 images) from ImageNET	3 hours
	Extracting the images with the background	2 hours
	Annotation of 2500 images in the dataset	5 hours
	Installing Darknet framework and adjusting annotation file (xml) to train in YOLO algorithm	0.5 hours
	First training with small part of dataset (300 images)	16 hours
	Second training with all annotated images (2500 images)	45 hours
	Writing a luggage detection program with created model in training by using Tensorflow	3 hours
	Writing a background subtraction program	0.5
	Writing human detection program by using Support Vector Machine & Histogram of Gradient Descents algorithms for further purposes	5 hours

Table 10: Sprint-1 Backlog Table

6.7.1.2 Sprint Burndown Chart

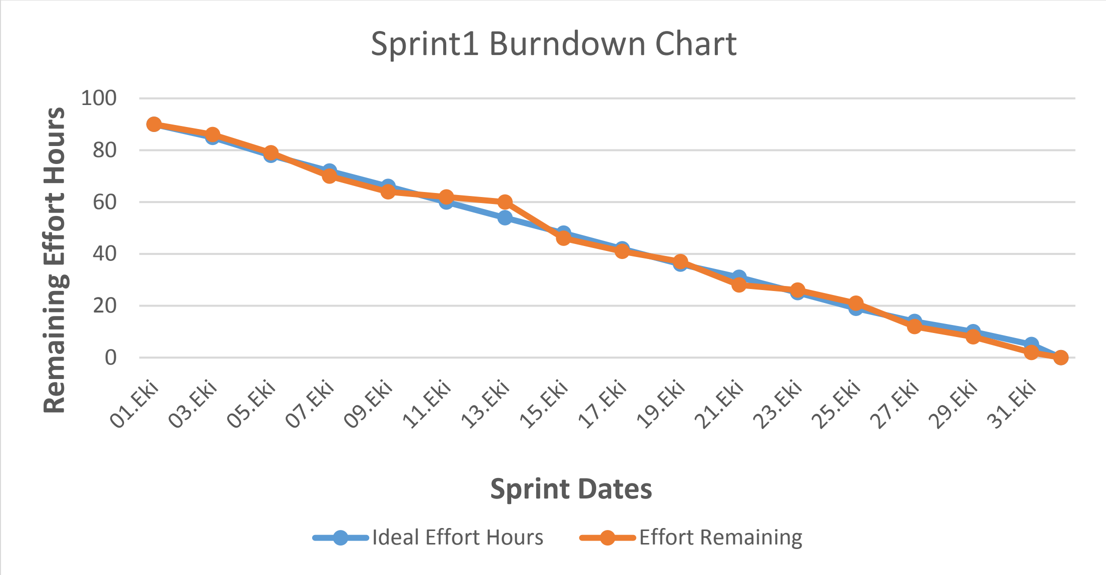


Figure 28: Sprint-1 Burndown Chart

6.7.1.3 Sprint Review

For first sprint, our plan was to define a platform to work on and complete all required setups. As a second goal, we wanted to create a large dataset for luggage since there is not any available dataset for this object. It took too much time of us but we managed to do this, additionally we created the annotation files for images with background. This was important for us to start training a model. To sum up, we managed more than we planned. Now, we have a trained model and a luggage detection program to improve.

According to result gained from first training, the accuracy on testing is 81%. However, in this training a small number of dataset (300 images) was used. In the second training (2500 images in dataset), we reached 85% accuracy. From now on, our algorithm (YOLO) seems suitable and can be optimizable. So, in the perspective of detection algorithm, we are on the right way. For background subtraction, there are more things to learn and search to find a best algorithm. These will be our next Sprint goals.

6.7.1.4 Sprint Retrospective

As mentioned above, this project is run by two people (Naciye and Damla). Naciye is responsible for computer vision algorithm search and implementation and Damla (Scrum Master) is responsible for machine learning part of the project. For the first sprint, our aim was to write a program that detect luggage from video, image and camera. We have done this part. Naciye is still searching for most appropriate algorithm for background subtraction and human detection, for now, there is not any implementation of them to get a base for our project.

For second sprint, we are expecting that Naciye will implement a background subtraction program and work on human detection. Damla wrote the program for detection and trained the models (by creating dataset & annotations). But there are other projects/lectures that we need to deal with, so our speed while working on project is getting a bit slower.

6.7.2 Sprint 2

6.7.2.1 Sprint Backlog

Sprint Backlog		
Sprint Term	Description	Effort in Hours
SPRINT 2 (01.11.17 – 01.12.17)	Wrote a Background Subtraction program with Gaussian Mixture Model	1 hour
	Create short-term background subtraction using Zivkovic parameters for Mixture of Gaussian	1 hour
	Crate long-term background subtraction using Zivkovic parameters for long-term Mixture of Gaussian	0.5 hour
	Extracting moving foreground from each backgrounds	5 hours
	Compare proposals from background extracted static object proposals	3 hours
	Testing background modeling with videos and webcam	1 hour
	Creating dataset with the images that have not one-colored background (in previous sprint we didn't separate the images with/without background.)	4 hours
	Annotation of new dataset	2.5 hours
	Preparing xml files to train in Faster R-CNN	1 hour
	Trained Faster R-CNN with 300 images (for the first training attempt) Tensorflow Object Detection API (ResNet model)	12 hours
	Trained Faster R-CNN with 2500 images (for the second training attempt) Tensorflow Object Detection API (inception model)	15 hours
	Write a program to test Faster R-CNN	0.5 hours
	Write a program with threading to test Faster R-CNN	1 hour
	Literature review about comparison of Faster R-CNN and YOLO	4 hours

Table 11: Sprint-2 Backlog Table

6.7.2.2 Sprint Burndown Chart

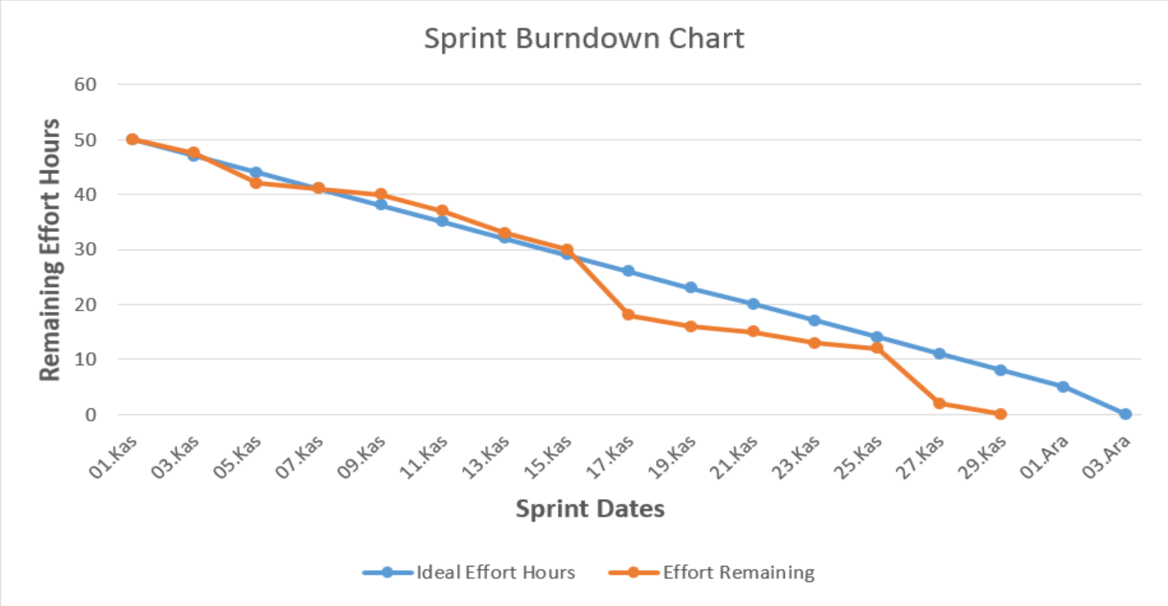


Figure 29: Sprint-2 Burndown Chart

6.7.2.3 Sprint Review

Within Sprint #2, we focused on existing projects and their implementations. While coming to end of this sprint, we are able to detect luggage in the coming frame and do background subtraction. Goal of this sprint was to analyze and criticize the existing projects in order to implement a better application compared to others. This sprint makes the program flow of the project clear to follow.

6.7.2.4 Sprint Retrospective

Due to workload in other courses, we had limited time for the implementation part. However, in the previous sprint, we were able to do more than we planned, this will not affect our studies. Naciye finished literature review for background modeling, but she will continue checking new papers and implementations. For the next sprint, extracted static objects will be sent to detection part. Damla continues to train the dataset with different number of images and algorithms. Our goal for the next sprint is to make a final decision about detection algorithm.

6.7.3 Sprint 3

6.7.3.1 Sprint Backlog

Sprint Backlog		
Sprint Term	Description	Effort in Hours
SPRINT 3 (01.12.17 – 01.01.17)	Combining background modelling and detection part	3 hours
	Testing YOLO and Faster R-CNN with different images, videos and webcam (real-time)	3 hours
	Training SSD with Object Detection with Tensorflow API	16 hours
	Writing program for optimizing bounding boxes for Faster R-CNN	1 hour
	Creating person dataset and annotating of each images	5 hours
	Detection program with person and luggage dataset trained model	1 hour
	Background modeling implementation with Gamma filter	1 hour
	Installing Tensorflow (gpu) and OpenCV on another computer in RZ-11 lab.	5 hours
	Threading on main program for real-time video tests	8 hours
	Writing program for distance between luggage and person	14 hours

Table 12: Sprint-3 Backlog Table

6.7.3.2 Sprint Burndown Chart

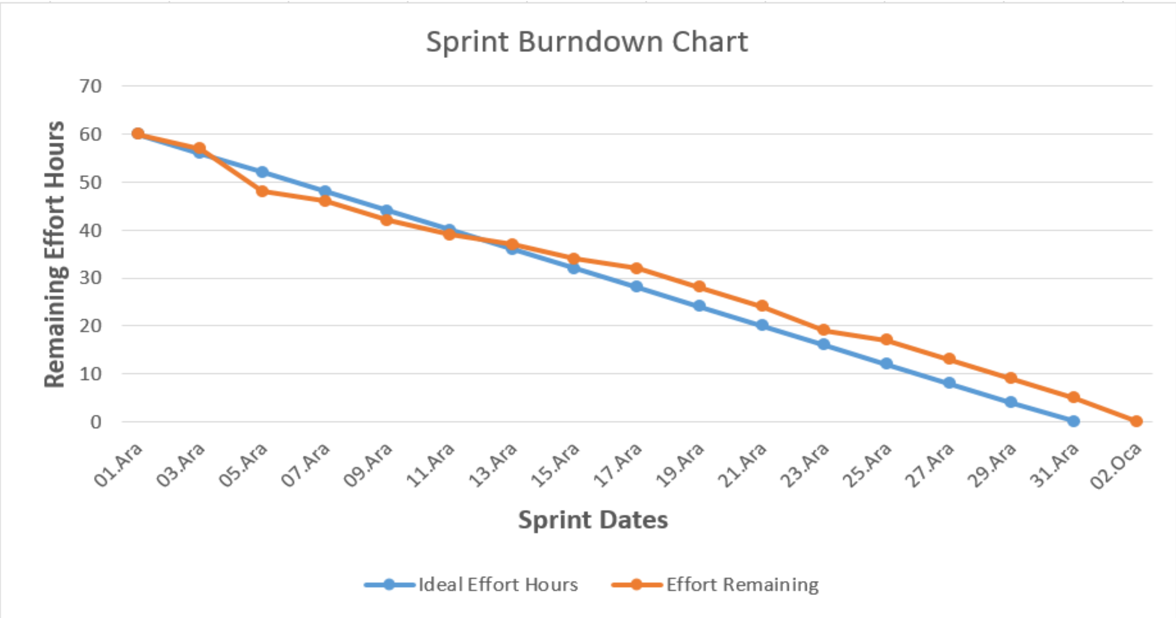


Figure 30: Sprint-3 Burndown Chart

6.7.3.3 Sprint Review

In this sprint, we made a final decision on detection algorithm. We have three algorithms that were trained on same dataset. These are SSD, YOLO and Faster R-CNN. For this application, accuracy is the most important thing and because of this fact, we chose Faster R-CNN. After that, combination of detection and background modelling part was finalized. However, we needed to improve the speed of the program for real-time usage since speed of Faster R-CNN is not as strong as YOLO. Hence, our concern was on threading. In python multiprocessing is a hard process to handle so it took very much time that we expected. Our other goal was to add "finding distance between luggage and person" feature to the distance but we could not finish this with our webcam camera since it requires a security camera with calibration feature.

6.7.3.4 Sprint Retrospective

This sprint is more about improving program speed and efficiency in terms of accuracy. We managed to complete threading in our program, and although our camera is not suitable for calibration and because of that common distance algorithm cannot be applied, we designed an approach but we did not have time to complete its implementation. For the next sprint, our aim is much event recognition oriented. Finding distance between luggage and person, alarm situations, warning states will be handled in next sprints.

Semester Burndown Chart

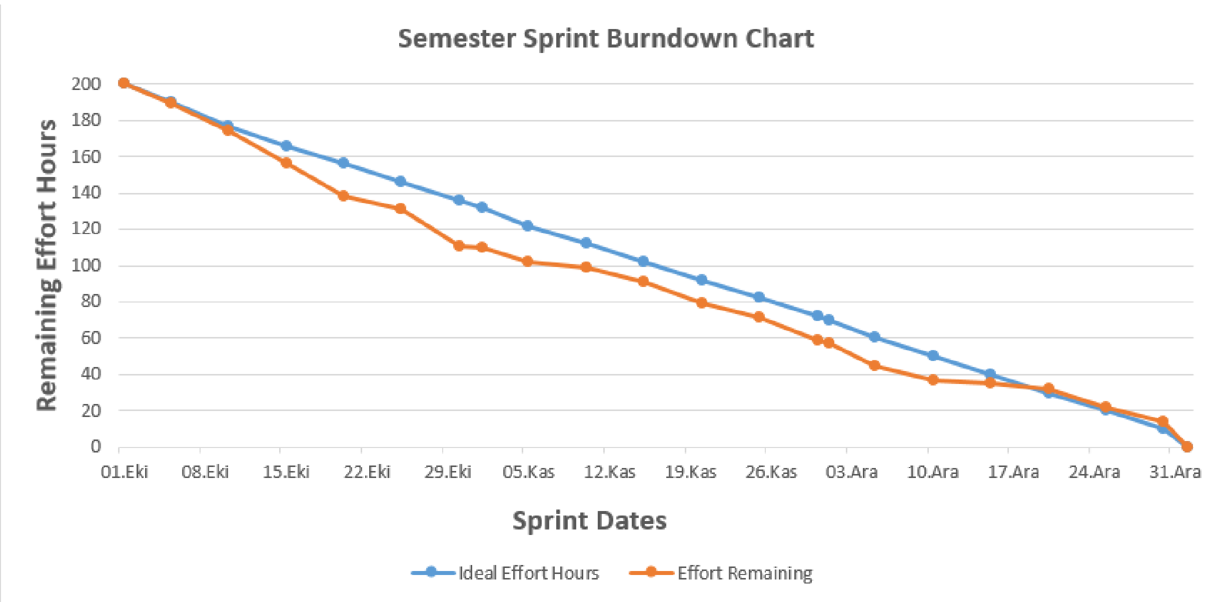


Figure 31: Semester Burndown Chart

6.8 Risk Management

1) Hardware unavailability:

Identification: Hardware which is essential in our case is 1 IP camera might not be delivered on time. Also we can count that if even it is delivered, equipment has a chance to be broken.

Analysis: It is highly possible since equipment order is made by school and delivery is made by cargo.

Risk planning: We have a webcam of our own which is not an IP camera but it can be used in case equipment delivery is not made.

Risk monitoring: We still have this possibility since we didn't get the camera yet.

2) Time line estimation (Size underestimate or overestimate):

Identification: There is a possibility of stay behind of the schedule on the project.

Analysis: Risk of time line estimations to be wrong is medium chance. We sometimes have problems that we cannot see before. Also, there is chance to be faster than we think.

Risk planning: We always update the schedule, and discuss our progress every week in scrum meetings.

Risk monitoring: If faster, do estimation again and give time to tasks might take longer than you think. If slower, then give take time from tasks that might take shorter time than you think.

3) Requirements change:

Identification: There might me large amount of changes to the requirements than anticipated at the beginning.

Analysis: Chance for this is very low, since we did all our research and requirement analysis at the beginning.

Risk planning: Avoid large number of changes at the project.

Risk monitoring: We always discuss progress and even minor changes in the project development. No big changes occurred yet and chance to be occurred is still very l

7. System Design

7.1. Architecture Models

Level 0:

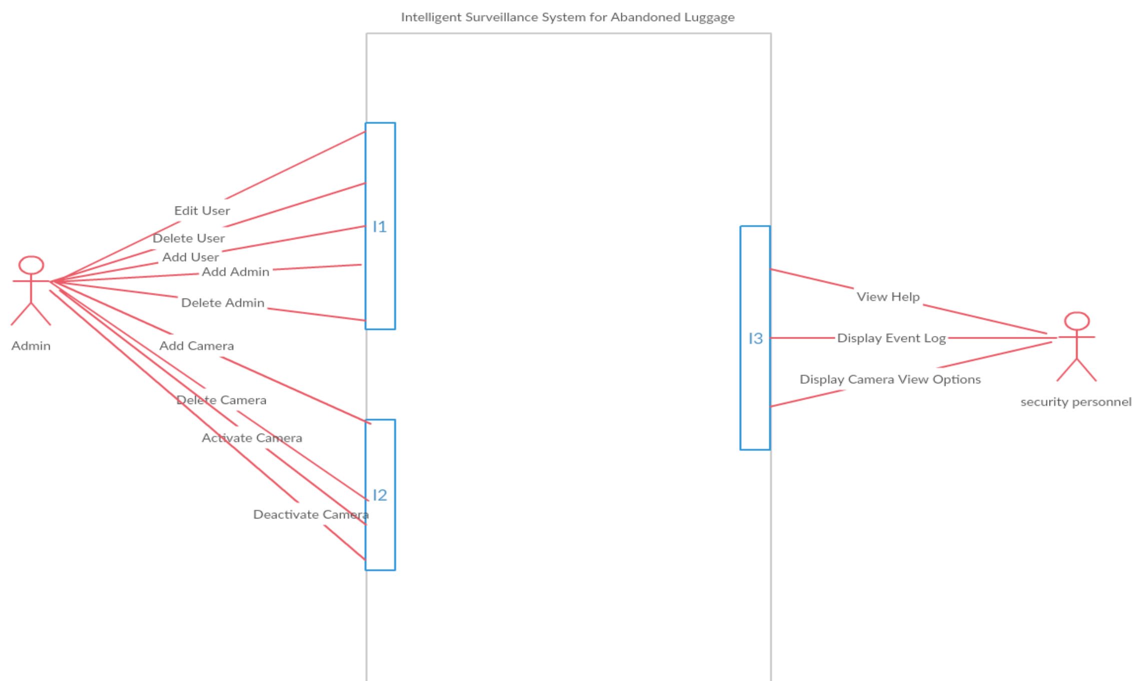


Figure 32: Architecture Level-0 Diagram

Level 1

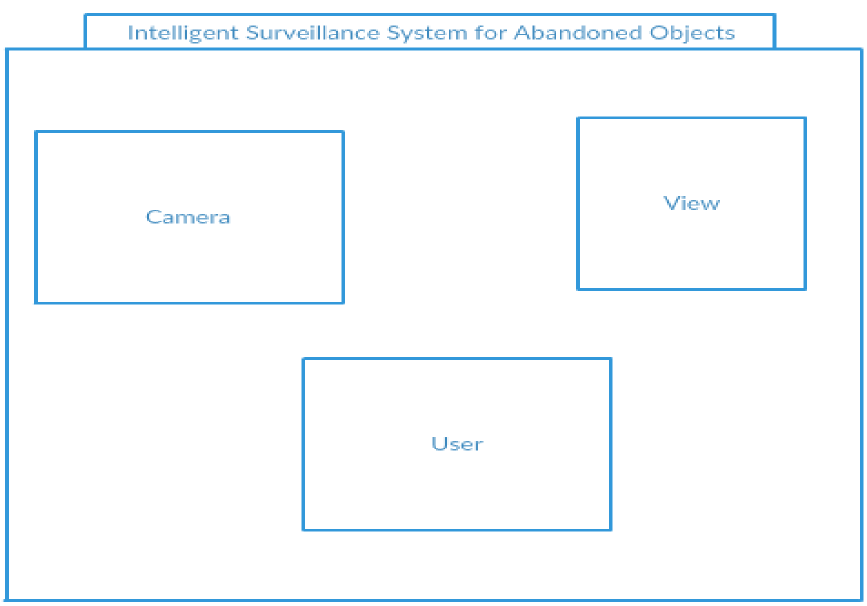


Figure 33: Architecture Level-1 Diagram

7.2. Database Design

7.2.1 Database Diagram

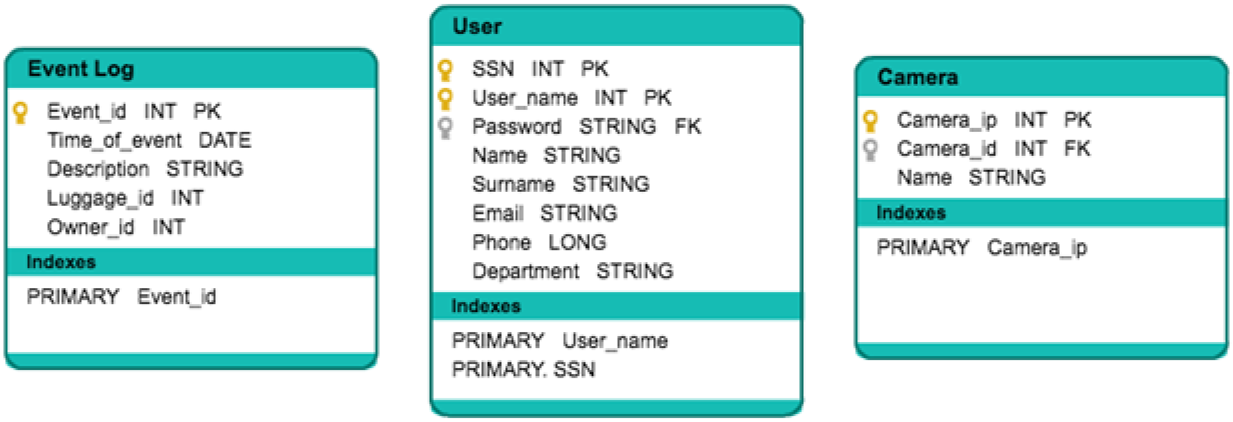


Figure 34: Database Diagram

In the database of the system, we have 3 tables. Event log represents if any luggage detection occurred in the system or not. Camera represents all cameras that our system has. Lastly, since we differentiated between administrator and security personnel we keep user also in the database.

7.2.2 ERD Diagram

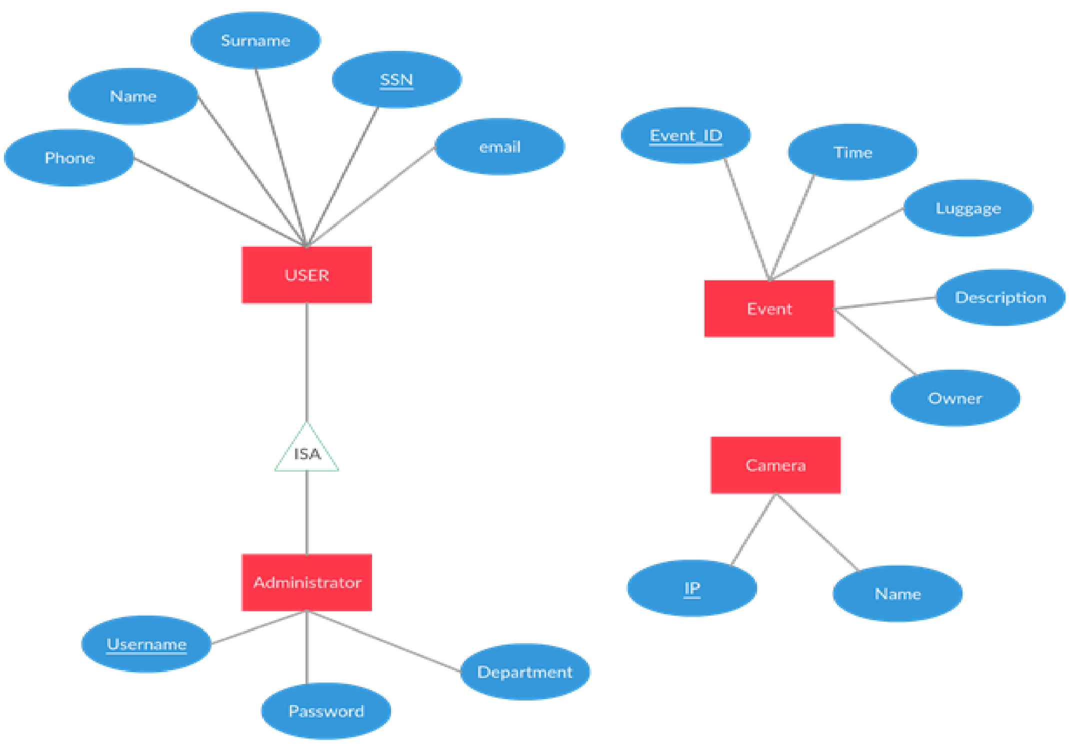


Figure 35: ERD Model

In the system, no database table has a relationship with each other. As stated previous section each table represents very different things in the system and no module have to know each other. The user is to keep username of the personnel, their passwords, and email addresses. Also, since event log keeps detection of an abandoned luggage, it has event id to differentiate between these events. In addition, it has time which represents exact time of the event. Furthermore, it keeps detected luggage and owner id which is for owner labeling. For users, it keeps a brief description of the event. This description is displayed to them also. In addition, the system has camera information to add and delete cameras later. It keeps camera IP address to connect to the camera, also to keep things simple for the user, it keeps camera id and a camera name.

7.3. Dataflow Diagram

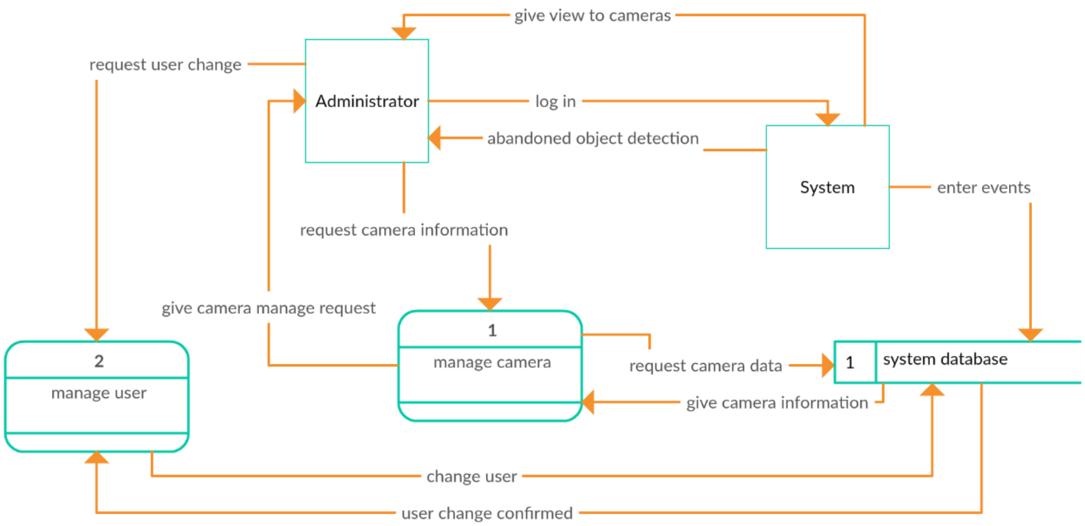


Figure 36: Dataflow Model Level-0

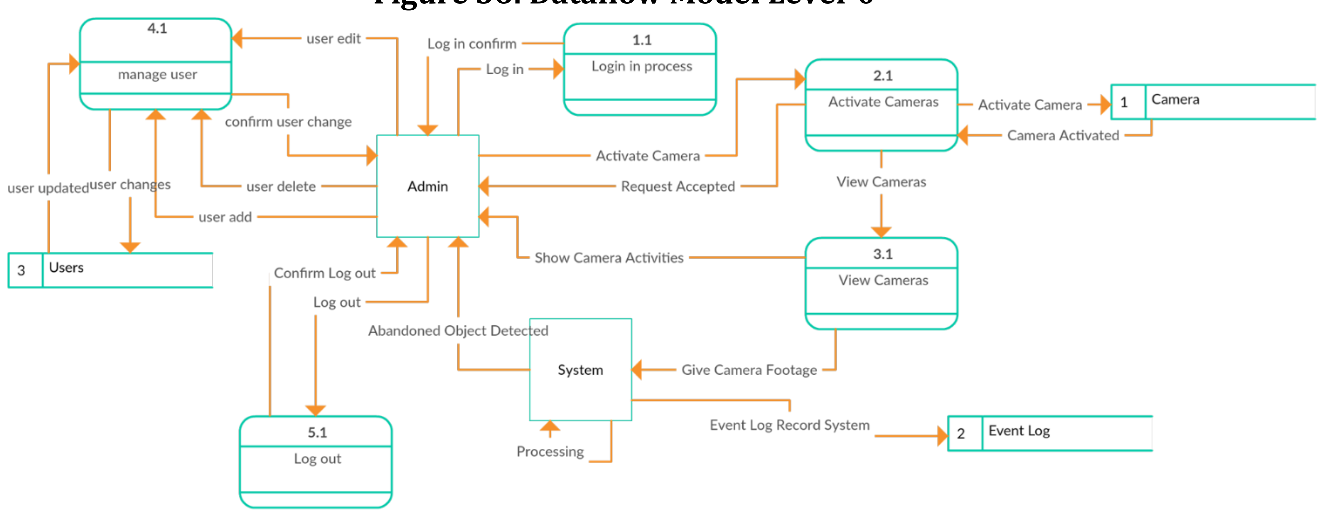


Figure 37: Dataflow Model Level-1

7.4. Sequence Diagrams

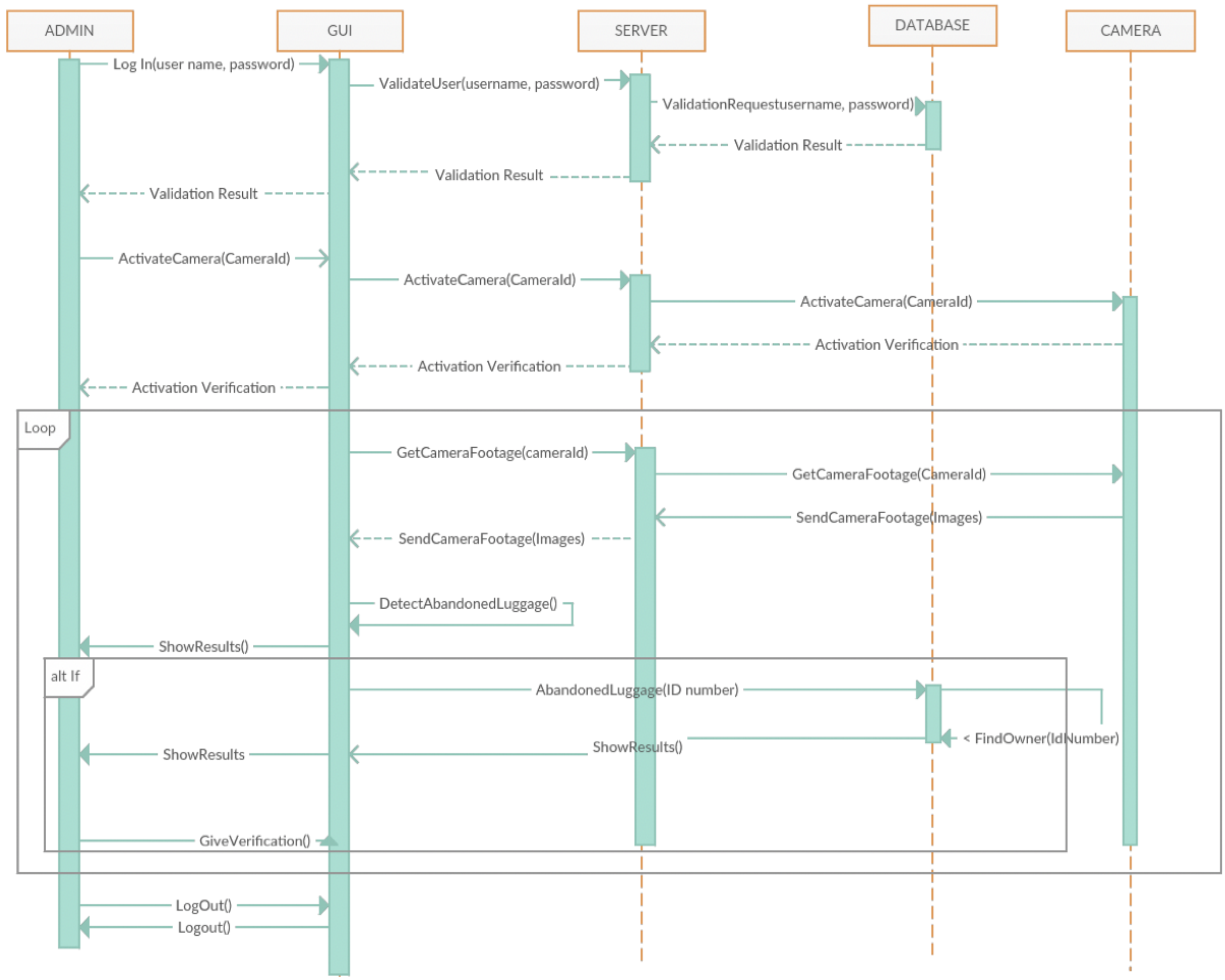


Figure 38: Sequence Diagram

In sequence diagram, we have 5 layers in the system that are communicate each other. Administrator uses GUI for all operations. Since there will be communication between camera and the system thorough IP address of the camera, and also we have database to communicate with each other we need internet. To represent that we have server among GUI, database, cameras.

7.5. Class Diagram and Definitions

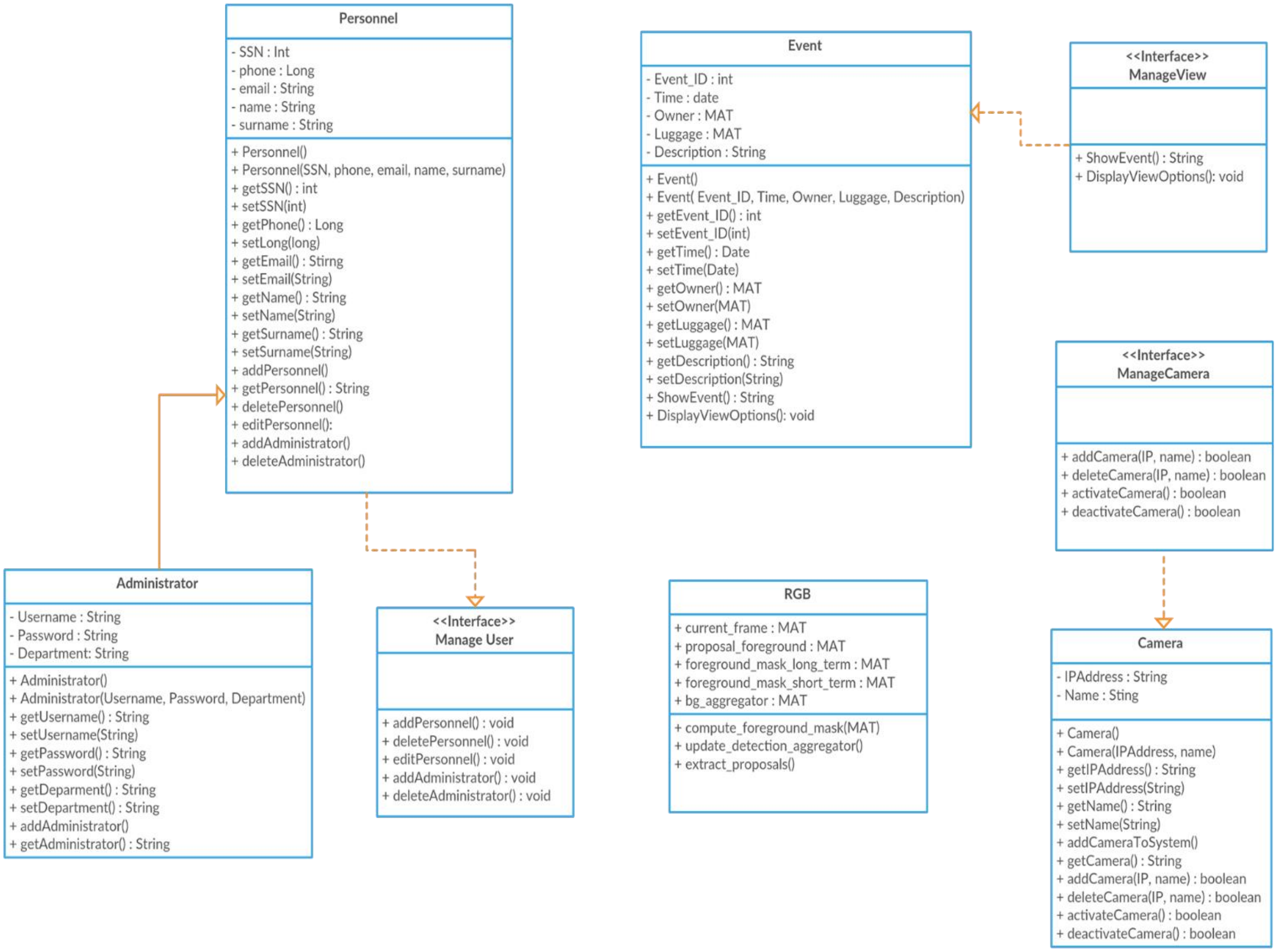


Figure 39: Class Diagram

Above diagram shows the all classes that we have in the system.

- Personnel class: This class is to represent users in the system. It will implement manage user interface and each function is for a user case in the requirements section. This will make possible to manage user part of the system
- Administrator class: An administrator is also personnel. Since only admins can control the system they have username and password. That is why we preferred to be different class that personnel.
- Camera class: This class will represent all cameras. It has IP and name of camera to create camera object in the system. Also, the class will implement manage camera interface for user which allows users to add/delete/activate/deactivate camera.
- Event class: This class will represent all events that is occurred in the system. It keeps event id, detected luggage, owner of the event, description. It also implements manage view part of the system.
- RGB class: RGB class is kept for main program. It keeps current frame and detections for that frame. This is used in the main part where we detect and implement create events.

7.6 Data Structures and Data Types

In our system, there are 2 main data structure that we used and continue to use. One of them is list, and other is Queue.

- **Array list:** Array list is used in main program where we do all detection and it will be used in the software part also. In main program we used it to store event for short term and also planning to use it to keep users, camera, so that we wouldn't have to connect database all the time.
- **Queue:** Queue data structure is also used in the system. It is mainly used in the detection part especially as process queue. So, it enables our main program to make parallel processing and share memories between these processes. It is very important part of our main implementation.

7.7 Interface Definition

```
I1: Manage User {
        Public void addPersonnel()
        Public void deletePersonnel()
        Public void editPersonnel()
        Public void addAdministrator()
        Public void deleteAdministrator()
}
I2: Manage Camera {
        Public Boolean addCamera()
        Public Boolean deleteCamera()
        Public Boolean void activateCamera()
        Public Boolean void deactivateCamera()
}

I3: Manage View {
        Public String showEvent()
        Public void displayViewOptions()
}
```

8. Conclusion

8.1 Critical evaluation of the project

As indicated in the Introduction & Background part, since this project has an important place for community safety, the main objective is to design a robust and reliable system. For this purpose, in every step of the project, we are trying to apply new technologies and methods which gives better results compared to older ones. For example, we use a modern object detection technique which uses machine learning algorithm: Faster R-CNN. Instead of using this algorithm alone, we are doing a pre-process on the frame that reduces the speed by reducing the region of interest for detection part.

Unlike most other research and projects, we continue our projects with event recognition and owner labeling which is one of the challenging part of the system because situations and conditions in the environments can vary and concerning events gets harder.

In these perspectives, our system is innovative, reliable, and robust. We can verify this by looking at the results that we get until now.

8.2 Retrospective of the first semester

At the beginning of the project, we did not have a dataset for luggage images, and also there is not any published one. So, we managed to create one to use in the training part. Almost 6000 images were collected from the Google Images and only 2500 were used for training part since they are the ones with background.

We trained our model in three algorithm. These are Single Shot Detection (SSD), You Only Look Once (YOLO), and Faster R-CNN. We compared them in terms of accuracy and detection speed with images, videos and real-time on webcam. At the end we chose the Faster R-CNN algorithm which gives the best accuracy result. These comparison process took lots of time while testing.

A background modelling program was written with the help of the paper of Porikli (2007) in order to detect static objects in the frame and parameters were adjusted according to our testing environments.

Coming to the end of this semester, we combined our background modelling part with detection part. As we expected, we needed to write a parallel program which runs the detection and background modelling part simultaneously. We spent too much with these since it is not only related with programming but also hardware. We faced too many problem with GPU, but at the end we managed to make our program work with multi-processing.

For the end of this semester, another aim was to complete the implementation of finding distance luggage and person. However, due to the time spent in multiprocessing, we could not able to finish the implementation but we designed an approach for it. In normal case, if we have a qualified security camera, the solution was almost ready to implement. But, because of the delay for getting camera, we needed to find an algorithm which can be applied with our webcam.

To sum up, for these four month duration, we are able to complete our tasks and aims that we planned at the beginning of the semester. The remaining work and our plan is mentioned in the following part.

8.3 Future Work

We are going to continue our studies on finding distance problem. After that we will work on event recognition part which is the final result of whole system. Here, we find out that luggage is abandoned, and it is in the warning or alarm state. For the final stage, we are going to present this project as desktop application. So, we will start to implement the graphical user interface of the application and its operations.

REFERENCES

[1] Dahivelkar, S. and Borse, S. (2017). Abandoned Object Detection Based on Statistics for Labeled Regions. International Journal of Engineering Research in Electronics and Communication, 4(8), pp.2-4.

[2] Hui Kong, Audibert, J. and Ponce, J. (2010). Detecting Abandoned Objects With a Moving Camera. IEEE Transactions on Image Processing, 19(8), pp.2201-2210.

[3] Sidyakin, S. and Vishnyakov, B. (2017). Real-time detection of abandoned bags using CNN. Automated Visual Inspection and Machine Vision II, [online] 10334, pp.1-13. Available at: http://SPIEDigitalLibrary.org/conference-proceedings-of-spie [Accessed 7 Oct. 2017].

[4] Liao, W., Yang, C., Ying Yang, M. and Rosenhahn, B. (2017). SECURITY EVENT RECOGNITION FOR VISUAL SURVEILLANCE. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-1/W1, pp.19-26.

[5] Porikli, F., Ivanov, Y., & Haga, T. (2007). Robust abandoned object detection using dual foregrounds. EURASIP Journal on Advances in Signal Processing, 2008(1), 197875.

[6] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. arXiv preprint arXiv:1612.08242. [online] Available at: https://arxiv.org/pdf/1506.02640.pdf [Accessed 9 Sept. 2017].

[7] PETS 2006. (2017). PETS 2006 Benchmark Data. [online] Available at: http://www.cvg.reading.ac.uk/PETS2006/index.html#workshop-organisation [Accessed 2 Nov. 2017].

[8] Xiao, T., Li, H., Ouyang, W. and Wang, X., 2016. Learning deep feature representations with domain guided dropout for person re-identification. arXiv preprint arXiv:1604.07528.

[9] Tian, Y. L., Feris, R., & Hampapur, A. (2008, October). Real-time detection of abandoned and removed objects in complex environments. In The Eighth International Workshop on Visual Surveillance-VS, 2008.

[10] M. Valera, S.A Velastin, “Intelligent distributed surveillance systems: a review”, IEE Proceedings - Vision, Image and Signal Processing, 152(2), pp. 192-204, 2005.

[11] M.S. Nagmode, A. J. Madhuri, M. S. Ashok, "A Novel approach to Detect and Track Moving Object using Partitioning and Normalized Cross Correlation", ICGST-GVIP Journal, 9(4), Aug. 2009.
[12] S. Ferrando, G. Gera, C. Regazzoni, "Classification of Unattended and Stolen Objects in Video Surveillance System", Proc. of IEEE Intl. Conf. on Advanced Video and Signal-Based Surveillance (AVSS'06), pp.21, Nov. 2006.
[13] T. T. Zin, P. Tin and T. Tori, "A Probability-based Model for Detecting Abandoned Objects in Video Surveillance Systems," in Proceedings of the World Congress on Engineering, London, U.K, 2012.
[14] Auvinet, E., Grossmann, E., Rougier, C., Dahmane, M., Meunier, J. "Left-luggage detection using hymnographies and simple heuristics." In: Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), New York, pp. 51–58 (2006).

[15] Chitra, M., Geetha, M. and Menaka, L. (2013). Occlusion and Abandoned Object Detection for Surveillance Applications. International Journal of Computer Applications Technology and Research, 2(6), pp.708-713.

[16] Bhandari, P., Ratanpara, T. and K.H, D. (2017). Abandoned Object Detection Using Dual Background Model from Surveillance Videos. International Journal of Engineering and Technology, 9(3S), pp.215-223.

[17] Aman, M. (2007). Preventing terrorist suicide attacks. Sudbury, MA: Jones and Bartlett, p.21.

[18] Ren, S., He, K., Girshick, R. and Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, [online] 39(6), pp.1137-1149. Available at: https://arxiv.org/abs/1506.01497 [Accessed 3 Oct. 2017].

[19] Gao, H. (2017). Faster R-CNN Explained – Hao Gao – Medium. [online] Medium. Available at: https://medium.com/@smallfishbigsea/faster-r-cnn-explained-864d4fb7e3f8 [Accessed 15 Nov. 2017].

[20] Python.org. (2018). Comparing Python to Other Languages. [online] Available at: https://www.python.org/doc/essays/comparisons/ [Accessed 5 Jan. 2018].

[21] Jones, C. (2012). Estimating software costs. New York: McGraw-Hill.

APPENDIX

CHANGE REQUEST FORM	
Change ID	002
Date	11.12.2017
Change Requester	Damla Gül Altunay
Change Analyzer	Okan Topçu Cem Direkoğlu Naciye Karademir
Requested Change	Detection algorithm has to be changed from You Only Look Once to Faster R-CNN since its accuracy results are better than former.
Changed Priority	High
Change Implementation:	It is going to use different format in term of TensorFlow model functions but idea will be same. As previous, we will continue with Python and necessarily Tensorflow and OpenCV library.
Estimated Effort	2 hours