ORTA DOĞU TEKNİK ÜNİVERSİTESİ

KUZEY KIBRIS KAMPUSU

CNG 491

GUI & Estimation Report
Intelligent Security System for Abandoned Luggage

Team Members:

Damla Gül Altunay – 2015881

Naciye Karademir – 2016095

Contents

List of Figures

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 . REQUIREMENTS

2.1 System Requirements Specification

2.1.1 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 and Security personnel 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 sometime, 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.

Camera	
C -01	Camera should be connected to the Internet.
C -02	Minimum FPS should be 20, and pixel resolution should be 640*380 for better detection

2.1.2 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. Error messages should give specific instructions for recovery. The help system will explain all functions and how to achieve common tasks.

ii) Reliability

The reliability of the wearable device 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 administrator of the system.

iii) Robustness

Problems in data delivery can occur anywhere with the communication between data sources in our case 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 user aware of what happened, then it could find a solution. Besides that, system is going to check every error and failure to find a solution for them.

iv) Performance

Image data transfer through Internet connection and live streaming makes performance measures critical. For desired performance, image capturing, transfered data size, speed of connection, response time, processing speed must be considered. System should work 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 value for FPS and minimum 640x380 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 and time will be stored.

v) Security

All data for camera and event log is going to be stored in 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.

2.2 System Models

2.2.1 Use Case Diagram

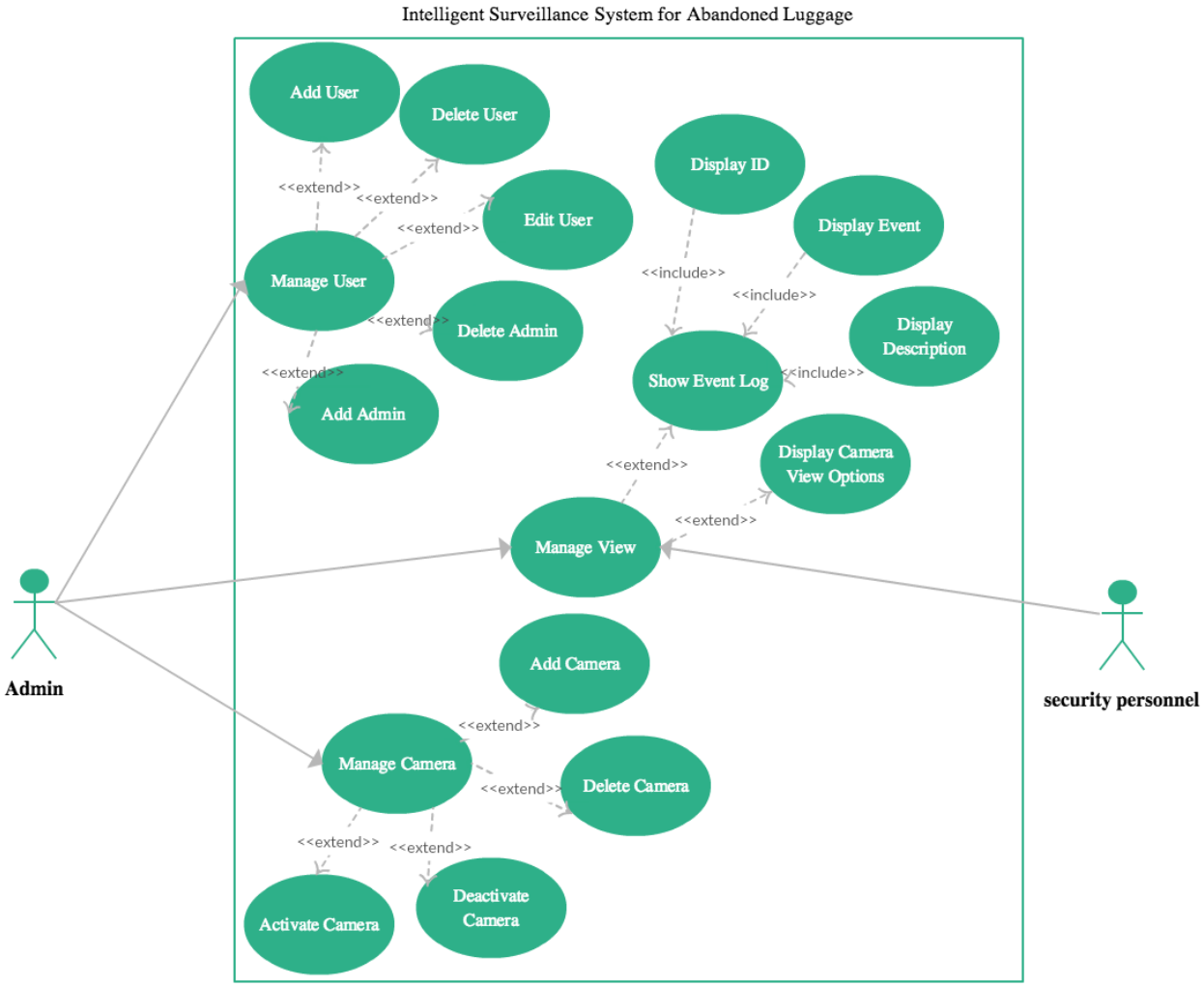


Figure 1: Use Case Diagram

2.2.2 Sequence Diagram

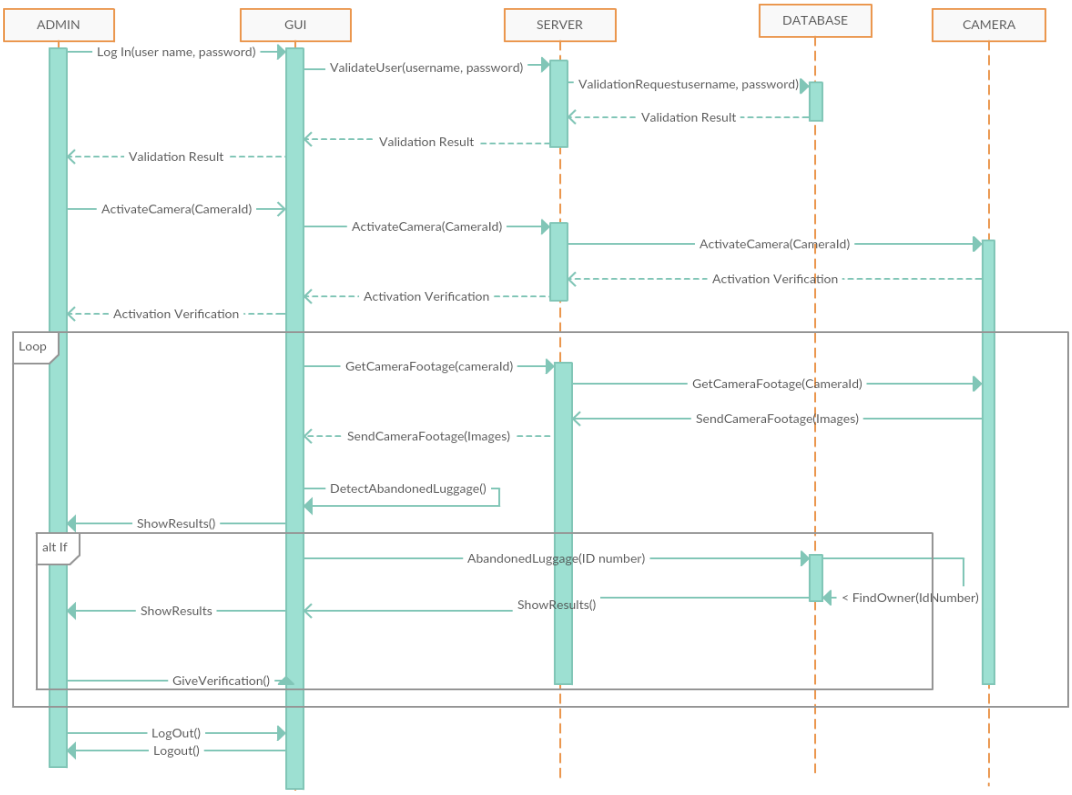


Figure 2: Sequence diagram

2.2.3 Database ER Diagrams

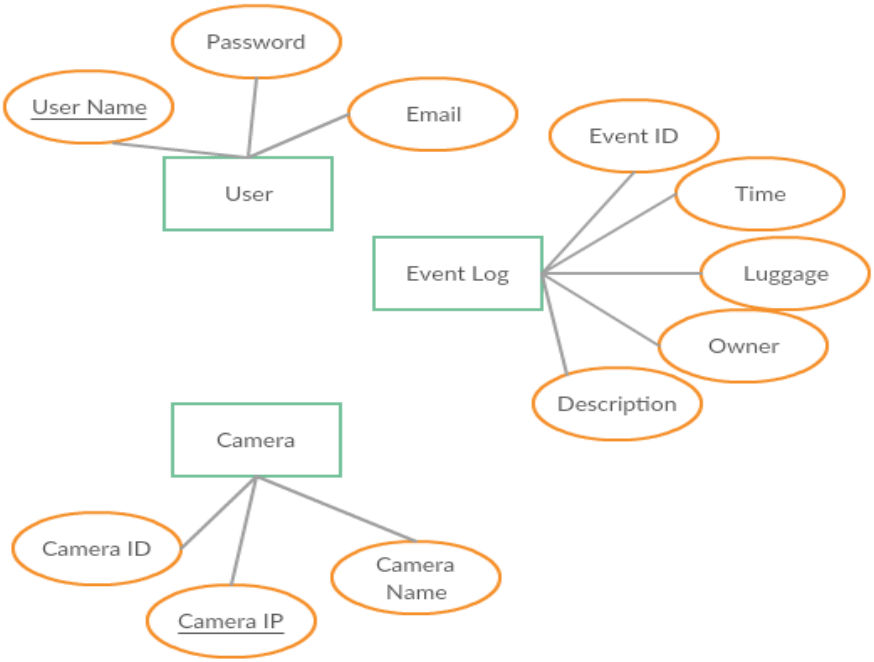


Figure 3: Database ER diagrams

2.2.4 Context Diagram:

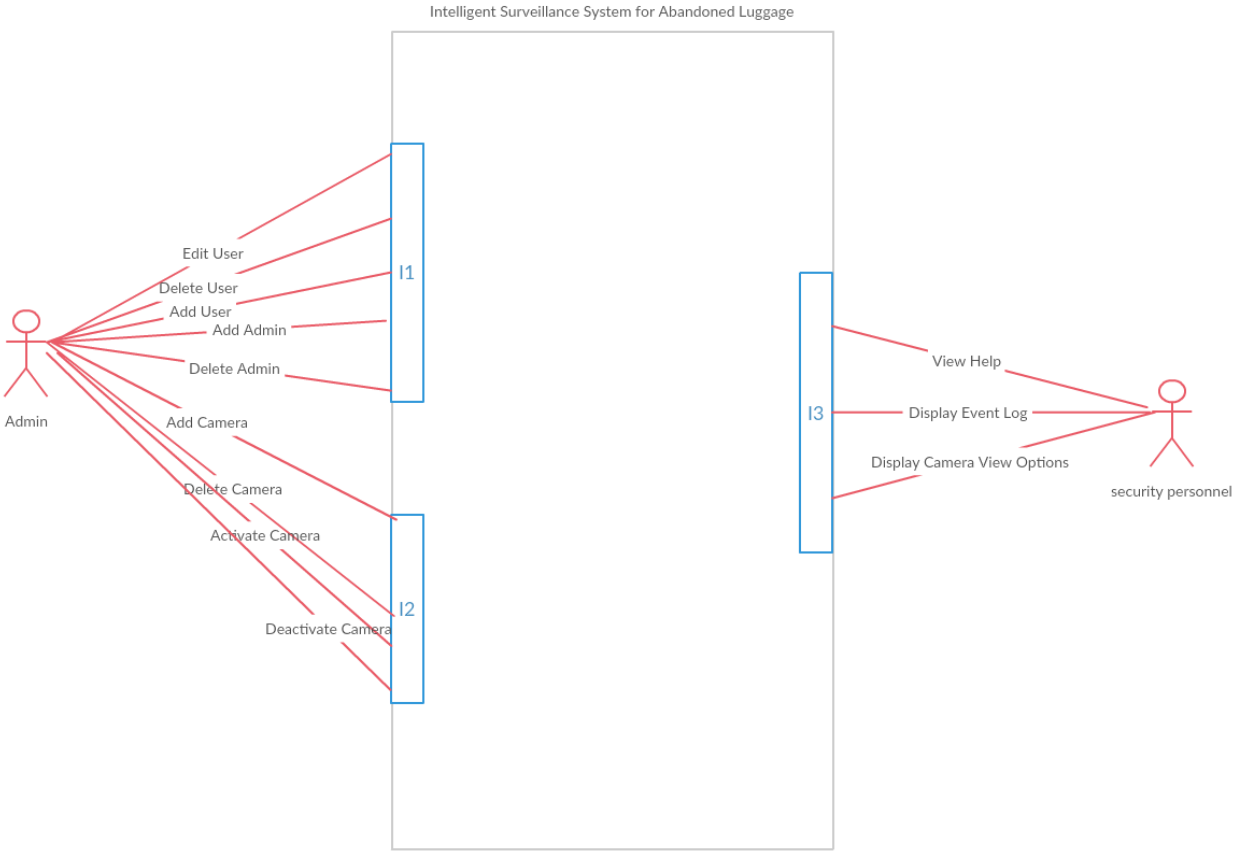


Figure 4: Context Diagram

2.2.5 Architectural diagram level-1

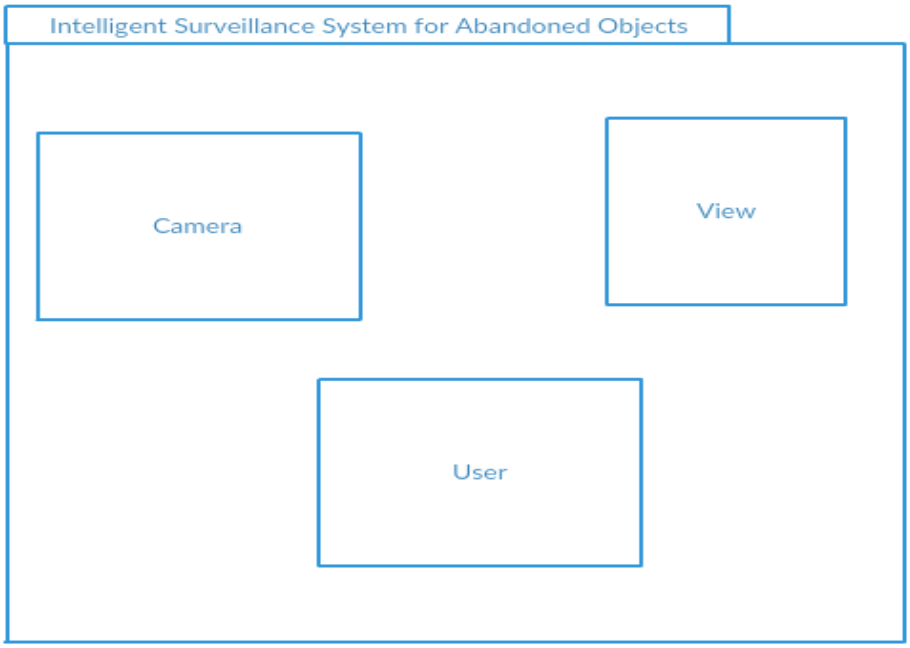


Figure 5: Architectural diagram

2.2.6 Process Model

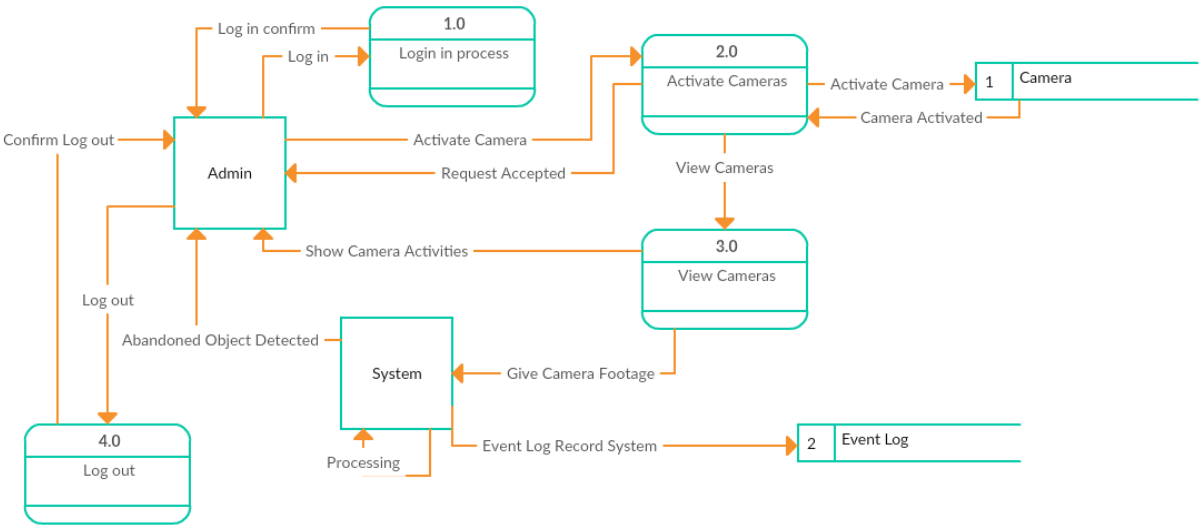


Figure 6: Process Model

3. SCRUM DETAILS

While running this project, we come together every two days and have a meeting for evaluating our progress about what we have done so far and what we are going to do for next meeting. We think that this makes project easy to control and manage. Each meeting lasts maximum 40 minutes.

Our platform is defined as Ubuntu 16.04 since while we were using Windows, we had to add dependencies of the program for each compilation so it wasted our time. Instead, in Ubuntu it is not required, just adding libraries into program is enough to compile successfully. On the other hand, for our first attempt detection algorithm - You Only Look Once (YOLO), we planned to train our model in Darknet framework. It did not work on Windows properly because of dependencies (OpenCV and Tensorflow).

3.1 Sprint Backlog

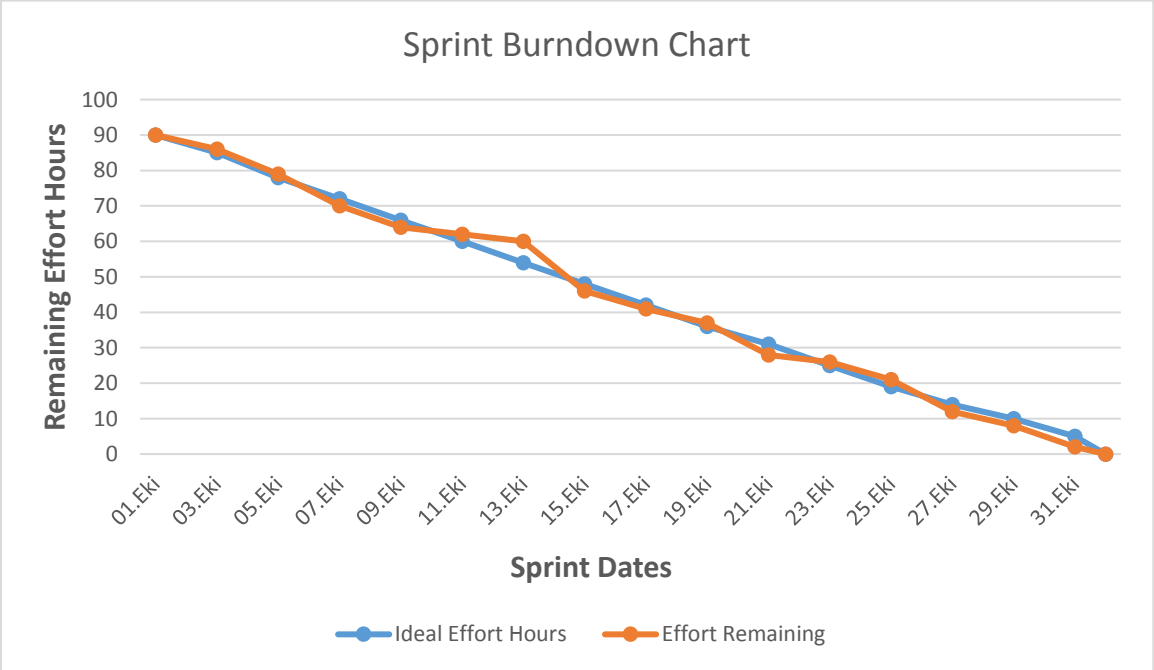
Sprint Backlog		
Sprint Term	Description	Effort in Hours
SPRINT 1 (01.10.17 – 01.11.17)	Installation of Ubuntu 16.04 Virtual Machine on Windows and Mac (team members' laptops)	0.5 hours
	Installation of OpenCV & Tensorflow libraries to team members' laptops	1.5 hours
	Setting up cuDNN and CUDA Development 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	5 hours

3.2 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.

As mentioned in Sprint#1 Backlog, our first program is luggage detection in a given image and video. According to result gained from first training, the accuracy on testing is 81%. However, in this training a small number of dataset was used. In the second training, 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.

3.3 Sprint Burndown Chart



3.4 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.

4. ESTIMATION

Function Point Estimation

Function Points			
Program Characteristic	Low Complexity	Medium Comple	High Complexity
Number of inputs	1 x 3	1 x 4	1 x 6
Number of outputs	2 x 4	0 x 5	2 x 7
Inquiries	1 x 3	2 x 4	1 x 6
Logical internal files	2 x 7	0 x 10	0 x 15
External interface files	0 x 5	0 x 7	0 x 10
Function Point Estimation			
Unadjusted total of function points		66	
Influence multiplier		1.07	
Adjusted total of function point		70.62	

Table 4: Sprint backlog table for MTS

Calculation of Influence Multiplier:

	Rating		
1- data communications	2	11- installation ease	4
2- Distributed Functions	1	12- operational ease	5
3- Performance	3	13- Multiple sites	1
4- Heavily used configurations	1	14- Facilitate change	3

5- Transaction Rate	3	TDI	42
6- online data entry	4		
7- end user efficiency	5		
8- online updates	5		
9- complex processing	2		
10- reusability	3		
11- installation ease	4		
12- operational ease	5		

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

ATFP = UTFP * IM = 66 * 1.07 = 70.62

Since we train our model in C, C for the Language Unit Size (LUS), which concludes us with 128 =>: most common value .

Line of Codes (LOC) = LUS * ATFP = 128 * 70.62 = 9039.36

COCOMO (Constructive Cost Model)

KDSI = Thousand delivered source instructions (KLOC)

KDSI = ATFP * Language unit size /1000 = 70.62 * 128 / 1000 = 9.03936

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 * $(9.03936)^{1.12}$

PM = 35.318

TDEV (Total Development Time) = 2.5 * (PM) ^{c}

TDEV = 2.5 * $(33.8755)^{(0.35)}$

TDEV = 8.70 months

5. GUI

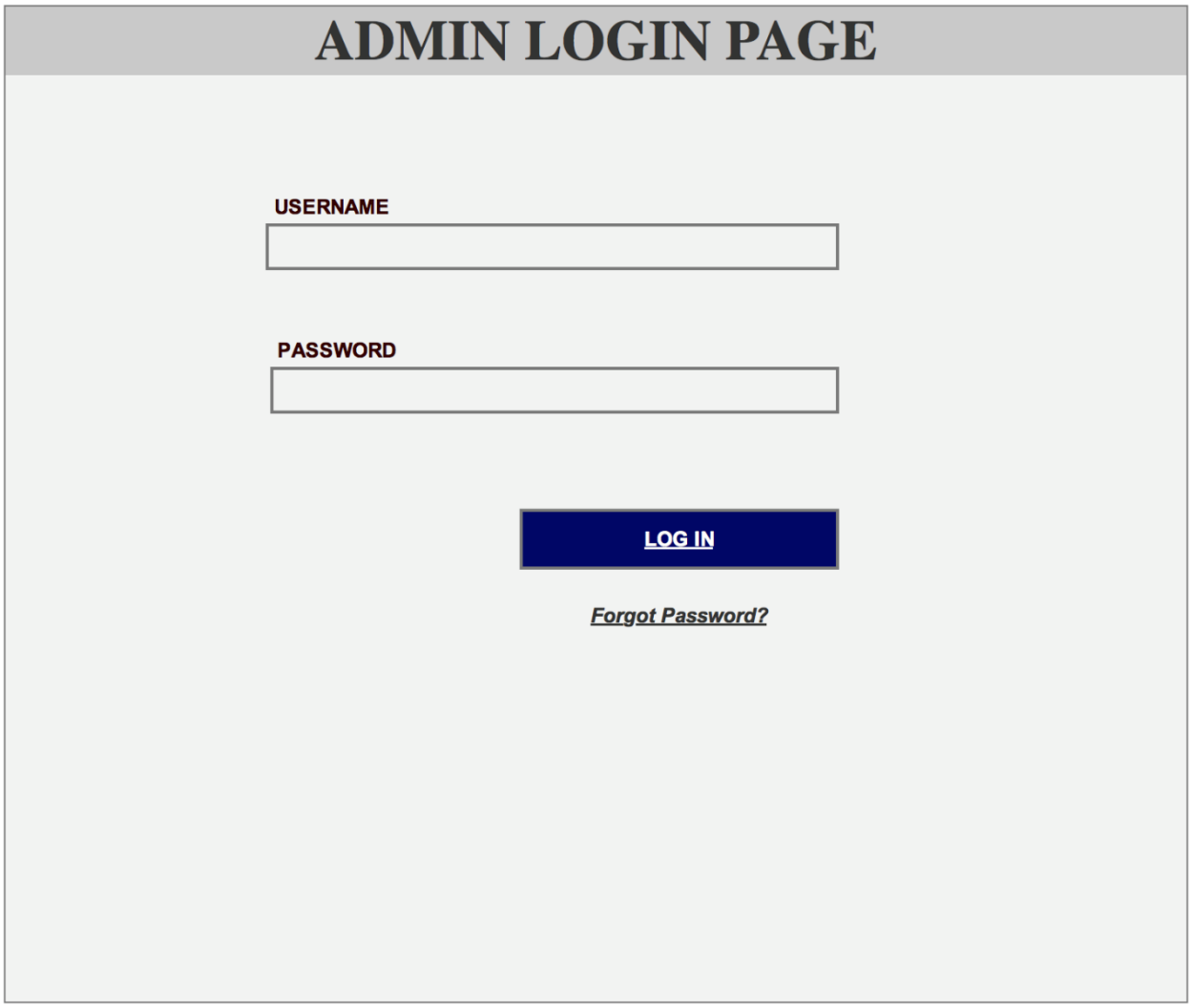


Figure 7: Login Page for Admin

GUI design for our surveillance system desktop application is planned to have multiple users. Firstly, we have should have an administrator to manage the application. An administrator should log in the system first. The page above is planned to use in the first step of our application. Admin will have a username and password to activate the system. If s/he tries to enter the system without a username and/or password, system will show an error message to the administrator. If the admin forgot the password and clicked the "Forgot Password?" button, it will send an email to registered email which is in the system. After administrator logs into the application, main page of it will show up.

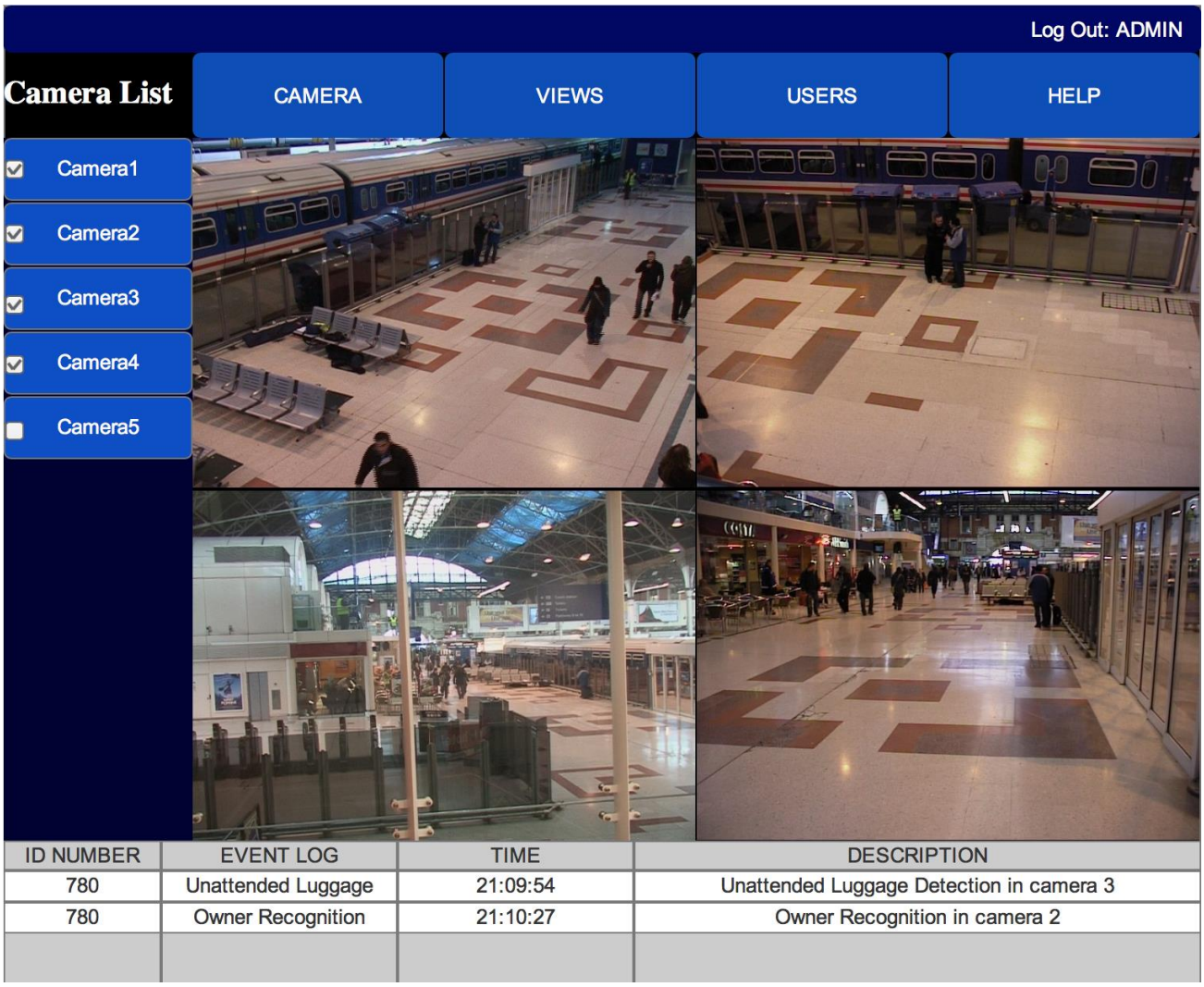


Figure 8: Main Page of Application [1]

Above figure shows the planned design for the main page of the application. In the left admin will be able to see cameras activated in the system. So, camera list basically shows the activated cameras for the security. Cameras which has a tick sign will be shown on the main page. Admin will be able to choose which cameras should be displayed.

In the below of the page, there will be event log flow. Our system will be able to detect any unattended luggage and then it will be able to find the owner also. In the figure, there is a small example to show how it will do and show this.

This flow also will assign an ID Number for luggage and its owner. Also, administrator can see the time of detection and description for the event. In the above of the main page, there will be multiple buttons for admin. Administrator will able to go camera page where s/he can handle camera management system. Below we can see camera page.

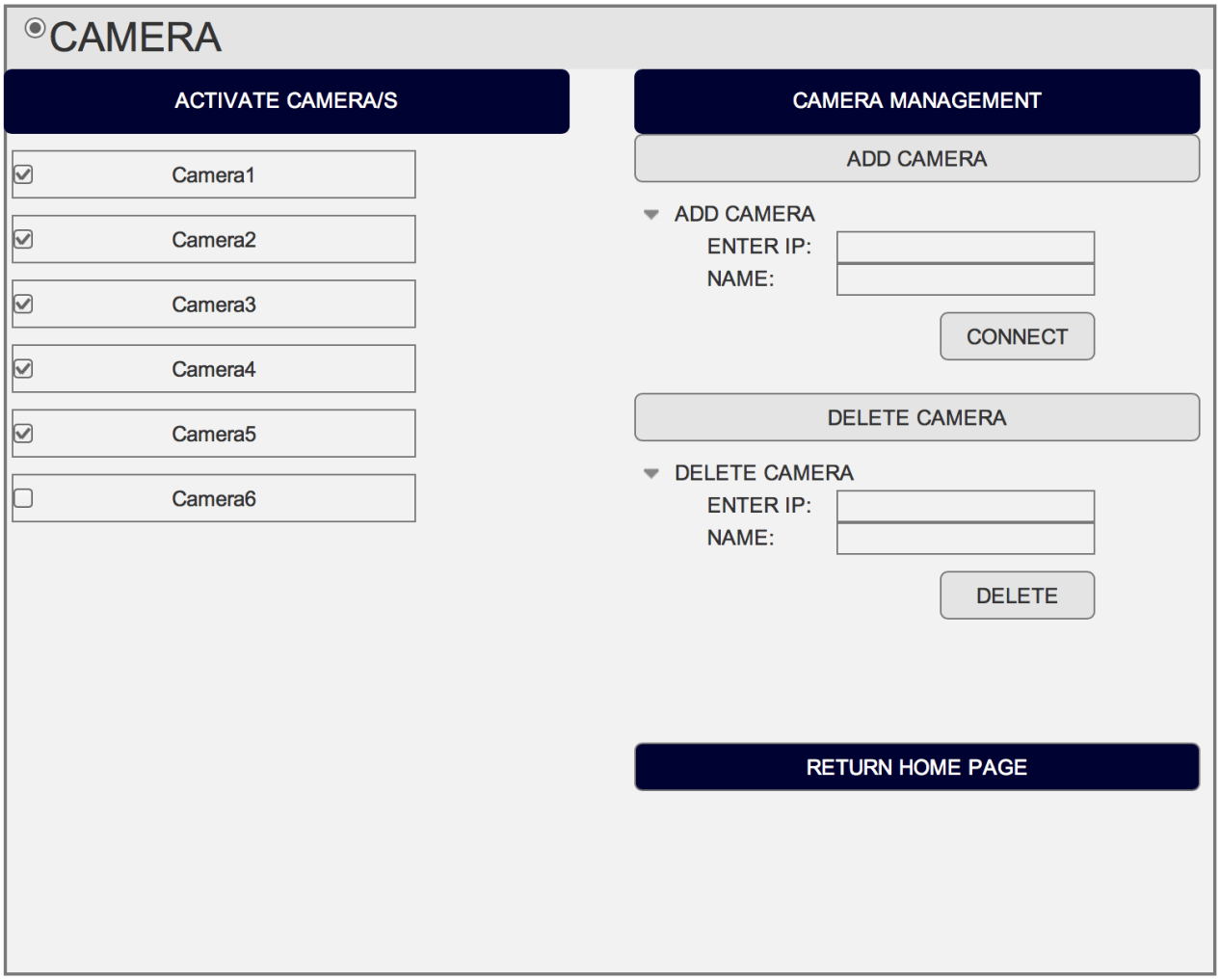


Figure 9: After clicking "Camera" link from Main Menu

In the camera management page, the administrator will be able to activate or deactivate any camera that they have. Also, s/he will be able to add a camera for the system also by IP address of the camera. Furthermore, delete camera option will be available on this page so admin also can remove any camera from the system. Admin can return the main page by clicking to the return home page button below the page.

In the main page admin will have 3 more button choices. One of them is view option. View option is to make admin able to manage camera viewing options. Admin can choose to view only one camera as well as view multiple cameras.

If admin clicks to “USERS” button, s/he will go to user management page in the application. In the user management page, application will have options to add a user. This user will not be admin; they will be security personnel who will follow the system in front of the camera. Also, admin will be able to delete any user from the system. To be clear, these users will only be able to see the video system. Since users will be actually security personnel, they will not have access to the user management or camera management pages like admins have. In addition to this, the administrator will be able to add or delete another administrator from this page also. This change will require an email verification from the new administrator. Two requirements will be requested from administrators, to choose a username and enter a password. We assume that at least one admin should be registered in the system. So if one admin will be deleted, there should be at least one more administrator other than the admin will be deleted.

Help button also available the right sight of the page. This help page will show the contact information and general information about how to use application of the system.

At the top most, there will be logged out page for the administrators. By clicking to this button, the admin will close the application and will be able to just leave.

REFERENCES

[1] 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].