# Software Requirements Specification

## for

# Drone Project

**Version 1.0**

**Prepared by**

**Group Name:** *< Group 9 >*

| | | |
|---|---|---|
| ENES KILIÇ | 2000006890 | 2000006890@stu.iku.edu.tr |
| NURETTİN BERKE DEMİREL | 1900005511 | 1900005511@stu.iku.edu.tr |
| İBRAHİM AĞCABAY | 1900002653 | 1900002653@stu.iku.edu.tr |
| İREM ERKAN | 1700003209 | 1700003209@stu.iku.edu.tr |
| BUKET ÇALŞİMŞEK | 1800002703 | 1800002703@stu.iku.edu.tr |
| ELA NUR ZAMUR | 1800002622 | 1800002622@stu.iku.edu.tr |
| OSAMA ABUHAJIB | 1900005042 | 1900005042@stu.iku.edu.tr |

**Instructor:** *AKHAN AKBULUT*

**Course:** SYSTEM ANALYSIS AND HUMAN COMPUTER INTERACTION DESIGN

**Lab Section:** *A*

**Teaching Assistant:** *BÜŞRA KOCAÇINAR*

**Date:**  **13.05.2022**

## Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
|---------|-------------------|------------------------|----------------|
| 1.0 | Enes Kılıç | The entire content of this document has been prepared | 25/05/21 |

# 1. INTRODUCTION

Our Drone Project basically consists of collecting data using image processing technologies, storing these data regularly and reporting for later analysis.

Although drones were originally built for military purposes, rapid growth and advances have been seen. Their original positions were weapons in the form of remote guided air missile deployers[1]. It was seen in 1839 when Austrian soldiers attacked the city of Venice with unmanned balloons filled with explosives.[2] The invention of winged aircraft changed everything for unmanned and manned vehicles.

From past to present, drones are used for a wide variety of functions, such as human aircraft, monitoring climate change, delivering goods, assisting in search and rescue operations, taking photographs, and recording video.

## 1.1. Purpose

The purpose of this document is to identify the features, requirements and interface of the Drone Project System. It will explain the scenario of the desired project and the necessary steps to be successful in the task. Throughout this document there will be a general description of the project for implementation and definitions and abbreviations related to the project. The preparation of this SRS will help to consider all of its requirements and needs before this project is commissioned and then reduce as much as possible visual redesign, recoding, retesting. This greatly reduces time, cost and work.

## 1.2. Scope

The name of the project to be presented in this document is Developing Autonomous Applications with Drone. We tried to keep the scope of the project as wide as possible, since our application has many features. In this way, we aimed to offer our users the best experience and comfort. For this purpose, we have reviewed, tested and evaluated all drone-related projects.

Concentrating our project on computer vision studies, we determined our features in this context.

We have listed all the features of the project and grouped all these features from general to specific. These groups enabled us to identify the subsystems of our project, the main modules of these subsystems, and the submodules of these main modules. Thus, the relations of all the features in our project with each other were determined on a regular basis.

Since the project does not have a complex interface, has high mobility, and is always used quickly and comfortably, we decided to manage all its features under the control of a simple mobile compatible website.

We recorded the data of the project as it should be, adhering to the database management system. We analyzed this data and prepared reporting studies by making the necessary associations.

We examined the scope of our project from general to specific and prepared all our module groups in line with the list given below..

### 1.2.1. Subsystems
**1.2.1.1.**      **Face Subsystem**
**1.2.1.2.**      **Object Detection Subsystem**

### 1.2.2. Main Moduls
**1.2.2.1.**      **First Face Detection**
**1.2.2.2.**      **Face Recognition**
**1.2.2.3.**      **Gender Age Detection**
**1.2.2.4.**      **Object Tracker Video/Live**

**1.2.2.5.**        **Object Counter Video/Live**
**1.2.2.6.**        **Object Counter Image**
**1.2.2.7.**        **Object Crop Images**

### 1.2.3.    SubModuls
**1.2.3.1.**        **Attendance**

Let's examine all system modules in order according to the module group they are connected to;

### 1.2.1.    Subsystems

We cover the most general features of our project in the subsystems group, which is the most general group of our drone project. These features are the part where the operations related to the human face are located and the part where the features related to object identification are located. Based on these scopes, we have developed our features.

#### 1.2.1.1.   Face Subsystem

It is a subsystem of our drone project, in which modules are connected to the main processes related to the human face in general. Our processes related to the human face that we have developed in this sub-system; The system recognizes the first person it sees and records it as a visitor, the system recognizes the people it knows before appearing on the system camera and performs related actions, and our last feature in this sub-system is gender and age recognition. It detects people's faces with our drone project camera and then estimates their gender and age.

#### 1.2.1.2.   Object Detection Subsystem

It is the subsystem of our drone project, in which modules are connected to the main processes related to object definition and related operations in general. We are defined by the drone camera and we make some features related to them; The first of these is that we track the living or inanimate objects we see in our drone camera by assigning a unique ID number according to the predetermined

classes depending on the object types. Another feature; We count the animate or inanimate objects we see with our drone camera according to the predetermined classes depending on the object types. Our last feature connected to this sub-system is that we crop each animate or inanimate object in the pictures we take with the drone camera according to their predetermined classes and unique identification numbers from the relevant picture and store them in a way that is connected to the picture..

### 1.2.2. Main Moduls

It is a module group that covers all the main features of our drone project. Under the subsystems to which they are connected, they have features that directly concern the subsystem. Some of these modules are face recognition, gender and age detection, object tracking based on unique ID numbers…

#### 1.2.2.1. First Face Detection

On the web design side of our Drone Project, the first face that the webcam sees immediately after registering the login information of our visitors who want to register to our system is the face of the visitor and is a main module with the features of recording the picture of the visitor..

#### 1.2.2.2. Face Recognition

In our Drone Project, it is a main module that includes the features of ensuring that the faces registered in the system (this can be in the form of a picture, can be in the form of video, can be database data) are recognized by the system by comparing the faces of the people using the relevant feature on the drone camera. One of these features is the Atttendance feature. It is introduced to the system as a submodule of the main module.

### 1.2.2.3. Gender-Age Detection

It is a main module of our Drone Project that has the features of estimating and recording the gender and age of human faces determined by the drone camera.

### 1.2.2.4. Object Tracker Video/Live

With the drone camera of our Drone Project, it determines the unique identification numbers of predetermined living or inanimate objects in a live image or in a pre-existing video recording, and it can record them in the desired video format. At the same time, it is a main module with the features of recording the resulting data..

### 1.2.2.5. Object Counter Video/Live

It is a main module that has the features of recording the predetermined living or inanimate objects in a pre-existing video recording with the drone camera of our Drone Project, determining their total number by determining their total number and recording them.

### 1.2.2.6. Object Counter Image

It is a main module that can determine the predetermined animate or inanimate objects according to their objects, determine the total number of the pictures taken with the drone camera of our Drone Project, and record them.

### 1.2.2.7.        Object Crop Image

It is a main module that has the features of determining the predetermined animate or inanimate objects according to the objects of the pictures taken by the drone camera of our Drone Project, cropping them depending on the object group and the main picture of the object, naming them according to this information, and recording the pictures and their associated data..

## 1.2.3.    Sub Molduls

It is the module group that our drone project includes sub-features related to its main features. Main features can be developed by separating them into sub-feature groups as well as on their own. In this context, our Attendance submodule is available in our project..

### 1.2.3.1.   Attendance

In our Drone Project, it is a sub-module that has the features of detecting people whose faces are determined according to their pictures in the system, and when they use the relevant feature on the drone camera, those people are detected by the matching control between the faces of the people present in the system and their own faces, and that these people are recorded..

## 1.2.4.    Project Users

Our users using our drone project are separated by certain structures. Our users are divided into three basic role groups. These are guest, systemOfficials, users roles..

### 1.2.4.1. Guest

A group of our users who use our drone project are visitors. They have the lowest authority group. Visitors are users who want to use our drone project but do not have any authority. Our users who want to use our drone project must register via the web platform. Thanks to our First face detection main module and the web platform, our visitors can access the standard user status by becoming a member..

### 1.2.4.2. System Owners

A group of our users who use our drone project are system owners. System owners are the user group that has full authority over our drone project. They can access all kinds of important data and interfaces within the system. They are responsible for many system development tasks such as system control, sustainability, usability, integrity, security..

### 1.2.4.3. Standard Users

A group of our users who use our drone project are standard users. Standard users are the group our drone project serves. These users have full access to the features provided by our drone project. They do not have access to pages on both the drone software and the web platform, where private, confidential data and data groups are located. They are isolated from these structures. Users' personal information has been used in our project for the purpose of providing services..

**Note:** Our done project is not involved in any data breaches. Our data is not shared with third parties. It is protected by law

**Not:** Anyone using our dorne project is deemed to have accepted our data privacy policiy.

## 1.3.   Objectives

The main purpose of the Drone Project described in this document is the integration of software types serving on different platforms within the scope of software engineering and providing a higher quality service with a single integrated software resulting from this integration. Within the scope of this project, we provide services to our users with the integrated software I mentioned, which we have obtained by making use of the features of our drone device, which we use as its own device, as well as the ability to take images with the camera, as well as providing services to our users with our data that we have recorded by adhering to the database management system that we offer both unprocessed and processed. It was ensured that they could easily reach the data analysis they targeted.

## 1.4. Definitions and Abbreviations

SRS: Software Requirements Specificiation

API: Application Programming Interface

## 1.5. References

1.5.1.    A Brief History of Drones: The Remote Controlled Unmanned Aerial Vehicles, https://interestingengineering.com/a-brief-history-of-drones-the-remote-controlled-unmanned-aerial-vehicles-uavs

1.5.2.    The History of Drones(The History Timeline From 1849 To 2019),          https://en.wikipedia.org/wiki/Jibbigo

## 1.6. Roles and Responsibilities

| Enes Kılıç(Project Supervisor) | |
|---|---|
| **Annotated Directive Given in the Project** | **Document Ownership According to the Related Directive** |
| 1 INTRODUCTION | 1.  INTRODUCTION<br>    1.1. Problem Definition |
| 1.1 Project Purpose and Scope, and Objectives | 1.2. Purpose<br>1.3. Scope<br>1.4. Objectives<br>1.5. Definitions and Abbreviations<br>1.6. References |
| 1.2 Roles and responsibilities | 1.7. Roles and Responsibilities |
| 1.3 Technical Assumptions and Constraints | 1.8. Overview<br>2.  OVERALL DESCRIPTION<br>    2.1. Product Perspective<br>    2.2. Product Functionalities<br>    2.3. Operating Environment<br>    2.4. Constraints, Assumptions and Dependencies |
| 1.4 Naming Conventions | 2.5. Naming Conventions<br>    2.5.1.  Document Formatting Rules<br>    2.5.2.  App(Coding) Naming Conventions |
| 4.3 Use Case View | 5.4. Use Case View |

| İbrahim Ağcabay - İrem | |
| --- | --- |
| **Annotated Directive Given in the Project** | **Document Ownership According to the Related Directive** |
| 2 Requirements<br>2.1 Functional Requirements | 3.  REQUIREMENTS<br>    3.1. Functional Requirements<br>    3.2. Products Requirements<br>    3.3. Data Requirements<br>    3.4. Security Requirements<br>    3.5. Performance Requirements<br>    3.6. Usability Requirements |
| 2.2 Non Functional Requirements<br>    2.2.1 Performance Requirements | 4.  NON FUNCTIONAL REQUIREMENTS<br>    4.1. Performance Requirements |
| 2.2.2 Safety and Security Requirements | 4.2. Safety and Security Requirements |
| 2.2.3 Software Quality Attributes | 4.3. Software Quality Attributes<br>    4.3.1.  Usability<br>    4.3.2.  Reliability<br>    4.3.3.  Availability<br>    4.3.4.  Portability<br>    4.3.5.  Testability<br>    4.3.6.  Scalability<br>    4.3.7.  Flexibility<br>    4.3.8.  Reusability<br>    4.3.9.  Maintainability<br>    4.3.10. Supportability<br>    4.3.11. Interoperability<br>    4.3.12. Performance<br>    4.3.13. Security |

| Nurettin Berke Demirel - Buket | |
| --- | --- |
| **Annotated Directive Given in the Project** | **Document Ownership According to the Related Directive** |
| 3 Other Requirements | |
| 4 System Architecture and Architectural Design | 5.  SYSTEM ARCHITECTUREL AND ARCHITECTURAL DESIGN<br>    5.1. Performance Requirements<br>        5.1.1 Waterfall<br>        5.1.2 Scrum<br>        5.1.3 Agile |
| 4.1 Logical View | 5.2. Logical View |
| 4.2 Deployment View | 5.3. Deployment View |

| OSAMA ABUHAJIB(soldan 5) – Ela Nur Zamur(soldan 4.3.1 ve 6) | |
| --- | --- |
| **Annotated Directive Given in the Project** | **Document Ownership According to the Related Directive** |
| 4.3.1 Use Case Scenarios | 5.5. Use Case Scenarios |
| 5 Design and Implementation | 6.  DESIGN AND IMPLEMENTATION |
| 6 Other Supporting Information | |

## 1.7. Overview

This section explains the purpose, scope and objectives of the product, related dependencies and assumptions. The next section explains the product perspective, product functions, product restrictions, dependencies and assumptions of the respective application. In the third part of the document, all the requirements of the application are mentioned. The fourth part describes the non-functional requirements. In the fifth chapter, the system architecture and architectural design of the application and finally the design and implementation of the application are mentioned.

# 2.OVERALL DESCRIPTION

This part gives information about product perspective, product functions and constraints, assumptions and dependencies respectively.

## 2.1.   Product Perspective

The drone project is a structure prepared by combining various software platforms based on computer vision applications.

This structure was determined by using our computer/software engineering skills as much as possible, as well as by experimenting and testing various similar products, applications, projects and structures.

The following paragraph contains basic information of the software we use;

- Operating System: Windows10/Linux,Android(Web),Ios(Web)

- Programming Language: Python 3.9, HTML , CSS , PHP, JAVAscript, MySQL

- Development Environment: Pycharm 2021.3.2(Professional Edition) , Notepad++ , Microsoft SQL server management studio 18

## 2.2.   Product Functionalities

The basic functions of our drone project are on the use of image processing applications. Therefore, the fps value of the drone camera must always be high. Since the drone device provides power from its own battery, not from any power port, the charge level of the battery should not be reduced below the minimum level to fly. Drone devices are made of light and sensitive materials in order to fly, so they are susceptible to breaking, crushing and fragmentation due to their general structure. For this reason, we must protect our drone device. The wifi device on the drone should be of high quality, since there is no disconnection between the web and the drone.

## 2.3.   Operating Environment

The product runs on any operating system that can run a web browser. In order to use the product, if the operating system on the computer is Windows, it must be at least Windows Vista. Linux must be at least the minimum MX Linux operating system. On the Mac side, you must have Mountain Lion – Mac OS X 10.8. On Android devices, at least Android 2.3 (Gingerbread) version, and Ios 5.0 on Ios devices. These minimum system requirements must be met to use the product.

## 2.4.   Constraints, Assumptions and Dependencies

In order to maintain the accuracy, following assumptions are made:

- The internet connection of the drone device should never be lost. Therefore, the necessary equipment should be used under appropriate conditions of use. The internet speed of the device should be at least 10 Mbps and the signal should be strong..

- Human face and object identification etc. in the drone device. In all features, if a live image is taken, care must be taken to capture the target objects of the camera angle with a quality resolution and to cover the objects. If object identification is to be made from the video recording, the video must be of good enough resolution to detect the objects and to cover the object...

- If the use of our Drone Project will be in a closed area, there should be an area that is wide and high enough for the drone device to fly and detect objects. If it is to be used outdoors, the weather should be favorable and the drone device should be used from a suitable location as the distance of the drone device in the open area will be unlimited. You should be cautious about negative situations that may occur.

- The drone device is a sensitive device due to its structure. It can hit or crush any object during use, so be very careful when using it..

## 2.5. Naming Conventions

This document has been prepared within certain naming conventions.

### 2.5.1. Document Formatting Rules

- ➢ Entire document should be justified.
- ➢ Convention for main title
  - Font face: Times New Roman
  - Font style: Bold
  - Font size: 16
- ➢ Convention for sub title
  - Font face: Times New Roman
  - Font style: Bold
  - Font size: 14
- ➢ Convention for sub(sub) title
  - Font face: Times New Roman
  - Font style: Bold
  - Font size: 13


- ➢ Convention for body
  - Font face: Verdana
  - Font size: 11

## 2.5.2. App(Coding) Naming Conventions

| Kind | Rule |
| --- | --- |
| Private field | _lowerCamelCase |
| Public field | UpperCamelCase |
| Protected field | UpperCamelCase |
| Internal field | lowerCamelCase |
| Property | UpperCamelCase |
| Method | UpperCamelCase |
| Class | UpperCamelCase |
| Interface | IUpperCamelCase |
| Local veriable | lowerCamelCase |
| Parameter | lowerCamelCase |

# 3. REQUIREMENTS

## 3.1. Functional Requirements

### 3.1.1 Table to keep personal information
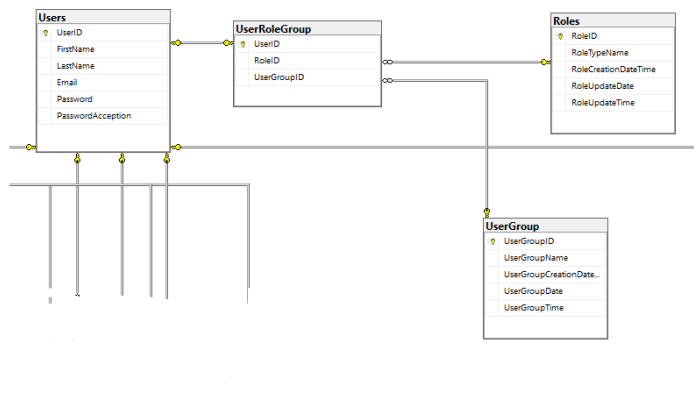


The tables in which the information of the persons are kept and the role id is assigned according to the authority of the persons and the group name are indicated in the figures.

### 3.1.2 Identification, Face analysis,Logging
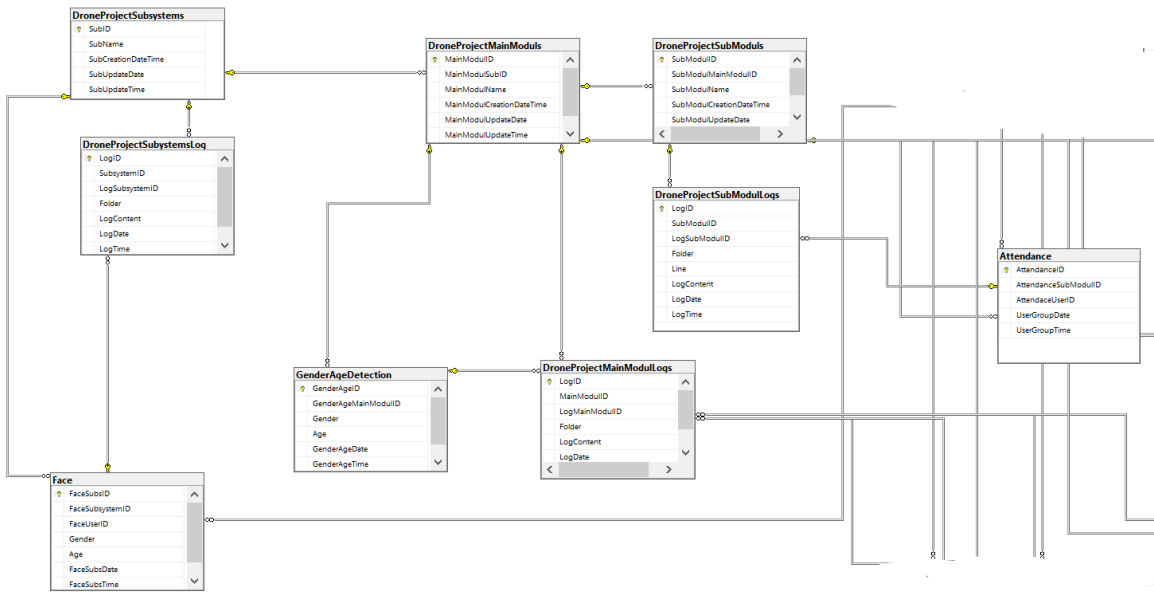


The gender and age analysis of the user, who shows his/her face in the tables specified in the figure, is made and this information is recorded in the relevant tables. In addition, the log information of the user logged into the system is also recorded in the table (attendance).

### 3.1.3  Video and picture analysis
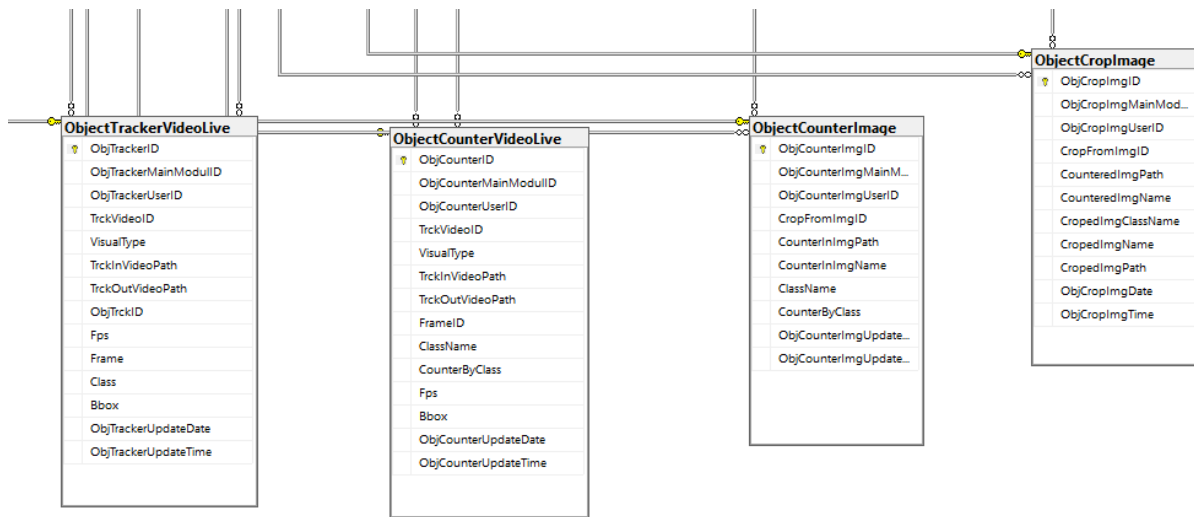


Records of every analysis made on photographs and videos are kept in these tables.

## 4.  NON FUNCTIONAL REQUIREMENTS

## 4.1 Performance Requirements

There must be a database version that is compatible with the system to which the database will be integrated. Thus, the connection can be established properly and correctly, and the transactions are carried out without any problems. So I used mssql.

## 4.2 Safety and Security Requirements

When there is an error or problem in the system, the database is not affected at all. There is no data loss or any change on the data.

## 4.3 Software Quality Attributes

### 4.3.1 AVAILABILITY

The database is well integrated with the system and the purposeful transactions take place dynamically.

### 4.3.2 CORRECTNESS

The data to be recorded in the project and the dates of the transactions are transferred directly to the database.

### 4.3.3 MAINTAINABILITY

The database design and the system connection were made properly and correctly. That's why the whole project runs fluently.

### 4.3.4. USABILITY

Since the database design looks pure and straightforward, another person can understand the database mechanism.

## 5. SYSTEM ARCHITECTURE AND ARCHITECTURAL DESIGN

In software engineering, the architecture and design of the project goes through certain stereotyped phases. This is called the Software Development Life Cycle (SDLC). Its phases are: Planning, Analysis, Design, Implementation, Testing and Maintenance.
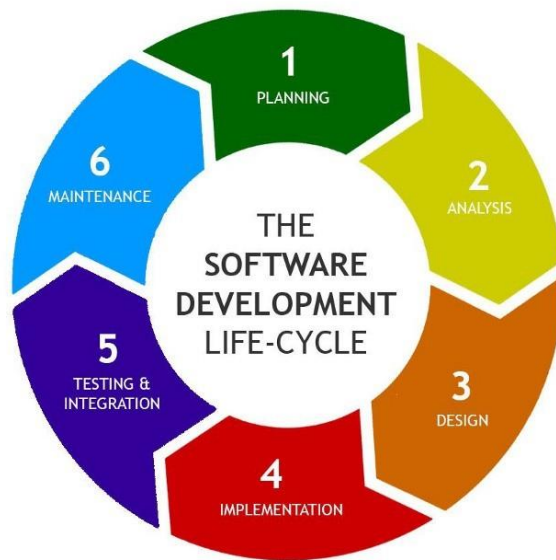
These phases have been used with different methods over the years. Each method serves a different purpose and each method has its own advantages. The important thing is to use the most appropriate method according to the needs.

## 5.1 SOFTWARE DEVELOPMENT METHODOLOGIES

### 5.1.1 Spiral Model

In the "Spiral Model" the project is generally divided into 4 phases: 1-Idetify objectives, 2-Perform risk analysis, 3-Develop and test, 4-review and evaluate.
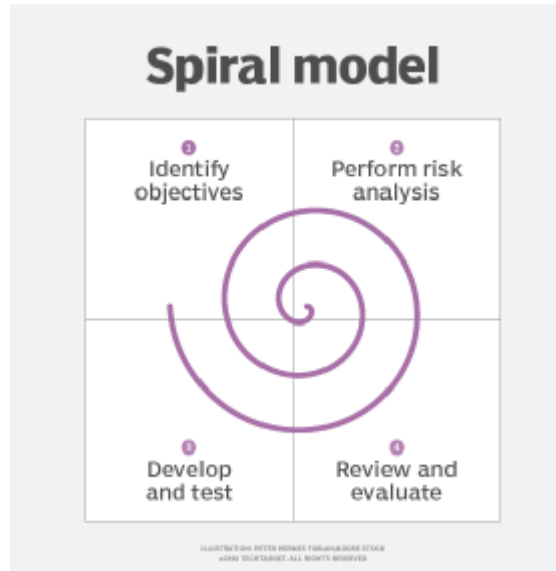


In the second part of these 4 phases, a prototype is created. It starts with a small and superficial part of the application. This prototype does the design and implementation of the application with a small waterfall technique. Then this small application passes to other phases. Prototypes are created again and again until the application reaches its final state.
 Since these 4 phases are constantly repeated, a dynamic management is ensured. It gives you the opportunity to think over and over again about the basic needs and problems of the application. Every phase of the project can be intervened very quickly. There is a traceable application at every phase.

The spiral model was used for our project as follows:
In the first prototype, the purpose of the system and expectations from the system were established.
In other prototypes, systematic studies were carried out together in the following areas: "web", "database", "drone's own software" and "software that will work independently of the drone".

1. On the web; In the first prototypes, "a website" with a simple interface such as login and registration was created. In the next prototypes; An interface has been created where python codes, database and device (camera) are connected and the user can access the necessary information.
2. On the database; In the first prototypes, general tables of the system were created. In the next prototypes; Detailed columns and rows were added to the tables, data entries were made.
3. On the device for software; In the first prototypes, the setup of the drone was created in the software field using its own libraries. In the next prototypes; Transferring the data received from the camera was carried out with various projects with external SDK.
4. On the  non-device software: In the first prototypes, general applications were made for face recognition and object detection. In the next prototypes; projects were detailed, database connection was made and tested with real applications.
5. All domains were combined with the final prototypes (firmware, non-device software, web, database)
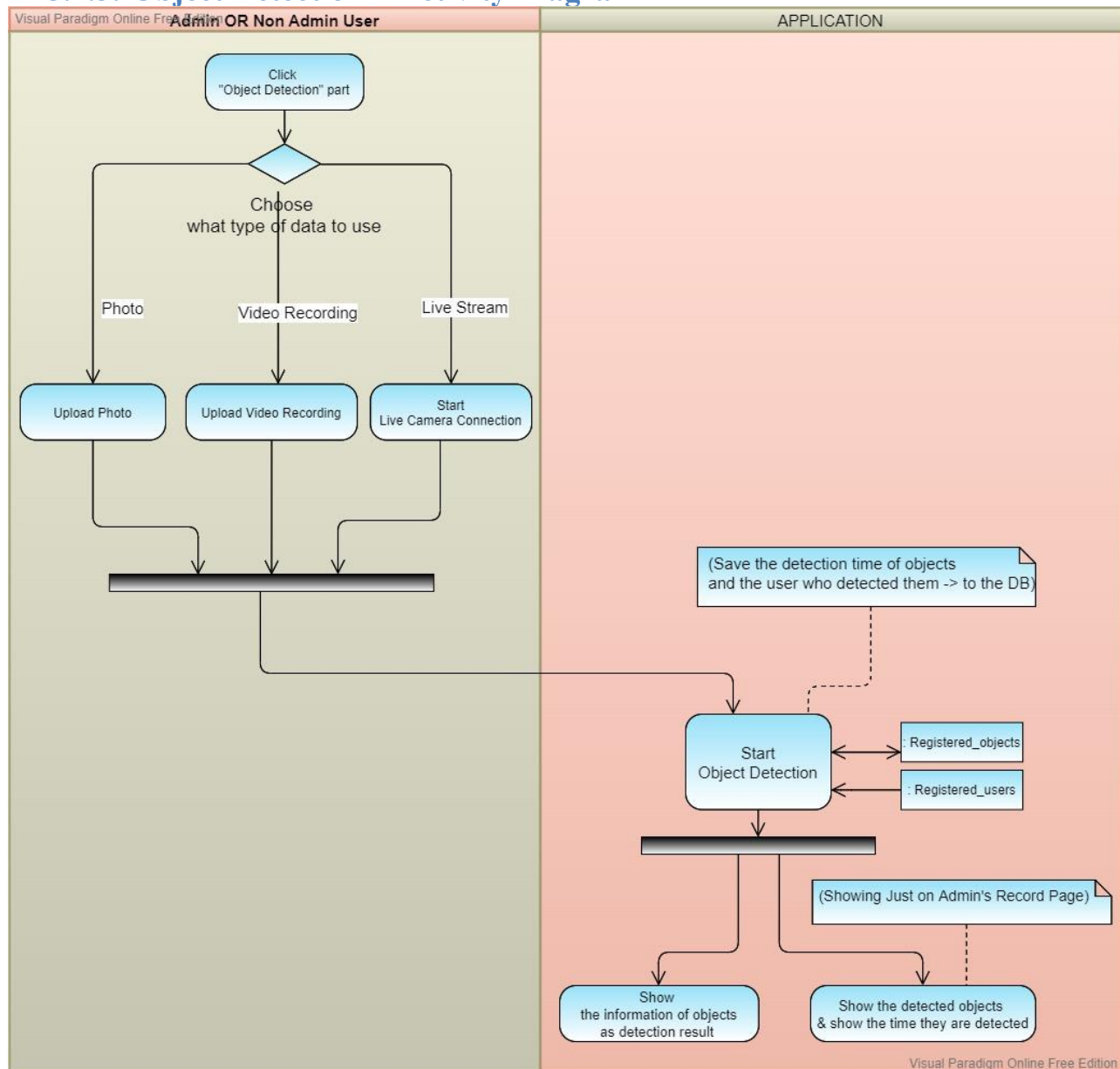
## 5.2 LOGICAL VIEW

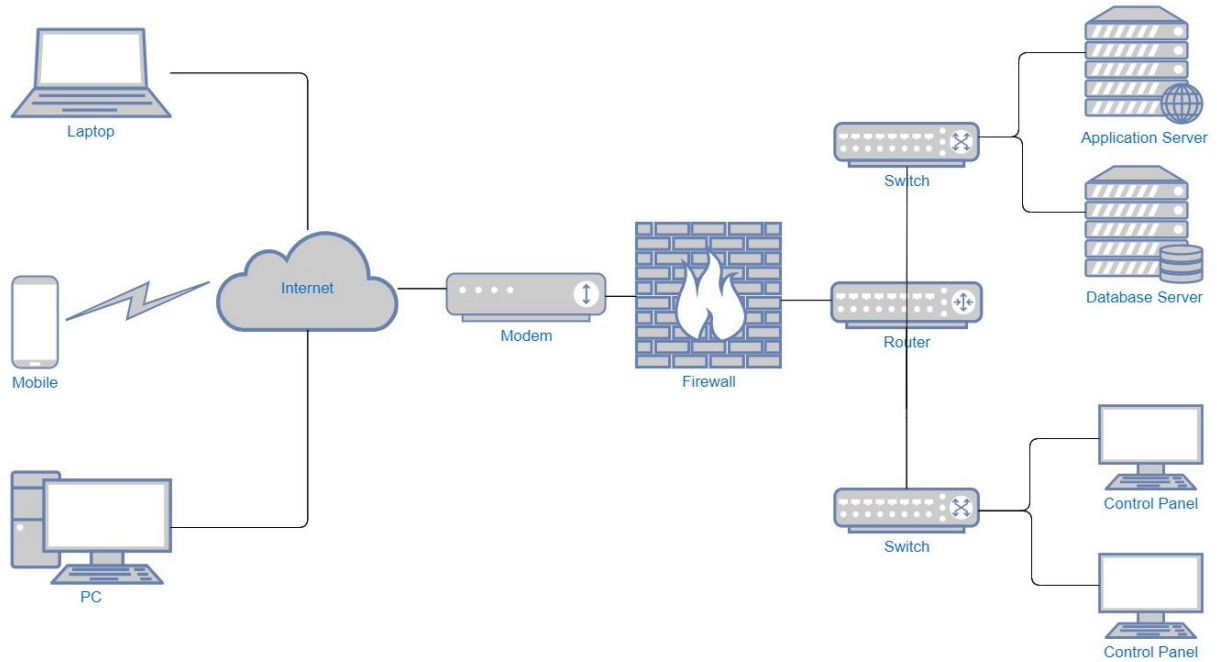## 5.2.1 Opening the Application – Activity Diagram
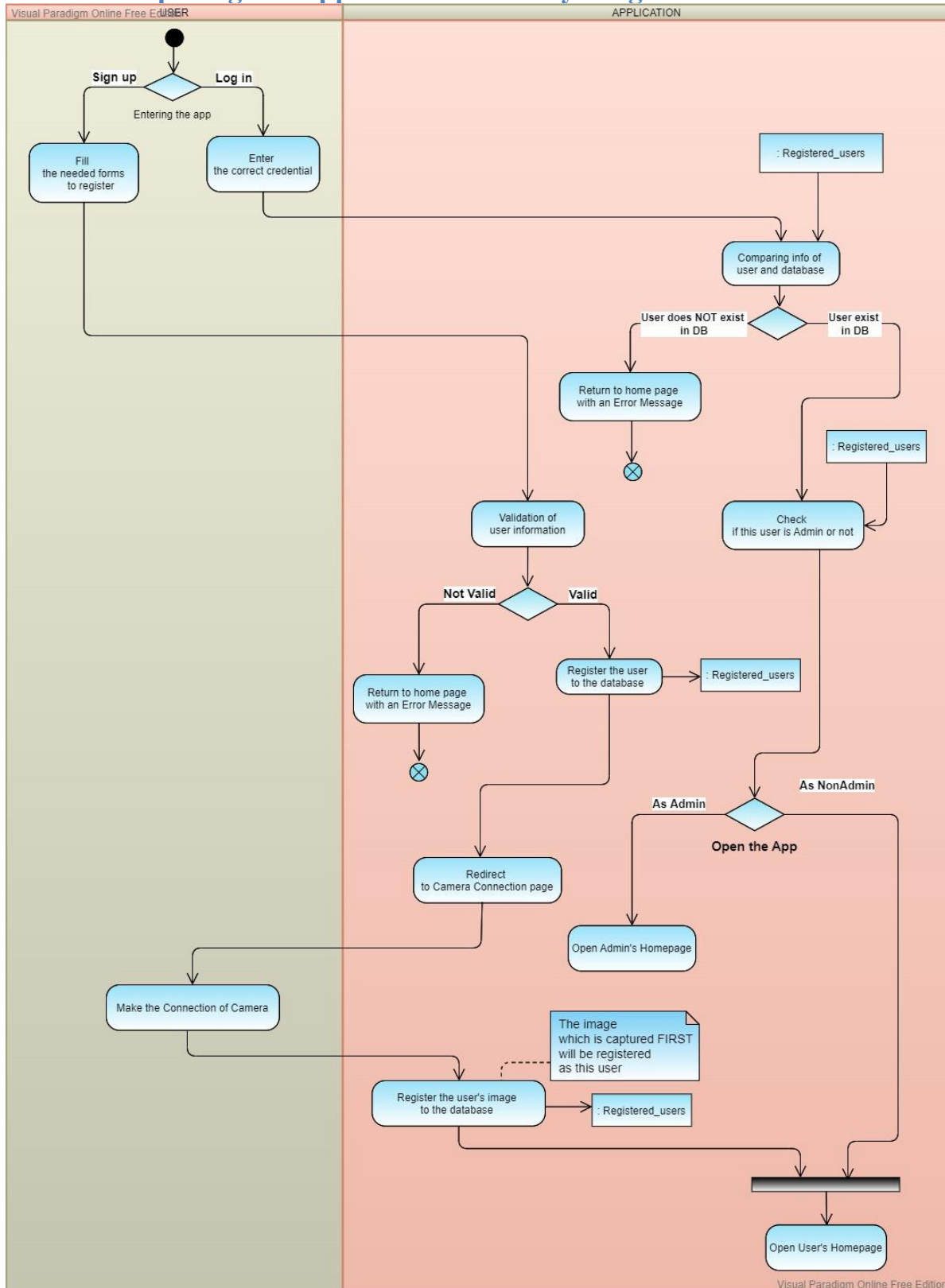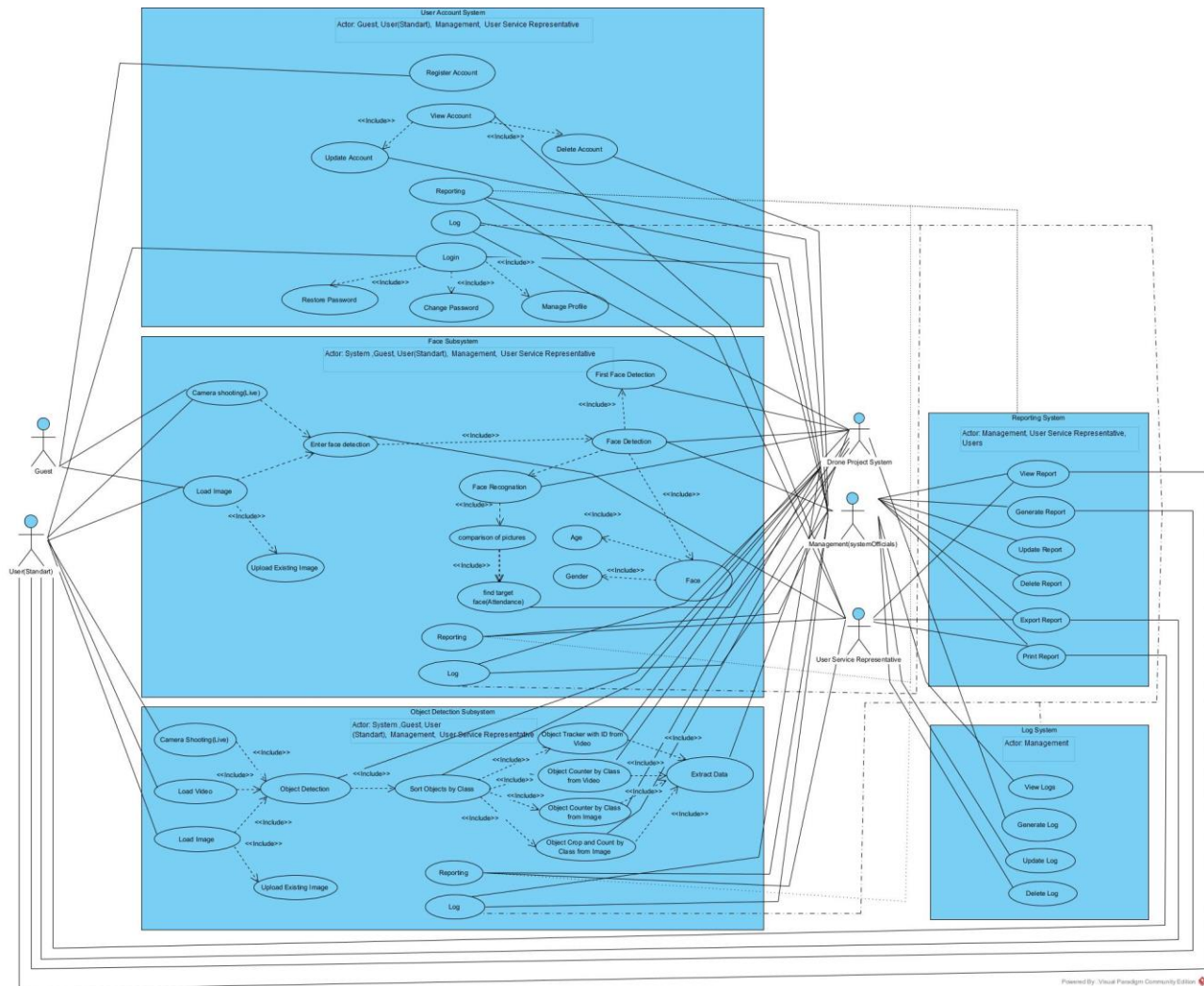
## 5.2.2. Face Recognition – Activity Diagram
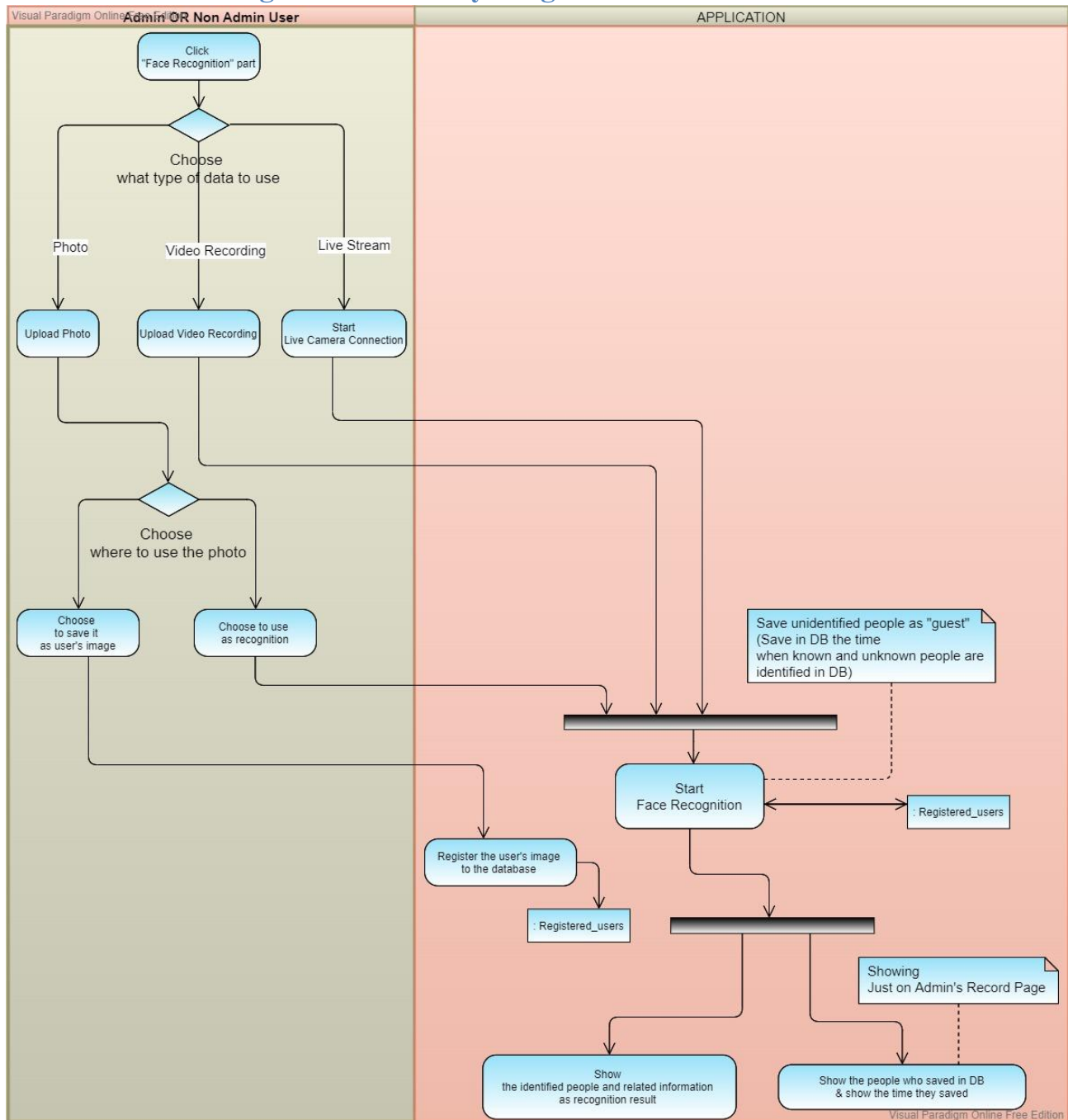
### 5.2.3. Object Detection – Activity Diagram

## 5.3. DEPLOYMENT VIEW

# 5.4. USE CASE VIEW

| User Account Subsystem | |
|---|---|
| **Use Case** | **Description** |
| Register Account | Create a user(standard) registration for the use of the Drone Project. |
| View Account | View / management of the created user(standard) account and use the usage features based on this situation. |
| Update Account | Updating the created user(standard) account. |
| Delete Account | Deleting the created user(standard) account. |
| Reporting | It is the status of all reporting studies of system users (standard) and system administration, about the user account subsystem. |
| Login | Created user(standard) can Login to the System and use the usage features based on this situation. |
| Restore Password | Renewing the password of the created user's(standard)  account |
| Change Password | Changing the password of the created user's (standard) account |
| Manage Profile | Managing the profile information of the created user's(standard)  account (name, surname, phone number, mail ..) |

| Actors | |
|---|---|
| **Actor** | **Description** |
| Guest | People who are likely to be included in the Drone Project. |
| User(Standart) | They are authorized persons who can use the features of the Drone Project. |
| User Service Representive | They are people who help users using the Drone Project to support their use. |
| Management | They are both the owners and the people responsible for the management and execution of the Drone Project. |
| System(App) | It is the only software of Project Drone integrated across platforms. |
| | |

| Face Subsystem | |
|---|---|
| **Use Case** | **Description** |
| Camera Shooting(Live) | Live video recording for the face subsystem of the drone camera |
| Load Image | Official uploads of the Drone Project's users (participation poll etc.) and visitors (when signing up) with their own faces |
| Upload Existing Image | Uploading already existing pictures of the Drone Project's users (participation polls, etc.) and visitors (when signing up) with their own faces |
| Enter Face Detection | In the Drone Project, the data taken by the system to determine the face of the face picture/live recording taken from the user or the visitor |
| Face Detection | Approval and acceptance of the data received by the system to determine the face of the face picture/live recording received from the user or the visitor in the Drone Project |
| First Face Detection | In the Drone Project, the first detected face in the webcam opened to record the faces of the users who want to register in the system, right after they enter the necessary information into the system. |
| Face | It is the use case that depends on the use cases of the operations to be performed on all the faces determined in the drone project. |
| Age | Finding age by looking at the designated face in the drone project |
| Gender | Finding gender by looking at the face identified in the drone project |
| Face Recognition | The first detected face in the webcam, which is opened to record the faces of the users who want to register in the Drone Project, immediately after entering the necessary information into the system. (The person trying to register in the system, that is, the face of the visitor) |
| Comparison of Pictures | Users registered to the system in the Drone Project compare their own faces (live/picture) |

| | with the pictures of previously registered faces in the system while using the attendance submodule. |
|---|---|
| Find Target Face | It is the usage situation of the users registered in the system in the Drone Project, when using the attendance submodule, comparing the image of their own face (live/picture) with the pictures containing the previously registered faces in the system, and the corresponding search for a match submodule. |
| Reporting | In the drone project, the data of the face subsystem and the main module and submodules to which it is connected are made into report files/sections by making necessary inquiries. |
| Log | It is the use case of the system created in order to use informative data for the purpose of informing the management side of the functions performed in the program while the program is running, of the face subsystem and the main module and submodules to which it is connected in the drone project. |

| Object Detection Subsystem | |
|---|---|
| **Use Case** | **Description** |
| Camera Shooting(Live) | It is the live video (live) recordings made with the drone camera to find the objects of the users registered in the system in the drone project. |
| Load Vide | Users registered to the system in the drone project upload the previously taken video recording to the system to find the objects. |
| Load Image | Users registered to the system in the drone project upload the previously taken picture to the system to find the object |
| Upload Existing Image | |
| Object Detection | Identification of objects in live recording, video or images uploaded to the system in the drone project |
| Sort Objects by Class | Grouping the recognized objects in the live recording, video or pictures uploaded to the system in the drone project according to their classes (Ex: 3 person, 3 bicycle, 1 dog) |
| Object Tracker with ID from Video | Grouping the recognized objects in the live recording or video uploaded to the system in the drone project according to their classes and specifying them by assigning a unique ID number. |
| Object Counter by Class from Video | Specifying the total number of recognized objects in the live recording or video uploaded to the system in the drone project by grouping them according to their classes |
| Object Counter by Class from Image | Specifying the total number of recognized objects in the picture uploaded to the system in the drone project by grouping them according to their classes |
| Object Crop and Count by Class from Image | In the drone project, the recognized objects in the picture uploaded to the system are grouped according to their classes, and all objects are cropped one by one, named according to their groups and unique ID (sequence number) numbers, and stored under the folder opened for the picture they belong to. |

| Extract Data | In the drone project, it is the use case of the data outputs that are created according to the main module and sub-modules (there is no sub-module under these main modules, but can be added in the future) in which the live video recording (live), previously shot video recording and pictures are used. |
| Reporting | In the drone project, the data of the object detection subsystem and the main module and submodules to which it is connected are made into report files/sections by making necessary inquiries. |
| Log | In the drone project, it is the use case of the system created for the purpose of using informative data for the object detection subsystem and the main module and submodules to which it is connected, with the aim of informing the management side of the functions performed in the program while the program is running. |

| Reporting System | |
|---|---|
| **Use Case** | **Description** |
| View Report | Ensuring that all subsystems of the drone project system (main system) and the main modules connected to them and the data groups created by the sub-modules connected to them when they are run are regularly displayed (listed) of the query results created in connection with dms (database management system). |
| Generate Report | Ensuring that all sub-systems of the drone project system (main system) and the main modules connected to them and the data groups created by the sub-modules connected to them when they are run, are regularly created based on dms (database management system). |
| Update Report | Ensuring that all subsystems of the drone project system (main system) and the main modules connected to them and the data groups created by the sub-modules connected to them are regularly updated according to the dms (database management system) query results. |
| Delete Report | Ensuring that all subsystems of the drone project system (main system) and the main |

|  | modules connected to them and the data groups created by the sub-modules connected to them are regularly deleted based on dms (database management system). |
| --- | --- |
| Export Report | All subsystems of the drone project system (main system) and the main modules connected to them and the data groups created by the sub-modules connected to them when they are run are regularly exported (pdf, excel, csv) of the query results created in connection with dms (database management system). ,sql) provisioning. |
| Print Report | All subsystems of the drone project system (main system) and the main modules connected to them and the data groups created by the sub-modules connected to them when they are run, the regular export of the query results created in connection with the dms (database management system) directly via the printer (pdf ,excel,csv,sql) provision. |

| Log System | |
| --- | --- |
| **Use Case** | **Description** |
| View Log | Preparing and displaying (listing) informative data about all subsystems of the drone project system (main system) and the main modules connected to them and the parts of the system created by the sub-modules connected to them, exclusively for the system management (system owners) |
| Generate Log | Preparing informative data about all sub-systems of the drone project system (main system) and the main modules connected to them and the parts of the system created by the sub-modules connected to them when they are run, exclusively for the system management (system owners) by the system |
| Update Log | Updating the informational data about all subsystems of the drone project system (main system) and the main modules connected to them and the parts of the system created by the sub-modules connected to them when they are run, only regular data prepared by the system specifically for the system management (system owners) |
| Delete Log | Deletion of informative data about all subsystems of the drone project system (main system) and the main modules connected to them and the parts of the system created by the sub-modules connected to them when they are run, regular data prepared by the system |

| | only for the system management (system owners) |
|---|---|

# 5.5.  USE CASE SCENARIOS

## 1.

CASE: REGISTER

- As a guest, I want to register A.R Drone 2.0 Website (ARD).

Acceptance Criteria:

- ✓ The website must be open in browser.
- ✓ Username, email, first name, last name and password must be provided.
- ✓ The guest must re-enter the password.

## 2.

CASE: LOGIN

- As a user, I want to login to use ARD Website.
- As a manager, I want to login to admin dashboard in website.

Acceptance Criteria:

- ✓ For a user login, username and password must be match with information guest provided in sign up page.
- ✓ The user must be registered on the website.
- ✓ For a manager, username and password must be match with information in database of the website.

## 3.

CASE: VIEW ACCOUNT

- As a manager, I want to see users account details to manage it.
- As a user service representative, I want to see users account details.

Acceptance Criteria:

- ✓ Fields must be viewable to manage information.
- ✓ The manager should be able to update the account whenever he/she want.
- ✓ The manager should be able to delete the account whenever he/she want.
- ✓ When the manager edits information, then the information should be edited.

## 4.

CASE: RESTORE / CHANGE PASSWORD

- As a user, I want to restore/change my password whenever I see it necessary.

Acceptance Criteria:

- ✓ User must provide new password.
- ✓ When the user edits information, then the information should be edited.

## 5.

CASE: MANAGE PROFILE

- As a user, I want to manage my profile when I see it necessary.

Acceptance Criteria:

- ✓ The areas where the information will be managed should be able to be displayed on the screen.
- ✓ Mandatory and optional fields should be manageable.
- ✓ The information should be able to be changed or deleted whenever the user wants.

## 6.

CASE: REPORTING

- As a manager, I want the authority to report the information received from the website.
- As a user service representative, I want to see reports when its needed.
- As a user, I want to see my reports in website.

Acceptance Criteria:

- ✓ Information required for reporting must be available to the manager for generate a report.
- ✓ Managers and user service representatives should be able to view a report for analyze it.
- ✓ Users should be able to generate and view their own reports.
- ✓ Managers should be able to update a report as needed.
- ✓ If a report no longer needs to be used, it should be deleted by the manager.
- ✓ The manager should be able to export or print a report when its needed.

✓ The user service representative should be able to export or print report for present it to their manager or client.
✓ Export and print reports should be provided for users too.

## 7.

| CASE: LOG |
| --- |

- The drone project system, the step-by-step follow-up of all the operations since the code was run, and the return notification messages for the manager to inform.
- As a manager I want to do operations on log system.

| Acceptance Criteria: |
| --- |

✓ Subsystems and main modules must be accessible to managers, and they should be able to view logs.
✓ After the code starts working, the data can be generated by the software itself, namely by the drone project system.
✓ Updating and deleting logs are very confidential and important operations, so only management should be able to do them.

## 8.

| CASE: CAMERA SHOOTING (LIVE) / UPLOAD IMAGE / UPLOAD VIDEO |
| --- |

- As a guest I want to upload images and videos on website or start live camera shooting.
- As a user I want to upload images or videos on website or start live camera shooting.
- As a manager I want to I want upload images or videos on website or start live camera shooting.

| Acceptance Criteria: |
| --- |

✓ Guests, users, and managers must allow the website to use their computer's web cam.
✓ The website must have the necessary buttons to connect the webcam and upload the images from existing images.
✓ All the uploaded files should be saved in database for face recognition, object recognition and face id login.

## 9.

| CASE: ENTER FACE DETECTION |
| --- |

- As a user service representative I want to check the images uploaded or shouted for face detection.

Acceptance Criteria:

- ✓ It should be checked whether there is any object (mask, glasses) in the user's picture or in the live camera shooting that may prevent face detection.
- ✓ Loaded or shouted images suitable for face detection should get approval from user service representative.

## 10.

CASE: FACE DETECTION

- The drone project system starts the face detection code after all the needed files provided from users or management.

Acceptance Criteria:

- ✓ Users and guests must be started live camera shooting or loaded image for face detection.
- ✓ Since the photos of the managers were uploaded to the face subsystem beforehand, they should be able to do face detection without uploading anything.
- ✓ User service representative can report usage of face subsystem.

## 11.

CASE: FIRST FACE DETECTION

- As a guest I want my face to be detected.
- The drone project system starts the face detection code after all the needed files provided from users or management.

Acceptance Criteria:

- ✓ The guest must open webcam and show his/her face.
- ✓ For guests, the drone project system should make the first face detection and record the face.

## 12.

| CASE: FACE RECOGNATION / FACE ID LOGIN |
|---|

- As a manager I want to login with face id and use face recognition board on website. Also, I want to have control on reports and logs for face subsystem.

- As a user I want to login with face id and use face recognition board on website.

| Acceptance Criteria: |
|---|

- ✓ Managers and users should be able to do face id login on login page by using webcam.
- ✓ Both user and admin dashboards need to provide face recognition boards which it has to be filled with webcam and upload image buttons.
- ✓ Users or managers should be already opened webcam or uploaded their picture for starting face recognition.
- ✓ Only if comparison of pictures and find target face success face recognition will be completed.
- ✓ Managers should be able to control and make operations on reports and log systems.

## 13.

| CASE: COMPARISION OF PICTURES |
|---|

- The drone project system compares the detected faces and makes face recognition according to the success rate.

| Acceptance Criteria: |
|---|

- ✓ All the needed files for comparison must be loaded in the system.

- ✓ Comparison of pictures should be done with recorded pictures and pictures shouted live or uploaded by users or managers.

## 14.

| CASE: FIND TARGET FACE (ATTENDANCE) |
|---|

- The drone project system uses its own submodules (attendance) for face matching.

| Acceptance Criteria: |
|---|

- ✓ The submodules must be added in software.
- ✓ The drone project system should start a search to find a face from the database that matches the face on the picture.

CASE: AGE AND GENDER PERCEPTION

- The drone project system perceives the age and gender of the users and managers.

Acceptance Criteria:

✓ All the needed files for perception must be already loaded in the system.

✓ Face detection must be already succeeded.

✓ Even if the perception of gender and age is an estimation, it should offer a realistic approach.

## 15.

CASE: OBJECT DETECTION

- The drone project system starts the object detection code after all the needed files provided from users or management.

Acceptance Criteria:

✓ Users or managers must be started live camera shooting or loaded image/video for object detection.
✓ The picture or video must have object inside such as car, dog, truck etc.

## 16.

CASE: SORT OBJECT BY CLASS

- After the drone project system starts the object detection code, classification begins.

Acceptance Criteria:

✓ Object detection must be already succeeded.
✓ Classification standards must be predetermined by the system.

## 17.

CASE: OBJECT TRACKER WITH ID FROM VIDEO

- The drone project system uses the standard of tracking by id in uploaded videos.

Acceptance Criteria:

✓ Manager or user must have uploaded video.
✓ Object detection must be already succeeded.
✓ The drone project system should be able to extract data after operation done.

✓ All the needed boards and buttons must be provided on website for users or managers to use object detection.

## 18.

CASE: OBJECT COUNTER BY CLASS FROM VIDEO/IMAGE

- The drone project system uses the standard of counting by class in uploaded images and videos.

Acceptance Criteria:

✓ User or manager must have uploaded a video or image has object inside to count.

✓ Object detection must be already succeeded.

✓ The drone project system should be able to extract data after operation done.
✓ All the needed boards and buttons must be provided on website for users or managers to use object detection.
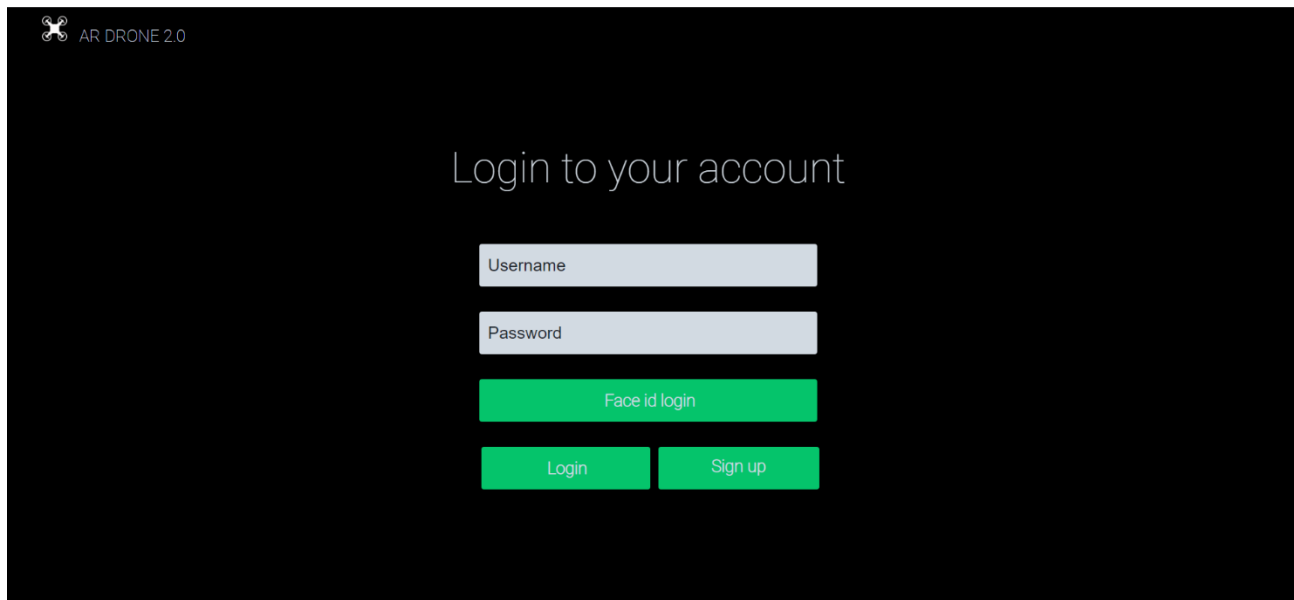
## 6. DESIGN AND IMPLEMENTATION

## Feed back from users

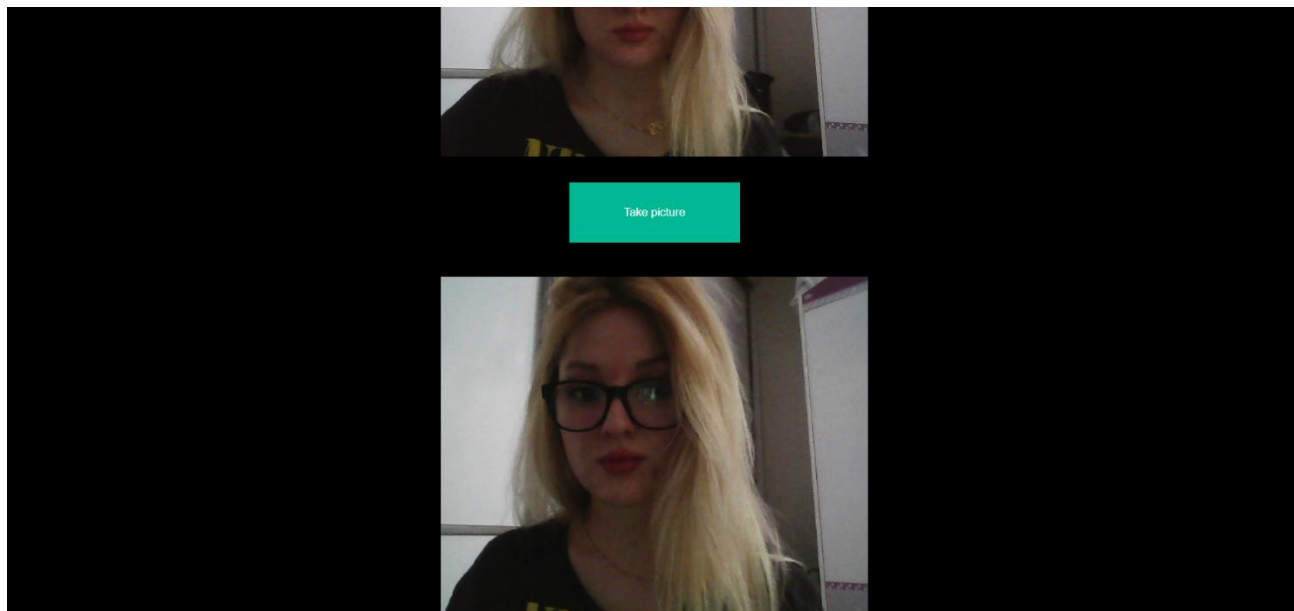| Username | Message | Time |
|----------|---------|------|
| ali | test | 2022-05-20 04:16:27pm |

| Username | Message | Time |
|----------|---------|------|
| osama | test | 2022-05-20 04:15:38pm |

## People seen

| Number of people seen | Location people seen in | Date | Time |
|------------------------|--------------------------|------|------|
| 7 | beach | 2022-05-13 | 08:44:54.0000000 |

| Number of people seen | Location people seen in | Date | Time |
|------------------------|--------------------------|------|------|
| 7 | beach | 2022-05-13 | 08:44:54.0000000 |

| Number of people seen | Location people seen in | Date | Time |
|------------------------|--------------------------|------|------|
| 7 | beach | 2022-05-13 | 08:44:54.0000000 |

## Objects seen

| Number of objects seen | Location objects seen in | Objects name | Date | Time |
|---|---|---|---|---|
| 1 | backgarden | bicycle | 2022-05-13 | 08:24:39.0000000 |

| Number of objects seen | Location objects seen in | Objects name | Date | Time |
|---|---|---|---|---|
| 1 | backgarden | dog | 2022-05-13 | 08:24:39.0000000 |

| Number of objects seen | Location objects seen in | Objects name | Date | Time |
|---|---|---|---|---|
| 1 | backgarden | truck | 2022-05-13 | 08:24:39.0000000 |

| Number of objects seen | Location objects seen in | Objects name | Date | Time |
|---|---|---|---|---|
| 6 | beach | kite | 2022-05-13 | 08:24:39.0000000 |

## Reports

**How many** [Objects ▾] **drone saw between?**
[22.05.2022 📅] [22.05.2022 📅]    [List] [Print / PDF]

**When did drone see** [Objects ▾] **in** [Location ▾]
[List] [Print / PDF]

**Who/What did drone see in** [location ▾] **at** [-- : -- 🕐]
[List] [Print / PDF]

**Where did drone see** [Objects ▾] **between?**
[22.05.2022 📅] [22.05.2022 📅]    [List] [Print / PDF]

## Records analysis

| Uploader name | File name | Date | Time |
|---|---|---|---|
| ela | | 2022-05-17 | 08:44:54.0000000 |

| Uploader name | File name | Date | Time |
|---|---|---|---|
| ela | el2.jpg | 2022-05-17 | 08:44:54.0000000 |

| Uploader name | File name | Date | Time |
|---|---|---|---|
| Osama | db2.PNG | 2022-05-17 | 08:44:54.0000000 |

| Uploader name | File name | Date | Time |
|---|---|---|---|
| ali | db21.PNG | 2022-05-17 | 08:44:54.0000000 |

## My records

| File name | Date | Time |
|---|---|---|
| | 2022-05-17 | 08:44:54.0000000 |

| File name | Date | Time |
|---|---|---|
| el2.jpg | 2022-05-17 | 08:44:54.0000000 |