

# VIRTUALEYE- LIFE GUARD FOR SWIMMING POOLS TO DETECT ACTIVE DROWNING

PROJECT REPORT

*Submitted by*

YUVA RAJA M (913119104124)  
HARI HARAN M (913119104028)  
KISHOR RI (9131191040048)  
SELVA KUMAR M (913119104091)

# **1.INTRODUCTION**

## **1.1 Project Overview**

Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. Swimming pools are found larger in number in hotels, and weekend tourist spots and barely people have them in their house backyard. Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident. Worldwide, drowning produces a higher rate of mortality without causing injury to children. Children under six of their age are found to be suffering the highest drowning mortality rates worldwide. Such kinds of deaths account for the third cause of unplanned death globally, with about 1.2 million cases yearly. To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life. By studying body movement patterns and connecting cameras to artificial intelligence (AI) systems we can devise an underwater pool safety system that reduces the risk of drowning. Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analyzing the video feeds to detect any anomalies. but AS a POC we make use of one camera that streams the video underwater and analyses the position of swimmers to assess the probability of drowning, if it is higher then an alert will be generated to attract lifeguards' attention.

## **1.2 Purpose**

A meticulous system is to be implemented along the swimming pools to save human life. By studying body movement patterns and connecting cameras to artificial intelligence (AI) systems we can devise an underwater pool safety system that reduces the risk of drowning. Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analyzing the video feeds to detect any anomalies. but AS a POC we make use of one camera that streams the video underwater and analyses the position of swimmers to assess the probability of drowning, if it is higher then an alert will be generated to attract lifeguards' attention.

# **2.LITERATURE SURVEY**

## **2.1 Existing Problem**

Swimming pools are found larger in number in hotels, and weekend tourist spots and barely people have them in their house backyard. Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident. Worldwide, drowning produces a higher rate of mortality without causing injury to children. Children under six of their age are found to be suffering the highest drowning mortality rates worldwide. Such kinds of deaths account for the third cause of unplanned death globally, with about 1.2 million cases yearly. To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life. By studying body movement patterns and connecting cameras to

artificial intelligence (AI) systems we can devise an underwater pool safety system that reduces the risk of drowning. Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analyzing the video feeds to detect any anomalies. but AS a POC we make use of one camera that streams the video underwater and analyses the position of swimmers to assess the probability of drowning, if it is higher then an alert will be generated to attract lifeguards' attention.

## **2.2 References**

Hengu Zhu's Fei Lei's, How-Lung Eng

Wai Kit Wongloe How Hui ,Abdel ilah N.Alshbatat

Jia-Xian Jian, Chuin-Mu Wang

Pavithra P, Nandini S,Nanthana A,Noor Tabreen Aslam, Praveen Kumar P

Muhammad Aftab HayatGoutian Yang

## **2.3 Problem Statement Definition**

Virtual Eye - Life Guard for Swimming Pools to Detect Active Drowning

Abstract:

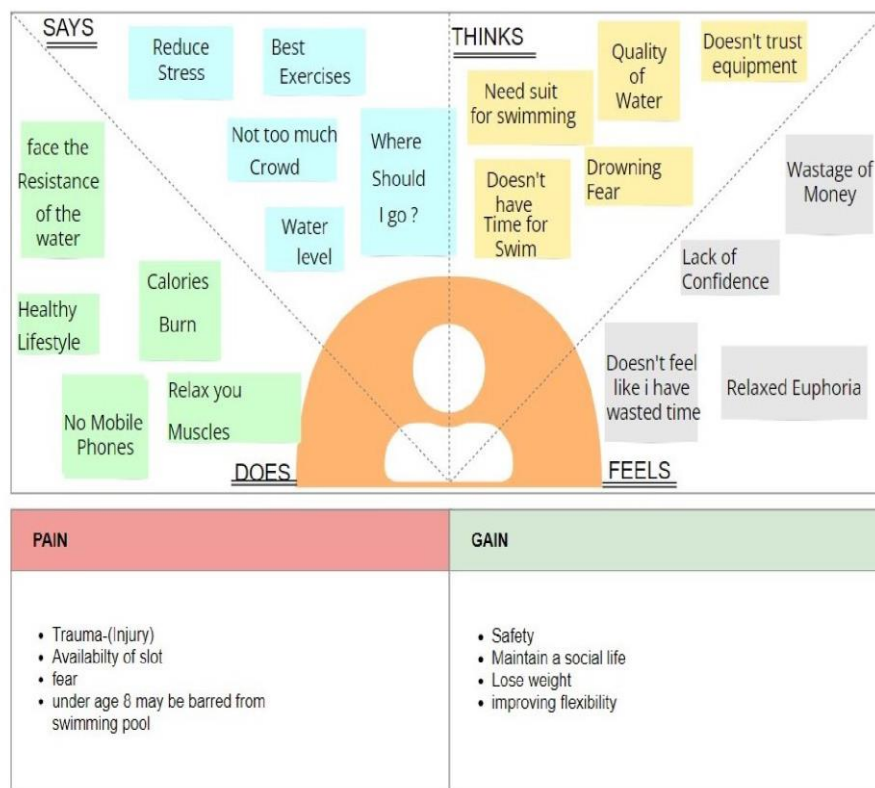
- Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle.
- Swimming pools are found larger in number in hotels, and weekend tourist spots.
- Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident.
- Children under six of their age are found to be suffering the highest drowning mortality rates worldwide.

PROBLEM - 1				
I am <b>Lifeguard</b>	I'm trying to <b>Save the people.</b>	But <b>I can't save those people without prior intimation.</b>	Because <b>There is no detection system.</b>	Which makes me feel <b>Helpless</b>
PROBLEM - 2				
I am <b>Beginner in swimming</b>	I'm trying to <b>Swim on the pool.</b>	But <b>It hesitates me a little.</b>	Because <b>I don't know swimming.</b>	Which makes me feel <b>Panic</b>
PROBLEM - 3				
I am <b>Parent</b>	I'm trying to <b>Get my kid into swimming.</b>	But <b>I can't leave him alone to swim.</b>	Because <b>Drowning is more possible.</b>	Which makes me feel <b>Fear</b>
PROBLEM - 4				
I am <b>Depressed person.</b>	I'm trying to <b>Relax my mind by swimming.</b>	But <b>I can't swim on my own</b>	Because <b>If I accidentally drown.</b>	Which makes me feel <b>Afraid</b>
PROBLEM - 5				
I am <b>Pool owner</b>	I'm trying to <b>Give high security.</b>	But <b>I can't ensure the safety.</b>	Because <b>More likely to drown.</b>	Which makes me feel <b>Pressure</b>

### 3.IDEATION AND PROPOSED SOLUTION

#### 3.1 Empathy Map Canvas

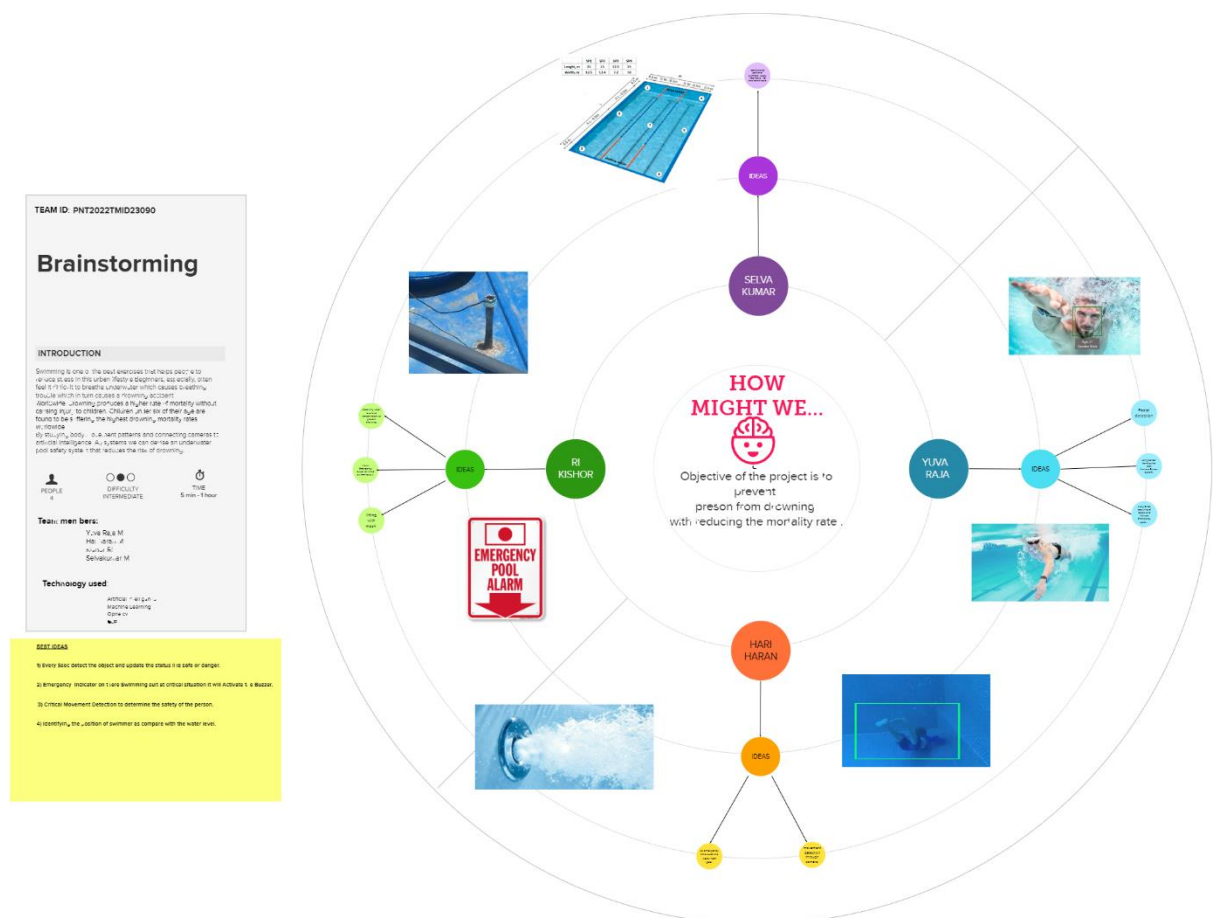
An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community. An Empathy Map consists of four quadrants. The four quadrants reflect four key traits, which the user demonstrated/possessed during the observation/research stage. The four quadrants refer to what the user: Said, Did, Thought, and Felt.



## 3.2 Ideation and Brainstorming

Ideation is the process where you generate ideas and solutions through sessions such as Sketching, Prototyping, Brainstorming, Brainwriting, Worst Possible Idea, and a wealth of other ideation techniques. Ideation is also the third stage in the Design Thinking process

Brainstorming is a method of generating ideas and sharing knowledge to solve a particular commercial or technical problem, in which participants are encouraged to think without interruption. Brainstorming is a group activity where each participant shares their ideas as soon as they come to mind.



### **3.3 Proposed Solution**

Your proposed solution should relate the current situation to a desired result and describe the benefits that will accrue when the desired result is achieved. So, begin your proposed solution by briefly describing this desired result.

#### **Project Design Phase-I Proposed Solution**

Date	24 September 2022
Team ID	<b>PNT2022TMID23090</b>
Project Name	Virtual Eye - Life Guard for Swimming Pools to Detect Active Drowning
Maximum Marks	2 Marks

#### **Proposed Solution:**

<b>S.No.</b>	<b>Parameter</b>	<b>Description</b>
1.	Problem Statement (Problem to be solved)	<p>People are visiting to swimming pools to practice or to learn swimming. There is a possibility of someone drowning as they are new to these activities.</p> <p>So, to detect the active drowning of the person we have designed a “VIRTUAL EYE” program which is installed in the Security Camera Available in the Pool and it is connected with Alarm and thus alerting the Rescue team about the drowning.</p> <p>Thus, a system is to be implemented along the swimming pools to save human life.</p>

### 3.4 Problem Solution Fit

Problem-solution fit is a term used to describe the point validating that the base problem resulting in a business idea really exists and the proposed solution actually solves that problem.

#### Project Design Phase-I - Solution Fit Template

Project Title: VIRTUAL EYE -LIFE GUARD SWIMMING POOLS TO DETECT ACTIVE DROWNING

Team ID: PNT2022TMD23090

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> <ul style="list-style-type: none"><li>➤ The person who swims in a pool are going to Monitor</li><li>➤ Based on Visual monitoring system.</li><li>➤ Every Frames are going to be monitored by trained model.</li></ul>	<b>6. CUSTOMER CONSTRAINTS</b> <span>CC</span> <ul style="list-style-type: none"><li>➤ Maintenance and cost of products</li><li>➤ Constant network connection</li><li>➤ Misunderstanding the behavior of swimmers</li></ul>	<b>5. AVAILABLE SOLUTIONS</b> <span>AS</span> <ul style="list-style-type: none"><li>➤ By installing camera and detect each and every person's actions in the pool and if any abnormal action is detected then alarm will notify the lifeguard.</li><li>➤ Detects and prevents active drowning.</li></ul>	Explore AS, differentiate
	<b>2. PROBLEMS / PAINS</b> <span>J&amp;P</span> <ul style="list-style-type: none"><li>➤ People visit to swimming pools to practice or learn swimming.</li><li>➤ As a beginner there is a lot of chance of drowning as they are new to these activities.</li><li>➤ Such activities may lead to increase in mortality rate in world wide.</li></ul>	<b>9. PROBLEM ROOT CAUSE</b> <span>RC</span> <ul style="list-style-type: none"><li>➤ People think that the camera that is set up to monitor the persons who are swimming are of no proper and accurate use.</li></ul>	<b>7. BEHAVIOUR</b> <span>BE</span> <ul style="list-style-type: none"><li>➤ People were trusting more in a manual monitoring system rather than visual monitoring system.</li><li>➤ They want to be safe while learning so always want to be surrounded by a lifeguard.</li></ul>	
	Focus on J&P, tap into BE, understand RC			

## **4.REQUIREMENT ANALYSIS**

### **4.1 Functional Requirement**

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it's important to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behaviour under specific conditions.

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration Via Email Registration Via phone number
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP Create and store the data
FR-3	Alarm system	Monitor and detect the drowning person Alert the lifeguard by trigger the alarm
FR-4	Output	Visual representation Image detection Report generation

### **4.2 Non-Functional Requirements**

Non functional requirements are requirements that define ‘how’ the app must perform a certain function. In essence, they are the quality attributes of an app that define the user experience of the app. They are also known as non-behavioral requirements and are to be implemented according to their priority to the app function. This makes them flexible to an extent, making it possible to skip a few in case of time, budget or technology constraints.

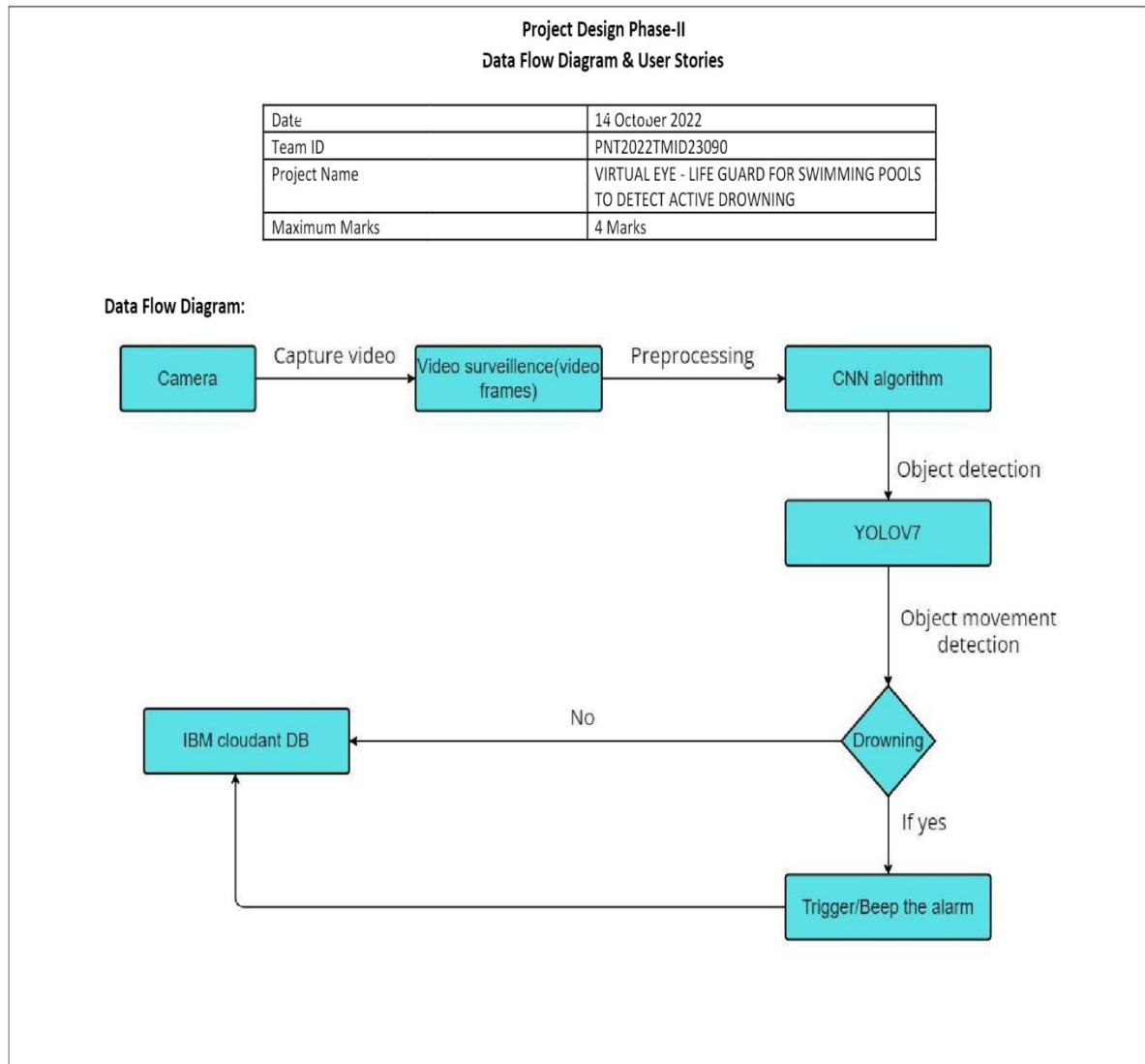
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	Eco – Friendly.
NFR-2	<b>Security</b>	Observing each and every body movement of the swimmers.
NFR-3	<b>Reliability</b>	Suitable for all the swimming pools.



## 5.PROJECT DESIGN

### 5.1 Data Flow Diagram

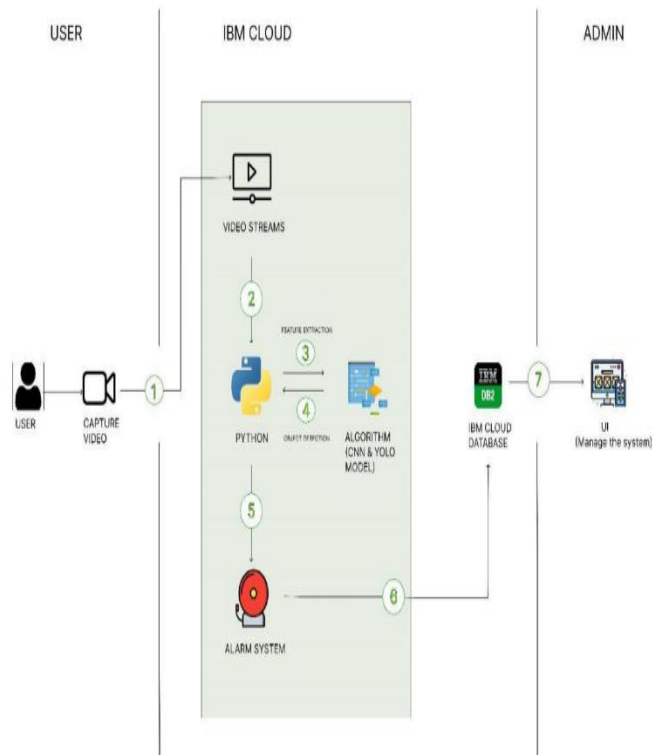


## 5.2 Solution and Technical Architecture

### Project Design Phase-II Technology Stack (Architecture & Stack)

Date	14 October 2022
Team ID	PNT2022TMID23090
Project Name	VIRTUAL EYE - LIFE GUARD FOR SWIMMING POOLS TO DETECT ACTIVE DROWNING
Maximum Marks	4 Marks

#### Technical Architecture:



### 5.3 User Stories

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	I can be able to Register with My phone number but only one registration for single number.	1	High	Selva Kumar M
		USN-2	I will receive confirmation OTP on the registered phone number.	2	Medium	M yuva raja
		USN-3	I can also register Through Gmail	2	Low	Hari Haran
	Login	USN-4	I can login into the application by entering phone number & password.	1	High	M yuva raja
		USN-5	In prediction page, the data uploaded will help the user to detect the drowning movements	2	Medium	R I Kishor

Sprint-1	Dataset collection	USN-6	The dataset collected will give high accuracy on the drowning details.	2	High	Hari Haran M
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Cloudant DB	USN-7	Create DB	4	High	M yuva raja
	Train the model	USN-8	We will train the model.	8	High	R I Kishor
		USN-9	We will test the model.	6	High	Selva Kumar M
Sprint-3	Detection	USN-10	The tested model will be loaded.	3	High	Hari Haran M
		USN-11	To identify the person by collecting real-time data.	5	Medium	M yuva raja
		USN-12	The data collected at present is checked with the pre-fed data.	8	High	Selva Kumar M
Sprint-4	Alert	USN-13	When the abnormal movement is detected the system will ring an alarm to notify the lifeguard to rescue the person.	7	High	Hari Haran M
		USN-14	We will be able to detect the drowning person.	3	Medium	R I Kishor

Sprint-4	Logout	USN-15	User can logout of the application.	2	Low	M yuva raja
----------	--------	--------	-------------------------------------	---	-----	-------------

#### Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)
Sprint-1	10	6 Days	24 Oct 2022	Oct 2022
Sprint-2	18	6 Days	31 Oct 2022	Nov 2022
Sprint-3	16	6 Days	07 Nov 2022	Nov 2022
Sprint-4	12	6 Days	14 Nov 2022	Nov 2022

## **6 Project planning & scheduling**

### **6.1 Sprint Planning & Estimation**

#### **PROJECT PLANNING PHASE** **MILESTONE & ACTIVITY LIST**

Date:	1 Nov 2022
Team id:	PNT2022TMID23090
Project Name:	Virtual Eye - LifeGuard For Swimming Pools To Detect Active Drowning
Maximum Mark:	4 marks

S.NO	MILESTONE	DESCRIPTION	DURATION
1	Prerequisites	It is very important for every execution of different phase of project	1WEEK
2	Create & Configure IBM cloud services	It provides solutions that enable higher levels security, management and having methods for rapid delivery for running critical workloads.	2WEEK
3	Develop the python script	A Python script is a set of commands included in a file that is intended to be run similarly to a program. The concept is that the file will be run or performed from the command line or from within a Python interactive shell to perform a particular	1WEEK

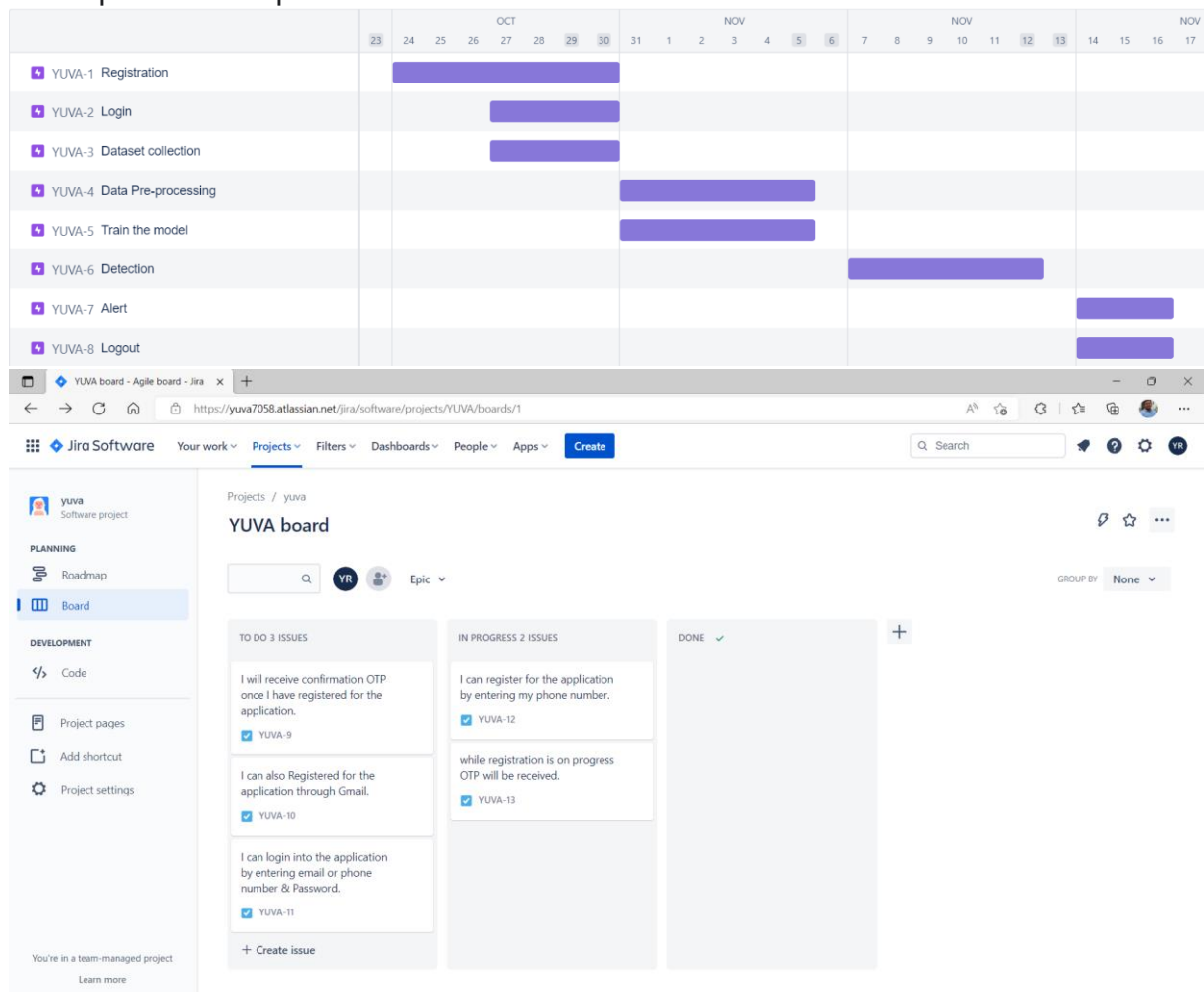
## 6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)
Sprint-1	10	6 Days	24 Oct 2022	Oct 2022
Sprint-2	18	6 Days	31 Oct 2022	Nov 2022
Sprint-3	16	6 Days	07 Nov 2022	Nov 2022
Sprint-4	12	6 Days	14 Nov 2022	Nov 2022

## 6.3 Reports From JIRA

Jira helps teams plan, assign, track, report, and manage work and brings teams together for everything from agile software development and customer support to start-ups and enterprises.



## **7.CODING AND SOLUTIONING**

### **(Explain the features added in the project along with code)**

A meticulous system is to be implemented along the swimming pools to save human life. By studying body movement patterns and connecting cameras to artificial intelligence (AI) systems we can devise an underwater pool safety system that reduces the risk of drowning. Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analysing the video feeds to detect any anomalies. but AS a POC we make use of one camera that streams the video underwater and analyses the position of swimmers to assess the probability of drowning, if it is higher, then an alert will be generated to attract lifeguards' attention.

#### **FEATURE 1:**

##### **LOGIN**

Algorithm:

1. Enter the credentials and hit enter (email and password).
2. If already logged in user is taken to home page
3. If wrong credentials enter, notification displayed to user and user stays in login page.
4. On correct credentials, user is taken to home page.

##### **SIGNUP**

Algorithm:

1. Enter the signup form fields (name, email, password, re-enter password, date of birth) and hit enter.
2. All credentials are validated at client side.
3. Email is checked if already registered or not in the database.
4. If already registered, notification displayed. Or else, the user is taken to the successful signup page.

#### **FEATURE 2:**

##### **DETECT DROWNING**

- Algorithm:
1. Detect a object using yolo model
  2. Check whether its person or not
  3. If it's a person, check if person moving or not
  4. If not moving for 10 secs, alert the life guard

##### **Code:**

```
import cvlib as cv
from cvlib.object_detection import draw_bbox
import cv2
import time
import numpy as np

webcam = cv2.VideoCapture(0)

if not webcam.isOpened():
    print("Could not open webcam")
```

```

    exit()
t0 = time.time()

centre0 = np.zeros(2)
isDrowning = False
s
while webcam.isOpened():

    status, frame = webcam.read()

    if not status:
        print("Could not read frame")
        exit()

    bbox, label, conf = cv.detect_common_objects(frame)

    if(len(bbox)>0):
        bbox0 = bbox[0]

        centre = [0,0]

        centre = [(bbox0[0]+bbox0[2])/2,(bbox0[1]+bbox0[3])/2 ]

        hmov = abs(centre[0]-centre0[0])
        vmov = abs(centre[1]-centre0[1])

        x=time.time()

        threshold = 10
        if(hmov>threshold or vmov>threshold):
            print(x-t0, 's')
            t0 = time.time()
            isDrowning = False

        else:

            print(x-t0, 's')
            if((time.time() - t0) > 10):
                isDrowning = True

        print('bbox: ', bbox, 'centre:', centre, 'centre0:', centre0)
        print('Is he drowning: ', isDrowning)

        centre0 = centre

    out = draw_bbox(frame, bbox, label, conf,isDrowning)
    cv2.imshow("Real-time object detection", out)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
webcam.release()
cv2.destroyAllWindows()

```

## **8.TESTING**

### **8.1 Test cases**

1. Login button click with wrong credentials entered.
2. Signup with already registered mail ID.
3. Signup with wrong form data entered.
4. Entering home page with logged out session.
5. Clicking home page buttons with logged out session.
6. Invalid data entered in change password page and requested for change in password.

### **8.2 User Acceptance Testing**

S.NO	TEST CASE	REQUIRED OUTPUT	RESULT OUTPUT	STATUS
1	Login button click with wrong credentials	Wrong credentials entered notification	Wrong credentials entered notification	ACCEPTED
2	Signup with already registered mail ID.	Email already registered notification	Email already registered notification	ACCEPTED
3	Signup with wrong form data entered.	Wrong credentials entered notification	Wrong credentials entered notification	ACCEPTED
4	Entering home page with logged out session.	Take user to login page	Take user to login page	ACCEPTED
5	Clicking home page buttons with logged out session.	Take user to login page	Take user to login page	ACCEPTED
6	Invalid data entered in change password page and requested for change in password.	Wrong form data entered notification	Wrong form data entered notification	ACCEPTED



## 9.RESULTS

### 9.1Performance Metrics

Test Case ID	Feature Type		Test Scenario	Steps TO execute	Test	Expected Result	Actual Result
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	1. Enter URL and click go 2. Click on My Account dropdown button 3. Verify login/Signup popup displayed or not	Login.html	Login/Signup popup should display	Working as expected
LoginPage_TC_002		Home page	Verify the UI elements in Login/Signup popup	1. Enter URL and click go 2. Click on My Account dropdown 3. Verify login/Signup popup with below UI elements. a. email text box b. password text box c. Login button d. New customer. Create account link e. Last password? Recovery password link	Login.html	Application should show below elements: a. email text box b. password text box c. Login button with orange colour d. New customer? Create account link e. Last password? Recovery password link	Working as expected
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials	1. Enter URL and click go 2. Click on My Account dropdown 3. Enter Valid username/email in Email text 4. Enter valid password in password text box 5. Click On login button	Username: lax@gmail password: lax26	User should navigate to prediction homepage	working as expected
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials	1. Enter URL and click go 2. Click on My Account dropdown button 3. Enter Invalid username/email in Email text box 4. Enter valid password in password text box 5. Click on login button	Username: lax password: lax26	Application should show 'Incorrect email or password' validation message.	working as expected
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials	1. Enter URL and click go 2. Click On My Account dropdown 3. Enter Valid username/email in Email text box 4. Enter Invalid password in password text box 5. Click on login button	username: lax26@mail password: lax26	Application should show 'Incorrect email or password' validation message.	working as expected
LoginPage_TC_005	Functional	Login page	Verify user is able to log into application with Invalid credentials	1. Enter URL and click go 2. Click on My Account dropdown 3. Enter Invalid username/email in Email text box 4. Enter Invalid password in password text box 5. Click on login button	username: lax26@mail password: 1803	Application should show 'Incorrect email or password' validation message.	working as expected
PredictionPage_TC_006	Functional	Prediction Page	Page should display whether the person is drowning or not	1. Camera should take pictures of people swimming in pools 2. It should predict the probability of drowning 3. It should show a bounding box displaying the probability of drowning	image Of people drowning	generate a alert to lifeguard if people are drowning	Working as expected

## **10.ADVANTAGES AND DISADVANTAGES**

### **ADVANTAGES**

Safety first. Active drowning detection.

Position and image of the drowning. When it comes to swimmers in trouble, every second counts.

Recording of events.

An additional level of security.

### **DISADVANTAGES**

Designed for whom has to guarantee every day the safety in public and intensive use pools, this life guard detects potential drownings and promptly notifies you. It features the latest artificial intelligence technology and adapts to the needs of the user. It's the ultimate drowning detection system for those who demand the ultimate in safety.

## **11.CONCLUSION**

Consistently numerous people, including kids, are suffocated or near suffocating in the deeps of the swimming pools, and the lifeguards are not prepared all around to deal with these issues. In this manner raises the necessities for having a framework that will thus recognize the suffocating people and alert the lifeguards at such hazard. It can be installed in International standardized schools where classes are held for training kids.

## **12.FUTURE SCOPE**

- 1) Test results show that the mean precision rate of drowning is 94.62%, the mean false rate is 1.43%, and the mean missing rate is 3.57%.
- 2) The mean precision rate of swimming is 97.86%, the mean false rate is 7.93%, the mean missing rate is 5.93%, and the average frame rate is 33f/s.
- 3) This study will attempt to identify superiority in trained lifeguards through the use of videoed pool swimming scenarios that vary in set size.

## **13.APPENDIX**

### **Source Code:**

```
import necessary packages
import cv2
import os
import numpy as np
from .utils import download_file
```

```
initialize = True
net = None
dest_dir = os.path.expanduser('~') + os.path.sep + '.cvlib' + os.path.sep
+ 'object_detection' + os.path.sep + 'yolo' + os.path.sep + 'yolov3'
classes = None
#colors are BGR instead of RGB in python
COLORS = [0,0,255], [255,0,0]
```

```
def populate_class_labels():
```

```
    #we are using a pre existent classifier which is more reliable and more
    efficient than one
```

```
    #we could make using only a laptop
```

```
    #The classifier should be downloaded automatically when you run this
    script
```

```
    class_file_name = 'yolov3_classes.txt'
```

```
    class_file_abs_path = dest_dir + os.path.sep + class_file_name
```

```
    url = 'https://github.com/Nico31415/Drowning-
    Detector/raw/master/yolov3.txt'
```

```
    if not os.path.exists(class_file_abs_path):
```

```
        download_file(url=url, file_name=class_file_name,
        dest_dir=dest_dir)
```

```
        f = open(class_file_abs_path, 'r')
```

```
        classes = [line.strip() for line in f.readlines()]
```

```
    return classes
```

```
def get_output_layers(net):
```

```
#the number of output layers in a neural network is the number of possible
```

```
#things the network can detect, such as a person, a dog, a tie, a phone...
```

```
layer_names = net.getLayerNames()
```

```
output_layers = [layer_names[i - 1] for i in net.getUnconnectedOutLayers()]
```

```
return output_layers
```

```
def draw_bbox(img, bbox, labels, confidence, Drowning, write_conf=False):
```

```
    global COLORS
```

```
    global classes
```

```
    if classes is None:
```

```
        classes = populate_class_labels()
```

```
    for i, label in enumerate(labels):
```

```
        #if the person is drowning, the box will be drawn red instead of blue
```

```
        if label == 'person' and Drowning:
```

```
            color = COLORS[0]
```

```
            label = 'DROWNING'
```

```
        else:
```

```
            color = COLORS[1]
```

```
        if write_conf:
```

```
            label += ' ' + str(format(confidence[i] * 100, '.2f')) + '%'
```

```
        #you only need to points (the opposite corners) to draw a rectangle. These points
```

```
        #are stored in the variable bbox
```

```
        cv2.rectangle(img, (bbox[i][0],bbox[i][1]), (bbox[i][2],bbox[i][3]), color, 2)
```

```
        cv2.putText(img, label, (bbox[i][0],bbox[i][1]-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
```

```

return img

def detect_common_objects(image, confidence=0.5, nms_thresh=0.3):

    Height, Width = image.shape[:2]
    scale = 0.00392

    global classes
    global dest_dir

    #all the weights and the neural network algorithm are already
    preconfigured
    #as we are using YOLO

    #this part of the script just downloads the YOLO files
    config_file_name = 'yolov3.cfg'
    config_file_abs_path = dest_dir + os.path.sep + config_file_name

    weights_file_name = 'yolov3.weights'
    weights_file_abs_path = dest_dir + os.path.sep + weights_file_name

    url = 'https://github.com/Nico31415/Drowning-
    Detector/raw/master/yolov3.cfg'

    if not os.path.exists(config_file_abs_path):
        download_file(url=url, file_name=config_file_name,
        dest_dir=dest_dir)

    url = 'https://pjreddie.com/media/files/yolov3.weights'

    if not os.path.exists(weights_file_abs_path):
        download_file(url=url, file_name=weights_file_name,
        dest_dir=dest_dir)

    global initialize
    global net

    if initialize:
        classes = populate_class_labels()

```

```

net = cv2.dnn.readNet(weights_file_abs_path,
config_file_abs_path)
initialize = False

blob = cv2.dnn.blobFromImage(image, scale, (416,416), (0,0,0), True,
crop=False)

net.setInput(blob)

outs = net.forward(get_output_layers(net))

class_ids = []
confidences = []
boxes = []

for out in outs:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        max_conf = scores[class_id]
        if max_conf > confidence:
            center_x = int(detection[0] * Width)
            center_y = int(detection[1] * Height)
            w = int(detection[2] * Width)
            h = int(detection[3] * Height)
            x = center_x - w / 2
            y = center_y - h / 2
            class_ids.append(class_id)
            confidences.append(float(max_conf))
            boxes.append([x, y, w, h])

indices = cv2.dnn.NMSBoxes(boxes, confidences, confidence,
nms_thresh)

bbox = []
label = []
conf = []

for i in indices:
    i = indices[i]
    box = boxes[i]
    x = box[0]

```

```
y = box[1]
w = box[2] h = box[3]
bbox.append([round(x), round(y), round(x+w), round(y+h)])
label.append(str(classes[class_ids[i]]))
conf.append(confidences[i])
return bbox, label, conf
```

## **INTERNET OF THINGS:**

The Internet of things (IoT) describes physical objects (or groups of such objects) with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks. Internet of things has been considered a misnomer because devices do not need to be connected to the public internet, they only need to be connected to a network and be individually addressable. The field has evolved due to the convergence of multiple technologies, including ubiquitous computing, commodity sensors, increasingly powerful embedded systems, and machine learning. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), independently and collectively enable the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems, cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. IoT is also used in healthcare systems. There are a number of concerns about the risks in the growth of IoT technologies and products, especially in the areas of privacy and security, and consequently, industry and governmental moves to address these concerns have begun, including the development of international and local standards, guidelines, and regulatory frameworks.

## **IBM WATSON IOT PLATFORM:**

IBM Watson IoT Platform for Bluemix provides a versatile toolkit that includes gateway devices, device management, and powerful application access. By using Watson IoT Platform, you can collect connected device data and perform analytics on real-time data. The IBM Watson IoT Platform is a fully managed, Cloud-hosted service that provides device management capabilities as well as data 36 collection and management in a time series format. As part of IBM's Platform as a Service offering, IBM Bluemix, you can use the IBM Watson IoT Platform to rapidly build IoT apps from the catalog of services available in IBM Bluemix. You can choose from such IoT app options as storage services, rules, analytics services, stream analytics, machine learning, visualization, and user apps (Web or mobile). You also can embed cognitive capabilities in your IoT apps by using IBM Watson services available in IBM Bluemix

## **SOURCE CODE LINK:**

[IBM-EPBL/IBM-Project-19730-1659705250: VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning \(github.com\)](https://github.com/IBM-EPBL/IBM-Project-19730-1659705250-VirtualEye-LifeGuardforSwimmingPools-to-DetectActiveDrowning)