PROJECT
REPORT

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

1. INTRODUCTION

1.1 Overview

In Modern metropolitan lifestyle,swimming in one of the best activity for stress reduction .Worldwide,drowning results in a higher mortality rate without harming children .The highest global drowning fatality rates are observed to be among children under the age of six.With around 1.2 million incident each year .These types of deaths rank third among all unexpected deaths worldwide

1.2 Purpose

The aim this project is to design a meticulous system which can be implemented among the swimming pools to save human life.An alarm will be issued to call he attention of the lifeguards .When the video is being streamed underwater and swimmer position is the being examined to determine the likehood of drowning .This will help in the reducing drowning .This will help in reduncing drowning rates and create a self environment

2. LITERATURE SURVEY

2.1 Existing Problem

Beginners, in particular, frequently struggle to breathe underwater, resulting in respiratory issues and, eventually, a drowning disaster. Drowning causes a higher mortality rate worldwide while causing no harm to children. Children under the age of six are found to have the highest global drowning fatality rates. These types of deaths rank third among all unexpected deaths worldwide, with approximately 1.2 million incident each year.

2.2 Reference

NAME OF THE PAPER: A Survey of Drowning Detection Techniques.
NAME OF THE AUTHOR: Nagato Konishi ,Yoh Ishigaki ,Seizi Iinuma ,Tsubasa Nakada,Taisuke Hoshino ,Wataru Nemoto ,Kazunori Ohkawara.

JOURNAL PUBLISHED: 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC).
MONTH AND YEAR PUBLISHED: 09 June 2021.
OBJECTIVE OF THE PROJECT: To track swimmers in a pool using machine learning techniques and prevent drowning accidents.
TECHNOLOGY USED: The concepts of image and video processing are used along with machine learning paradigms.
NAME OF THE PAPER: Automated Vision-based Surveillance System to Detect Drowning Incidents in Swimming Pools.
NAME OF THE AUTHOR: Abdel llah N. Alshbatat,Shamma Alhameli ,Shamsa Almazrouei
Salama Alhameli ,Wadhha Almarar.
JOURNAL PUBLISHED : 2020 Advances in Science and Engineering Technology International Conferences (ASET).
MONTH AND YEAR PUBLISHED : 16 June 2020
OBJECTIVE OF THE PROJECT : To track swimmers in a pool using machine learning techniques and prevent drowning accidents.
TECHNOLOGY USED : Raspberry Pi with the Raspbian operating system,based on the color based algorithm.

2.3 Problem Statement definition

Swimming is a great urban stress-reliever. Hotels and tourist spots have more swimming pools than private homes. Beginners have trouble breathing underwater, causing breathing problems and drowning.

Drowning increases global mortality without harming children. Under-6-year-olds have the highest drowning mortality rates globally.These deaths are the third cause of unplanned death globally, with 1.2 million cases yearly.

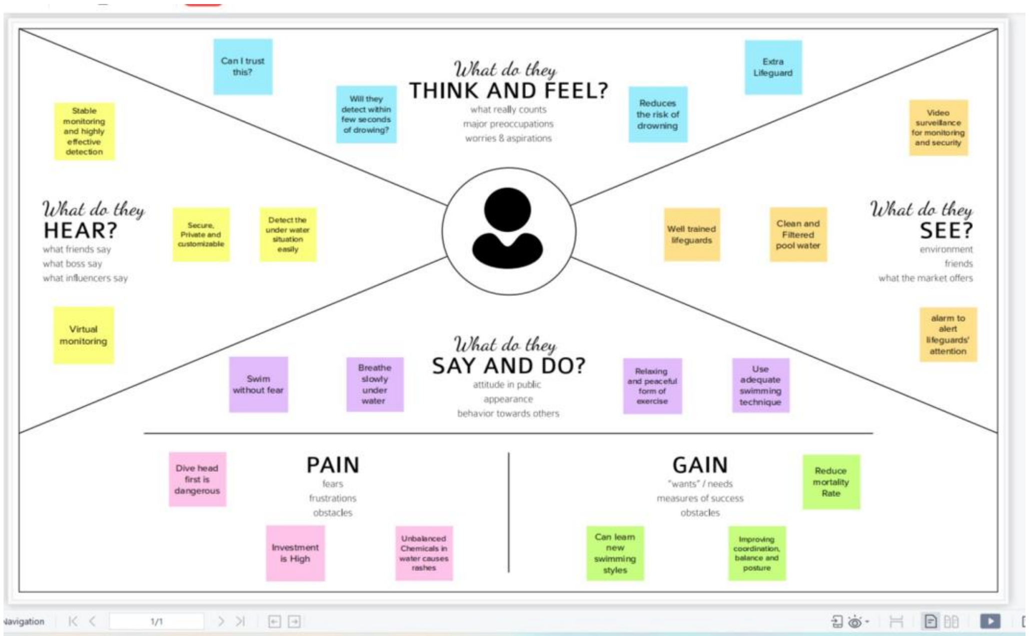
To save lives, a meticulous system must be implemented along swimming pools. By studying body movement patterns and connecting cameras to AI System ,We can create a safer underwater pool.

Installing 16 underwater and ceiling cameras and analysing video feeds can create such systems.

As a POC,we use one camera that streams underwater video and analyses swimmers' positions to assess drowning risk; if it's high, an alert is generated to alert lifeguards.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map canvas


What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Can I trust this?
Will they detect within few seconds of drowing?
Reduces the risk of drowning
Extra Lifeguard
What do they
HEAR?
what friends say
what boss say
what influencers say
Stable monitoring and highly effective detection
Secure, Private and customizable
Detect the under water situation easily
Virtual monitoring
What do they
SEE?
environment
friends
what the market offers
Video surveillance for monitoring and security
Well trained lifeguards
Clean and Filtered pool water
alarm to alert lifeguards' attention
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Swim without fear
Breathe slowly under water
Relaxing and peaceful form of exercise
Use adequate swimming technique
PAIN
fears
frustrations
obstacles
Dive head first is dangerous
Investment is High
Unbalanced Chemicals in water causes rashes
GAIN
"wants" / needs
measures of success
obstacles
Reduce mortality Rate
Can learn new swimming styles
Improving coordination, balance and posture
Navigation
1/1

3.2 Ideation & Brainstorming

3.3 Proposed Solution

SNO	Parameter	Description
1	Problem Statement (Problem to be solved)	Drowning detection system that detects every dangerous situation and accident. This software works in close integration with the cameras installed in the pool to continuously scan the pool. This system can also able to record all the activities in the pools and to classify critical situations from normal ones in order to keep track of what happened. The built-in notification system produces alarms within 10 seconds on onsmartwatches, phones, flashing lights and other configurable devices. Thus 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 a poolsafety system that reduces the risk of drowning
2	Idea / Solution description	This system by analyzing the movement and shape, evaluates swimmers' condition based on visual based monitoring device and analarm to alert the lifeguards and provides solution in detecting drowning incidents.
3	Novelty / Uniqueness	Virtual eye has developed a novel idea of alerting the ambulance and another life guard if there is any delay in saving the person to death

4	Social Impact / Customer Satisfaction	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, 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.
5	Business Model (Revenue Model)	There are many products currently available this regard. Our solution, once developed well, has enough possibility to become a good product to save drowning victims.
6	Scalability of the Solution	Our proposed solution is very scalable i.e., in future, there are a lot of rooms for evolving our present model by Adding new features to enhance our system in the future.

4. REQUIREMENT ANALYSIS

4.1 Functional requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Installation	Needed to be fixed under the water without Creating any disturbance to the people in the swimming pool.
FR-2	User registration	Register via Email/Phone number andget verified for further use
FR-3	Deduction	Either not moving or in unconscious state
FR-4	Support	Take swim tubes or take the help of rescuer
FR-5	Alert	Set alarm and send message throughthe application to life guard.
FR-6	Output	Vision based monitor Image, position and movement detection Drowning is detected Recue drowning people by LifeGuard

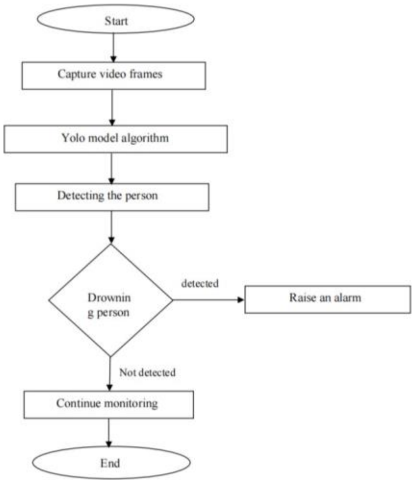
4.2 Non-Functional Requirement

S No.	**Non-Functional Requirement**	**Description**
1	**Usability**	To ensure the safety of each and every person present in the pool. A Lifeguard should be present all the time in the pool.
2	**Security**	Lifeguards should be aware of the alertmessage to save the life of the swimmer
3	**Reliability**	Virtual eye lifeguard triggers an immediate prior alarm if a swimmer is in peril, helping to avoid panic even in critical situations.

4	**Performance**	The alarm is triggered when the swimmer is detected as drowning
5	**Availability**	Equipment and accessories include lifesaverrings, inflatable vests, aShepherd's Crook, life hooks, spine boards, rescue tubes, and a first aid kit. Remember to keep them accessible to quickly pull someone from the water water safely
6	**Scalability**	Virtual eye lifeguard detects potentialdrownings and promptly notifies you. It features the latest artificial intelligence technology and adapts to the needs of the user.

5. PROJECT DEESIGN

5.1 Data Flow Diagram



5.2 Solution & Technical Architecture

5.3 User Stories

User Type	**Functional Requirement**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
Customer (Supervisor)	Installation	USN-1	Theyset camera and install and configure the system in swimming pools	The software is installed and cameras are setup	High	Sprint-1
	Pre-processing	USN-2	Train and test the model	Trainthe model byusing datasets	High	Sprint-1
	Detection of drowning	USN-3	The swimmers can monitored by cameras	Camera surveillance	High	Sprint -2
		USN-4	Swimmers can be detected through the actions Detection of drowning	Detection of drowning	High	Sprint-2
	Alarm rings	USN -5	Alarm rings When the system detects drowning person	Alert the lifeguard	High	Sprint-3

User Type	Functional Requirements	User Story Number	User story/Task	Acceptance criteria	Priority	Release
Lifeguard	Saves the person	USN-6	The Lifeguard saves the swimmer who isdrowning once the alarm rings	Saves the lifeof people	High	Sprint-3
Administrator	Register	USN-7	Register into the application	Admin can access the account	Medium	Sprint-2
	Login	USN-8	Login and manage the application	Managesystem	Medium	Sprint-2
		USN-9	Stores the database	Storage the database	Medium	Sprint-2

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Krishna Kumar S
Sprint-1	Registration	USN-2	As a user, I wil receive confirmation email once I have registered for the application	1	High	Subash V
Sprint-1	Registration	USN-3	As a user, can register for the application through through Facebook	2	Low	Kamesh H
Sprint-1	Registration	USN-4	As a user, can register for the application through Gmail	2	Medium	Mahesh G
Sprint-1	Login	USN-5	As a user, can log into the application by entering email & password	1	High	Subash V

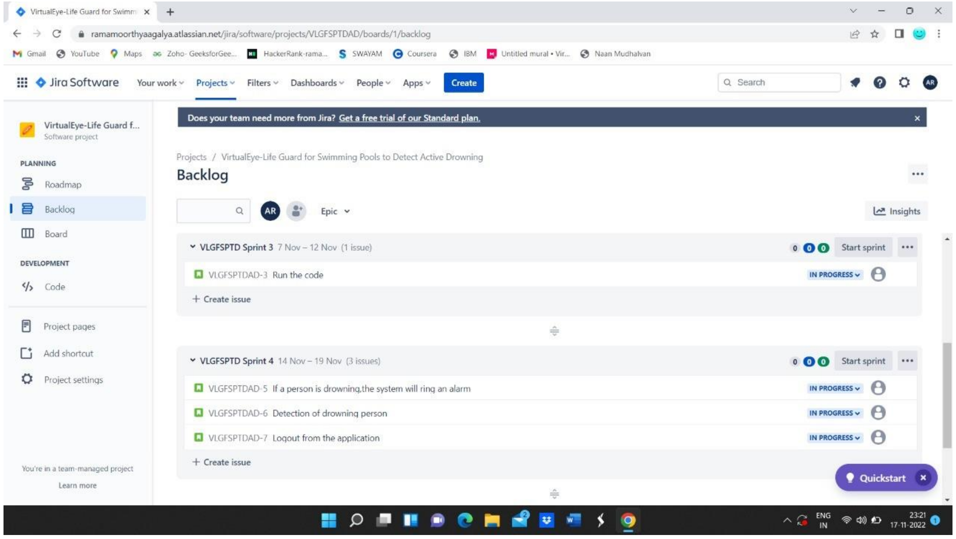
Sprint-2	Dataset Collect	USN-6	Collect number of datasets and get accuracy	2	Medium	Kamesh H
Sprint-2	Pre-processing	USN-7	The dataset is extracted	2	High	Subash V
Sprint-2	Train the model	USN-8	Train the model.	4	High	Mahesh G
Sprint-2	Test the model	USN-9	Test the model	6	High	Krishna Kumar S
Sprint-3	Detection	USN-10	Load the trained model.	3	High	Subash V
Sprint-3	Detection	USN-11	Identify the person by collecting real-time data through a webcam.	5	Medium	Kamesh H
Sprint-3	Detection	USN-12	classify it by usinga trained model to predict the output	8	High	Subash V
Sprint-4	Detection	USN-13	If person is drown the system will ring alarm to give signal	7	High	Mahesh G
Sprint-4	Detection	USN-14	As a User, I can detect the drowning person.	3	Medium	Krishna Kumar S
Sprint-4	Logout	USN-15	As a User, I can logout application	2	Low	Subash V

6.2 Sprint Delivery Schedule

Sprint	Total Story Point	Duration	Sprint Start Date	Sprint End Date(Planned)	Story point Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	8	6 Days	24 Oct 2022	29 Oct 2022	6	29 Oct 2022
Sprint-2	14	6 Days	31 Oct 2022	05 Nov 2022	12	05 Nov 2022
Sprint-3	16	6 Days	07 Nov 2022	12 Nov 2022	11	12 Nov 2022
Sprint-4	12	6 Days	14 Nov 2022	19 Nov 2022	12	19 Nov 2022

6.3 Reports from JIRA

Screen Shot 1:



Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / VirtualEye-Life Guard for Swimming Pools to Detect Active Drowning

Backlog

VLGFSPTD Sprint 3 7 Nov – 12 Nov (1 issue) 0 0 0

VLGFSPTDAD-3	Run the code	IN PROGRESS

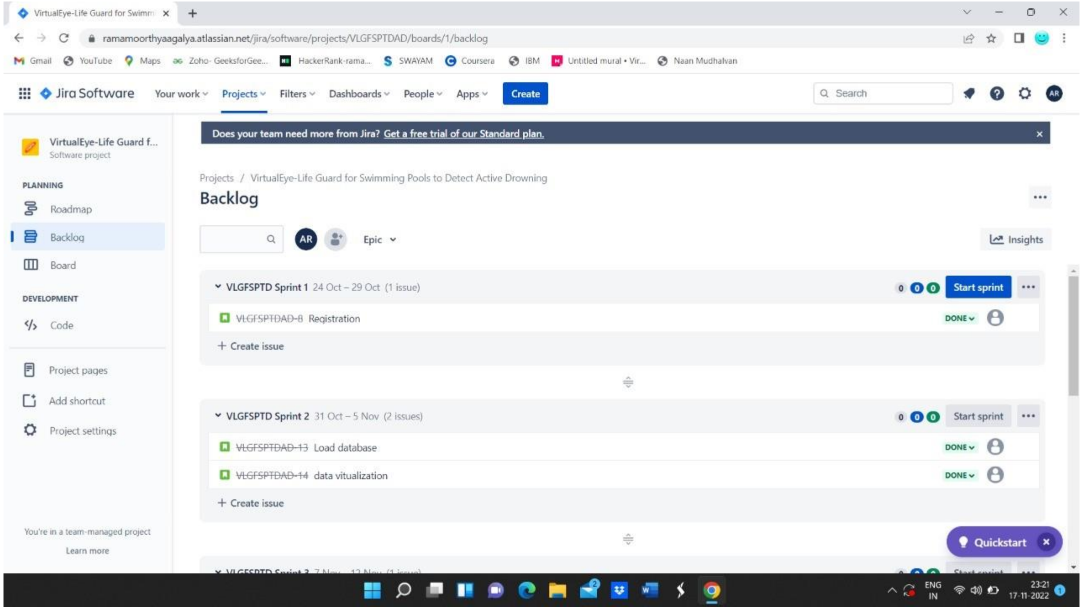
+ Create issue

VLGFSPTD Sprint 4 14 Nov – 19 Nov (3 issues) 0 0 0

VLGFSPTDAD-5	If a person is drowning,the system will ring an alarm	IN PROGRESS
VLGFSPTDAD-6	Detection of drowning person	IN PROGRESS
VLGFSPTDAD-7	Logout from the application	IN PROGRESS

+ Create issue

Screen Shot 2:



Projects / VirtualEye-Life Guard for Swimming Pools to Detect Active Drowning

Backlog

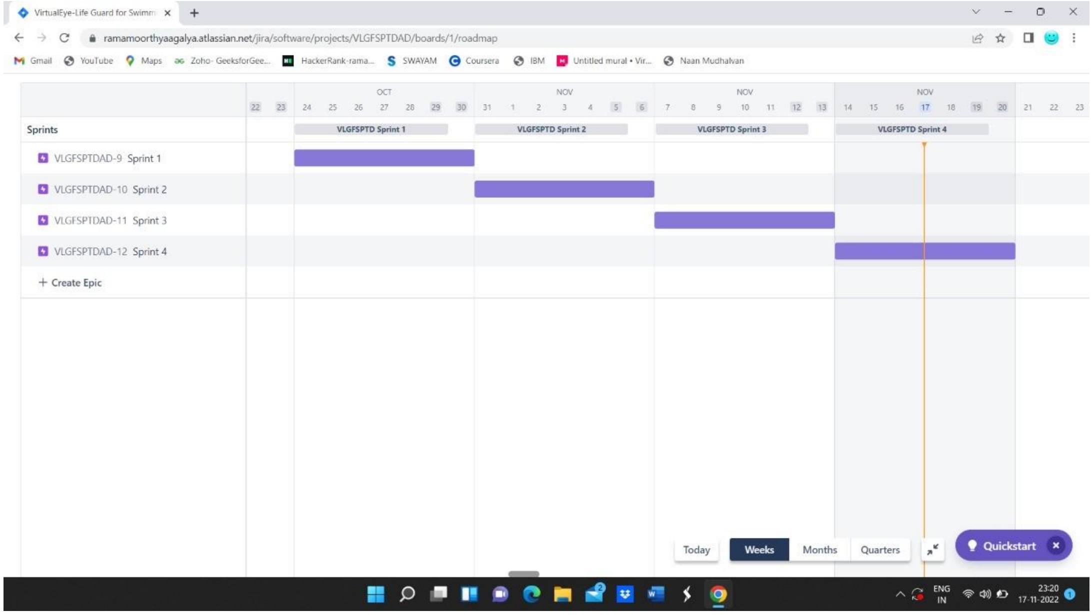
VLGFSPTD Sprint 1 24 Oct – 29 Oct (1 issue) 0 0 0

- ~~VLGFSPTDAD-8~~ Registration DONE
- + Create issue

VLGFSPTD Sprint 2 31 Oct – 5 Nov (2 issues) 0 0 0

- ~~VLGFSPTDAD-13~~ Load database DONE
- ~~VLGFSPTDAD-14~~ data vitualization DONE
- + Create issue

Screen Shot:3



Sprints	VLGFSPTD Sprint 1	VLGFSPTD Sprint 2	VLGFSPTD Sprint 3	VLGFSPTD Sprint 4
VLGFSPTDAD-9 Sprint 1				
VLGFSPTDAD-10 Sprint 2				
VLGFSPTDAD-11 Sprint 3				
VLGFSPTDAD-12 Sprint 4				
+ Create Epic				

7.

CODING & SOLUTIONING

7.1 Feature 1

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.

Software Requirements:

tensorflow

Keras

IBM Cloudant

Flask

OpenCVpython

imutils

flask

progress bar

play sound

Hardware Requirement:

Processor	- Intel core i5
Hard Disk Space	- Minimum 100GB
RAM	- 4GB
Display	-14.1"colour monitor (LCD,CRT or LED)
Clock Speed	- 1.67 GHz

7.2 Feature 2

To design a system in an economically viable and easily accessible way that acts as a virtual eye to detect the drowning person in the swimming pool and alert the lifeguard using alarms to save the drowning person.

The system can be deployed in house,hotels,resorts,and swimmingpoolcenters.The result is predicated in real-time, thus it can be used in emergency situations.

8. TESTING

8.1 Testcase

Test case	**Feature type**	**Page**	**Test scenario**	**Step To Execute**	**Test**	**Expected Result**	**Actual Result**
Login page_TC _001	Functional	Home page	Verify user is see the login/sign up popup when user clicked on my	1.Enter URL and click go 2.click on my account dropdo	Login.html	login/sign popup should display	Result Working

			Account	wn 3.verify login/signup popup display or not			
Login page_TC_002	Functional	Home page	Verify the UI element in login/signup popup	1.Enter URL and click go 2.Click on My account dropdown 3.Verify login/signup below UI a.Mail text box. b.password text box c.L% in button d.New customer?Create an account e.Lost passwor	Login.html	Application should show below element: a.email text box b.password text box c.login button with orange colour d.New Customer? Create account e.Last password ?Recovery password	Working

				d? Recovery password			
Login page_TC_003	Functional	Home page	Verify user is able to log into application with valid credentials	1.Enter URL and click go 2.Click my account dropdown 3.Enter valid user name	User name:swetha@gmail.c om Password :swtha21	User Should navigateto prediction homepage	Working
Login page_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials	1.-EnterURL and click go2.Click on my account dropdown 3.Enter Valid username/email in Emailtext box	Username:swetha@gmail.com password:swetha21	Application should show'Incorrect email or password' validation message	Working

				4.Enter Invalid password in passwo rd text box 5.Click on in button			
Login page_TC_005	Functional	Login page	Verify user is able to 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 passwordin password textbox 5.click on I in	Username:swetha@gmail.com password:swetha21	Application should show'Incorrect email or password' validation message.	Working

				button			
Login page_TC_006	Functional	Prediction page	Page should display wheather the person is drowning or not	1.Cameras should take pictures of drowning in pools 2.It should predict the probability of drowning 3.It should show a bounding box diplayaing the probability of drownin	Image of people drowning	Generate a alert to lifeguard if people are drowning	Working

8.2 User Acceptance testing

1. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Section	**Test Case**	**Not Testd**	**Fail**	**Pass**
Print Engine	2	0	0	2
Client Application	2	0	0	2
Security	1	0	0	1
Outsource Shipping	1	0	0	1
Exception Reporting	2	0	0	2
Final Report Output	1	0	0	1

2. Test case Analysis

This report shows the nmber of test case that have been passed,failed,untested

Resolution	Severty 1	Severty 2	Severty 3	Severty 4	Severty 5
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

10. ADVANTAGES & DISADVANTAGES

Advantages

The use of deep learning gives accurate results after training the model.

YOLOv3 model is fast and can process up to 45 frames per second.

Disadvantages

YOLO has low recall value and struggles to detect very close objects.

11. CONCLUSION

In this project, we have developed a deep learning system using YOLOv3 model to predict if a person is drowning or not. The system is connected to IBM cloud services the user can access through a web application along with the alarm feature system to notify the lifeguard.

12.FUTURE SCOPE

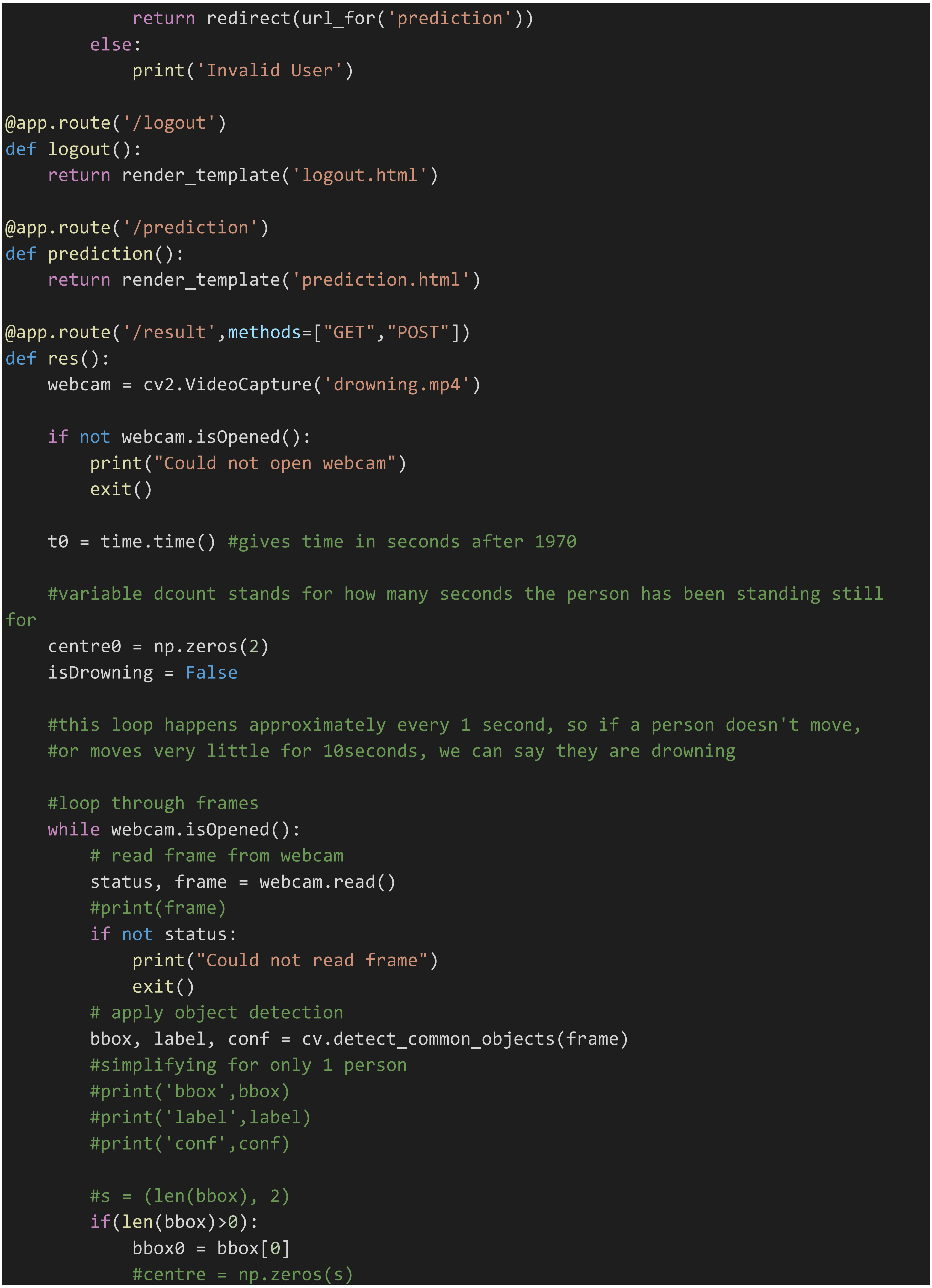
The project can be further extended by deploying multiple cameras underwater to improve accuracy of prediction. The processing speed of the model can be improved to produce the result faster.

13. APPENDIX

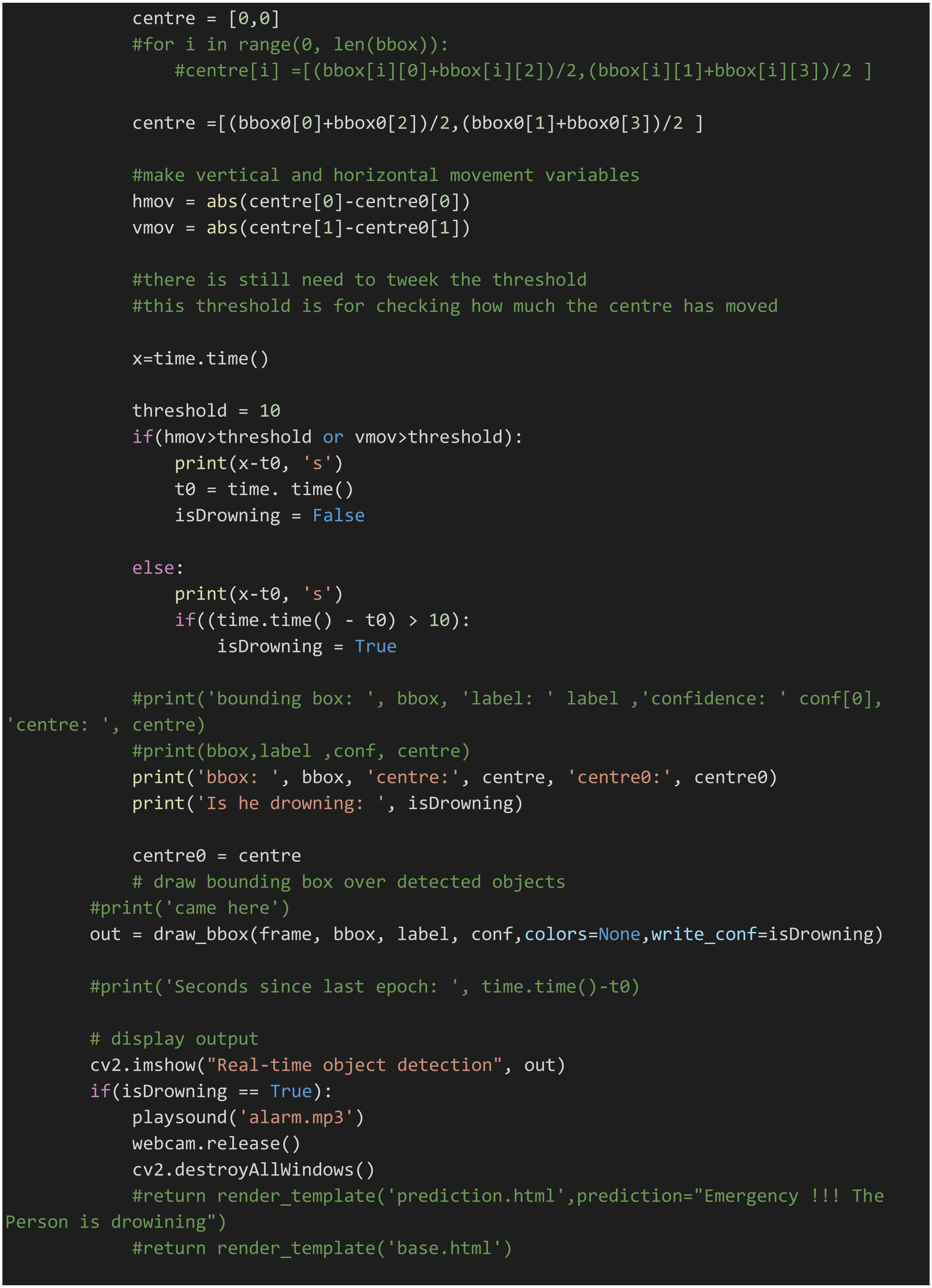
Source Code

```
import re
import numpy as np
import os
from flask import Flask, app, request, render_template, redirect, url_for
from tensorflow.keras import models
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.python.ops.gen_array_ops import concat
import cvlib as cv
from cvlib.object_detection import draw_bbox
import cv2
import time
from playsound import playsound
import requests

#Loading the model

from cloudant.client import Cloudant

# Authenticate using an IAM API key
client = Cloudant.iam('57f444d5-dfbd-4fc0-b752-dea54005c3cc-
bluemix','HTLp9_GkWGDyMR9VHruMMwi_qzZ43qaI3UVR77GOI2GX', connect=True)


# Create a database using an initialized client
my_database = client.create_database('my_database')

app=Flask(__name__)

#default home page or route
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/index.html')
def home():
    return render_template("index.html")

#registration page
@app.route('/register')
def register():
    return render_template('register.html')
```

```
@app.route('/afterreg', methods=['POST'])
def afterreg():
    x = [x for x in request.form.values()]
    print(x)
    data = {
    '_id': x[1], # Setting _id is optional
    'name': x[0],
    'psw':x[2]
    }
    print(data)

    query = {'_id': {'$eq': data['_id']}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))

    if(len(docs.all())==0):
        url = my_database.create_document(data)
        #response = requests.get(url)
        return render_template('register.html', pred="Registration Successful, please
login using your details")
    else:
        return render_template('register.html', pred="You are already a member, please
login using your details")

#login page
@app.route('/login')
def login():
    return render_template('login.html')

@app.route('/afterlogin',methods=['POST'])
def afterlogin():
    user = request.form['_id']
    passw = request.form['psw']
    print(user,passw)

    query = {'_id': {'$eq': user}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))

    if(len(docs.all())==0):
        return render_template('login.html', pred="The username is not found.")
    else:
        if((user==docs[0][0]['_id'] and passw==docs[0][0]['psw'])):
```



```
            return redirect(url_for('prediction'))
        else:
            print('Invalid User')

@app.route('/logout')
def logout():
    return render_template('logout.html')

@app.route('/prediction')
def prediction():
    return render_template('prediction.html')

@app.route('/result',methods=["GET","POST"])
def res():
    webcam = cv2.VideoCapture('drowning.mp4')

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

    t0 = time.time() #gives time in seconds after 1970

    #variable dcount stands for how many seconds the person has been standing still
for
    centre0 = np.zeros(2)
    isDrowning = False

    #this loop happens approximately every 1 second, so if a person doesn't move,
    #or moves very little for 10seconds, we can say they are drowning

    #loop through frames
    while webcam.isOpened():
        # read frame from webcam
        status, frame = webcam.read()
        #print(frame)
        if not status:
            print("Could not read frame")
            exit()
        # apply object detection
        bbox, label, conf = cv.detect_common_objects(frame)
        #simplifying for only 1 person
        #print('bbox',bbox)
        #print('label',label)
        #print('conf',conf)

        #s = (len(bbox), 2)
        if(len(bbox)>0):
            bbox0 = bbox[0]
            #centre = np.zeros(s)
```



```
            centre = [0,0]
            #for i in range(0, len(bbox)):
                #centre[i] =[(bbox[i][0]+bbox[i][2])/2,(bbox[i][1]+bbox[i][3])/2 ]

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

            #make vertical and horizontal movement variables
            hmov = abs(centre[0]-centre0[0])
            vmov = abs(centre[1]-centre0[1])

            #there is still need to tweek the threshold
            #this threshold is for checking how much the centre has moved

            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('bounding box: ', bbox, 'label: ' label ,'confidence: ' conf[0],
'centre: ', centre)
            #print(bbox,label ,conf, centre)
            print('bbox: ', bbox, 'centre:', centre, 'centre0:', centre0)
            print('Is he drowning: ', isDrowning)

            centre0 = centre
            # draw bounding box over detected objects
        #print('came here')
        out = draw_bbox(frame, bbox, label, conf,colors=None,write_conf=isDrowning)

        #print('Seconds since last epoch: ', time.time()-t0)

        # display output
        cv2.imshow("Real-time object detection", out)
        if(isDrowning == True):
            playsound('alarm.mp3')
            webcam.release()
            cv2.destroyAllWindows()
            #return render_template('prediction.html',prediction="Emergency !!! The
Person is drowining")
            #return render_template('base.html')
        # press "Q" to stop
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    # release resources
    webcam.release()
    cv2.destroyAllWindows()
    return render_template('prediction.html',prediction="Emergency !!! The Person is drowning")

""" Running our application """
if __name__ == "__main__":
    app.run(debug=False)
```