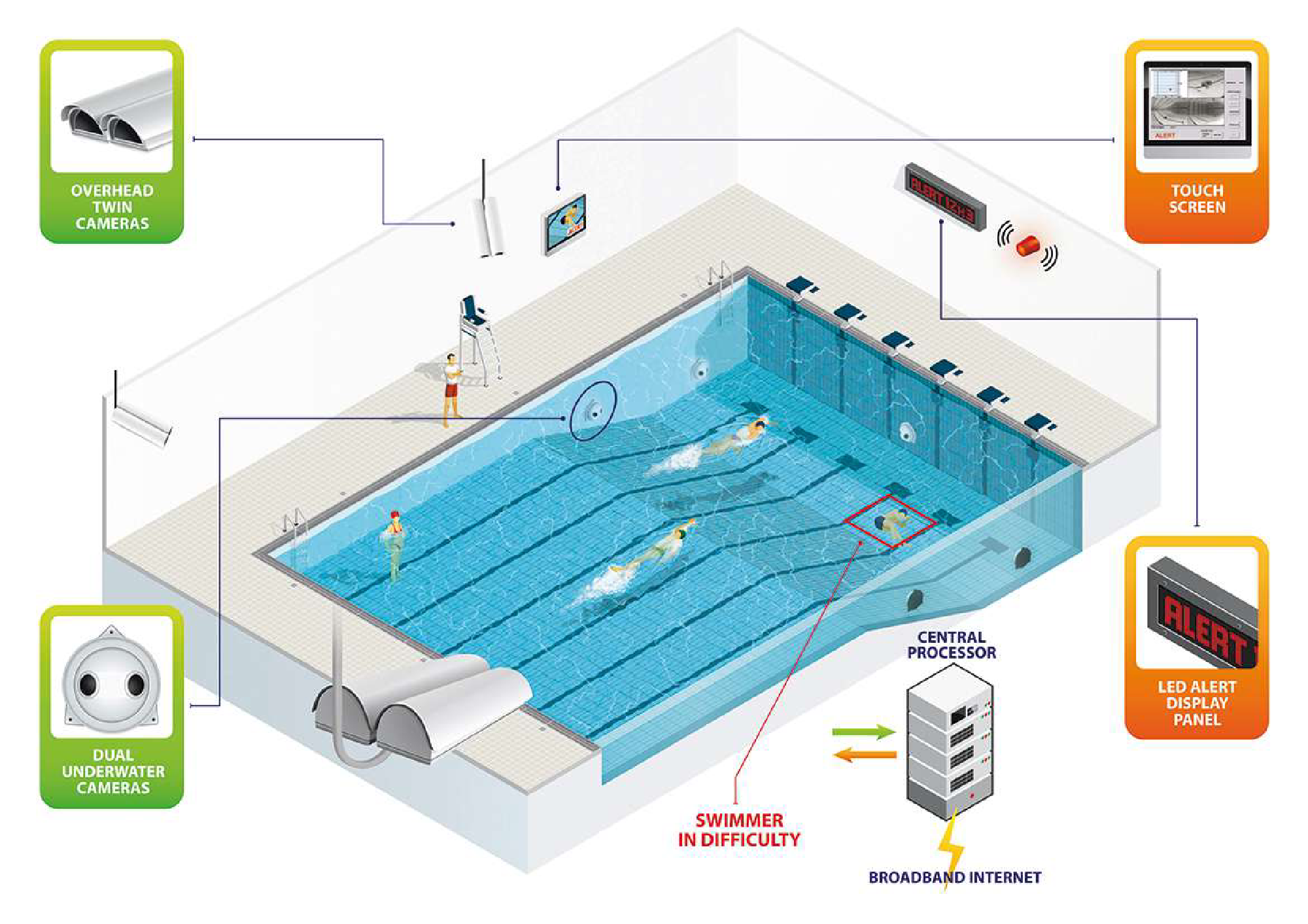
|  |
| --- |
| **Final Documentation** |
| VirtualEye –Life Guard for Swimming Pools to Detect Active Drowning |
| **TEAMID: PNT2022TMID52387**  **Team Members**  Tamizhavi K [Team leader]  Devaprasath R  Surya B  Mohamedmushraf S |
|  |
|  |
|  |
|  |
|  |

CONTENT

1. INTRODUCTION1.1 Project Overview  
   1.2 Purpose  
   2. LITERATURE SURVEY2.1 Existing problem  
   2.2 References  
   2.3 Problem Statement Definition  
   3. IDEATION & PROPOSED SOLUTION3.1 Empathy Map Canvas  
   3.2 Ideation & Brainstorming  
   3.3 Proposed Solution  
   3.4 Problem Solution fit  
   4. REQUIREMENT ANALYSIS4.1 Functional requirement  
   4.2 Non-Functional requirements  
   5. PROJECT DESIGN5.1 Data Flow Diagrams  
   5.2 Solution & Technical Architecture  
   5.3 User Stories  
   6. PROJECT PLANNING & SCHEDULING6.1 Sprint Planning & Estimation  
   6.2 Sprint Delivery Schedule  
   6.3 Reports from JIRA  
   7. CODING  
   8. TESTING8.1 Test Cases  
   8.2 User Acceptance Testing  
   9. RESULTS9.1 Performance Metrics  
   10. ADVANTAGES & DISADVANTAGES11. CONCLUSION12. FUTURE SCOPE13. APPENDIXSource Code  
   GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project overview:

1.2 Purpose:

The system is not designed to replace a lifeguard or other human monitor, but to act as an additional tool. “It helps the lifeguard to detect the underwater situation where they can’t easily observe.

2. LITERATURE SURVEY

2.1 Existing problem

Whilst literature on DDS mostly agrees on areas such as the risks and issues associated with DDS performance, there are other areas where sources offer differing points of view, for example, DDS and their co- existence with lifeguards. There is debate around whether DDS can be helpful or harmful towards lifeguarding practices and how DDS may change the landscape of traditional lifeguarding, as well as some disagreement on whether they serve as justification for reducing lifeguard numbers. The term ‘blended lifeguarding’ or ‘modern lifeguarding’ has been newly coined to describe the concept of traditional lifeguarding practices being blended with technology for drowning detection (Swimming Pool Scene, 2017).

Currently, there is little qualitative or quantitative research analysing the experiences of lifeguards themselves relating to this concept.

2.2 References

* + - AngelEye.(2019).AngelEye–Distributors.Retrievedfrom: https://[www.angeleye.it/news.](http://www.angeleye.it/news) php?id=28&newscat=10
    - Aquatics International. (2007). Traumatic Experiences – Should we make our youngest lifeguards come face to face with death? Retrieved from: <https://www.aquaticsintl.com/facilities/traumaticexperiences_o>
    - British Standards Institution. (2018). BS EN 15288-1, Swimming pools for public use. Safety requirements for design. Retrieved from: <https://shop.bsigroup.com/ProductDetail/?pid=000000000030360254>
    - British Standards Institution 1. (2018). BS EN 15288-2, Swimming pools for public use. Safety requirements for operation. Retrieved from: https://shop.bsigroup.com/ProductDetail/?p id=000000000030360257
    - Drowning Prevention. (2017). The Need. Retrieved from: <https://www.drowningprevention.com.au/>
    - German Institute for Standardization. (2019). German national guideline DGfdB R 94.15 “Test methods for camera-based drowning detection systems under operational conditions” (German Association for Public Swimming Pools).
    - Haizhou Li, Haizhou Li, Kar-Ann Toh and Liyuan Li. (2012). Advanced Topics in Biometrics, World Scientific Publishing Co. Pte. Ltd., ISBN-13 978-981-4287-84-5
    - Health and Safety Executive. (2018). HSG179, Health and safety in swimming pools (Fourth edition).
  1. Problem Statement Definition

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.

2.Applying the CNN algorithm to the dataset.Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident.

3.To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life.

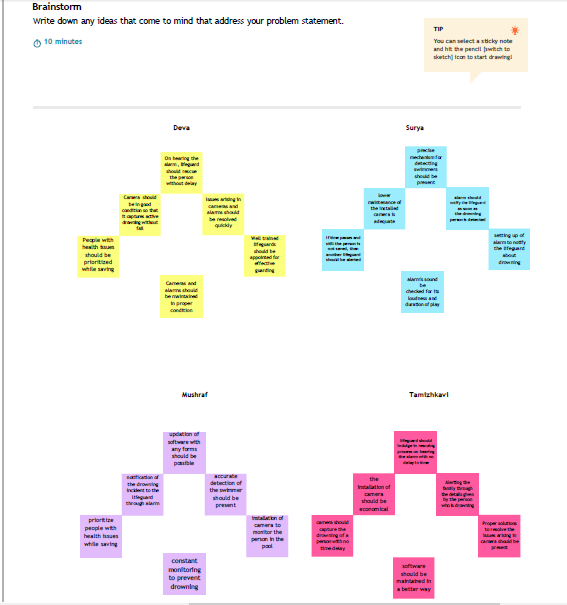
3. IDEATION & PROPOSED SOLUTION

In this paper we have proposed a method for automatic real-time detection of a person drowning in the swimming pools. The overview of the proposed algorithm in this paper is presented.

Our system is based on real time video analysis of the cameras installed around the swimming pool in a way which the entire swimming pool can be covered. Each camera is mounted on pool walls oriented downwards  
with a sharp angle, so that it can minimize the effect of lightening system which causes occlusions and foreshadowing. In this work, a ODROID-XU as a distributed system is installed in the swimming pool to collect all the video signals collected from cameras and process them using computer vision methods. The used hardware including the distributing system known as ODROID-XU, and our Logitech HD Pro C920 webcam used to record all the video sequences in this paper is illustrated in. The system is used to firstly detect the background of the pool and then decide to send an alarm to rescue team if a previously detected person is missing in video frames for an specific and defined  
period of time. In the next sections of this paper, we try to explain the concepts we used to detect and track  
individuals in swimming pools. 3.1 Empathy Map Canvas

****

3.2 Ideation & Brainstorming

****

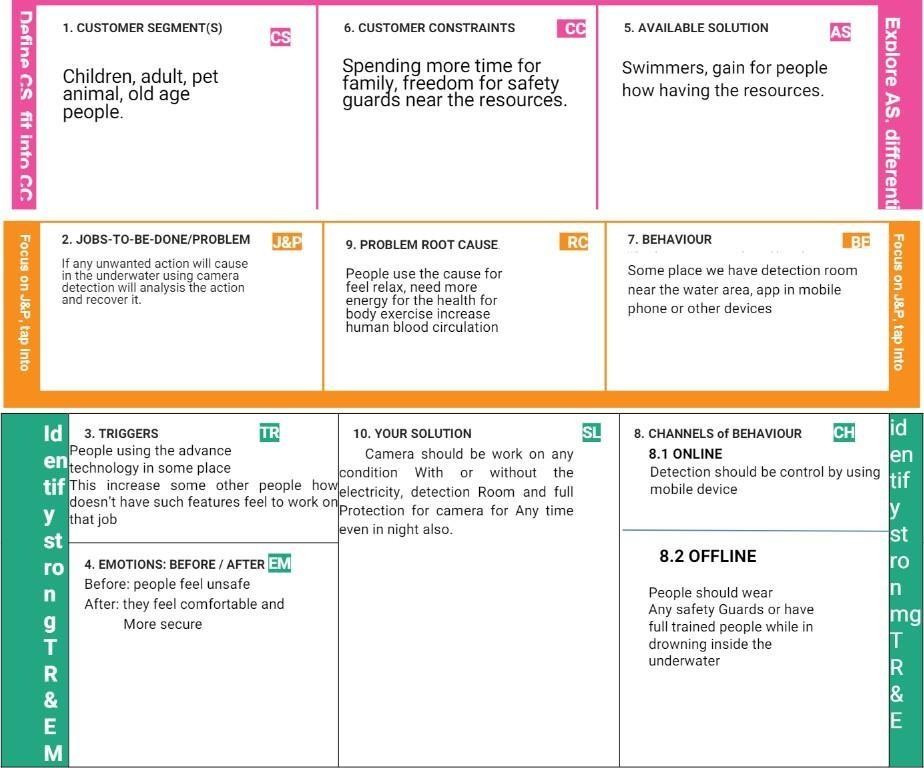
Ideations



3.3 Proposed Solution

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
| 1. | Problem Statement (Problem to be solved) | People use the swimming for enjoyment, health Exercise but for all age of the people pool is really dangerous we need lifeguard, in duty  swimming pools are very dangerous in the underwater. |
| 2. | Idea / Solution description | In this project, using Artificial intelligence technology, using the camera help we can detect the people action and positions and also we check breathing level of the people inside the underwater and use of any alarms system  we can detect the some of them are in the problem |
| 3. | Novelty / Uniqueness | The uniqueness of the our system is track the people position and body condition in the  drowning using YOLO Algorithm. It is fast and very speed in the detection |
| 4. | Social Impact / Customer Satisfaction | In world most of them are unexcepted cause very serious death in the underwater not only in the city but most occurs in the rural area in  the public places (well, lakes) we should avoid the accident in the underwater drowning |
| 5. | Business Model (Revenue Model) | In the software field this well increase good income. Safety innovation in the swimming  related issues this makes attractive for end users to use our software product |
| 6. | Scalability of the Solution | IBM cloud server will collect all the data and stored in the server. This will more safe and  secure |

3.4 Problem Solution fit



**4.** REQUIREMENT ANALYSIS

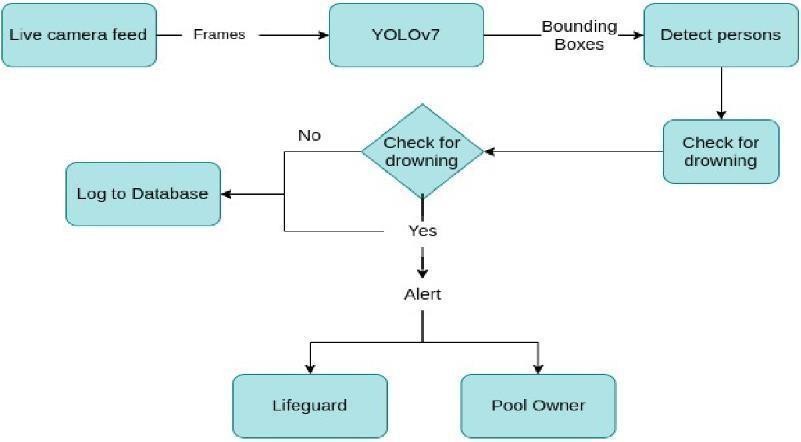
4.1 Functional requirement

|  |  |  |
| --- | --- | --- |
| **FR**  **No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | Installation | Install the camera inside the underwater,  connect necessary app in the phone or otherdevice |
| FR-2 | Detection | Near swimming pool area use detectionroom for monitor or use IBM  cloud forstorage purpose of the details |
| FR-3 | Audio | Give the alert signal for the people  enterinto the underwater and leaving into underwater |
| FR-4 | Support | Extra support from the lifeguard if any  person pulse rate will decrease inside the water |
| FR-5 | Prior alert | Extreme level problem should be  occursgive the alert signal for the entire pool |

4.2 Non-Functional requirements

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | A Lifeguard should be present in all the time near pool |
| NFR-2 | **Security** | Alert message or signal should be give by the lifeguard of swimmer |
| NFR-3 | **Reliability** | Triggers if any immediate needs of theswimmer inside the pool |
| NFR-4 | **Performance** | If any unwanted position changes and thepulse rate will decrease this will detect it. |
| NFR-5 | **Availability** | Equipment and other requirement should bechecked by the lifeguards |
| NFR-6 | **Scalability** | Virtual eye lifeguard detects potential  drownings and it should be notifies you. |

5.Project Design

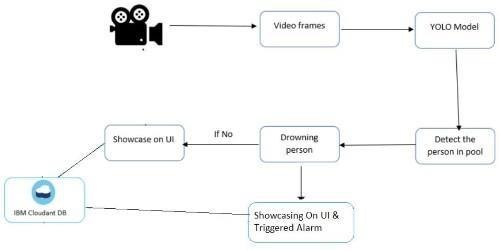
5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

* + - To find underwater movement while person in drowning they have any

Problem or anything else we will find the solution using the Artificial Intelligence (AI) detection technology.

* + - Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analyzing the video feeds to detect any anomalies. AS a POC we make use of one camera that streams the video underwater and analyses the positionof swimmers to assess the probability of drowning, if it is higher then an alert will be generated to attract lifeguards' attention.



5.3Users Stories

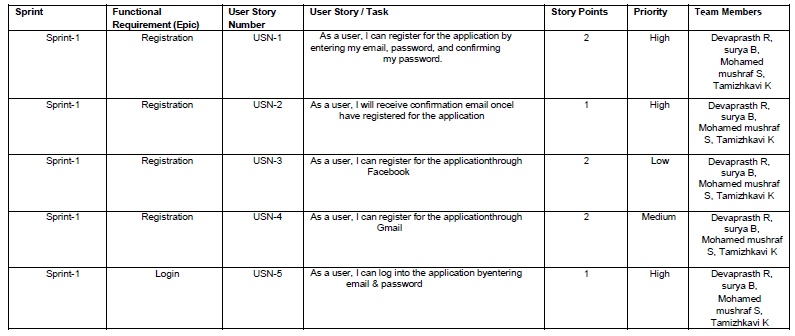
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer (Pool owner) | Installation | USN-1 | Install the camera inside the underwater, connect necessary app in the phone or  other device | I can cameras to the IBM cloud DB | High | Sprint-1 |
| Customer (Lifeguard | Secure thepeople | USN-2 | As a user, I can secure the drowning personsfrom the  pool | I can save the drowning person | High | Sprint-1 |
| Customer (swimmers) | safety | USN-3 | As a user, I can swim inside the underwater without  fear of the Drowning | I can swim safely | medium | Sprint-2 |
| Customer care (Executive) | Contact | USN-4 | As a user, I Can resolve if any problem occurs with any device technically | I can contact the customer care executiveto  resolve any issues | Medium | Sprint-3 |
| Administrator | Dashboard | USN-5 | Management of the drowning detection systemand database  management | I can access the  system’s  logs and any  other data instantly | High | Sprint-4 |

6.PROJECT PLANNING & SCHEDULING

6.1SPRINT PLANNING & ESTIMATION

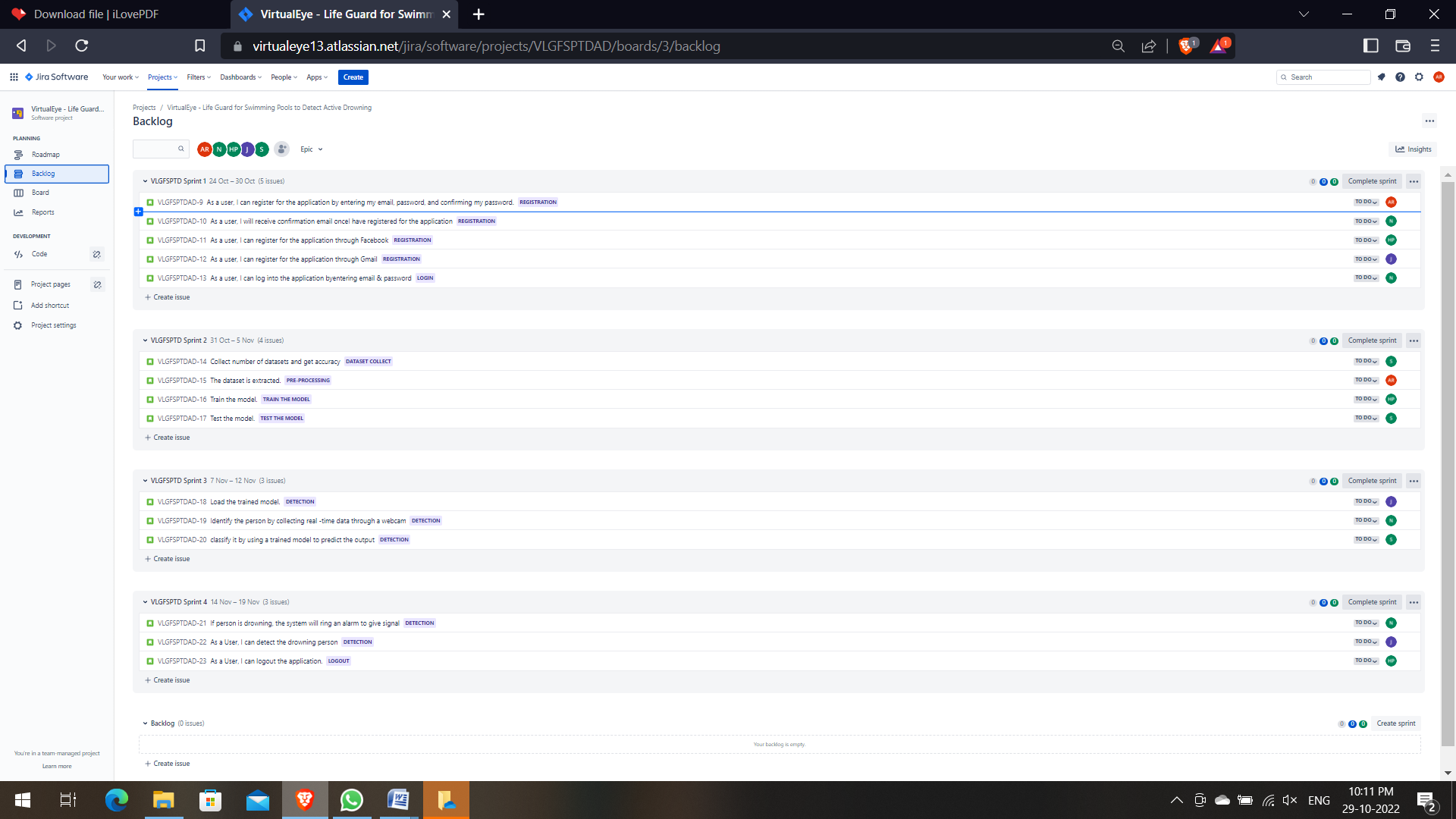
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned End Date)** | **Sprint Release Date(Actual)** |
| Sprint-1 | 8 | 6 Days | 24 Oct 2022 | 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 |

6.2 Sprint Delivery Schedule

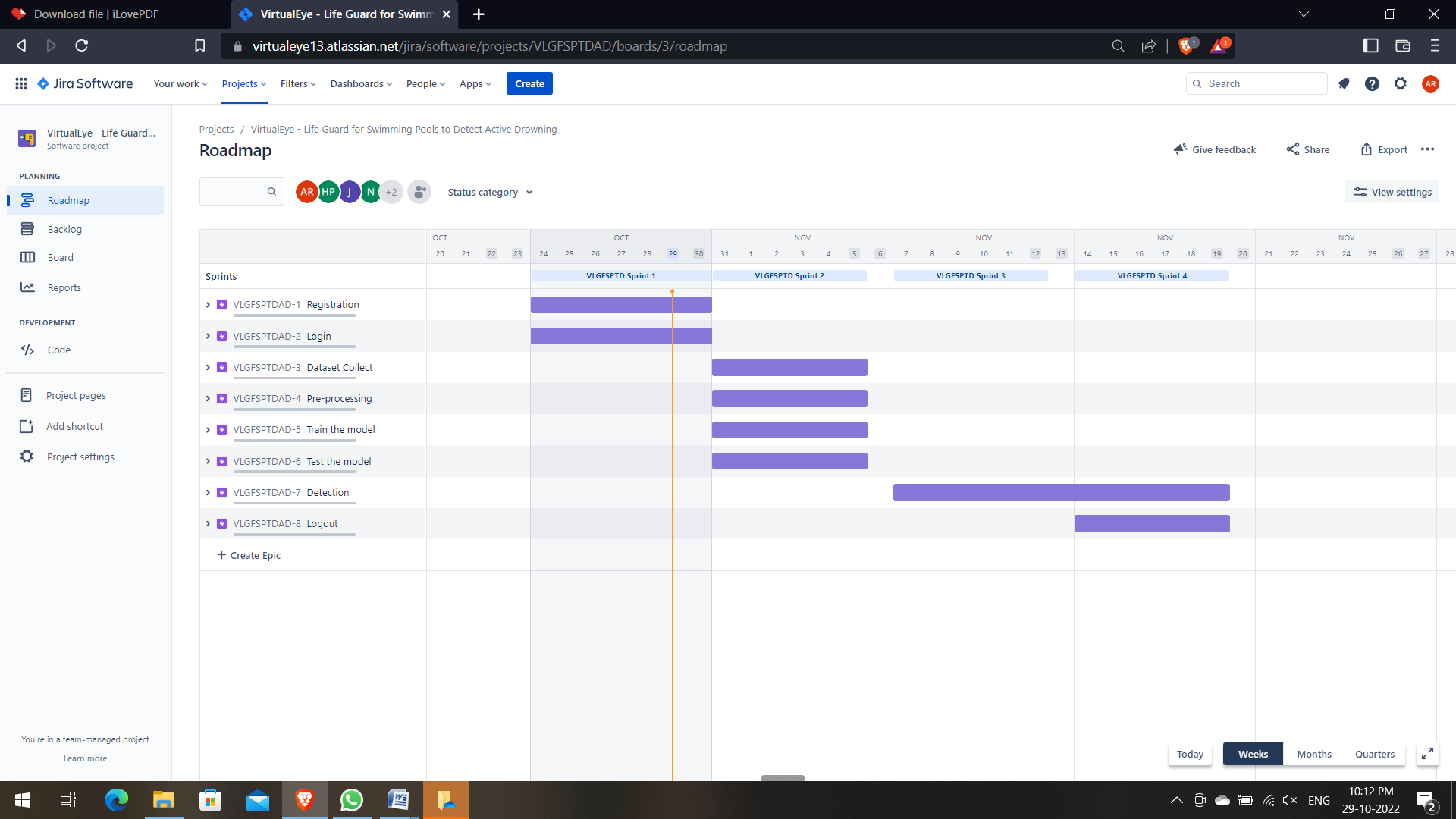


6.3 Reports from JIRA

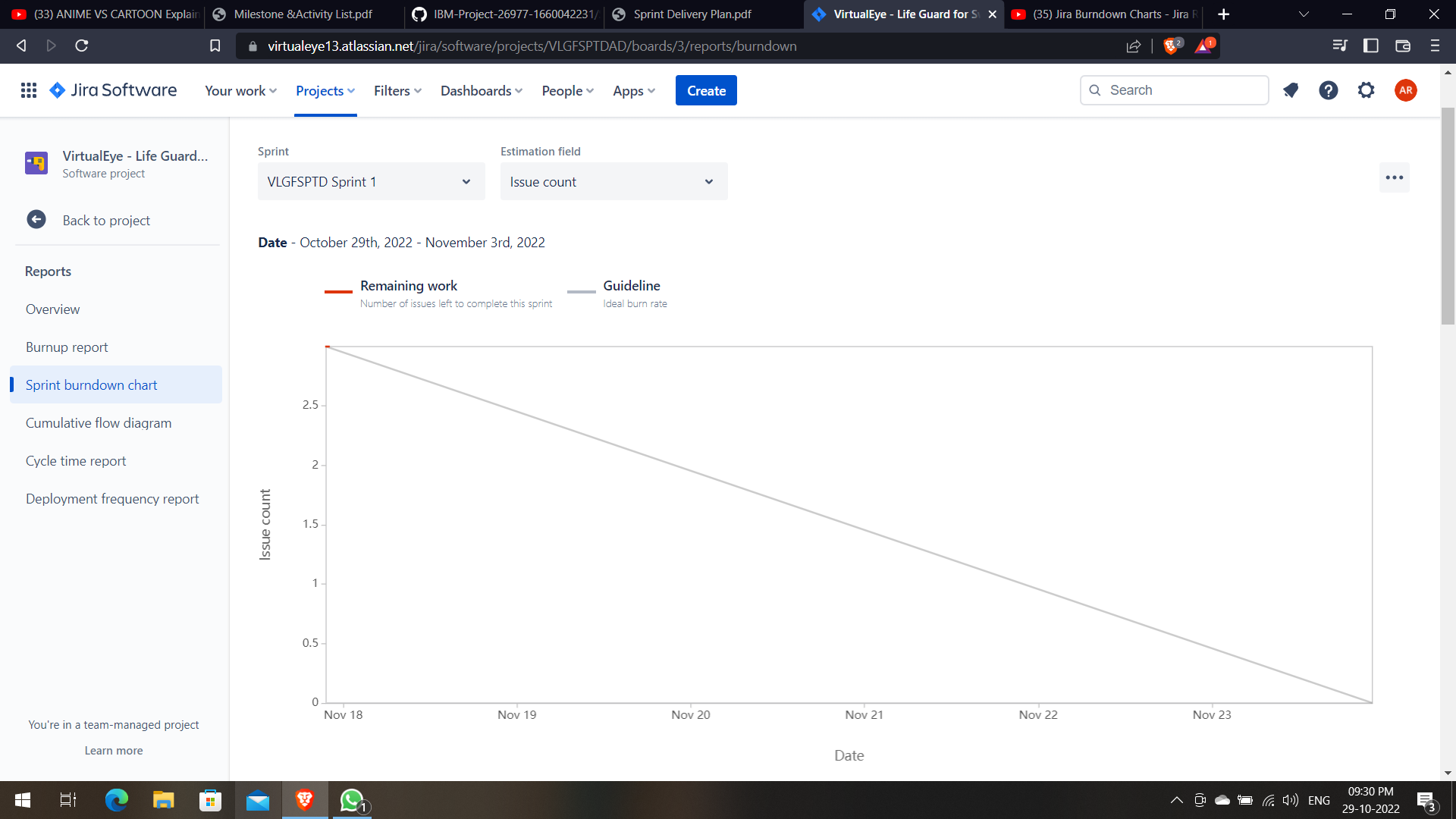
**Backlog**



Roadmap



Burndown Chart



**7. CODING**

import time  
  
import cv2  
import numpy as np  
from cloudant.client import Cloudant  
from flask import Flask, request, render\_template, redirect, url\_for  
from playsound import playsound  
  
import cvlib as cv  
from cvlib.object\_detection import draw\_bbox  
  
# Loading the model  
  
# Authenticate using an IAM API key  
client = Cloudant.iam('8780b82a-5a3b-4da0-a180-a0e1516479f9-bluemix', 'TzYs8u0Q5eoj204gDo2eOEDAuGRhj0fG\_9rlZr5SsJUH',  
 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()  
  
 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  
  
 # 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  
  
 out = draw\_bbox(frame, bbox, label, 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',)  
  
  
""" Running our application """  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run(debug=True)

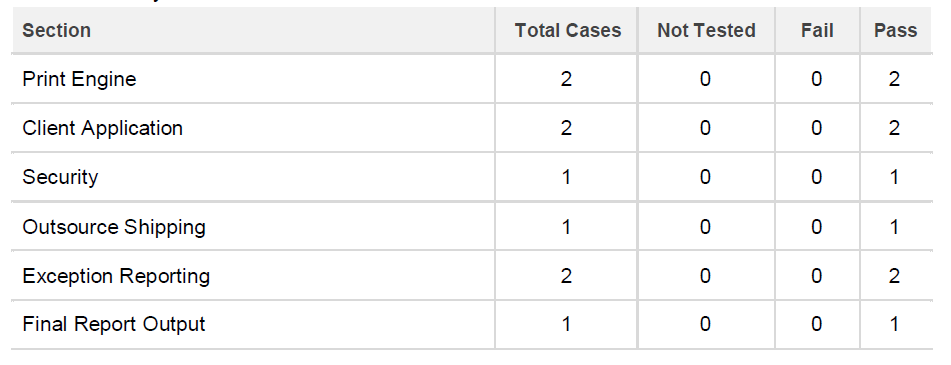
**8. TESTING**

**8.1 Test cases**

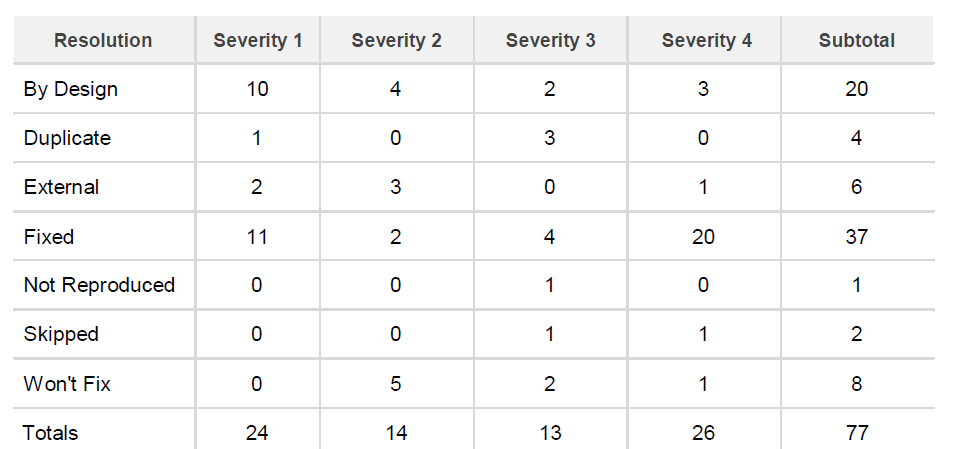


8.2 User Acceptance Testing

Defect analysis



Test analysis



**9. RESULTS**

**9.1Performance metrics**

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>High Quality Facial Recognition</title>

<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min. css" rel="stylesheet">

<script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js">

</script>

<script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js">

</script>

<script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js">

</script>

<link href="{{ url\_for('static', filename='css/main.css') }}"rel="stylesheet">

<style>

.bg-dark {

background-color: #42678c!important;

}

#result {

color: #0a1c4ed1;

</head>

}

</style>

<body style="background-color:black";>

<header id="head" class="header">

<section id="navbar">

<h1 class="nav-heading"></i>Virtual Eye</h1>

<div class="nav--items">

<ul>

<li><a href="{{ url\_for('index')}}">Home</a></li>

<li><a href="{{ url\_for('logout')}}">Logout</a></li>

<!-- <li><a href="#about">About</a></li>

<li><a href="#services">Services</a></li> -->

</ul>

</div>

</section>

</header>

<div class="container">

<div id="content" style="margin-top:2em">

<div class="container">

<div class="row">

<div class="col-sm-6 bd" >

<h2><em style="color:white;">High Quality Facial Recognition</em></h2>

<br>

<p><h5><i style="color:white;">Emotion Detection Through Facial Feature Recognition</i></h5></p>

<img src="https://130e178e8f8ba617604b- 8aedd782b7d22cfe0d1146da69a52436.ssl.cf1.rackcdn.com/facial- recognition- use-triggers-gdpr-fine-showcase\_image-10-a- 12991.jpg" style="height:240px"class="img-rounded" alt="Gesture">

</div>

<div class="col-sm-6">

<div>

Image Here</h4>

<h4 style="color:white;">Upload

<form action = "http://localhost:5000/" id="upload-file"

method="post" enctype="multipart/form-data">

<label for="imageUpload" class="upload-

label">

</label>

Choose Image

<input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg,.pdf">

</form>

<div class="image-section" style="display:none;">

<div class="img-preview">

<div id="imagePreview">

</div>

</div>

<div>

<button type="button" class="btn btn-info btn-lg " id="btn-predict">Analyse</button>

</div>

</div>

<div class="loader" style="display:none;"></div>

<h3>

</h3>

<span id="result"> </span>

</div>

</div>

</div>

</body>

</div>

</div>

</div>

<footer>

<script src="{{ url\_for('static', filename='js/main.js') }}" type="text/javascript"></script>

</footer>

</html>

# Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!--Bootstrap -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/boo tstrap.min.css" integrity="sha384-

Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGg FAW/dAiS6JXm" crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384- KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpG FF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/ popper.min.js" integrity="sha384-

ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPsk vXusvfa0b4Q" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootst rap.min.js" integrity="sha384-

JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5

+76PVCmYl" crossorigin="anonymous"></script>

<script src="https://kit.fontawesome.com/8b9cdc2059.js" crossorigin="anonymous"></script>

<link href="https://fonts.googleapis.com/css2?family=Akronim&family= Roboto&display=swap" rel="stylesheet">

<link rel="stylesheet" href="../static/style.css">

<!-- <script defer src="../static/js/main.js"></script> -->

<title>Virtual Eye</title>

</head>

<body>

<header id="head" class="header">

<section id="navbar">

<h1 class="nav-heading"></i>Virtual Eye</h1>

<div class="nav--items">

<ul>

<li><a href="{{ url\_for('index')}}">Home</a></li>

<li><a href="{{ url\_for('login')}}">Login</a></li>

<li><a

href="{{ url\_for('register')}}">Register</a></li>

<li><a href="{{ url\_for('login')}}">Demo</a></li>

</ul>

</div>

</section>

<section id="slider">

<div id="carouselExampleIndicators" class="carousel" data-ride="carousel">

<ol class="carousel-indicators ">

<li data-target="#carouselExampleIndicators" data-slide- to="0" class="active "></li>

<li data-target="#carouselExampleIndicators" data-slide-to="1"></li>

<li data-target="#carouselExampleIndicators" data-slide-to="2"></li>

</ol>

<div class="carousel-inner">

<div class="carousel-item active">

<img class="d-block w-100" src="../static/img/1.png"alt="First slide">

</div>

<div class="carousel-item">

<img class="d-block w-100" src="../static/img/second.jpg" alt="Second slide">

</div>

<div class="carousel-item">

<img class="d-block w-100" src="../static/img/third.jpg"alt="Third slide">

</div>

</div>

<a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">

<span class="carousel-control-prev-icon" aria- hidden="true"></span>

<span class="sr-only">Previous</span>

</a>

<a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">

<span class="carousel-control-next-icon" aria- hidden="true"></span>

<span class="sr-only">Next</span>

</a>

</div>

</section>

</header>

<section id="about">

<div class="top">

<h3 class="title text-muted"> ABOUT PROJECT

</h3>

<div class="line"></div>

</div>

<div class="body">

<div class="left">

<h2>Problem:</h2>

<p>

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 the hotels, weekend tourist spots and barelypeople have in their house backyard. Beginners, especially oftenfeel it difficult to breathe under water and causes breathing trouble which in turn cause 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 besuffering 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.

</p>

</div>

<div class="left">

<h2>Solution:</h2>

<p>

To overcome the conflict, a meticulous system is to be implemented along the swimming pools to save the human life. Bystudying body movement patterns and connecting cameras to an artificial intelligence (AI)system we can devise an underwater poolsafety 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 thanan alert will be generated to attract lifeguards attention.

</p>

</div>

</div>

<div class="bottom">

<p ><b>

Note : The system is not designed to replace a lifeguard or other human monitor, but to act as an additional tool. â€œIt helpsthe lifeguard to detect the underwater situation where they canâ€™t easily observe.

</b></p>

</div>

</section>

<section id="footer">

<p>Copyright Â© 2022. All Rights Reserved</p>

<div class="social">

<a href="#" target="\_blank"><i class="fab fa-2x fa-twitter-square"></i></a>

<a href="#" target="\_blank">

<i class="fab fa-2x fa-linkedin"></i></a>

<a href="#">

<i class="#"></i>

</a>

</div>

</section>

</body>

</html>

# Logout.html

<!DOCTYPE html>

<html >

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Virtual Eye</title>

<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>

<style>

.header {

top:0; margin:0px;

left: 0px; right: 0px; position: fixed;

background-color: #28272c; color: white;

box-shadow: 0px 8px 4px grey; overflow: hidden;

padding-left:20px;

font-family: 'Josefin Sans'; font-size: 2vw;

width: 100%; height:8%;

text-align: center;

}

.topnav {

overflow: hidden; background-color: #333;

}

.topnav-right a { float: left; color: #f2f2f2;

text-align: center; padding: 14px 16px; text- decoration: none; font- size: 18px;

}

.topnav-right a:hover { background-color: #ddd; color: black;

}

.topnav-right a.active { background-color: #565961; color: white;

}

.topnav-right { float: right;

padding-right:100px;

}

.login{

margin-top:-70px;

}

body {

background-color:#ffffff; background-repeat: no-repeat; background-size:cover; background-position: 0px 0px;

}

.main{

margin-top:100px; text-align:center;

}

form { margin-left:400px;margin-right:400px;}

input[type=text], input[type=email],input[type=number],input[type=password] { width: 100%;

padding: 12px 20px; display: inline-block; margin-bottom:18px; border: 1px solid #ccc;

box-sizing: border-box;

}

button {

background-color: #28272c; color: white;

padding: 14px 20px; margin-bottom:8px; border: none; cursor: pointer; width: 20%;

}

button:hover { opacity: 0.8;

}

.cancelbtn { width: auto;

padding: 10px 18px; background-color: #f44336;

}

.imgcontainer { text- align: center;

margin: 24px 0 12px 0;

}

img.avatar { width: 30%;

border-radius: 50%;

}

.container { padding: 16px;

}

span.psw { float: right;

padding-top: 16px;

}

/\* Change styles for span and cancel button on extra small screens

\*/

@media screen and (max-width: 300px) { span.psw {

display: block; float: none;

}

.cancelbtn { width: 100%;

}

}

</style>

</head>

<body style="font-family:Montserrat;">

<div class="header">

<div style="width:50%;float:left;font-size:2vw;text- align:left;color:white; padding- top:1%">Virtual eye</div>

<div class="topnav-right" style="padding-top:0.5%;">

<a href="{{ url\_for('home')}}">Home</a>

<a href="{{ url\_for('login')}}">Login</a>

<a href="{{ url\_for('register')}}">Register</a>

</div>

</div>

<div class="main">

<h1>Successfully Logged Out!</h1>

<h3 style="color:#4CAF50">Login for more information<h3>

<a href="{{ url\_for('login') }}"><button type="submit">Login</button></a>

</form>

</div>

</body>

</html>

# Prediction.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!--Bootstrap -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/boo tstrap.min.css" integrity="sha384-

Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGg FAW/dAiS6JXm" crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-

KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpG FF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/ popper.min.js" integrity="sha384-

ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPsk vXusvfa0b4Q" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootst rap.min.js" integrity="sha384-

JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5

+76PVCmYl" crossorigin="anonymous"></script>

<script src="https://kit.fontawesome.com/8b9cdc2059.js" crossorigin="anonymous"></script>

<link href="https://fonts.googleapis.com/css2?family=Akronim&family= Roboto&display=swap" rel="stylesheet">

<link rel="stylesheet" href="../static/style.css">

<script defer src="../static/js/JScript.js"></script>

<title>Prediction</title>

</head>

<body>

<header id="head" class="header">

<section id="navbar">

<h1 class="nav-heading"></i>Virtual Eye</h1>

<div class="nav--items">

<ul>

<li><a href="{{ url\_for('index')}}">Home</a></li>

<li><a href="{{ url\_for('logout')}}">Logout</a></li>

<!-- <li><a href="#about">About</a></li>

<li><a href="#services">Services</a></li> -->

</ul>

</div>

</section>

</header>

<!-- dataset/Training/metal/metal326.jpg -->

</br>

<section id="prediction">

<h2 class="title text-muted">Virtual Eye- Life Guard forSwimming Pools to Detect Active Drowning</h1>

<div class="line" style="width: 900px;"></div>

</section>

</br>

<section id="about">

<div class="body">

<div class="left">

<p>

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 the hotels, weekend tourist spots and barelypeople have in their house backyard. Beginners, especially oftenfeel it difficult to breathe under water and causes breathing trouble which in turn cause 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 besuffering 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.

</p>

</div>

<div class="left">

<div class="prediction-input">

<img class="d-block w-100" src="../static/img/second.jpg"alt="Second slide">

</br>

<form id="form" action="/result" method="post" enctype="multipart/form-data">

Demo">

</div>

</div>

</section>

<input type="submit" class="submitbtn" value="ClickMe! For a

</form>

</div>

<h5 style="text-color:Red">

<b style="text-color:Red">{{prediction}}<b>

</h5>

</br></br>

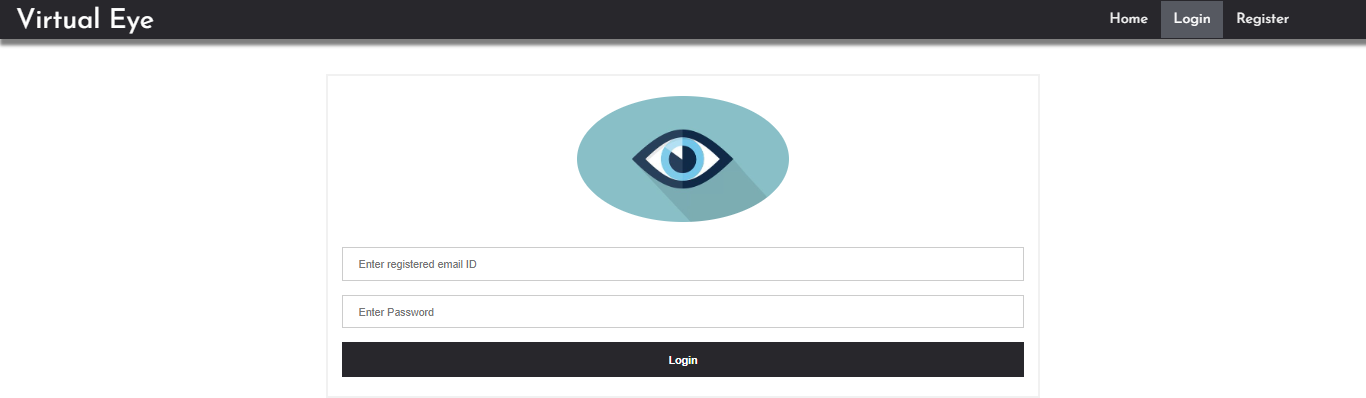
<section id="footer">

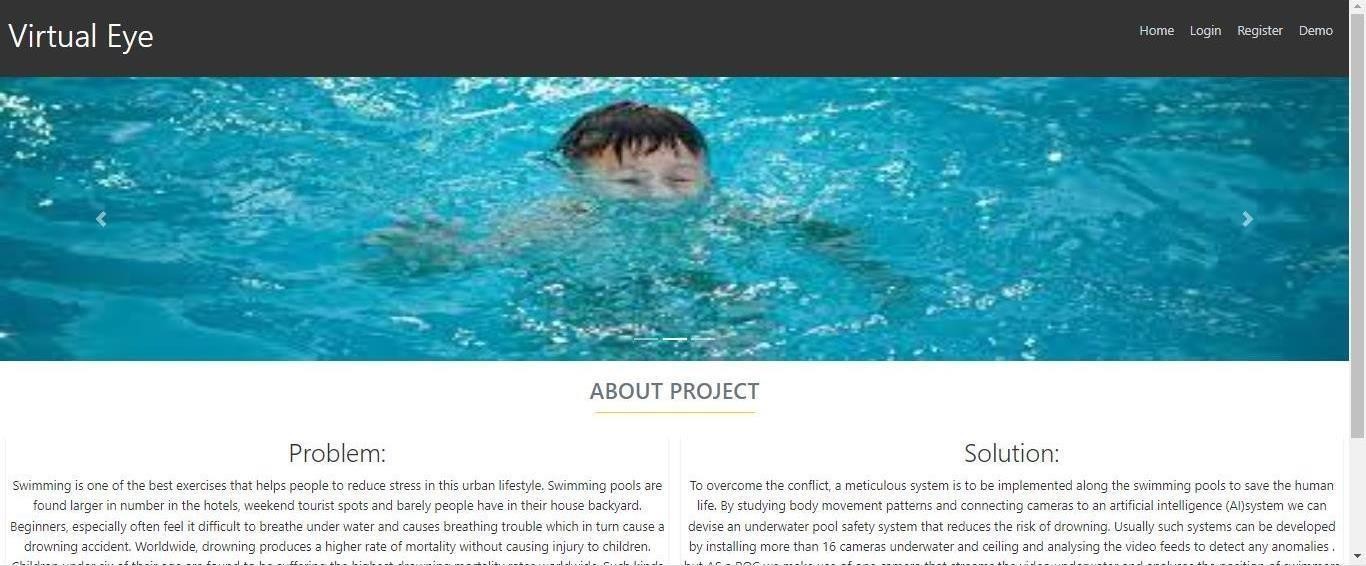
<p>Copyright Â© 2021. All Rights Reserved</p>

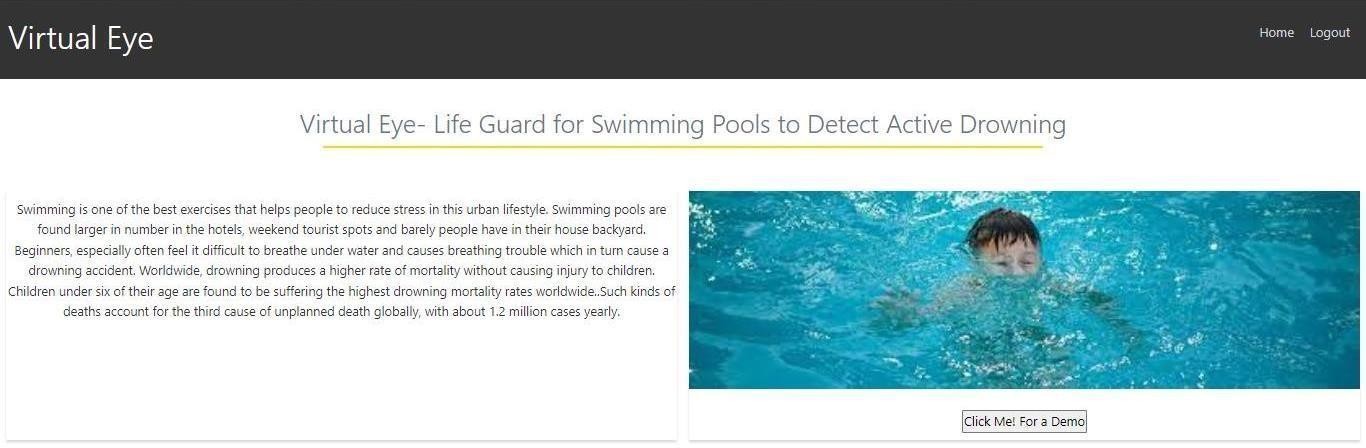
</section>

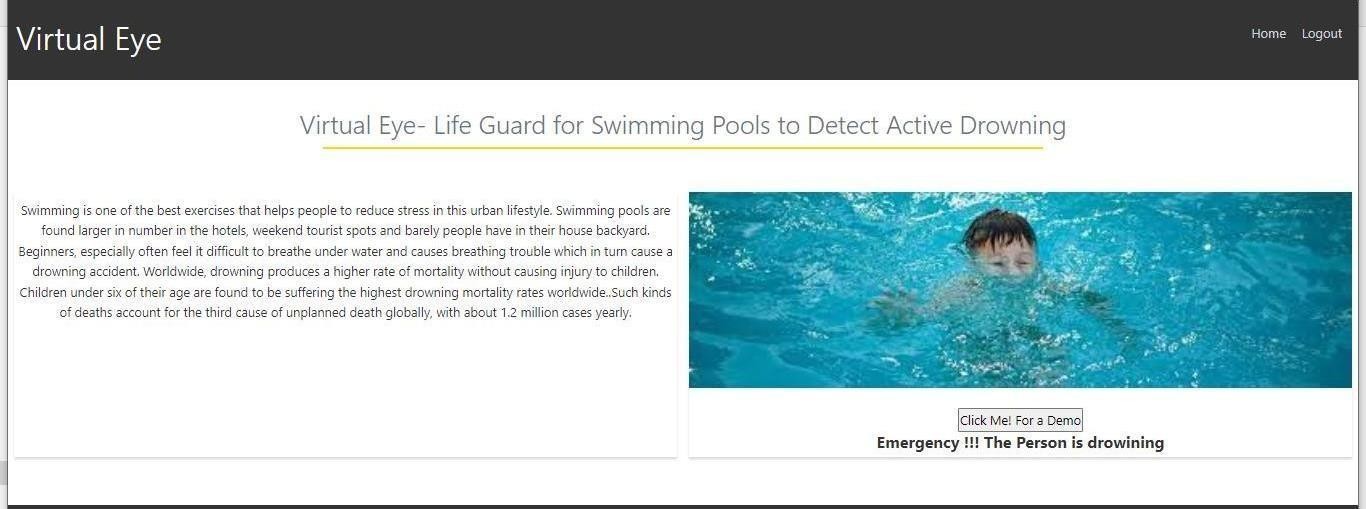
</body>

</html>









**10. ADVANTAGES & DISADVANTAGES**

# ADVANTAGES:

1. user feel comfortable and more secure
2. Children, adult, pet animal , old age people are used
3. spending more time for family, freedom for safety guards near the Swimming pool
4. Swimmers, resort are gain in the financal
5. drowning should be monitored

# DISADVANTAGE:

1. For uneducated people will suffer from this technology
2. Electricity will be required
3. Software and hardware requirement will need

**11. CONCLUSION**

* This section will draw from three core documents: ISO\_20380, HSG179, and the recently published German guideline, DGfdB R 94.15. A summary of each is given, outlining the key messages they disseminate and what this means for those involved with DDS.

ISO\_20380 This document focuses on the requirements for the installation, operation, maintenance and performance of DDS, the testing methods, and the information required from the supplier in the operating manual. These international standards do not apply to systems used in domestic pools or pools smaller than 150m2 .

* Prior to the installation of any DDS, ‘a technical study shall be carried out by the supplier in consultation with or based on information provided by the swimming pool’s owner/operator’. This is to establish the quantity and positioning of the equipment making up the system such as cameras, central processing unit, alarm tools, and other related equipment. The technical study must also provide a technical drawing of the pool basin, showing areas of ‘coverage’ and ‘non-coverage’, as well as the minimum lighting levels required above and below the water surface for the DDS to operate within performance requirements. To carry out the study, a list of factors to consider are given, outlining the variables that make each pool unique such as the architecture, and alarm reception coverage area of mobile devices to be used with the system. With this information all in one document, the technical study can be used to help optimise performance of the system, and forms part of the contract between the supplier and the pool operator. The next area of the standard is the performance requirements. This outlines the requirements needed to pass the regular maintenance testing and performance requirements for normal operation. This section covers the alarm set off time for operational performance, which is to be 15 seconds or less and displayed on the system interface. It also states that the alarm set off time must be built-in and shall not be changeable by staff. The section also discusses the areas covered by the DDS and highlights that each trained staff member must be aware of these areas. Another coverage-related requirement is that the DDS must be able to temporarily create areas where detection is disabled, to manage specific activities such as rescue drills.

12.Future Scope

This lifeguard system consists of three main components, i.e., the drowning detection, the rescuing drone, and the hazardous activity detection. All three components combined will create a system capable of detecting drowning victims, dispatching an inflatable tube using a drone (as depicted in Fig.9) and detecting hazardous activities—eventually becoming an entity that could assist a lifeguard. The system is accessible to its primary user, presumably a pool owner or a lifeguard, in the form of an interface with a sound alarm and an android mobile service that holds the capabilities of receiving Firebase notifications. Confined with a few of the hardware limitations, such as the use of a single camera and the Jetson Nano at the presence of better-quality hardware, could affect the speed and accuracy of the overall system is becoming a state-of-theart.

This limitation could be omitted with the use of multiple cameras that could be placed over the premises in several ground coordinates, increasing the accuracy of the computer vision algorithms. Moreover, due to the inability to fly a drone in extreme weather conditions such as rain, strong winds or lightning, the system is limited to be used under few specifications. As swimming in extreme weather conditions is not preferred either, the system could be further improved to emit a warning signal if a person was to swim in any of the above weather conditions, bypassing the need to fly the drone. Additionally, all the processing is done on the clientside of the applications on the Jetson Nano board, preventing any security and privacy issues that might arise due to the sensitive information inputted through the cameras. For future developments convenience wise, the system could benefit by having an additional set of cameras to identify and verify a drowning or a hazardous activity on the premises. Accessibility could also be improved by extending the Android service to be an application both in Android and iOS platforms that could hold the details of each premise individually, making a centralized system that watches over the decentralized pool premises. Both drown and hazardous activity detection could be improved by gathering a night time dataset that increases the accuracy of the data in low light.

13.Appendix

Source code

[net]

# Testing# batch=1

# subdivisions=1# Training batch=64 subdivisions=16 width=608 height=608 channels=3 momentum=0.9 decay=0.0005 angle=0 saturation = 1.5 exposure = 1.5hue=.1

learning\_rate=0.01 burn\_in=1000 max\_batches = 500200policy=steps steps=400000,450000

scales=.1,.1

[convolutional] batch\_normalize=1 filters=32 size=3 stride=1

pad=1 activation=leaky

# Downsample

[convolutional] batch\_normalize=1 filters=64 size=3 stride=2

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=32 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=64 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear#

Downsample

[convolutional] batch\_normalize=1 filters=128 size=3 stride=2

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=64 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=128 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=64 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=128 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear # Downsample

[convolutional] batch\_normalize=1

filters=256size=3 stride=2 pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=128 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=256 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=128 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=256 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=128 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=256 size=3

stride=1 pad=1

activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=128 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=256 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=128 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=256 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=128 size=1 stride=1

pad=1 activation=leaky

[convolutional]

batch\_normalize=1 filters=256 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=128 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=256 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=128 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=256 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear#

Downsample

[convolutional] batch\_normalize=1 filters=512 size=3 stride=2

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=256 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=512 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=256 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=512 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=256 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=512 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=256 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=512 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=256 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=512 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=256 size=1 stride=1

pad=1 activation=leaky

[convolutional]

batch\_normalize=1 filters=512 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=256 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=512 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=256 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=512 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear#

Downsample

[convolutional] batch\_normalize=1 filters=1024 size=3

stride=2 pad=1

activation=leaky

[convolutional] batch\_normalize=1 filters=512 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=1024 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=512 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=1024 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=512 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=1024 size=3 stride=1

pad=1

activation=leaky

[shortcut]from=- 3

activation=linear

[convolutional] batch\_normalize=1 filters=512 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1 filters=1024 size=3 stride=1

pad=1 activation=leaky

[shortcut]from=- 3

activation=linear ######################

[convolutional] batch\_normalize=1 filters=512 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1size=3 stride=1 pad=1 filters=1024 activation=leaky

[convolutional] batch\_normalize=1 filters=512 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1size=3 stride=1 pad=1 filters=1024

activation=leaky

[convolutional] batch\_normalize=1 filters=512 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1size=3 stride=1 pad=1 filters=1024 activation=leaky

[convolutional]size=1 stride=1

pad=1 filters=255 activation=linear

[yolo]

mask = 6,7,8

anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,

156,198, 373,326

classes=80 num=9 jitter=.3 ignore\_thresh = .7

truth\_thresh = 1random=1

[route] layers = -4

[convolutional] batch\_normalize=1 filters=256 size=1 stride=1

pad=1 activation=leaky

[upsample] stride=2

[route]

layers = -1, 61

[convolutional]

batch\_normalize=1 filters=256 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1size=3 stride=1 pad=1 filters=512 activation=leaky

[convolutional] batch\_normalize=1 filters=256 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1size=3 stride=1 pad=1 filters=512 activation=leaky

[convolutional] batch\_normalize=1 filters=256 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1size=3 stride=1 pad=1 filters=512 activation=leaky

[convolutional]size=1 stride=1

pad=1 filters=255 activation=linear

[yolo]

mask = 3,4,5

anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,

156,198, 373,326

classes=80 num=9 jitter=.3 ignore\_thresh = .7

truth\_thresh = 1random=1

[route] layers = -4

[convolutional] batch\_normalize=1 filters=128 size=1 stride=1

pad=1 activation=leaky

[upsample] stride=2

[route]

layers = -1, 36

[convolutional] batch\_normalize=1 filters=128 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1size=3 stride=1 pad=1 filters=256 activation=leaky

[convolutional] batch\_normalize=1 filters=128 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1size=3 stride=1

pad=1 filters=256 activation=leaky

[convolutional] batch\_normalize=1 filters=128 size=1 stride=1

pad=1 activation=leaky

[convolutional] batch\_normalize=1size=3 stride=1 pad=1 filters=256 activation=leaky

[convolutional]size=1 stride=1

pad=1 filters=255 activation=linear

[yolo]

mask = 0,1,2

anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,

156,198, 373,326

classes=80 num=9 jitter=.3 ignore\_thresh = .7

truth\_thresh = 1 random=1

**Source code(ii)**

#import necessary packagesimport cv2

import os

import numpy as np

from .utils import download\_file

initialize = Truenet

= 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 scriptclass\_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[0] - 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 blueif 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 = i[0]

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

Github link

<https://github.com/IBM-EPBL/IBM-Project-42896-1660710969>

Demo link

<https://drive.google.com/drive/folders/1S8QFmMeKThA8H03btPe1unMV5Fgz5Keq>