

FINAL DELIVERABLE PROJECT DOCUMENTATION

Date	19 November 2022
Team ID	PNT2022TMID40673
Project Name	VirtualEye-Lifeguard for Swimming Poolsto Detect the Active Drowning

SUBMITTED BY

TEAM LEADER :SUSITHRA.S(610919104093)

**TEAM MEMBERS :ANUSHA.S(610919104008)
JANCYRANI(610919104030)**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

JAYALAKSHMI INSTITUTE OF TECHNOLOGY

THOPPUR,DHARMAPURI

INDEX

1. INTRODUCTION

- 1.1.Project Overview
- 1.2.Purpose

2. LITERATURE SURVEY

- 2.1.Existing problem
- 2.2.References
- 2.3.Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- 3.1.Empathy Map Canvas
- 3.2.Ideation & Brainstorming
- 3.3.Proposed Solution
- 3.4.Problem Solution fit

4. REQUIREMENT ANALYSIS

- 4.1.Functional requirement
- 4.2.Non-Functional requirements

5. PROJECT DESIGN

- 5.1.Data Flow Diagrams
- 5.2.Solution & Technical Architecture
- 5.3.User Stories

6. PROJECT PLANNING & SCHEDULING

- 6.1.Sprint Planning & Estimation
- 6.2.Sprint Delivery Schedule
- 6.3.Reports from JIRA

7. CODING & SOLUTIONING

- 7.1.Feature 1
- 7.2.Feature 2

8. TESTING

- 8.1.Test Cases
- 8.2.User Acceptance Testing

9. RESULTS

- 9.1.Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12.FUTURE SCOPE

13. APPENDIX

13.1.Source Code

13.2.GitHub & Project Demo Link

CHAPTER 1

1.INTRODUCTION

Recently, there has been growing interest around the topic of drowning detection systems (DDS) in the sport and leisure industry both across the UK and globally. Advancements in technology, coupled with the importance of pool safety, has led to its growing prominence, with mention of DDS now in documents such as HSG179 - the latest UK standards document for health and safety in swimming pools (Health and Safety Executive, 2018). However, the topic is a debated area for various reasons explored in this review.

Whilst there are plenty of academic articles dedicated to the technology and design behind these products in the fields of biometrics, computer science and electronic engineering, there is limited academic research investigating their application to real-world scenarios. Furthermore, there is uncertainty around their use alongside traditional lifeguarding; whether international testing standards (ISO standards) are robust enough; and general risks affecting the effectiveness of these products. This includes factors such as water clarity, high pool occupancy, lighting, glare and attractions such as water slides and wave machines.

This piece will begin with an overview of the different definitions of DDS, followed by an explanation of the aims and methodology of this review. It will then discuss what the current DDS standards are alongside legislation and guidance available around DDS, , and provide a summary of the shared responsibilities towards the effective operation of DDS. Following this, the literature review will examine the co-existence between DDS and traditional lifeguarding, provide an analysis of its impact so far, and conclude with recommendations on the direction of future DDS research.

Project Overview



Purpose

>> Establish and outline what is known on Drowning Detection Systems. >> Evaluate the current literature on Drowning Detection Systems, including their use in indoor pool environments along with interaction with traditional lifeguarding. >> Better understand where DDS are positioned in the health and safety landscape of indoor swimming pools.

The value that can be generated from these aims stem from the recognition that currently, there are no published documents drawing together all the current DDS research. The literature review aims to contribute as independent research in this field and hopes to signpost the potential future direction of DDS research.

CHAPTER 2

2 LITERATURE SURVEY

Of the differing definitions of DDS, most outline three defining elements:

- 1) surveillance,
- 2) detection of a pool user in difficulty, and
- 3) raising an alarm

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 coexistence 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).

REFERENCES

- [1]AngelEye. (2019). AngelEye – Distributors. Retrieved from:
<https://www.angeleye.it/news.php?id=28&newscat=10>
- [2]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
- [3]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>
- [4]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/?pid=000000000030360257>

PROBLEM STATEMENT DEFINITION

1. Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle.

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.

CHAPTER 3

3.IDEATION & PROPOSED SOLUTION

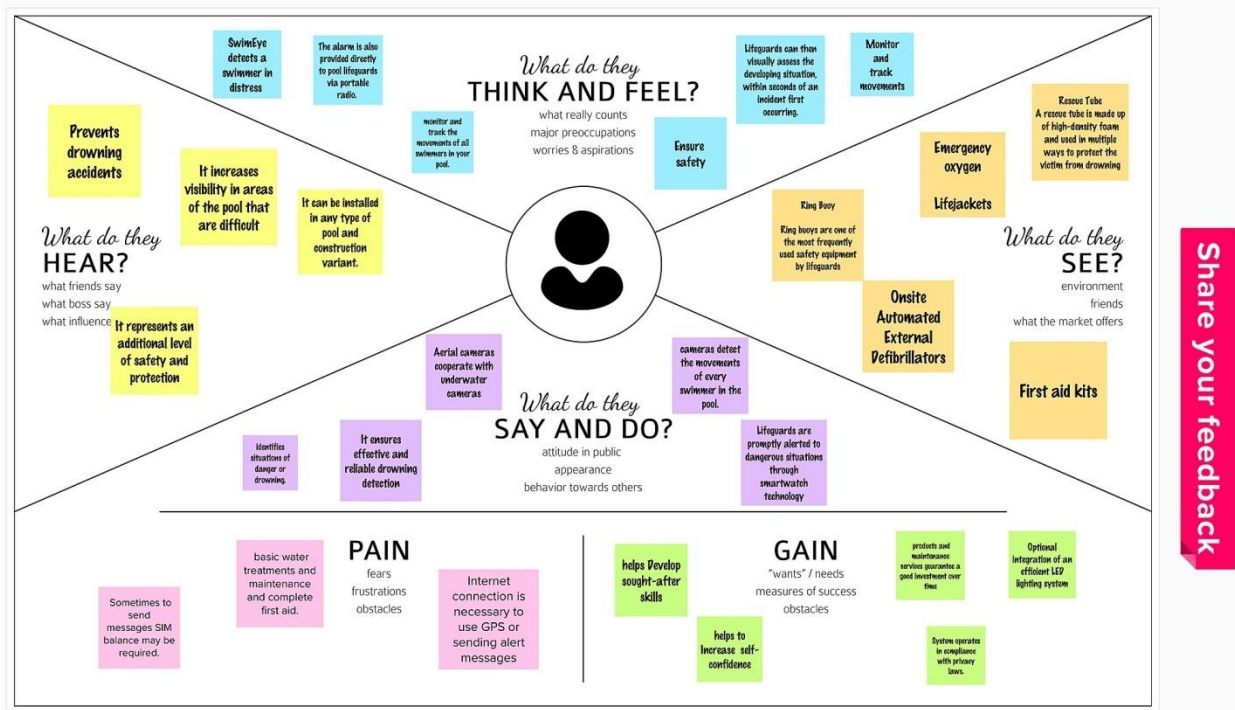
EMPATHY MAP CANVAS

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.



IDEATION & BRAINSTORMING

[illegible]

PROPOSED SOLUTION

Proposed Solution Template:

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 our system is track the people position and body condition in the drowning using YOLO

		Algorithm.
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 we should avoid the accident
	Business Model (Revenue Model)	Safety innovation in the swimming related issues this makes attractive for end users to use our software product
5.	Scalability of the Solution	IBM cloud server will collect all the data and stored in the server. This will more safe and secure

PROBLEM SOLUTION FIT

Problem-Solution fit canvas 2.0 Purpose / Vision

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Who is your customer? Designed for whom has to guarantee every day the safety in public and intensive-use pools, VirtualEye -LifeGuard detects potential drownings and promptly notifies you.	6. CUSTOMER CC What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices. 1. Teach basic swimming and water safety skills to school-age children aged 6 years and older. 2. Training bystanders in safe rescue and resuscitation. 3. Use life guard while swimming.	5. AVAILABLE SOLUTIONS AS Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pros and cons is no alternative to digital monitoring. 1. There are many actions to prevent drowning. Installing barriers (e.g. covering wells, using doorway barriers and playpens, fencing swimming pools etc.) to control access to water hazards, or removing water hazards entirely greatly reduces water hazard exposure and risk.	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS J&P Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. 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.	9. PROBLEM ROOT CAUSE RC What is the real reason that this problem exists? What is the backstory behind the need to do this job? 1. The most common cause of drowning is not knowing how to swim. Many adults and children will attempt to get into the water without proper swim training. 2. Drowning is the 3rd reason for the highest unintentional deaths, and that's why it is necessary to create trustable security mechanisms.	7. BEHAVIOUR BE What does your customer do to address the problem and get the job done? i.e. directly related; find the right solar panel installer, calculate usage and benefits; indirectly In order to quickly help lifesavers judge whether people are drowning in the swimming pool,	
3. TRIGGERS TR What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news. Drowning Awareness through the Advertisements and other social medias like Instagram, Facebook, LinkedIn & News etc. 4. EMOTIONS BEFORE / AFTER EM How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it as your communication strategy & design. 1. (i) struggle to keep the airway clear of the water, (ii) initial submersion and breath-holding, (iii) aspiration of water, (iv) unconsciousness, (v) cardio-respiratory arrest and (vi) death - inability to revive. 2. In order to quickly help lifesavers judge whether people are drowning in the swimming pool.	10. YOUR SOLUTION SL If you are working on an existing business, write down your current solution first fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour. 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.	8. CHANNELS OF BEHAVIOUR CH 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 Search engines are considered to be the best marketing channel for acquiring new leads for drowning detection system. Social media continues to grow every year as a powerhouse marketing channel. 8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. Local media advertising- Advertising on both television and radio is still very effective in creating brand awareness about drowning in the swimming pool.	Extract online & offline CH of BE	

Identify strong TR & EM

Focus on J&P, fit into BE, understand RC

Extract online & offline CH of BE

CC BY NC ND

Copyright © 2020 Amaltama. All rights reserved. NoDerivatives 4.0 license Created by Dana Niyalathna / Amaltama.com

AMALTAMA

CHAPTER 4

4.REQUIREMENT ANALYSIS FUNCTIONAL REQUIREMENT

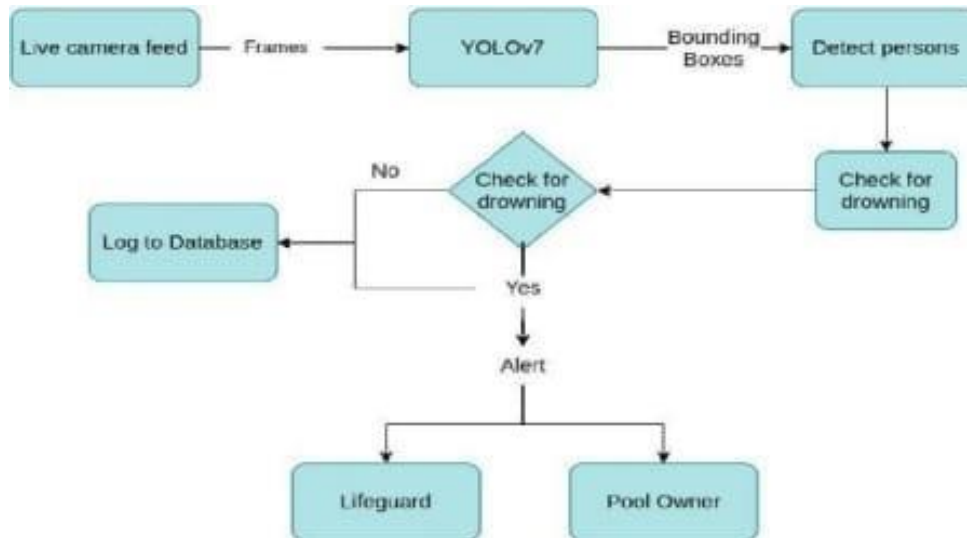
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Installation	Install the camera inside the underwater, connect necessary app in the phone or other device
FR-2	Detection	Near swimming pool area use detection room for monitor or use IBM cloud for storage purpose of the details
FR-3	Audio	Give the alert signal for the people entering into 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 occurred give the alert signal for the entire pool

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 given by the lifeguard of swimmer
NFR-3	Reliability	Triggers if any immediate needs of the swimmer inside the pool
NFR-4	Performance	If any unwanted position changes and the pulse rate will decrease this will detect it.
NFR-5	Availability	Equipment and other requirement should be checked by the lifeguards
NFR-6	Scalability	Virtual eye lifeguard detects potential drownings and it should be notified to you.

CHAPTER 5

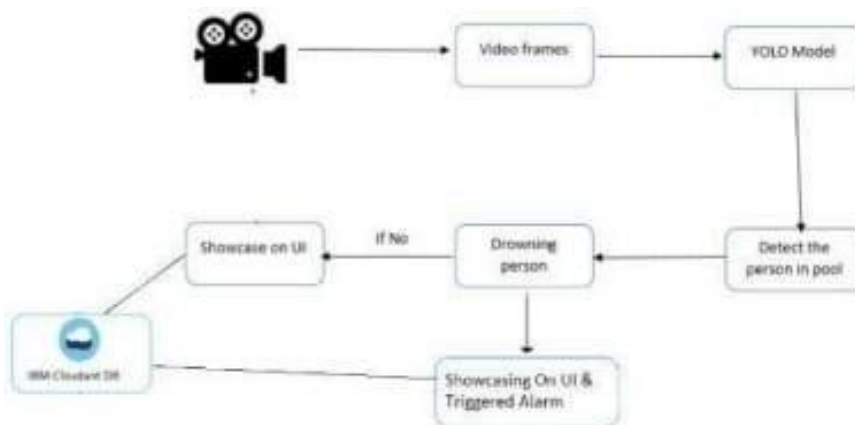
5.PROJECT DESIGN DATA FLOW DIAGRAMS



SOLUTION & TECHNICAL ARCHITECTURE

Solution Architecture:

❖ Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analyzing the video feedsto 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



USER 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 the people	USN-2	As a user, I can secure the drowning persons from 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 executive to resolve any issues	Medium	Sprint-3
Administrator	Dashboard	USN-5	Management of the drowning detection system and database management	I can access the system's logs and any other data instantly	High	Sprint-4

CHAPTER 6

6. PROJECT PLANNING & SCHEDULING SPRINT 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

SPRINT DELIVERY SCHEDULE

sprint	functional requirement (epic)	user story number	user story/task	story points	priority	team members
Sprint-1	Registration	VLGFSPTDA-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	high	Susithra.S
Sprint-1	Registration	VLGFSPTDA-2	As a user, I	1	high	Anusha.S

			will receive confirmati on email once I have registered for the application			
Sprint-1	Registration	VLGFSPTDA-3	As a user, I can register for the application through Gmail	2	low	Jancyrani.V
Sprint-1	Registration	VLGFSPTDA-4		2	medi um	Susithra.S
Sprint-1	Login	VLGFSPTDA-5	As a user, I can register for the application through Gmai	1	high	Anusha.S
Sprint-2	Dataset collect	VLGFSPTDA-6	Collect number of databasets and get accuracy	2	high	Jancyrani.V
Sprint-2	Pre- processing	VLGFSPTDA-7	The dataset is extracted	2	medi um	Susithra.S
Sprint-3	Train the	VLGFSPTDA-8	Train the	4	high	Anusha.S

	model		model			vx
Sprint-3	Test the model	VLGFSPTDA-9	if the person is drowning the system will ring the alarm	6	high	SUsithra.S
Sprint-4	Detection	VLGFSPTDA-10	system trained the model to predict the output	3	high	Jancyrani.V
Sprint-4	Detection	VLGFSPTDA-11	I detect drowning	7	high	Anusha.S

REPORT FROM JIRA

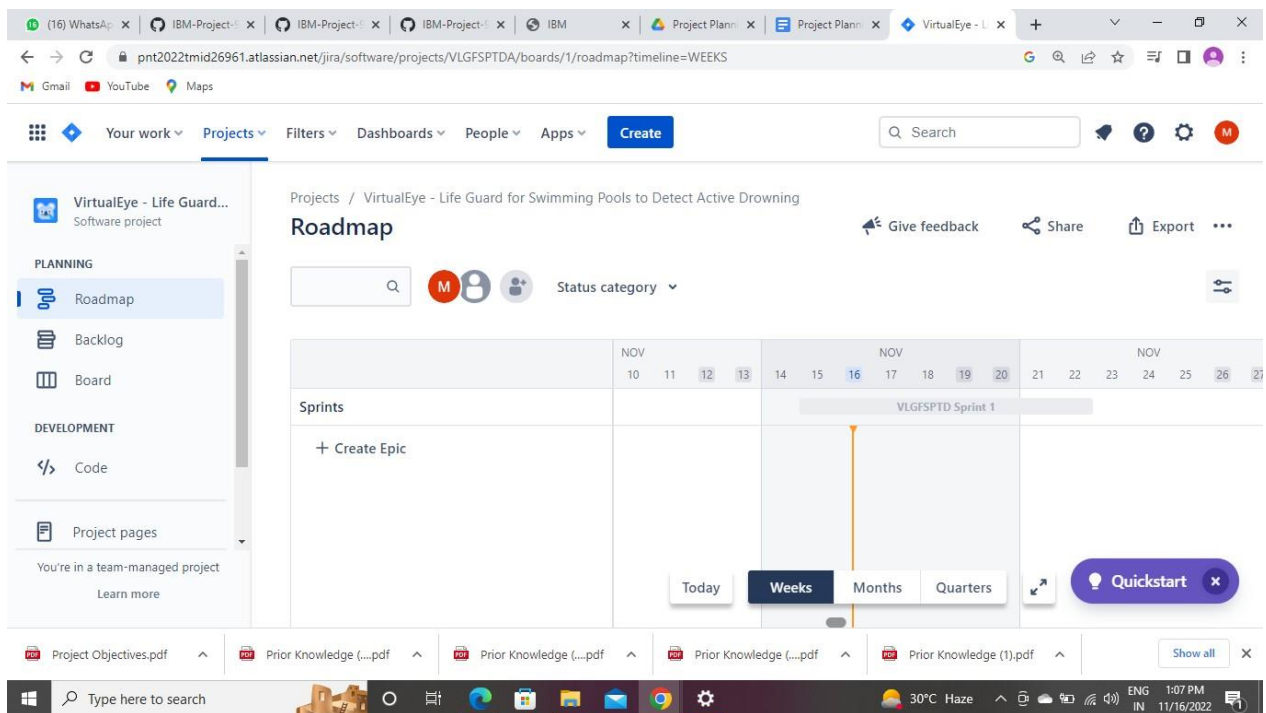
BACKLOG

The screenshot displays the Jira backlog for the project "VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning". The backlog is organized into a sprint named "VLGFSPTD Sprint 2" running from 31 Oct to 5 Nov, containing 4 issues. The issues listed are:

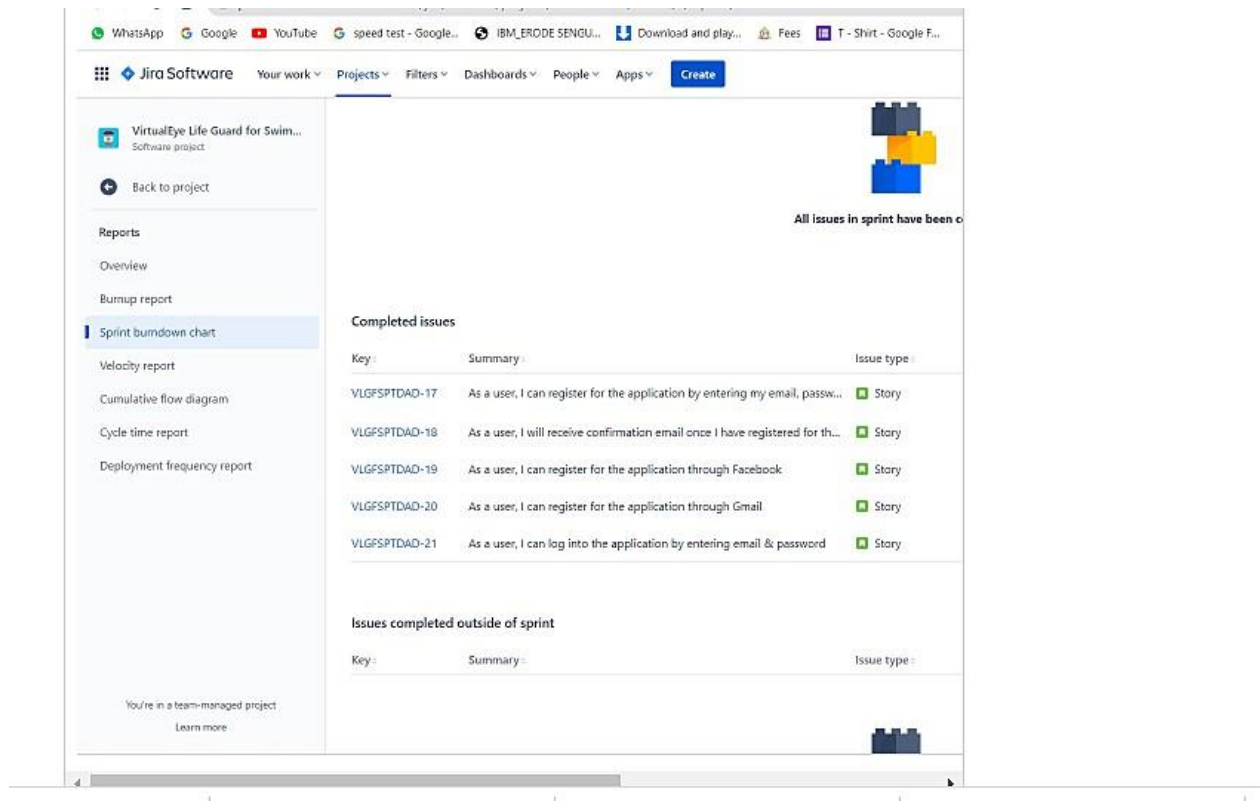
- VLGFSPTDA-8 dataset collect (Status: DONE)
- VLGFSPTDA-9 pre-processing
- VLGFSPTDA-10 Train the Model

The interface includes a sidebar with navigation options: PLANNING (Roadmap, Backlog, Board) and DEVELOPMENT (Code). The top navigation bar shows "Your work" and "Projects" with a search bar. A "Quickstart" button is visible in the bottom right corner of the backlog view.

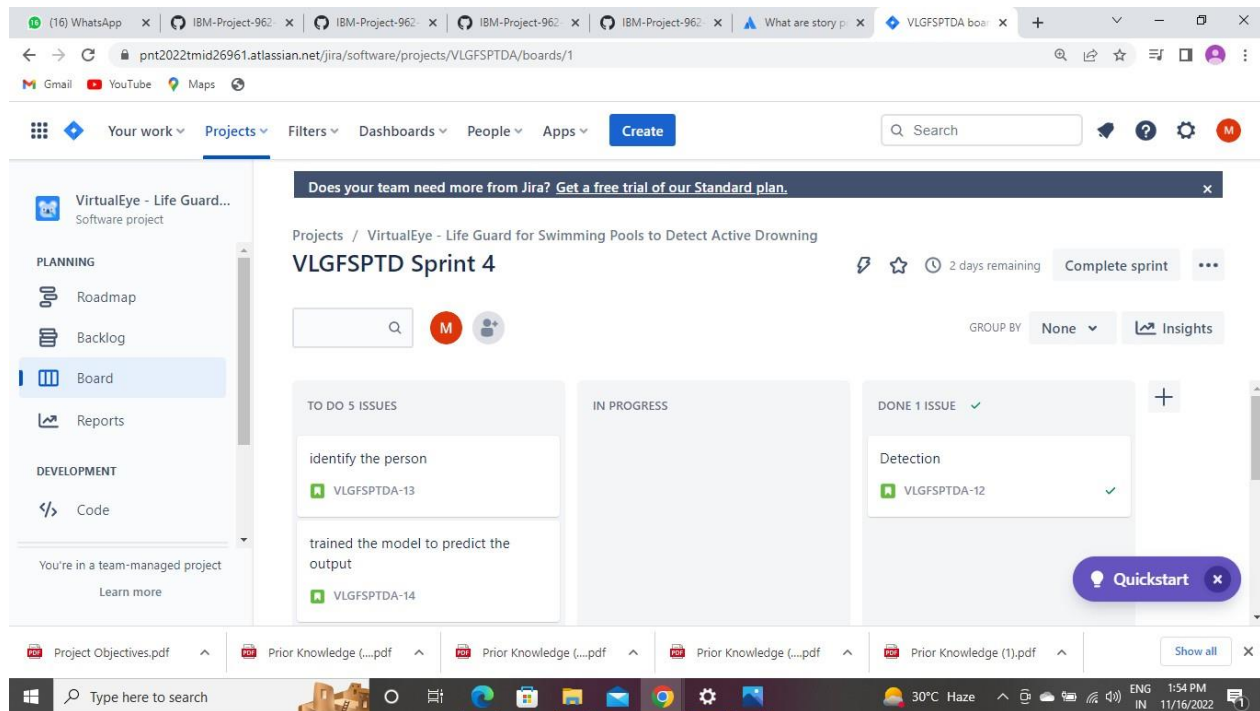
ROAD MAP



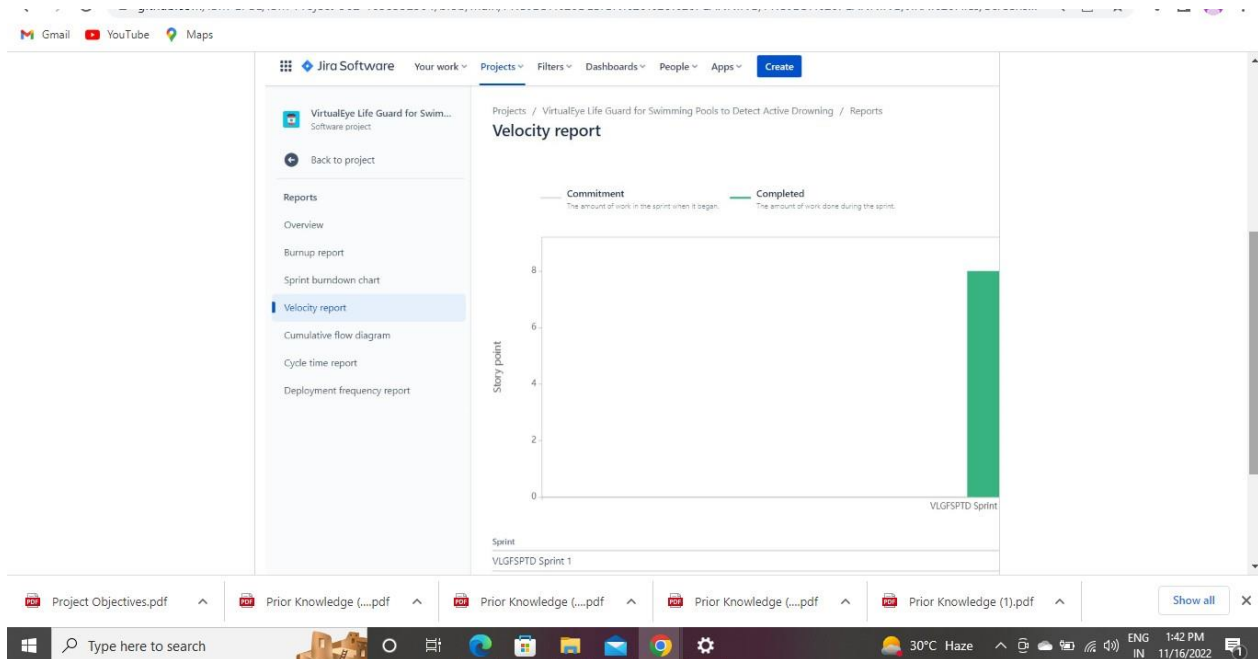
CHART



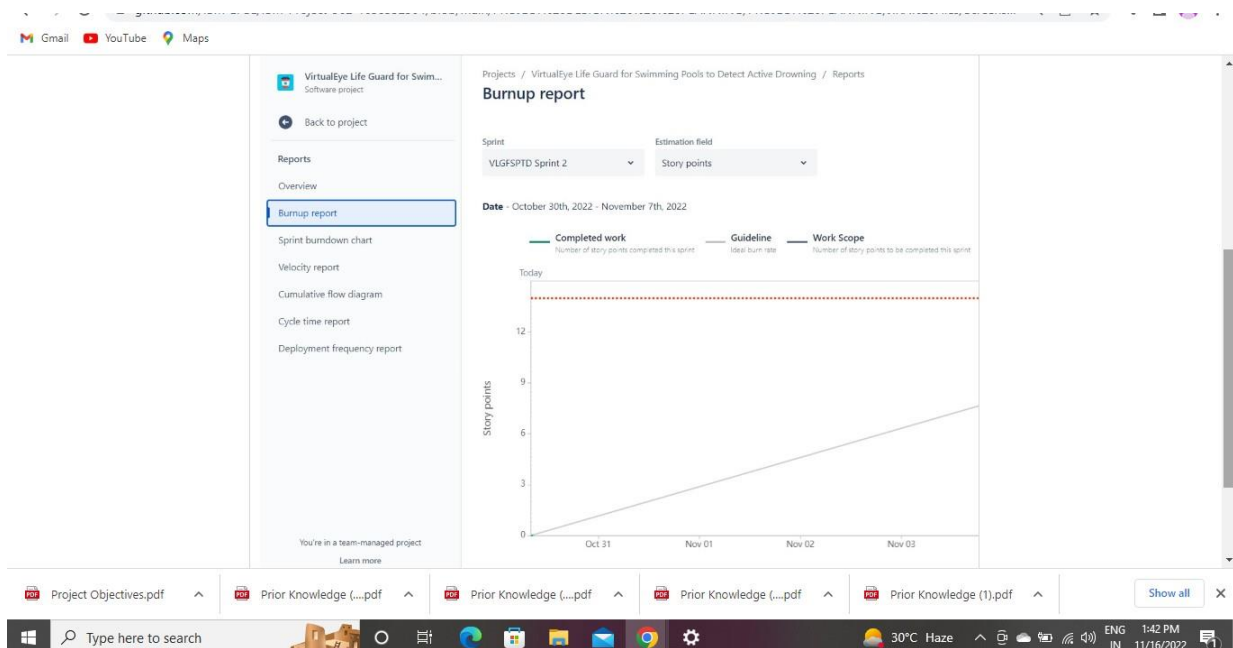
BOARD(KANBAN)



VELOCITY REPORT



BURNUP REPORT



CHAPTER 7

7.CODING & SOLUTION

FEATURE 1

INDEX.HTML

```
<!-- NAVIGATION MENUS -->
<!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" />
<style>
* {
margin: 0;
padding: 0;
box-sizing: border-box;
}
```

```
body {
font-family:cursive;
}
a {
text-decoration:none;
}
li{
list-style:none;
}
.navbar {
display:flex;
align-items:center;
justify-content:space-between;
padding:20px;
background-color:teal;
color:#fff;
}
.nav-links a {
color:#fff;
}
/* LOGO */
.logo {
font-size:32px;
}
/* NAVBAR MENU */
.menu {
display:flex;
gap: 1em;
font-size:18px;
}
.menu li:hover {
background-color:#4c9e9e;
border-radius:5px;
transition:0.3s ease;
}
.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;
}
.menu li {
padding:5px 14px;
}
.services {
position:relative;
}
.dropdown {
background-color:rgb(1, 139, 139);
padding: 1em 0;
position:absolute; /*WITH RESPECT TO PARENT*/
display:none;
border-radius:8px;
top:35px;
}
.dropdown li + li {
margin-top:10px;
}
.dropdown li {
padding:0.5em 1em;
width:8em;
text-align:center;
}
.dropdown li:hover {
background-color: #4c9e9e;
```

```
}
.services:hover
.dropdown {
display: block;
}
#example1 {
background: url();
}
#swim
{
height: 200px;
width: 50%;
}
</style>
<title>Document</title>
</head>
<body>
<nav class="navbar">
<!-- LOGO -->
<div class="logo">VIRTUAL EYE</div>
<!-- NAVIGATION MENU -->
<ul class="nav-links">
<!-- USING CHECKBOX HACK -->
<div class="menu">
<li><a href="/">Home</a></li>
<li><a href="/">About</a></li>
<li class="services">
<a href="/">Services</a>
</li>
<li><a href="/register">Register</a></li>
<li><a href="/login">Login</a></li>
</div>
</ul>
</nav>
<div class="swim">

<a href="/result"><button style = "position:absolute;
right:60px; bottom:45px; height:40px; width:500px;
color:cyan; background:black;">TRY THIS PROJECT IN DEMO
VERSION (CLICK HERE)</button></a>
</div>
</body>
</html>
```

LOGIN.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title</title>
<style>
* {box-sizing: border-box}
/* Add padding to containers */
.container {
padding: 16px;
}
/* Full-width input fields */
input[type=text], input[type=password] {
width: 100%;
padding: 15px;
margin: 5px 0 22px 0;
display: inline-block;
border: none;
background: #f1f1f1;
}
input[type=text]:focus, input[type=password]:focus {
background-color: #ddd;
outline: none;
}
/* Overwrite default styles of hr */
hr {
border: 1px solid #f1f1f1;
```



```

margin-bottom: 25px;
}
/* Set a style for the submit/register button */
.registerbtn {
background-color: #04AA6D;
color: white;
padding: 16px 20px;
margin: 8px 0;
border: none;
cursor: pointer;
width: 100%;
opacity: 0.9;
}
.registerbtn:hover {
opacity:1;
}
/* Add a blue text color to links */
a {
color: dodgerblue;
}
/* Set a grey background color and center the text of the
"sign in" section */
.signin {
background-color: #f1f1f1;
text-align: center;
}
</style>
</head>
<body>
<form>
<div class="container">
<h1>LOGIN</h1>
<p>Login with your credentials.</p>
<hr>
<label for="email"><b>Email</b></label>
<input type="text" placeholder="Enter Email"
name="email" id="email" required>

```

```
<label for="psw"><b>Password</b></label>
<input type="password" placeholder="Enter Password"
name="psw" id="psw" required>
<p><a href="/index" class="registerbtn">LOGIN</a>.</p>
</div>
</form>
</body>
</html>
```

REGISTER.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title</title>
<style>
* {box-sizing: border-box}
/* Add padding to containers */
.container {
padding: 16px;
}
/* Full-width input fields */
input[type=text], input[type=password] {
width: 100%;
padding: 15px;
margin: 5px 0 22px 0;
display: inline-block;
border: none;
background: #f1f1f1;
}
input[type=text]:focus, input[type=password]:focus {
background-color: #ddd;
outline: none;
}
/* Overwrite default styles of hr */
hr {
border: 1px solid #f1f1f1;
margin-bottom: 25px;
```

```

}
/* Set a style for the submit/register button */
.registerbtn {
background-color: #04AA6D;
color: white;
padding: 16px 20px;
margin: 8px 0;
border: none;
cursor: pointer;
width: 100%;
opacity: 0.9;
}
.registerbtn:hover {
opacity:1;
}
/* Add a blue text color to links */
a {
color: dodgerblue;
}
/* Set a grey background color and center the text of the
"sign in" section */
.signin {
background-color: #f1f1f1;
text-align: center;
}
</style>
</head>
<body>
<form>
<div class="container">
<h1>Register</h1>
<p>Please fill in this form to create an account.</p>
<hr>
<label for="email"><b>Email</b></label>
<input type="text" placeholder="Enter Email"
name="email" id="email" required>
<label for="psw"><b>Password</b></label>

```

```

<input type="password" placeholder="Enter Password"
name="psw" id="psw" required>
<label for="psw-repeat"><b>Repeat Password</b></label>
<input type="password" placeholder="Repeat Password"
name="psw-repeat" id="psw-repeat" required>
<hr>
<p><a href="/index" class="registerbtn">Register</a>.</p>
</div>
<div class="container signin">
<p>Already have an account? <a href="/login">Sign
in</a>.</p>
</div>
</form>
</body>
</html>

```

FEATURE 2

```

#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[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 blue
        if label == 'person' and Drowning:
            color = COLORS[0]
        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/DrowningDetector/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

```

CHAPTER 8

8.TESTING TEST CASES

				18-Nov-22 PNT022TMD26961 Virtual eye-lifeguard for swimming 4 marks			
Test case ID	Feature Type		Test Scenario	Steps To Execute	Test	Expected	Result
	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
		Home Page					
LoginPage_TC_002		Home Page	Verify the UI elements 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 with orange colour d. New customer? Create a 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 a account link e. Last password? Recovery password link	Working as expected
	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 in button	Username: lax@gmail password: lax26	User should navigate to prediction homepage	working as
	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 in button	Username: lax password: lax26	Application should show 'incorrect email or password ' validation message.	working as
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 in button	username: lax26@gmail password: lax26	Application should show 'incorrect email or password ' validation message.	working as
	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 password in password text box 5. Click on in button	username: lax26@gmail password: 1803	Application should show 'incorrect email or password ' validation message.	working as
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

USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT)

1. Defect Analysis

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

Section	Total Cases	Not Tested	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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
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

Version Control	1	0	0	1
-----------------	---	---	---	---

CHAPTER 9

9.RESULT

PERFORMANCE METRICS

LOGIN.HTML

<!DOCTYPE html>

```
<html >
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,initialscale=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='t
ext/css'>
<link
href='https://fonts.googleapis.com/css?family=Hind:300'rel='stylesheet'typ
e='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;  
}  
.login{  
margin-top:100px;  
}
```

```
form {border:3px solid #f1f1f1; 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:100%;
font-weight:bold;
}
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%;
```

LOGOUT.HTML

```
<!DOCTYPE html>
<html >
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initialscale=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'
t'>
<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;
eft: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 forspan 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>

```

BASE.HTML

```

<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initialscale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>High Quality 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"></scri
pt>
<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"></scrip
t>
<link href="{{url_for('static',filename='css/main.css')}}"rel="stylesheet">

```

```

<style>
.bg-dark{
background-color:#42678c!important;
}
#result {
color:#0a1c4ed1;
}
</style>
</head>
<body style="background-color:black";>
<headerid="head"class="header">
<section id="navbar">
<h1 style="color:white;">Virtual Eye</h1>
<div class="nav--items">
<ul>
<!-- <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" >
<h2><em style="color:white">High Quality Drowning
Recognition</em></h2>
<br><p><h5><i style="color:white;">Drowning Detection Using Virtual
Eye</i></h5></p>

</div>
<div class="col-sm-6">
<div>
<h4 style="color:white;">Upload Image Here</h4>
<form action = "http://localhost:5000/"id="upload-file" method="post"

```



```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initialscale=1.0">
<!--Bootstrap -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.mi
n.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJISAWiGgFAW/d
AiS6JXm" crossorigin="anonymous">
<script src="https://code.jquery.com/jquery3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkVYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93h
XpG5KkN" crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.
min.js" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvf
a0b4Q" crossorigin="anonymous"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.j
s" integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PV
CmYI" 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>
```

```
</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">

</div>
<div class="carousel-item">

</div>
<div class="carousel-item">

</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">
```


Next

</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 barely people have in their house backyard. Beginners, especially often feel 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 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.

</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. By studying body movement patterns and connecting cameras to an artificial intelligence (AI)system 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 than an alert will be generated to attract lifeguards attention.

</p>

</div>

</div>

<div class="bottom">

<p >

Note : 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 easily observe.

</p>

</div>

</section>

<section id="footer">

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

<div class="social">

<i class="fab fa-2x fa-twitter-square"></i>

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

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

</div>

</section>

</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, initialscale=1.0">


```
<!--Bootstrap -->
<link
rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/cs
s/bootstrap.min.css"integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJISAWiGgFAW/d
AiS6JXm" crossorigin="anonymous">
<script src="https://code.jquery.com/jquery3.2.1.slim.min.js"integrity="sha384-
KJ3o2DKtIkVYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93h
XpG5KkN" crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.
min.js" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvf
a0b4Q" crossorigin="anonymous"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.j
s" integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PV
CmYI" 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">
<div class="nav--items">
<ul>
<!-- <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 class="header">Virtual Eye- Life Guard for
Swimming 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 barely people have in their house
backyard. Beginners, especially often feel 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 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.
</p>
</div>
<div class="left">
<div class="prediction-input">

</br>
<form id="form" action="/result"method="post"enctype="multipart/formdata">
<input type="submit"class="submitbtn"value="Click Me! For a Demo">
</form>
</div>
<h5 style="text-color:Red">
```

```
<b style="text-color:Red"><b>
</h5>
</div>
</div>
</section>
</br></br>
<section id="footer">
<p>Copyright Â© 2021. All Rights Reserved</p>
</section>
</body>
</html>
```

REGISTER.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+Conde
nsed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.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;  
}  
.login{  
margin-top:100px;  
}  
form {border: 3px solid #f1f1f1; 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:100%;  
}  
button:hover {  
opacity:0.8;  
}  
.cancelbtn {  
width:auto;
```

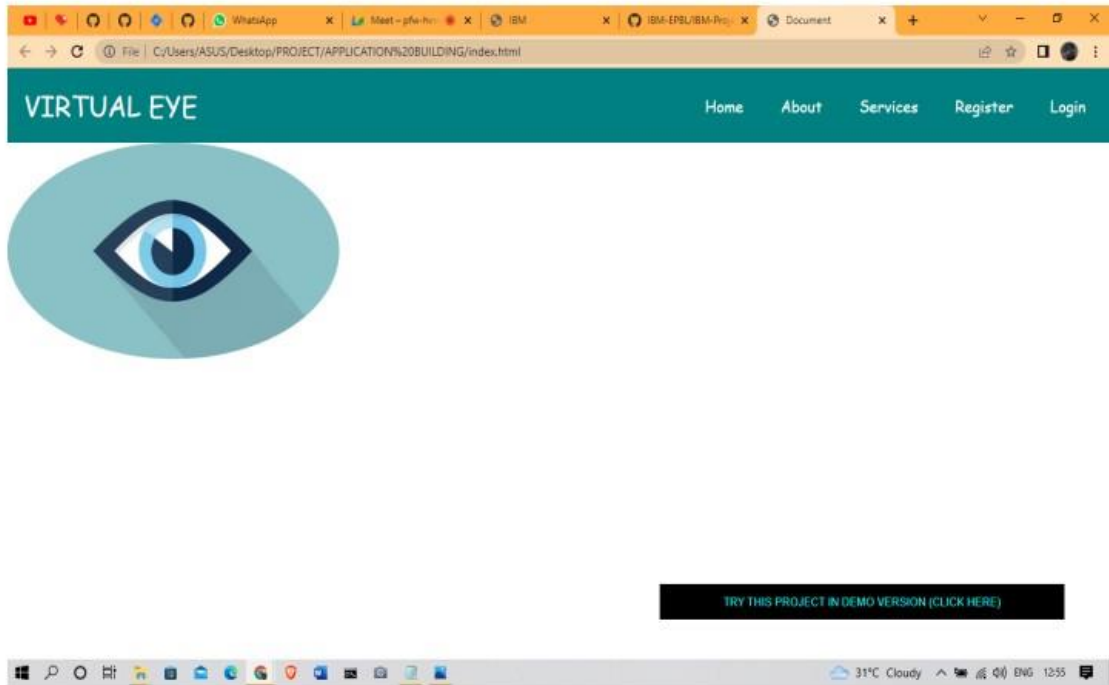
```

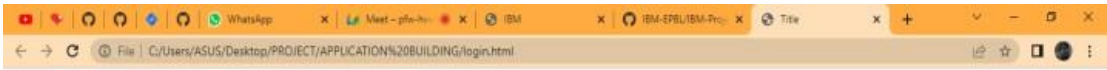
padding:10px 18px;
background-color:#f44336;
}
.imgcontainer {
text-align:center;
margin:24px 0 12px 0;
}
img.avatar {
width:30%;
border-radius:50%;
}
.container {
padding:16px;
}
span.psw {
float:right;
padding-top:16px;
}
/* Change styles for span and cancel button on extra small screens*/
@media screen 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" >
<a href="{{ url_for('home')}}">Home</a>
<a href="{{ url_for('login')}}">Login</a>

```

```
<a class="active"href="{{ url_for('register')}}">Register</a>
</div>
</div>
<div id="login" class="login">
<form action="{{url_for('afterreg')}}"method="post">
<div class="imgcontainer">

</div>
<div class="container">
<input type="text" placeholder="Enter Name" name="name" required><br>
<input type="email" placeholder="Enter Email ID" name="_id" required><br>
<input type="password" placeholder="Enter Password" name="psw"
required>
<button type="submit">Register</button><br>
</div>
<div class="container"style="background-color:#f1f1f1">
<div class="psw">Already have an account?&nbsp;&nbsp; <a href="{{
url_for('login') }}">Login</a></div>
</div>
</form>
</div>
</body>
</html>
```



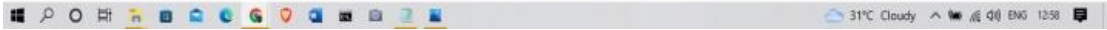


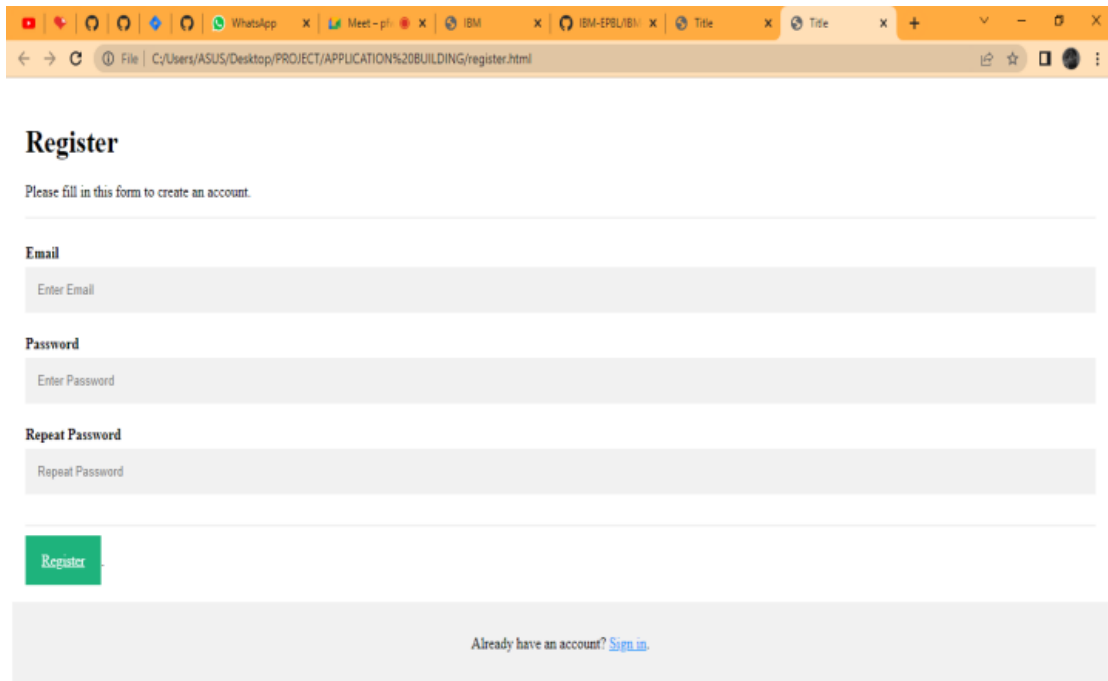
LOGIN

Login with your credentials.

Email

Password





The screenshot shows a web browser window with multiple tabs. The active tab displays a registration form titled "Register". The form includes a heading "Register", a subheading "Please fill in this form to create an account.", and three input fields labeled "Email", "Password", and "Repeat Password". Each input field has a placeholder text "Enter Email", "Enter Password", and "Repeat Password" respectively. A green "Register" button is located below the input fields. At the bottom of the form, there is a link that says "Already have an account? [Sign in.](#)".

CHAPTER 10

10.ADVANTAGES & DISADVANTAGES

ADVANTAGES

- (i) user feel comfortable and more secure
- (ii) Children, adult, pet animal , old age people are used

- (iii) spending more time for family, freedom for safety guards near the Swimming pool
- (iv) Swimmers, resort are gain in the financial
- (v) drowning should be monitored

DISADVANTAGES

- (i) For uneducated people will suffer from this technology
- (ii) Electricity will be required
- (iii) Software and hardware requirement will need

CHAPTER 11

11.CONCLUSION

This section will draw from three core documents: ISO_20380, HSG179, and the recently published German guideline, DGf dB 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.

CHAPTER 12

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-the-art. 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.

CHAPTER 13

13. APPENDIX

(i) SOURCE CODE

Init.py

```
from .object_detection import detect_common_objects  
Object_detect.py
```

```

#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/DrowningDetector/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 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.
    #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/DrowningDetector/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
Detect.py
import cvlib as cv
from cvlib.object_detection import draw_bbox
import cv2
import time
import numpy as np
from playsound import playsound
#for PiCamera
#from picamera Import PiCamera
#camera = PiCamera
#camera.start_preview()
# open webcam
webcam = cv2.VideoCapture(0)
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
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')
# press "Q" to stop
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
# release resources
webcam.release()
cv2.destroyAllWindows()
```

GITHUB LINK:<https://github.com/IBM-EPBL/IBM-Project-52225-1660991533>