12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

VirtualEye - Lifeguard for swimming pools to detect active drowning

1. INTRODUCTION

1.1 PROJECT OVERVIEW

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

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

1.2 PURPOSE

VirtualEye is a computer vision detection system for the prevention of drowning incidents in swimming pools. VirtualEye works like an “extra lifeguard” under the water of your pool. Our object recognition software tracks the movements of all swimmers in a pool. And in the event of a serious drowning incident, SwimEye will provide an alarm to pool lifeguards. This will help lifeguards improve their reaction-time, as they initiate a rescue.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

Object detection using different techniques:

It is claimed that usage of convolutional neural network architecture in deep neural networks has added a significant shift in learning more complicated informative characteristics in images as compared to older techniques. Furthermore models such as Fast R-CNN, Faster R-CNN and YOLO have been constructed since the region based convolutional neural network architecture proposal. Fast R-CNN improves bounding box improves bounding box regression and classification. Faster R-CNN which generates area suggestions using an extra subnetwork and YOLO which detects objects using a fixed grid regression is used to recognize generic objects based on CNN architectures. Local contrast enhancement and pixel level segmentation on the other hand are used to recognize salient objects. The techniques used in detecting objects will be crucial as they establish the groundwork for the methodologies used to identify drowning and hazardous activities.

Drowning detection and tracking:

To avoid drowning events utilizing an alert system there was a proposed system that integrated vision based monitoring system consisting of a raspberry pi, two pixy cameras and an Arduino board. They employed two cameras to detect and monitor swimmers by measuring their positions and the swimmers were obliged to wear passive yellow vests. Neptune is

another technology that uses statistical image processing of video sequences to detect drowning victims are based on the variables created by statistical image processing. Another system called VIBE uses background extraction to detect and track drowning victims and updates the motion area by taking the frame difference using VIBE algorithm which primarily evaluates the swimmers position when making judgements. There was some difficulties in spotting drowning victims in a watery environment and offer an automatic detection surveillance system. The key items are water ripples and splashes as well as background movements of reflective zones. When it comes to recognizing swimmers occulusions are also mentioned as a challenging difficulty.

Activity Detection using Computer Vison: Current work on human motion prediction has been focused on two independent but complementary subtasks. Short term prediction which is quantitatively evaluated by measuring the mean squared error over a short period and long term motion prediction qualitatively evaluated by visual inspections of samples over a long period .Short term models would be valuable in motion tracking applications because these jobs are applicable in several domains of work. On the other hand long term models might be valuable for creating computer graphics tools due to their broad applicability . additionally both models could be useful in human gait analysis , kinematics research and human computer interaction.

2.2 REFERENCES

[1] https://link.springer.com/content/pdf/10.1007/3-540-47979-1_20.pdf

[2]https://scholar.archive.org/work/k6feepj5z5dfxmwar32flg2jqq/access/wayback/http://www.es.sdu.edu.cn/project/doc/papers/icess09-papers/15-1.pdf

[3] https://www.angeleye.tech/en/

[4] Long-term Recurrent Convolutional Network (LRCN) - Home (kobiso.github.io)

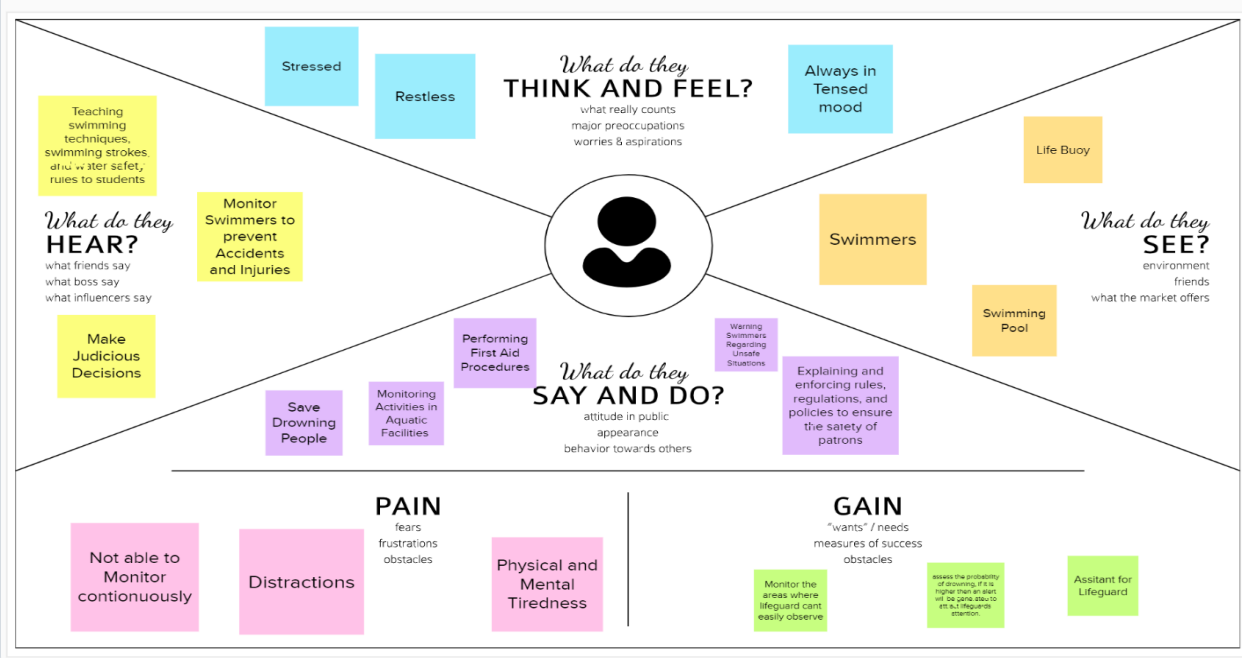
[5]https://cseweb.ucsd.edu//classes/wi19/cse291-g/student_presentations/Image_Caption_LRCN.pdf

2.3 PROBLEM STATEMENT DEFINITION

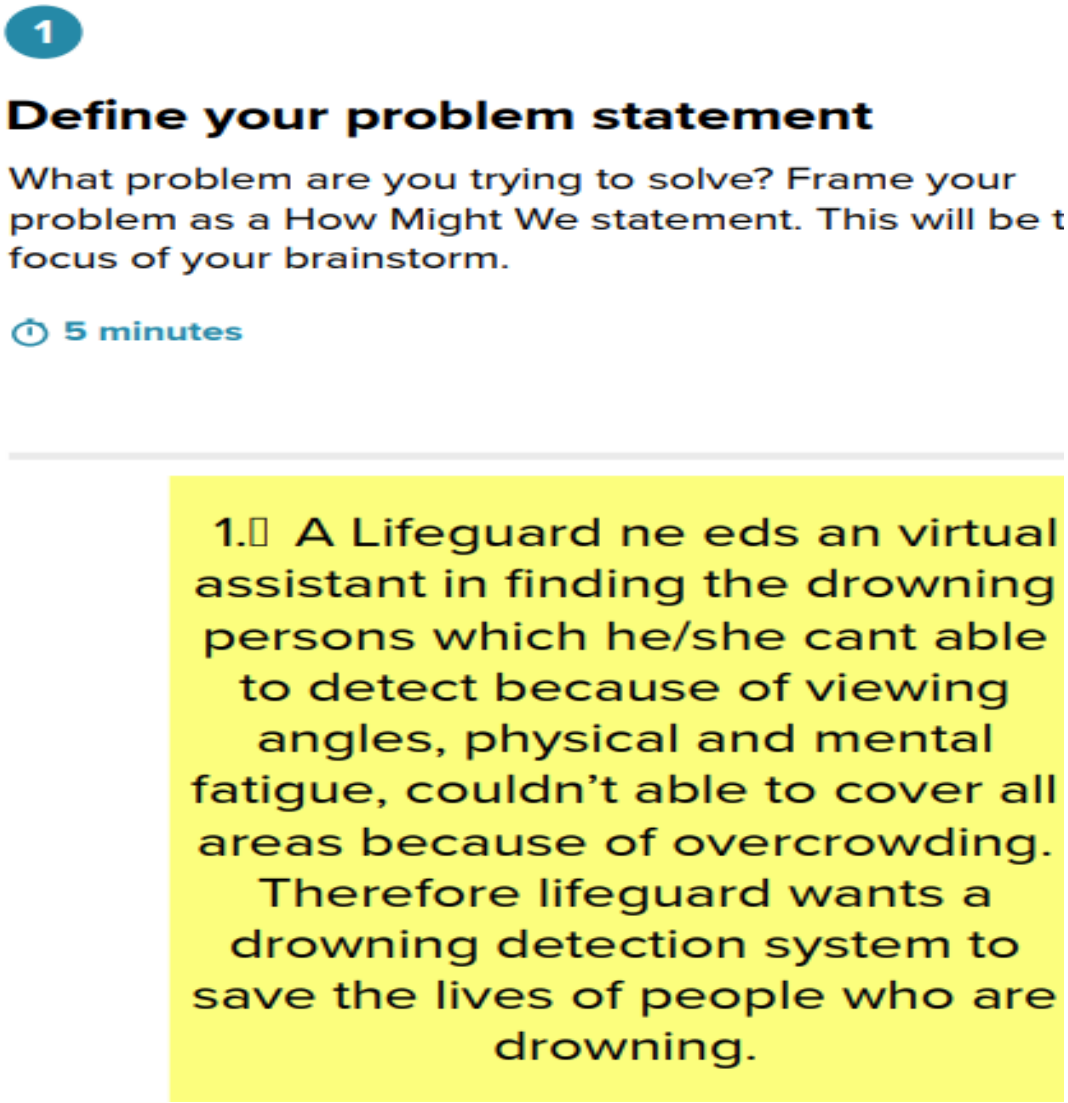
A Lifeguard needs an virtual assistant in fnding the drowning persons which he/she cant able to detect because of viewing angles, physical and mental fatigue, couldn't able to cover all areas because of overcrowding. Therefore lifeguard wants a drowning detection system to save the lives of people who are drowning.

3. IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Akshaya.J

- Monitor Breathing Rate
- Impose Barriers in Swimming depths
- Surveillance using underwater cameras
- provide age limits on swimming
- Monitor the swimming patterns
- use wearables to send data to the lifeguards

Indumathi.R

- place cameras on top of the swimming pool
- identify medical conditions of the swimmers
- impose time constraints on swimming
- Monitor the oxygen level of the swimmers
- set constant depth in pools based on age limit
- segregate professional and non professional swimmers

Vinisha.J

- list out good and bad swimming practises
- guidelines for providing first aid to drowned persons
- predict the probability of passive swimming accidents
- medical assitance from certified professionals
- provide alert to the divers
- monitor heart rate

Pavithra.V

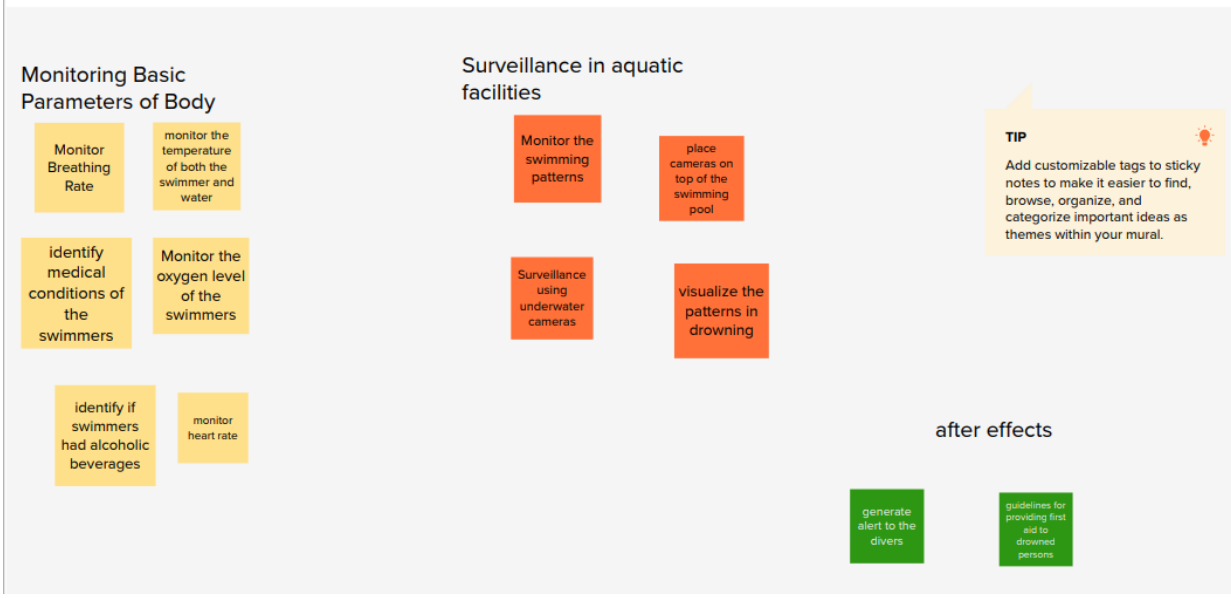
- monitor the temperature of both the swimmer and water
- creates fence around the pool
- visualize the patterns in drowning
- prevent crowding in aquatic facilities
- generate alert to the divers
- identify if swimmers had alcoholic beverages

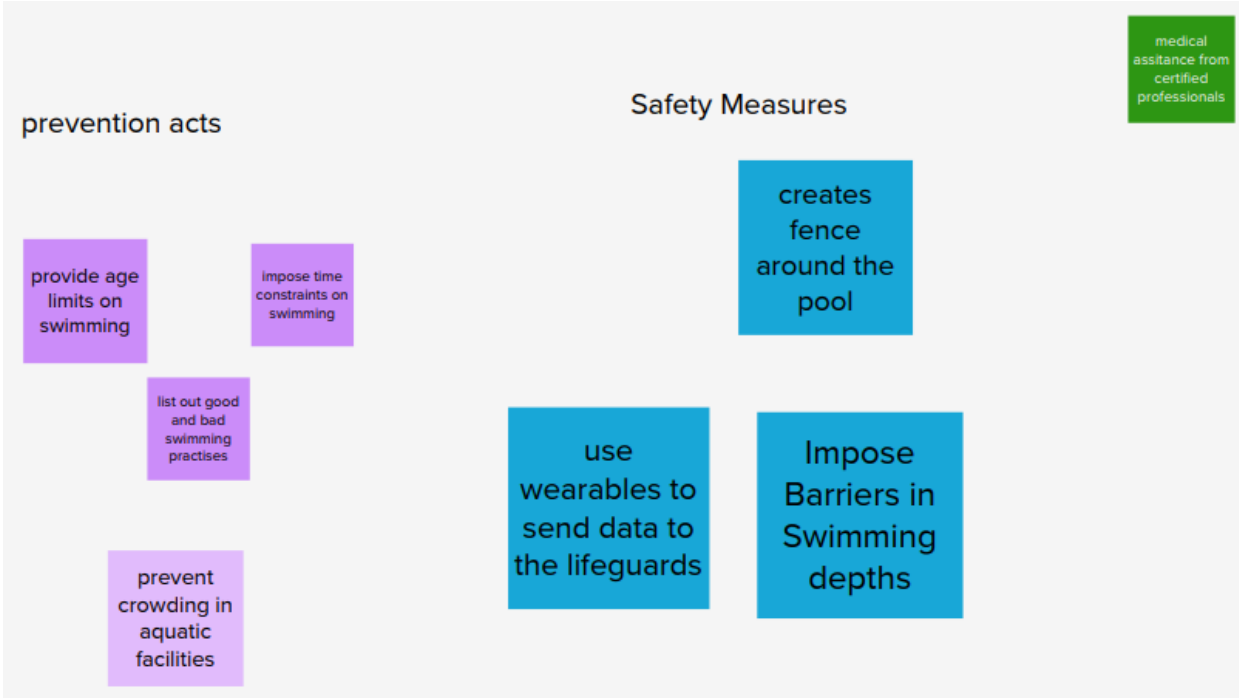
3

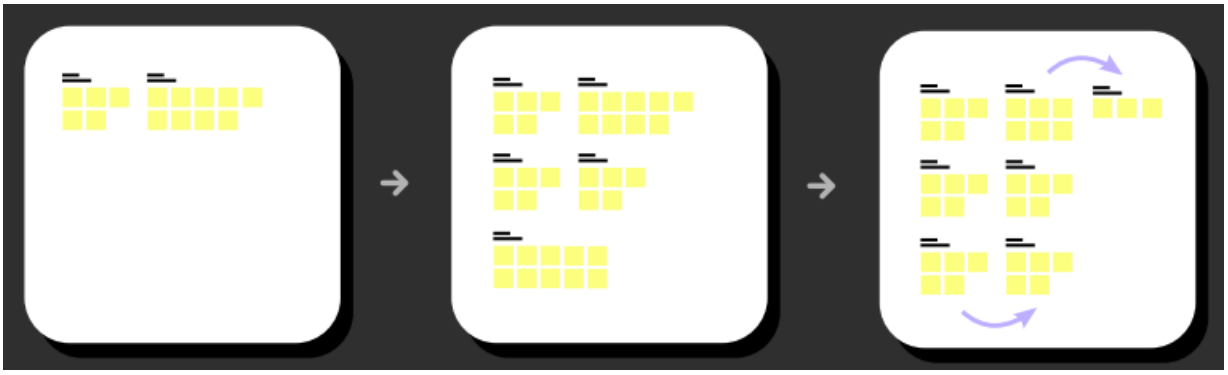
Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

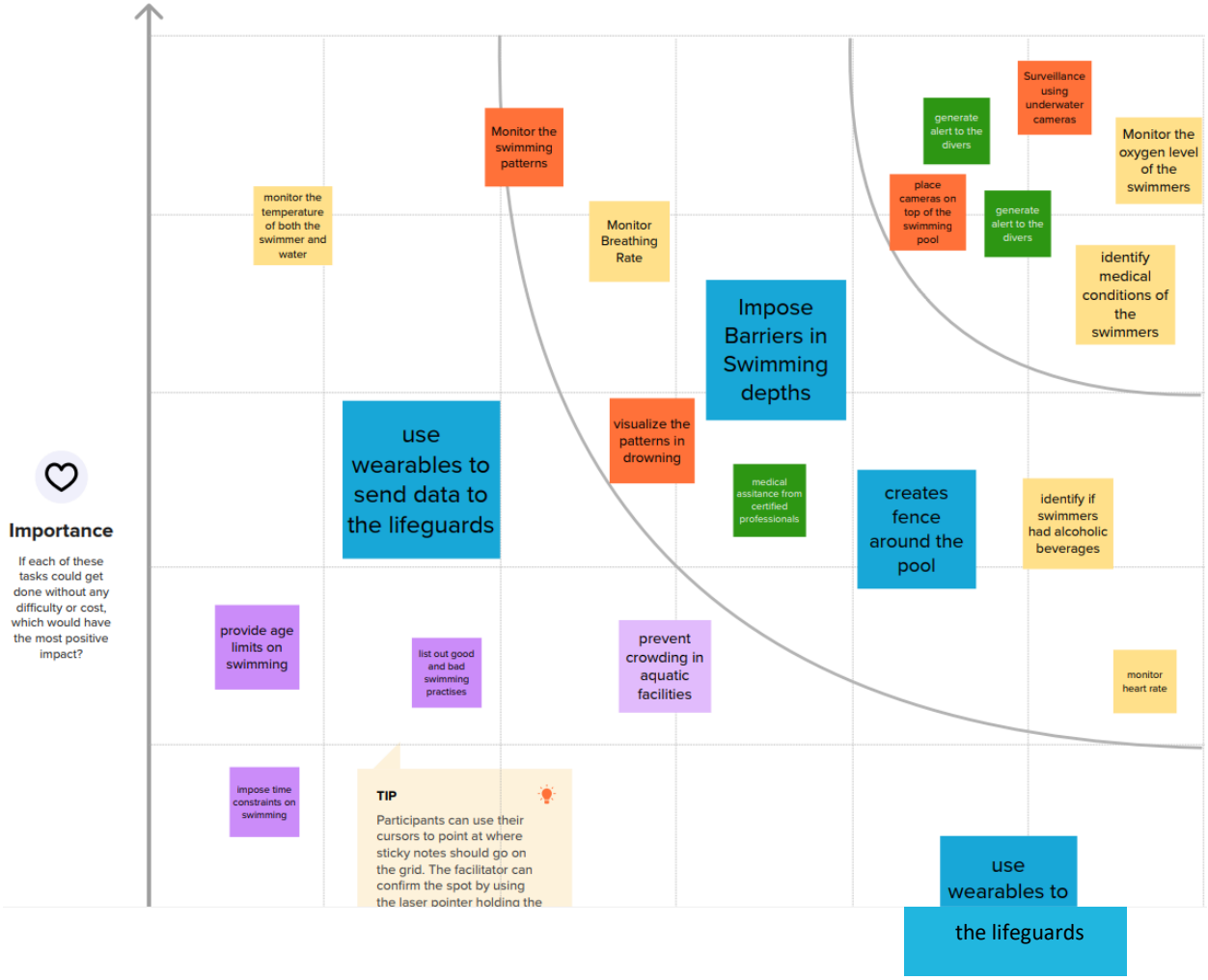
20 minutes







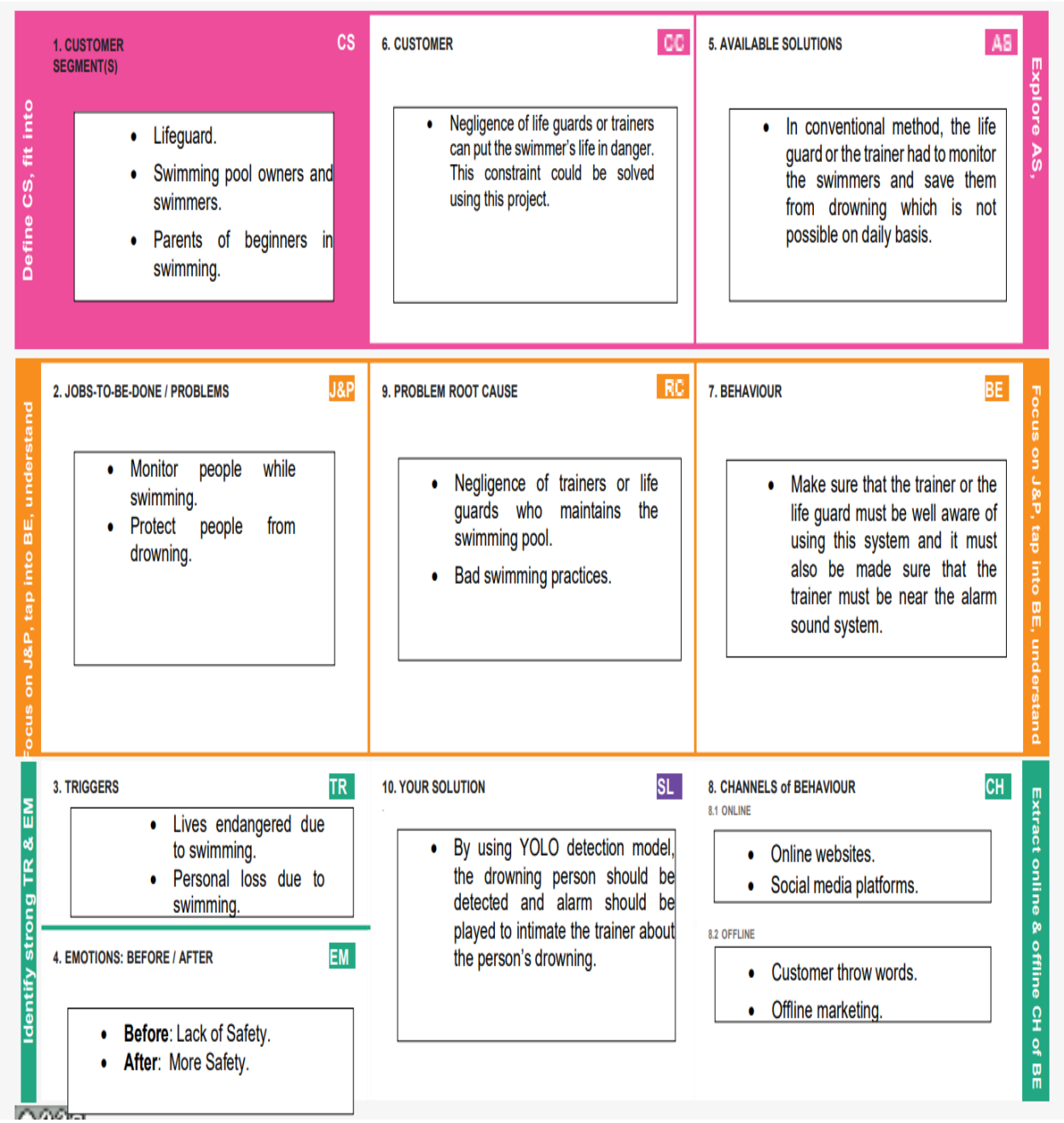
4.Prioritize



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	A Lifeguard needs an virtual assistant in finding the drowning persons which he/she can't able to detect because of viewing angles, physical and mental fatigue, couldn't able to cover all areas because of overcrowding. Therefore, lifeguard wants a drowning detection system to save the lives of people who are drowning.
2.	Idea / Solution description	By using YOLO method, the drowning person must be detected using machine learning and artificial intelligence techniques and once a person is detected the trainer will be alerted and the drowning person can be saved.
3.	Novelty / Uniqueness	Using YOLO method even if the person is detected once he will be notified to the trainer and alarm sound is played to save the person. By using human detection system if a person is found he will be saved.
4.	Social Impact / Customer Satisfaction	Crucial life of swimmers can be saved. It also increases the confidence of swimmers.
5.	Business Model (Revenue Model)	Once this model becomes a success. This model can be sold to private swimming pools and it can be sold for whooping prices. As it can support the trainers to look for drowning swimmers.
6.	Scalability of the Solution	It can be implemented to efficiently save people's life and can be used as a business model to generate revenue from private swimming pool.

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

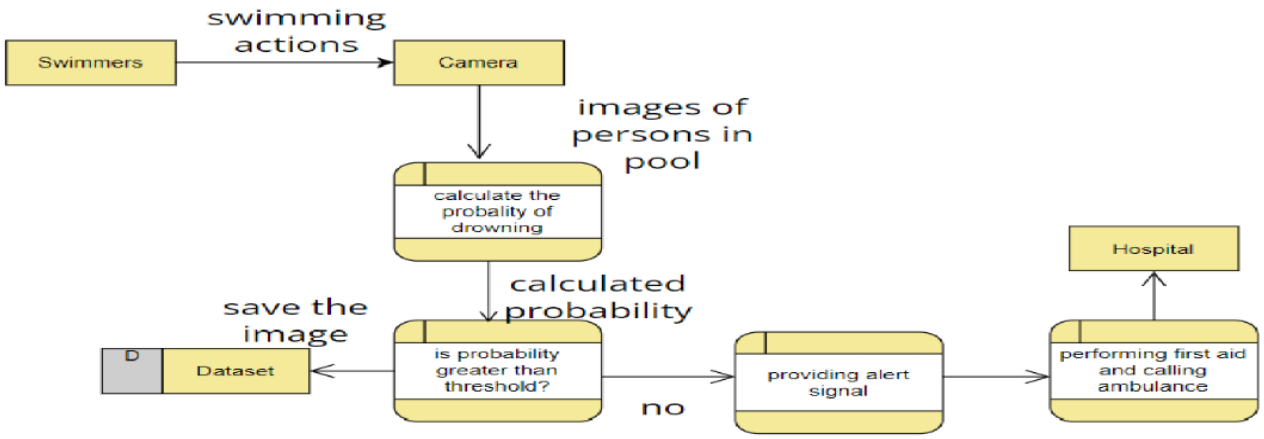
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Linkedin
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Monitoring Aquatic Facilities	Detecting Temperature Detecting Breathing Rate Detecting Oxygen Level
FR-4	Detecting Drowning	Prediction of drowning person Sending alert to hospitals
FR-5	First Aid measures	Video demonstrating CPR activities
FR-6	Doctor Consultation	Virtual meeting with doctor for further first aid safety measures

4.2 NON FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Alert of person Drowning is delivered in the means of speech synthesis in native language of lifeguards
NFR-2	**Security**	Data transmitted to lifeguards and hospitals is protected from third parties by means of encryption techniques
NFR-3	**Reliability**	Model is trained rigorously by providing huge dataset of drowning persons
NFR-4	**Performance**	Data is stored in cloud to provide faster bandwidth
NFR-5	**Availability**	Extending the dataset is carried autonomously and model is trained with new data's within an hour.
NFR-6	**Scalability**	Application is deployed on cloud to overcome the drawbacks of physical servers

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

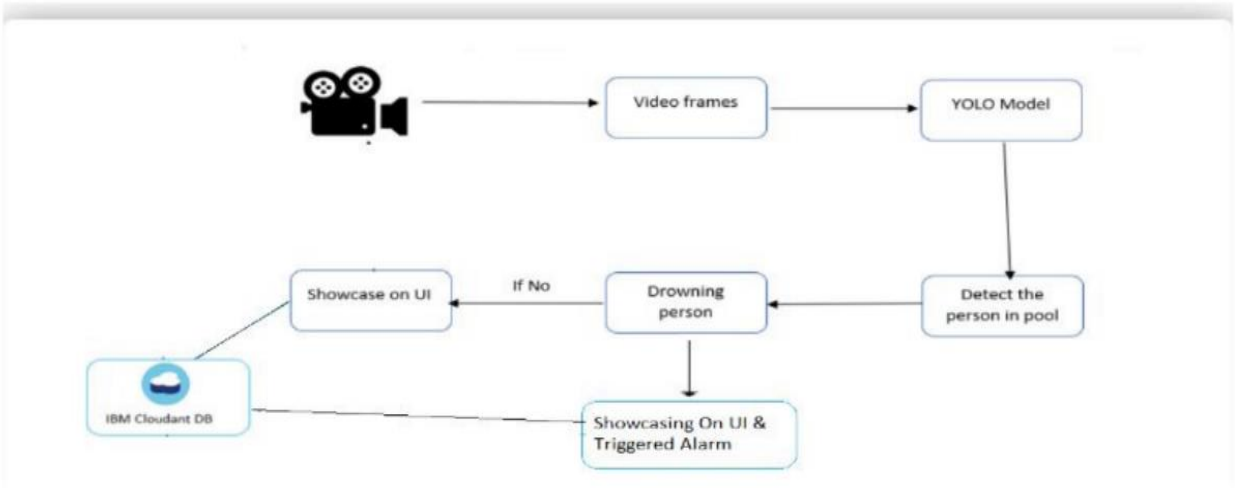


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interact with the application	HTML, JavaScript, CSS
2.	Application Logic-1	Extracting frames from live video feed	Python
3.	Application Logic-2	Person Detection	Python
4.	Application Logic-3	Detect drowning	Python
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM Cloudant DB
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Machine Learning Model	Detect humans	Object Detection Model (YOLOv7)
9.	Infrastructure (Server / Cloud)	Application Deployment on Cloud	Cloud Foundry, Docker.

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Anaconda Navigator, PyTorch, Flask,	Technology of Open-source framework
2.	Security Implementations	Security / access controls	IAM Controls
3.	Scalable Architecture	Whether demand increases gradually or abruptly, scalable web architecture can accommodate any load without compromising the application's integrity.	Microservies, Progressive Web Apps (PWA)
4.	Availability	Availability of applications like load balancers, distributed servers etc.	IBM Cloud

S.No	Characteristics	Description	Technology
5.	Performance	Designing the system software that can monitor a wide range of swimming pool at a time without any delay and to provide accurate predictions	IBM instance

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
(Aquatic facility owner)	Installation of camera	USN-1	As an aquatic facility owner i can place the cameras into aquatic facilities	I can install camera	High	Sprint-1
Lifeguard	Monitoring	USN-2	As a lifeguard I can able to monitor the people swimming in the pools	I can monitor people swimming in aquatic facilities	High	Sprint-1
Lifeguard	Detection	USN-3	As a lifeguard I can able to detect the person drowning in the aquatic facilities.	I can predict the drowning person in the pool by getting timely alerts	Low	Sprint-2
Lifeguard	First Aid	USN-4	As a lifeguard I can provide appropriate first and safety measures	Providing first aid	Medium	Sprint-2
Swimmer	Swimming practise	USN-5	As a swimmer I can swim without the fear of being drowned	Free from fear of drowning	High	Sprint-3
Hospital Authorites	Rescue	USN-6	As a doctor I can able to save the life of drowned person in time.	Saving Life	High	Sprint-4

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Installation of Camera	USN-1	As an aquatic facility owner I can place the cameras into aquatic facilities	2	High	Vinisha.J
Sprint-1	Monitoring	USN-2	As a lifeguard I can able to monitor the people swimming in the pools	1	High	Akshaya.J
Sprint-2	Detection	USN-3	As a lifeguard I can able to detect the person drowning in the aquatic facilities.	2	Low	Indumathi.R
Sprint-2	First Aid	USN-4	As a lifeguard I can provide appropriate first and safety measures	2	Medium	Pavithra.V
Sprint-3	Swimming Practise	USN-5	As a swimmer I can swim without the fear of being drowned	1	High	Akshaya.J
Sprint-4	Rescue	USN-6	As a doctor I can able to save the life of drowned person in time	2	High	Pavithra.V

6.2 SPRINT DELIVERY SCHEDULE

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	20	6 Days	24 Oct 2022	29 Oct 2022	19	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	19	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	10	19 Nov 2022

Velocity:

For Sprint-1 the Average Velocity (AV) is: AV = Sprint Duration / velocity = 20 / 18 = 1.1V

For Sprint-2 the Average Velocity (AV) is: AV = Sprint Duration / velocity = 20 / 19= 1.05V

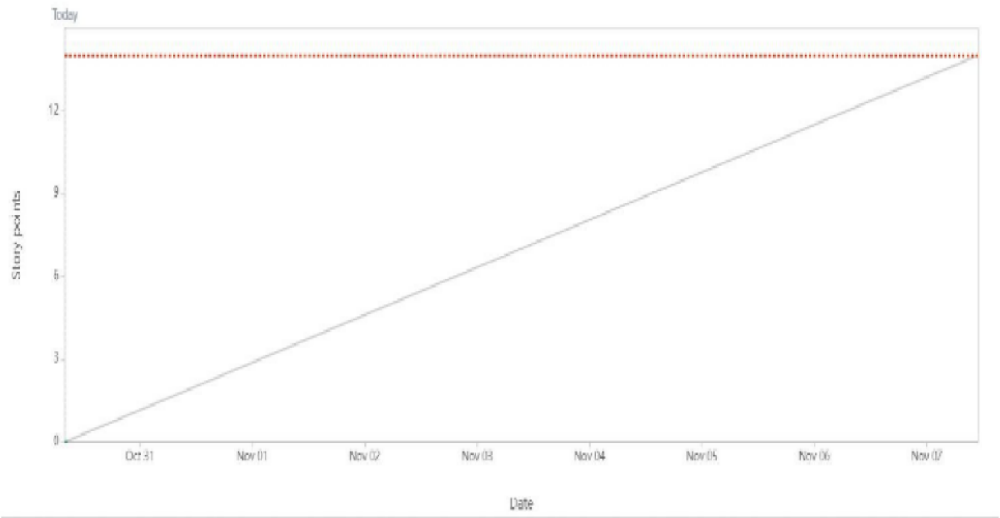
For Sprint-3 the Average Velocity (AV) is: AV = Sprint Duration / velocity = 20 / 10 = 2.0V

For Sprint-4 the Average Velocity (AV) is: AV = Sprint Duration / velocity = 20/ 10 = 2.0

6.3 REPORTS FROM JIRA

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



7. CODING AND SOLUTIONING

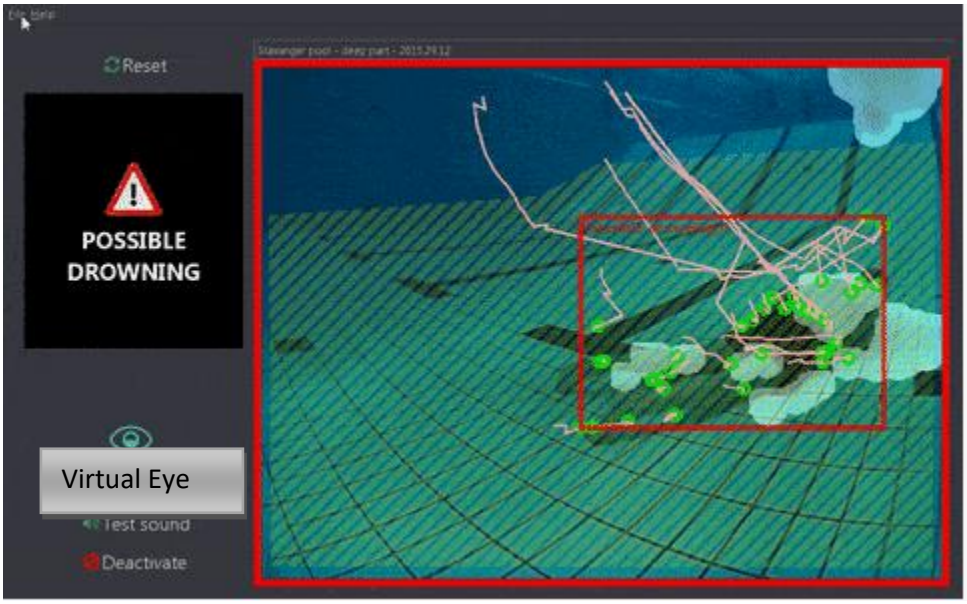
7.1 FEATURE 1

The live video stream from our underwater cameras is automatically monitored by our object recognition software. By studying body movement patterns and connecting cameras to artificial intelligence (AI) systems we can devise an underwater pool safety system that reduces the risk of drowning. Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analyzing the video feeds to detect any anomalies. But as a poc we make use of one camera that streams the video underwater and analyses the position of swimmers to assess the probability of drowning, if it is higher then an alert will be generated to attract lifeguards' attention.

7.2 FEATURE 2

When VirtualEye detects a swimmer in distress on the bottom of the pool, it will raise a radio alarm to pool lifeguards and an visual alarm to our Monitoring & Control Station.

Lifeguards can visually assess the developing situation within seconds of the event first occurring, and initiate their rescue procedure when necessary.



8.TESTING

8.1 TESTCASES

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
LoginPage_TC_OO1	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/Singup popup displayed or not	login.html	Login/Signup popup should display	Working as expected	Pass
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	login.html	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	pass
LoginPage_TC_OO3	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 button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: indu06 password: Testing123	User should navigate to prediction homepage	working as expected	pass
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: indu password: Testing123	Application should show 'Incorrect email or password ' validation message.	working as expected	pass
LoginPage_TC_OO4	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 Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: indu06 password: abcd	Application should show 'Incorrect email or password ' validation message.	working as expected	pass
LoginPage_TC_OO5	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 Invalid password in password text box 5.Click on login button	Username: xyz password: 1234	Application should show 'Incorrect email or password ' validation message.	working as expected	pass

Predictionpage_TC_OO_6	Functional	Prediction Page	Page should display whether the person is drowning or not		1. Camera should take pictures of people swimming in pools 2. It should predict the probability of drowning 3. It should show a bounding box displaying the probability of drowning	image of people drowning	generate a alert to lifeguard if people are drowning	Working as expected	pass

8.2 USER ACCEPTANCE TESTING

1. Defect Analysis

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

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
Totals	24	14	13	26	77

2. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

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
Version Control	1	0	0	1

9. RESULTS

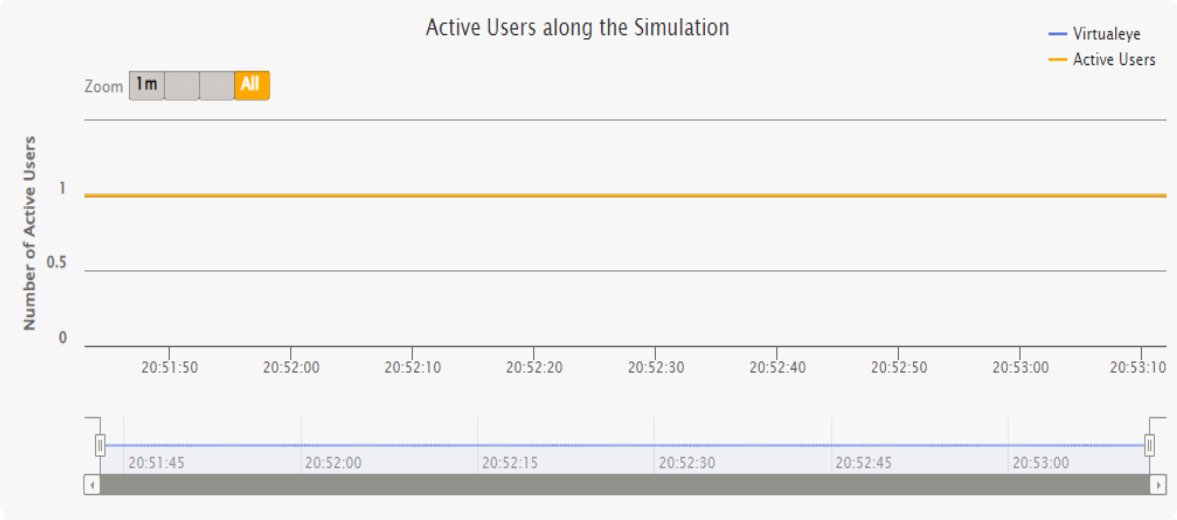
9.1 PERFORMANCE METRICS

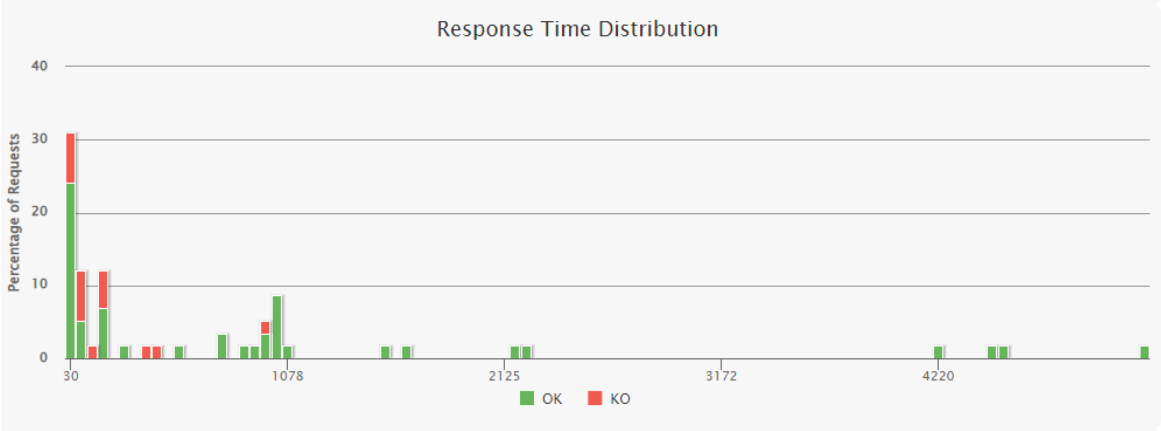
Requests	Executions					Response Time (ms)							
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
All Requests	58	43	15	26%	0.644	4	200	997	4248	4825	5241	734	1191
request_0	1	1	0	0%	0.011	20	20	20	20	20	20	20	0
jquery-3.2.1.slim.min.js	1	1	0	0%	0.011	5241	5241	5241	5241	5241	5241	5241	0
css2?family=Akronim&family=Roboto&display=swap	1	1	0	0%	0.011	749	749	749	749	749	749	749	0
popper.min.js	1	1	0	0%	0.011	789	789	789	789	789	789	789	0
8b9cdc2059.js	2	2	0	0%	0.022	988	2597	3401	4044	4173	4205	2597	1609
style.css	5	0	5	100%	0.056	4	75	194	376	412	421	141	155
1.png	1	0	1	100%	0.011	83	83	83	83	83	83	83	0
second.jpg	3	0	3	100%	0.033	15	86	143	189	198	200	100	76
request_1	1	1	0	0%	0.011	88	88	88	88	88	88	88	0
third.jpg	1	0	1	100%	0.011	87	87	87	87	87	87	87	0
request_2	1	1	0	0%	0.011	94	94	94	94	94	94	94	0
bootstrap.min.css	1	1	0	0%	0.011	4512	4512	4512	4512	4512	4512	4512	0
bootstrap.min.js	1	1	0	0%	0.011	4492	4492	4492	4492	4492	4492	4492	0
request_3	1	1	0	0%	0.011	20	20	20	20	20	20	20	0
request_4	1	1	0	0%	0.011	20	20	20	20	20	20	20	0
request_5	1	1	0	0%	0.011	1640	1640	1640	1640	1640	1640	1640	0
request_6	1	1	0	0%	0.011	305	305	305	305	305	305	305	0
request_7	1	1	0	0%	0.011	1015	1015	1015	1015	1015	1015	1015	0
request_8	1	1	0	0%	0.011	1524	1524	1524	1524	1524	1524	1524	0
request_10	1	1	0	0%	0.011	4	4	4	4	4	4	4	0
css?family=Pacifico	1	1	0	0%	0.011	547	547	547	547	547	547	547	0
css?family=Hind:300	1	1	0	0%	0.011	1010	1010	1010	1010	1010	1010	1010	0
css?family=Arimo	1	1	0	0%	0.011	1004	1004	1004	1004	1004	1004	1004	0
css?family=Josefin Sans	3	0	3	100%	0.033	146	451	723	941	984	995	531	351
css?family=Merriweather	1	1	0	0%	0.011	1005	1005	1005	1005	1005	1005	1005	0
css?family=Open+Sans+Condensed:300	1	1	0	0%	0.011	1007	1007	1007	1007	1007	1007	1007	0

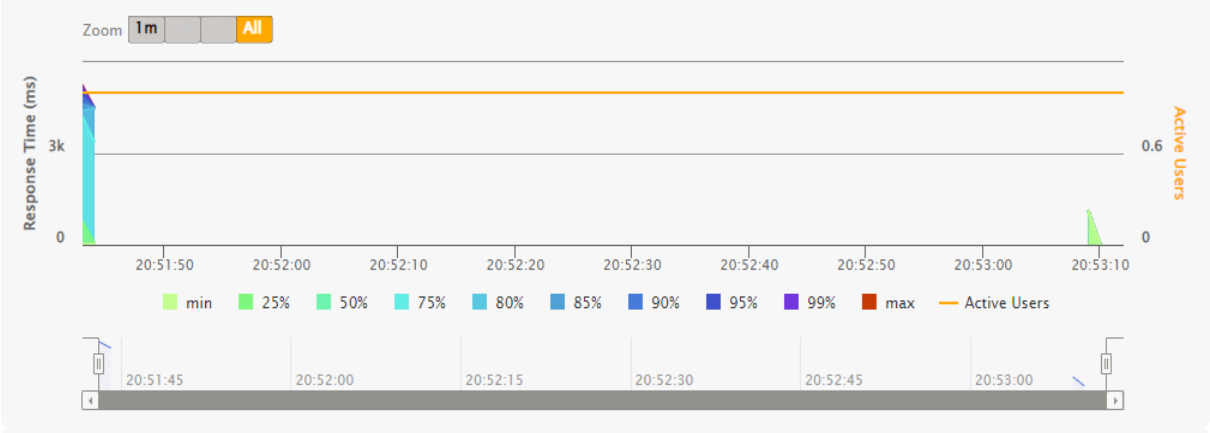
request_12	1	1	0	0%	0.011	997	997	997	997	997	997	997	0
request_13	1	1	0	0%	0.011	863	863	863	863	863	863	863	0
eye_image_generic_555.jpg	1	1	0	0%	0.011	934	934	934	934	934	934	934	0
css?family=Montserrat	1	1	0	0%	0.011	100	100	100	100	100	100	100	0
request_11	1	1	0	0%	0.011	203	203	203	203	203	203	203	0
request_15	1	1	0	0%	0.011	2162	2162	2162	2162	2162	2162	2162	0
request_19	1	1	0	0%	0.011	6	6	6	6	6	6	6	0
request_23	1	1	0	0%	0.011	2231	2231	2231	2231	2231	2231	2231	0
request_23 Redirect 1	1	1	0	0%	0.011	7	7	7	7	7	7	7	0
JScript.js	2	0	2	100%	0.022	14	107	153	190	197	199	107	93
request_24	1	1	0	0%	0.011	202	202	202	202	202	202	202	0
request_26	1	1	0	0%	0.011	199	199	199	199	199	199	199	0
request_25	1	1	0	0%	0.011	200	200	200	200	200	200	200	0
request_31	1	1	0	0%	0.011	1085	1085	1085	1085	1085	1085	1085	0
request_32	1	1	0	0%	0.011	10	10	10	10	10	10	10	0
request_34	1	1	0	0%	0.011	11	11	11	11	11	11	11	0
request_33	1	1	0	0%	0.011	12	12	12	12	12	12	12	0
request_35	1	1	0	0%	0.011	13	13	13	13	13	13	13	0
request_36	1	1	0	0%	0.011	6	6	6	6	6	6	6	0
request_37	1	1	0	0%	0.011	26	26	26	26	26	26	26	0
request_38	1	1	0	0%	0.011	23	23	23	23	23	23	23	0
request_39	1	1	0	0%	0.011	26	26	26	26	26	26	26	0

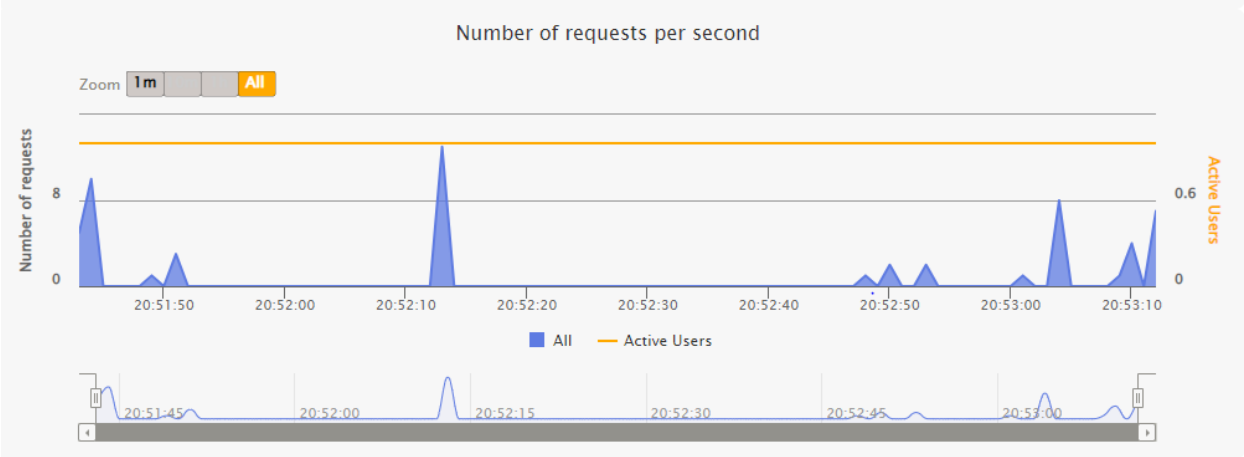
Errors

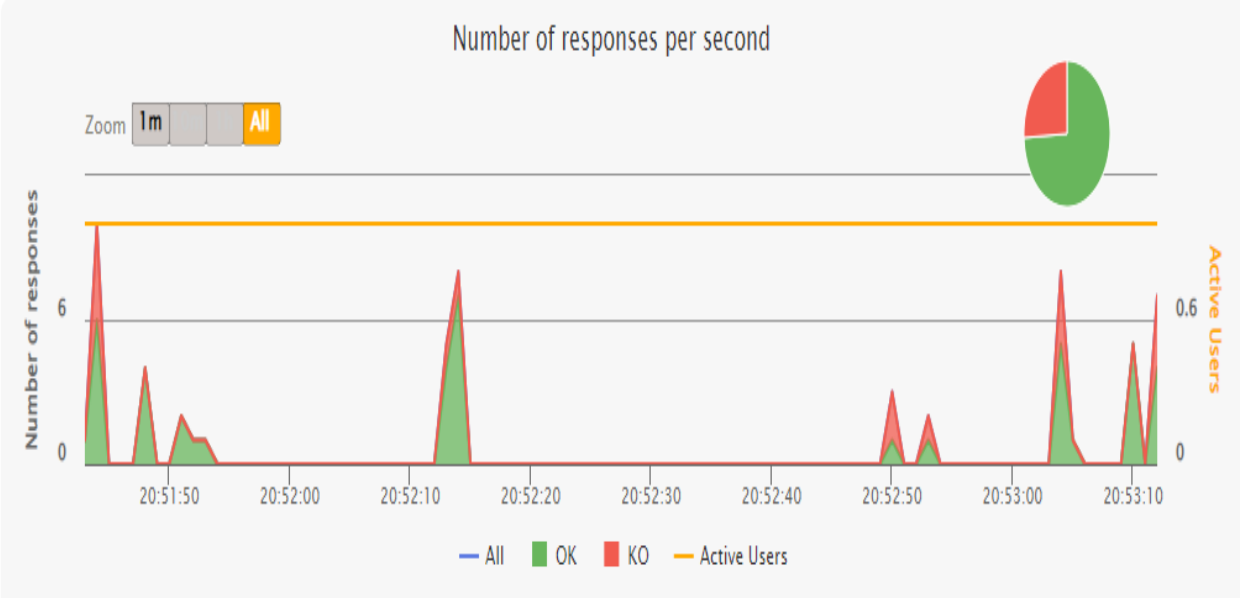
Error	Count	Percentage
status.find.in([200, 209], 304), found 404	12	80 %
status.find.in([200, 209], 304), found 400	3	20 %




Response Time Distribution
Percentage of Requests
40
30
20
10
0
30
1078
2125
3172
4220
OK
KO


Zoom
1m
All
Response Time (ms)
3k
0
Active Users
0.6
0
20:51:50
20:52:00
20:52:10
20:52:20
20:52:30
20:52:40
20:52:50
20:53:00
20:53:10
min
25%
50%
75%
80%
85%
90%
95%
99%
max
Active Users
20:51:45
20:52:00
20:52:15
20:52:30
20:52:45
20:53:00


Number of requests per second
Zoom
1m
All
Number of requests
8
0
Active Users
0.6
0
20:51:50
20:52:00
20:52:10
20:52:20
20:52:30
20:52:40
20:52:50
20:53:00
20:53:10
All
Active Users
20:51:45
20:52:00
20:52:15
20:52:30
20:52:45
20:53:00



10. ADVANTAGES

It increases computing and resource efficiency.
It helps to assess human conditions and behaviours.
It helps to learn from experts delivering skills.
It makes technology more intuitive.
It helps to communicate with machines in order to automate manual tasks.
It increases user experience and performance in playing games.

DISADVANTAGES

It is expensive technology due to costly hardware requirements. It does not work with few users who wear contact lenses or have long eye lashes.

It requires some calibration time before it gives satisfactory results. Hence

few users deviate themselves from using it.
Eye movements of some users are often un-intentional. This results into unwanted responses by the system.
It is difficult to control eye position accurately all the times unlike mouse.
Eye tracker provides instable output when it does not get appropriate image of the eye in consecutive frames.

11. CONCLUSION

Once we have the working drowning detection model we can feed live video footage of the swimming pool to it so that it can keep detecting continuously for any drowning activities. If drowning is detected it will be highlighted on the system screen as well as alarms will be raised to alert security guards so that they can initiate rescue.

12. FUTURE SCOPE

Availability of better dataset, modern methodologies, and technologies with high computational power accompanied by high-quality surveillance cameras, will help to improve the accuracy of drowning detection & even can be used in adverse conditions.

After the implementation of all these essentials, this system also can be used on sea beaches for drowning detection

13. APPENDIX

SOURCE CODE

Import the libraries

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

```
import cvlib as cv
from cvlib.object_detection import draw_bbox
import cv2
import time
import numpy as np
from playsound import playsound
import requests
from flask import Flask, request, render_template, redirect, url_for
#Loading the model

from cloudant.client import Cloudant
```

Create a database using an initiated client.

```
from cloudant.client import Cloudant

# Authenticate using an IAM API key
client = Cloudant.iam('2eb40045-a8d6-450d-9d24-52cc7cbb2810-bluemix','Ud0wunTPOI_8h5ZtEqi1IXk1gIKeYLmpUsCn0EeO8T4z', connect=True)


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

```python
#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')
```

```
        #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()
```

```
        # 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',)
```

GITHUB AND PROJECT DEMO LINK

IBM-EPBL/IBM-Project-9152-1658983893: VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning (github.com)

https://drive.google.com/file/d/1pOrnv5pOC1nDRaN6DdQ7Vw1y3jnc7_U3/view?usp=share_link