VIRTUALEYE - LIFE GUARD FOR SWIMMING POOLS TO DETECT ACTIVE DROWNING

PROJECT BASED LEARNING

Submitted by

ISHWARYA G K (910619104029)

JEYADARSHINI P (910619104034)

JEEVAPRIYA G M (910619104033)

DIVYA S (910619104017)

GOPIKA R (910619104022)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K.L.N. COLLEGE OF ENGINEERING, POTTAPALAYAM
ANNA UNIVERSITY : CHENNAI 600 025

APRIL 2020

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning”** is the bonafide work of **“ISHWARYA G K (910616104029)”,“JEYADARSHINI P (910616104034)”,“JEEVAPRIYA G M (910616104033)”,“DIVYA S (910619104017)”**and **“GOPIKA R (910619104022)”** who carried out the project under my supervision.

SIGNATURE

Mr.D.PRABHU

ASSISTANT PROFESSOR

Computer science and engineering,

K.L.N. College of Engineering,

Pottapalayam,

Sivagangai-630 612.

SIGNATURE

Dr.S.MIRUNA JOE AMALI

HEAD OF THE DEPARTMENT

Computer science and engineering,

K.L.N. College of Engineering,

Pottapalayam,

Sivagangai-630 612.

Submitted for the project viva-voce conducted on

MENTOR EVALUATOR

ABSTRACT

Safety is paramount in all swimming pools.The current systemsexpected to address the problem of ensuring safety at swimming poolshave significant problems due to their technical aspects such asunderwater camera and methodological aspects such as the need orhuman intervention in the rescue mission.The VirtualEye software works in closeintegration with the cameras installed in the pool to continuously to scanthe swimming pool. The First, by analyzing the spatial distribution ofswimming pool when swimmers are normally swimming, the data labelingand swimmer detect methods are determined. Second, a behaviorrecognition framework of swimmers on the basis of YOLOv3 algorithm (BRYOLOv3)is proposed.The spatial relationship between the locationinformation of the target and swimming/drowning area of swimming poolis analyzed to determine the swimmer’s drowning or swimming behavior. Itintroduces a revolutionary technology that identifies drowning victims in aminimum amount of time and dispatches an automated drone to savethem. Using convolutional neural network (CNN) models, it can detect adrowning person in three stages. Whenever such a situation like this isdetected, the inflatable tube-mounted self-driven drone will go on a rescuemission, sounding an alarm to inform the nearby lifeguards. The systemalso keeps an eye out for potentially dangerous actions that could result indrowning. This system's ability to save a drowning victim in under a minutehas been demonstrated in prototype experiments' performanceevaluations.The live video stream from our underwater cameras isautomatically monitored by our “state-of-the-art” object recognitionsoftware. When it 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 ourMonitoring & Control Station.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSION
CNN	Convolutional Neural Network
YOLO	You Only Look Once
DNN	Deep Neural Networks
MSE	Mean Squared Error
HSV	Hue, Saturation, Value
GPS	Global Positioning System

1. INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1 Project Overview

To understand the 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. AProof Of Concept (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

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.

2.LITERATURE SURVEY

CHAPTER 2
LITERATURE SURVEY

2.1 Existing problem

Video-based systems and wearable sensor-based systems aretwo types of existing drowning detection technologies. It will use of Objectdetection using different techniques will usage of Convolutional NeuralNetwork (CNN) architecture in Deep Neural Networks (DNNs) has added asignificant shift in learning more complicated, informative characteristics inimages as compared to older techniques. Then, Drowning Detection andTracking to avoid drowning events utilizing an alert system. ActivityDetection using Computer Vision has Current work on human motionprediction has been focused on two independent but complementary subtasks,according to Anand Gopalkrishnan, Short-term motion prediction,which is quantitatively evaluated by measuring the mean squared error(MSE) over a short period, and long-term motion prediction, qualitativelyevaluated by visual inspections of samples over a long period. Short-termmodels would be valuable in motion tracking applications because thesejobs are applicable in several domains of work. On the other hand, longtermmodels might be valuable for creating computer graphic tools due totheir broad applicability. Additionally, both models could be useful in humangait analysis, kinematics research, and human-computer interaction.

Another existing system, there has been an interest in integrating computervision in swimming pool surveillance systems. Automating such a processwill provide the communities with an efficient way of detecting drowningincidents that may occur while swimming. a hybrid system that willautomatically detect a drowning person and then set off an alarm to alertthe lifeguards has been developed. The system mainly consists of threemodules: a vision module, an event-inference module and an event-drivenmodule. The vision module is responsible for monitoring and detecting theposition of the person who is drowning. The event-inference module isresponsible for determining a swimmer’s position, velocity, and path of themovement. The event-driven module is responsible for initiating the rescueby sending an alarm alerting the lifeguard.The main contribution of thisproject is to develop a system for monitoring swimming pool to prevent theonset of a drowning incident.

Drowning detectors detect the drowning by analysis the various readings exhibited during drowning distress, by the victim. This could be like monitoring the waves generated due to panic to monitoring the irregular pressure variations from the gadget, used by the victim. Especially children get easily disturbed by placing any sensors very closer to the mouth and nose. They may also try to remove it because of the disturbance.

2.2 References

[1] Golan, "Drowning Detection gets real with Coral Manta 300.[Accessed 25 November 2020].

[2] N. Alshbatat, "Automated Vision-based Surveillance System to Detect Drowning Incidents in Swimming Pools.," in In Proceedings of the 2020 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, UAE, 2020.

[3] SwimEye, "A drowning detection system for any pool ,"Swimeye.com" [Accessed 25 November 2020].

[4] AngelEye,"Safeguard," AngelEye.tech [Accessed 25 November 2020].

[5] Zhang, "A Novel Camera-Based Drowning Detection Algorithm," in IGTA 2015: Advances in Image and Graphics Technologies pp 224-233, Berlin, 2015.

[6] L. Bugeja, "An analysis of stratagems to reduce drowning deaths of young children in private swimming pools and spas in Victoria, Australia," in Int J InjContrSafPromot, 2013.

[7] A. S. Joost Bierensa, "Drowning in swimming pools," in Microchemical Journal, Rome, 2013.

[8] Hartmann, "Drowning and Beach-SafetyManagement (BSM) along the Mediterranean Beaches of Israel: A Long-Term Perspective," in Journal of Coastal Research, Beer-Sheva, 2006.

[9] L. Wenmiao, "A vision-based approach to early detection of drowning incidents in swimming pools," in IEEE transactions on circuits and systems for video technology 14.2 159-178., 2004.

2.3 Problem Statement Definition

Video surveillance can be used as a tool for monitoring andsecurity. The visual monitoring capabilities can be employed in manydifferent locations to help people live more safely. Videobased surveillancesystems are designed and installed in places such as railway stations,airports, and even dangerous environments. Image processing, patternrecognition and machine-vision based methods are efficient ways for realtimeintelligent monitoring of the objects or events. The existingsurveillance systems deliver valued information in monitoring of largeareas. Applying intelligence in video surveillance systems allows realtimemonitoring of places, people and their activities. The tracking approach can change with varying targets and can change from a single camera tomultiple camera configurations. Tracking methods in video surveillance usedifferent parameters such as objects motion, position, path of movementand velocity, biometrics such as skin color or clothes color and many more.The tracking must be robust and overcome occlusion and noise which are common problems in monitoring.One important environment that the needfor monitoring systems is crucially sensed is the swimming pool. Each yearmany people including children are drowned or very close to drowning inthe deeps of the swimming pools, and the life guards are not trained wellenough to handle these problems. This raises the need for having a systemthat will automatically detect the drowning person and alarm the lifeguards of such danger.

Real-time detection of a drowning person in swimmingpools is a challenging task that requires an accurate system. The challengeis due to the presence of water ripples, shadows and splashes andtherefore detection needs to have high accuracy.In swimming poolmonitoring intelligent systems, different approaches have been proposed.Most methods perform background processing on input video frames.Some apply background subtraction and image denoising to detect thedrowning person.

In a Gaussian Mixture Model is used for describing thepixels and the parameters of the model are updated with the EM algorithm. Also, neural networks can be trained to classify near-drowning and normal swimming patterns. However this requires to have a large dataset of both groups of behavior. The dataset is obtained in by attaching a pressure sensor to a swimmer imitating drowning behavior and normal swimming.Pattern recognition algorithms are also very useful in swimmer detection.In a background model that has prior knowledge about swimming pools isemployed. This hierarchical model operates on behavioral traits common inalmost all troubled swimmers. It uses movement and intensity informationfrom image frames. In the YCbCr color model is selected for detection ofthe water polo players in water where luminance is separated and the Cband Cr components are analyzed. Moreover, underwater ultrasonic sensors can detect drowning people up to 70 meters below water in the swimming pool along with a underwater video detection unit that locates and finds thevictims. This research presents a vision-based approach for detecting adrowning person and alarming the life guards of such situations.

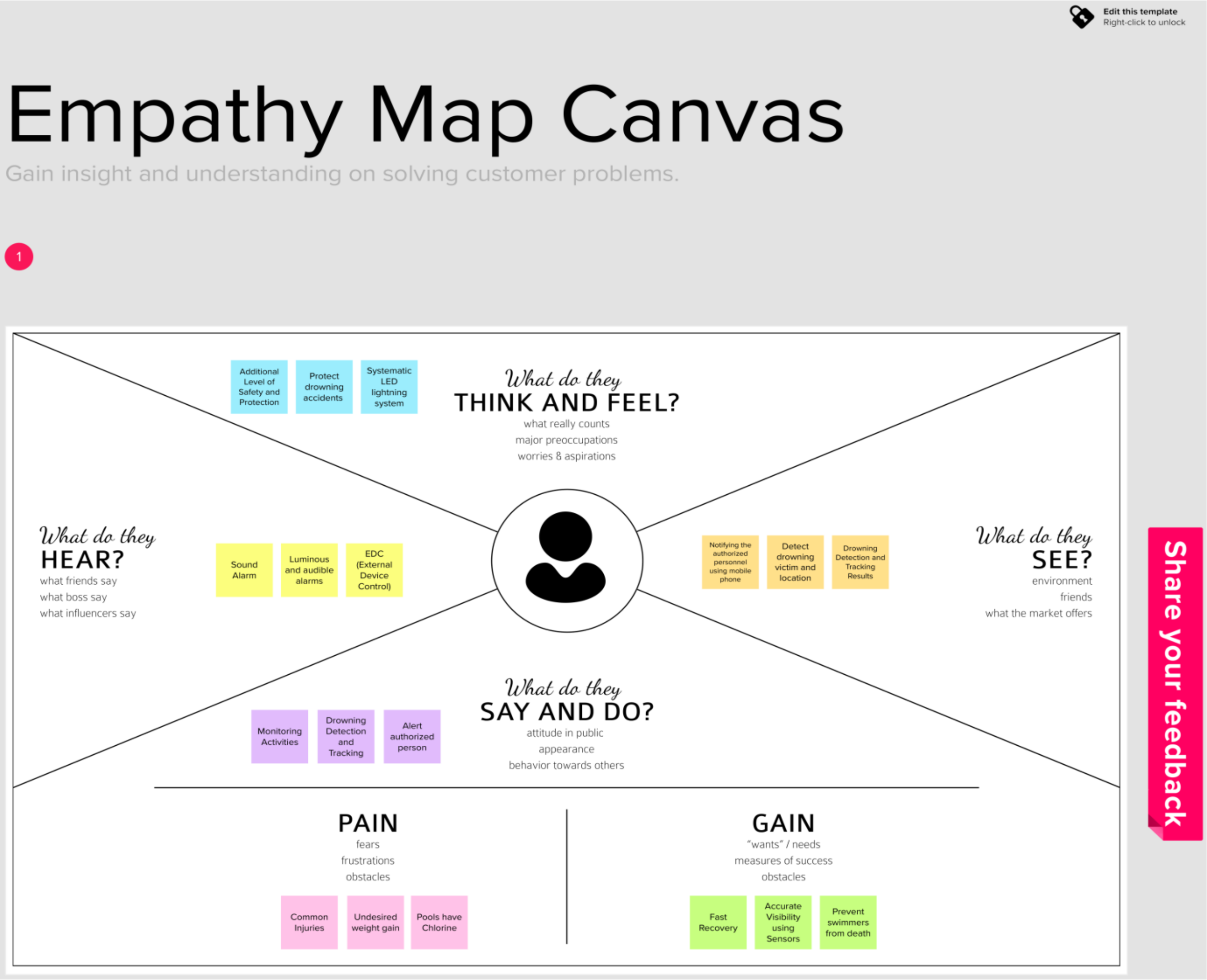
Theperson swimming in the pool is detected and tracked using the HSV colorspace properties and contour-based methods. As soon as the moving target remains under water for more than a determined period of time, analarm is sent to the lifeguard rescues. The HSV color space is selected overother color spaces because it is more effective in segmenting the swimmerin various light conditions from the background. The drowning detectioncomponent detects drowning victims through a custom CNN model, which detects drowning in three stages and immediately informs the user through an audio alert. The second component is the rescue drone,activated according to the drowning detection command and sent tothe victim's location coordinates. This procedure uses a customconfiguredx and y coordinate block system to link to ground GPScoordinates. At the same time, potentially dangerous activities, including running around the swimming pool and drinking, will be notified to authorized personnel in the premises through mobile alarms by thehazard detection component.

3.IDEATION & PROPOSED SOLUTION

CHAPTER 3
IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users.Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



Edit this template
Right-click to unlock

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

- Additional Level of Safety and Protection
- Protect drowning accidents
- Systematic LED lightning system

What do they HEAR?
what friends say
what boss say
what influencers say

- Sound Alarm
- Luminous and audible alarms
- EDC (External Device Control)

What do they SEE?
environment
friends
what the market offers

- Notifying the authorized personnel using mobile phone
- Detect drowning victim and location
- Drowning Detection and Tracking Results

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

- Monitoring Activities
- Drowning Detection and Tracking
- Alert authorized person

PAIN
fears
frustrations
obstacles

- Common Injuries
- Undesired weight gain
- Pools have Chlorine

GAIN
"wants" / needs
measures of success
obstacles

- Fast Recovery
- Accurate Visibility using Sensors
- Prevent swimmers from death

Share your feedback

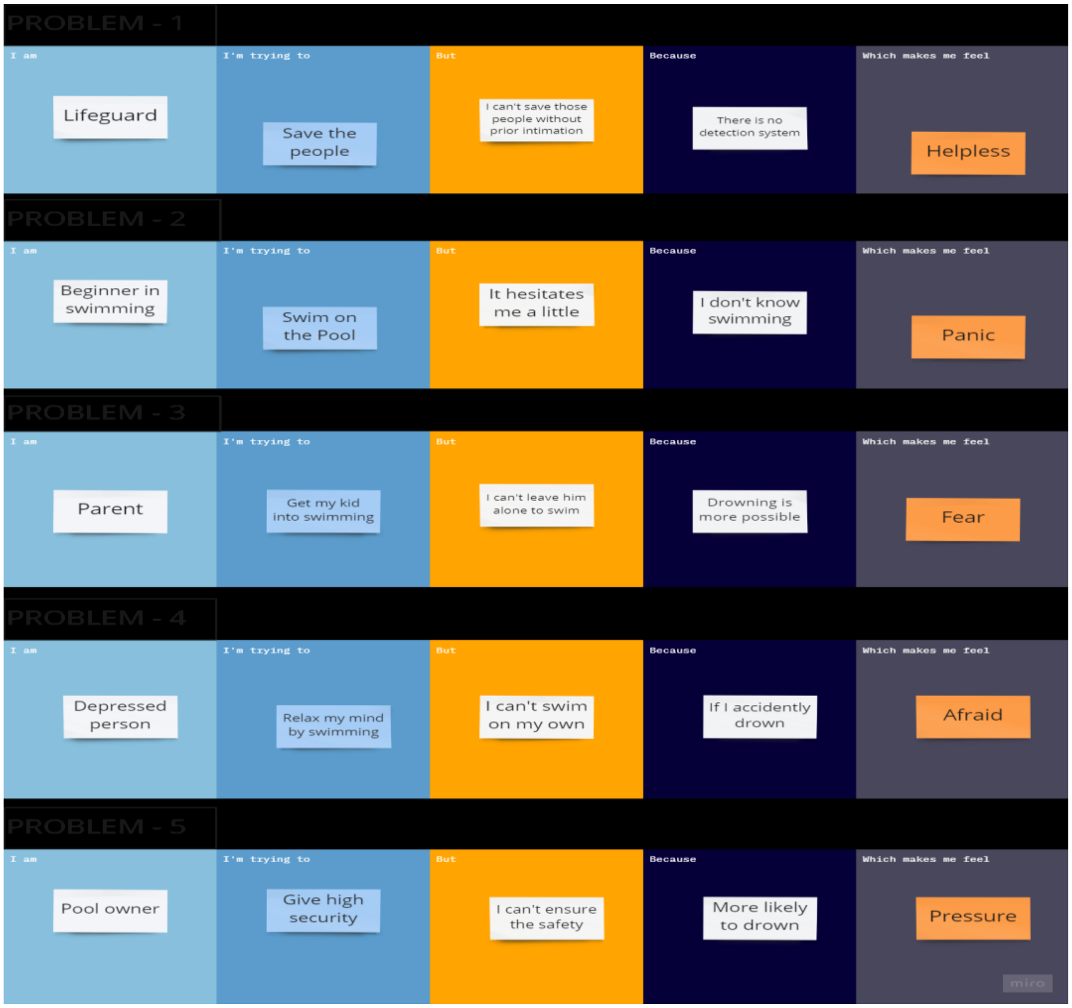
Fig 3.1 EMPATHY MAP

3.2IDEATION & BRAINSTORMING

IDEATION:

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you’ll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.



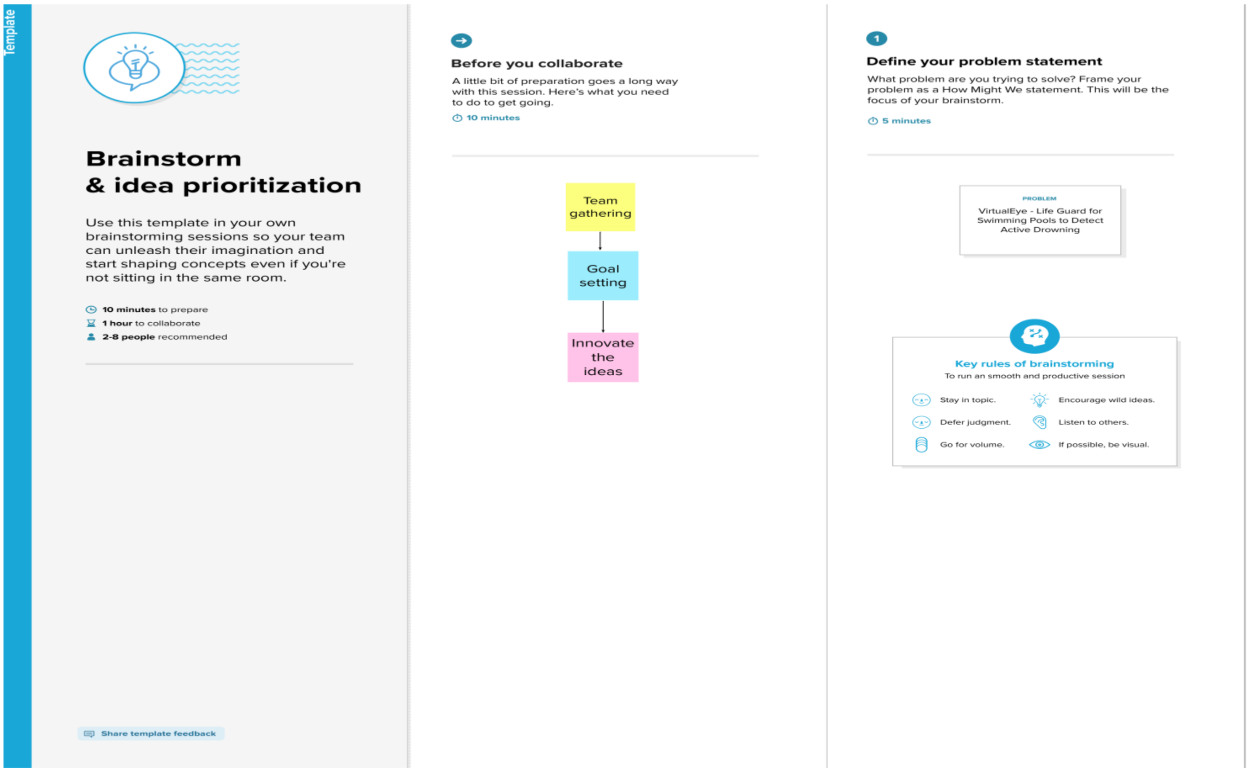
	I am	I'm trying to	But	Because	Which makes me feel
PROBLEM - 1	Lifeguard	Save the people	I can't save those people without prior intimation	There is no detection system	Helpless
PROBLEM - 2	Beginner in swimming	Swim on the Pool	It hesitates me a little	I don't know swimming	Panic
PROBLEM - 3	Parent	Get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

Fig 3.2 IDEATION

Brainstorm & Idea Prioritization:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Team gathering → Goal setting → Innovate the ideas

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

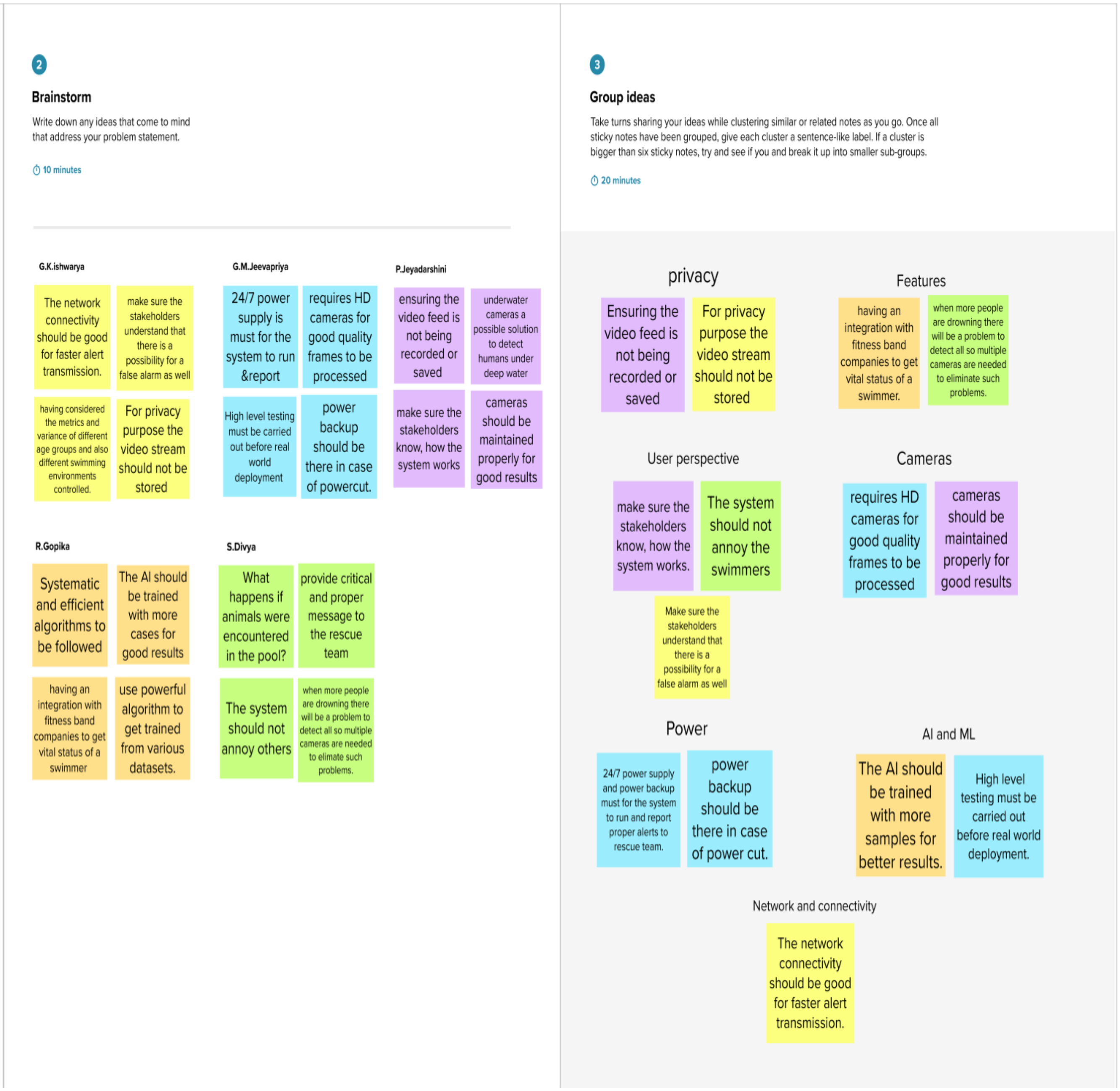
VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

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

10 minutes

G.K.ishwarya

- The network connectivity should be good for faster alert transmission.
- make sure the stakeholders understand that there is a possibility for a false alarm as well
- having considered the metrics and variance of different age groups and also different swimming environments controlled.
- For privacy purpose the video stream should not be stored

G.M.Jeevapriya

- 24/7 power supply is must for the system to run &report
- requires HD cameras for good quality frames to be processed
- High level testing must be carried out before real world deployment
- power backup should be there in case of powercut.

P.Jeyadarshini

- ensuring the video feed is not being recorded or saved
- underwater cameras a possible solution to detect humans under deep water
- make sure the stakeholders know, how the system works
- cameras should be maintained properly for good results

R.Gopika

- Systematic and efficient algorithms to be followed
- The AI should be trained with more cases for good results
- having an integration with fitness band companies to get vital status of a swimmer
- use powerful algorithm to get trained from various datasets.

S.Divya

- What happens if animals were encountered in the pool?
- provide critical and proper message to the rescue team
- The system should not annoy others
- when more people are drowning there will be a problem to detect all so multiple cameras are needed to elimate such problems.

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

privacy

- Ensuring the video feed is not being recorded or saved
- For privacy purpose the video stream should not be stored

Features

- having an integration with fitness band companies to get vital status of a swimmer.
- when more people are drowning there will be a problem to detect all so multiple cameras are needed to eliminate such problems.

User perspective

- make sure the stakeholders know, how the system works.
- The system should not annoy the swimmers
- Make sure the stakeholders understand that there is a possibility for a false alarm as well

Cameras

- requires HD cameras for good quality frames to be processed
- cameras should be maintained properly for good results

Power

- 24/7 power supply and power backup must for the system to run and report proper alerts to rescue team.
- power backup should be there in case of power cut.

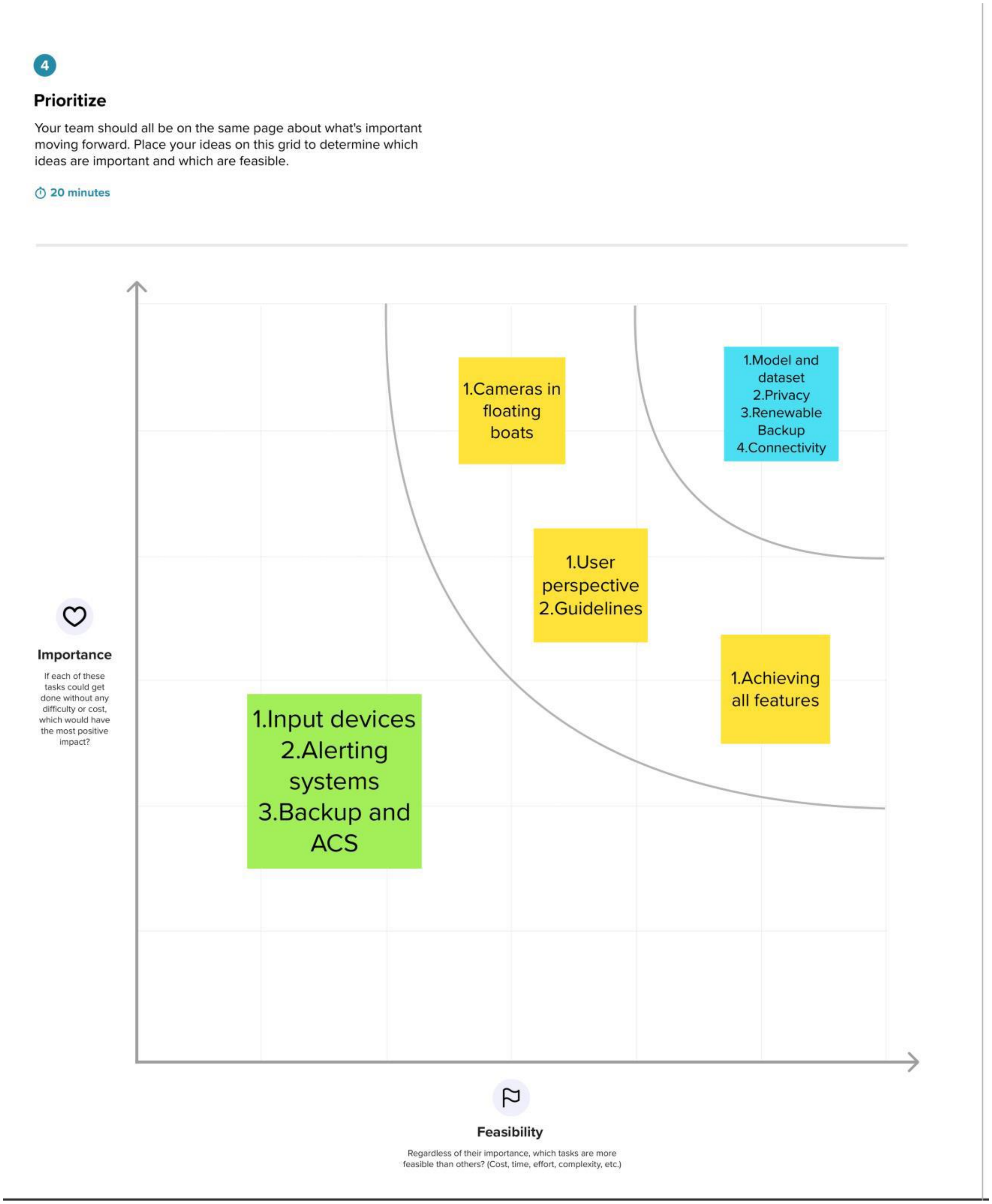
AI and ML

- The AI should be trained with more samples for better results.
- High level testing must be carried out before real world deployment.

Network and connectivity

- The network connectivity should be good for faster alert transmission.

Step-3: Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

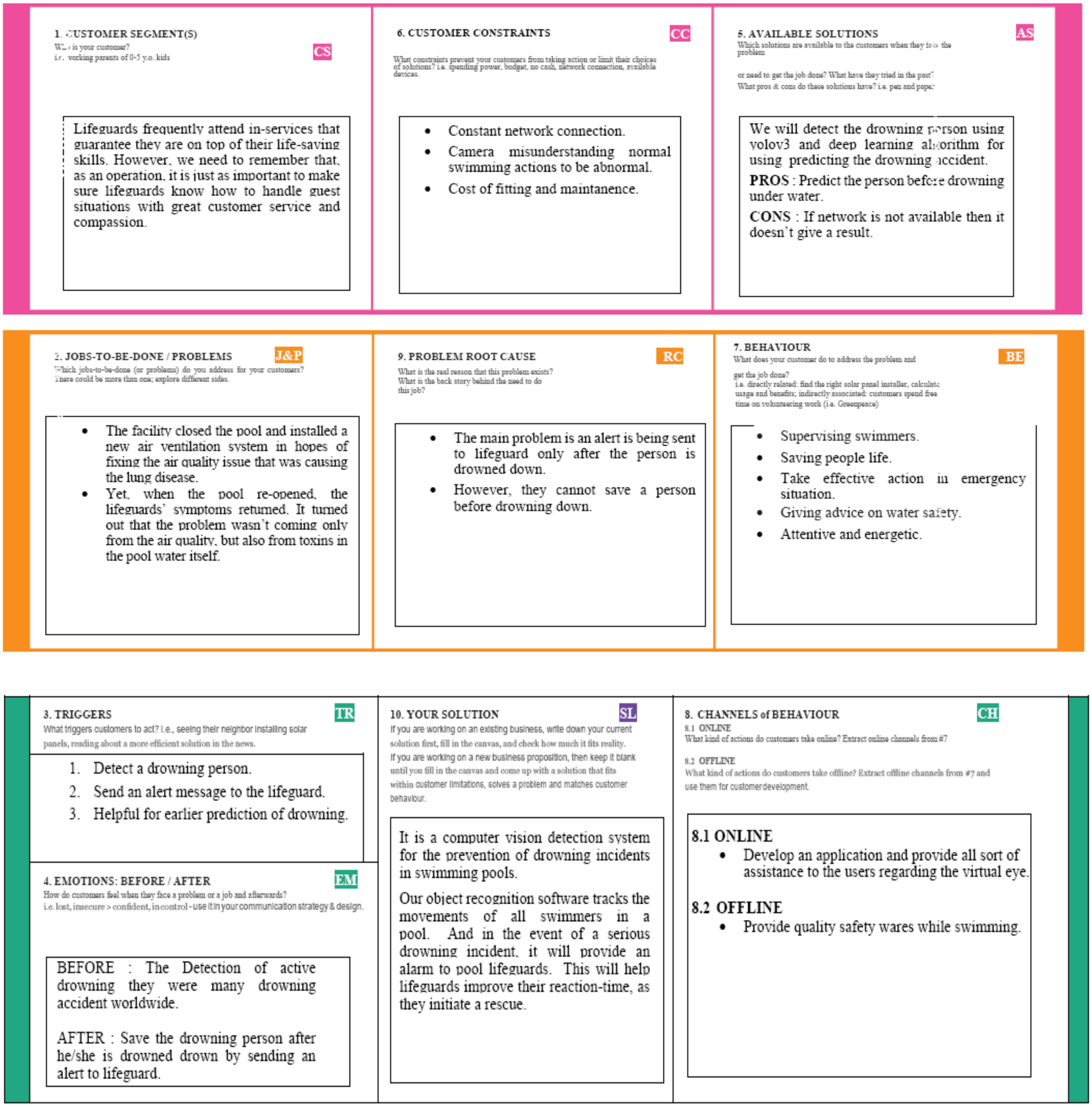
1.Cameras in floating boats
1.Model and dataset
2.Privacy
3.Renewable Backup
4.Connectivity
1.User perspective
2.Guidelines
1.Achieving all features
1.Input devices
2.Alerting systems
3.Backup and ACS
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Virtual Eye - Life Guard for Swimming Pools to Detect Active Drowning
2.	Idea / Solution description	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, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident. In this project, we will use Artificial Intelligence. We install the cameras in underwater to detect the drowning people. Using deep learning, image can be recognized. If the image is detected, it triggers the alarm to alert the Life Guard who rescue the drowning peoples.
3.	Novelty / Uniqueness	The uniqueness of our system software to track the position and the location of a drowning person. We use YOLO Algorithm. Because of its high accuracy and fast detection

		speed. So it helps lifeguard to save people within seconds.
4.	Social Impact / Customer Satisfaction	In case of an incident it is possible to extract and store not only the videos but also pulse rate of a victim so it will be useful to identify the reason behind his/her drowness.
5.	Business Model (Revenue Model)	Can generate a revenue from direct customers, like Life Guard and other swimming pool authorities.
6.	Scalability of the Solution	Our software system can be used by the company driver who manages the pools. We use the IBM cloud server to collect and maintain the data. We will ensure the safety of the swimmers.

3.4 Problem Solution fit



1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Lifeguards frequently attend in-services that guarantee they are on top of their life-saving skills. However, we need to remember that, as an operation, it is just as important to make sure lifeguards know how to handle guest situations with great customer service and compassion.

6. CUSTOMER CONSTRAINTS 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.

- Constant network connection.
- Camera misunderstanding normal swimming actions to be abnormal.
- Cost of fitting and maintanence.

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. pen and paper

We will detect the drowning person using yolov3 and deep learning algorithm for using predicting the drowning accident.

PROS : Predict the person before drowning under water.

CONS : If network is not available then it doesn't give a result.

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.

- The facility closed the pool and installed a new air ventilation system in hopes of fixing the air quality issue that was causing the lung disease.
- Yet, when the pool re-opened, the lifeguards' symptoms returned. It turned out that the problem wasn't coming only from the air quality, but also from toxins in the pool water itself.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?

- The main problem is an alert is being sent to lifeguard only after the person is drowned down.
- However, they cannot save a person before drowning down.

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 associated: customers spend free time on volunteering work (i.e. Greenpeace)

- Supervising swimmers.
- Saving people life.
- Take effective action in emergency situation.
- Giving advice on water safety.
- Attentive and energetic.

3. TRIGGERS TR

What triggers customers to act? i.e., seeing their neighbor installing solar panels, reading about a more efficient solution in the news.

1. Detect a drowning person.
2. Send an alert message to the lifeguard.
3. Helpful for earlier prediction of drowning.

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 in your communication strategy & design.

BEFORE : The Detection of active drowning they were many drowning accident worldwide.

AFTER : Save the drowning person after he/she is drowned drown by sending an alert to lifeguard.

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.

It is a computer vision detection system for the prevention of drowning incidents in swimming pools.

Our object recognition software tracks the movements of all swimmers in a pool. And in the event of a serious drowning incident, it will provide an alarm to pool lifeguards. This will help lifeguards improve their reaction-time, as they initiate a rescue.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

8.1 ONLINE

- Develop an application and provide all sort of assistance to the users regarding the virtual eye.

8.2 OFFLINE

- Provide quality safety wares while swimming.

4. REQUIREMENT ANALYSIS

CHAPTER 4
REQUIREMENT ANALYSIS

4.1 Functional requirement

FRNo.	Functional Requirement (Epic)	SubRequirement(Story/Sub-Task)
FR-1	Installation	Needed to be install under the water without Annoying to the swimmer in the swimming pool.
FR-2	Deduction	Either horrified or unconscious
FR-3	Audio	Shout for help or keep calm if the person is unconscious.
FR-4	Support	Take swim tube or take the help of rescue team.
FR-5	PriorAlert	Send alert message to the rescue team.

4.2 Non-Functional requirements

FRNo.	Non-FunctionalRequirement	Description
NFR-1	Usability	To ensure the safety of each and every person Present in the pool. A life guard should be present all the time in the pool.
NFR-2	Security	Rescue team should be aware of the alert message to save the life of the swimmer.
NFR-3	Reliability	Virtual eye lifeguards triggers an immediate prior alarm if a swimmer is in peril, helping to avoid panic even in critical situations.

NFR-4	Performance	The alarm is triggered when the swimmer's Pulse rate is decreasing.
NFR-5	Availability	Equipment and accessories include life saver rings, ashepherd's crook, life hooks, spine boards, rescue tubes, and a first aid kit. Important to keep them accessible to quickly pull someone from the water safely.
NFR-6	Scalability	Virtual eye lifeguard finds potential drowning and promptly notifies you. It features the latest artificial intelligence technology and adapts to the needs of the user.

5. PROJECT DESIGN

CHAPTER 5
PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

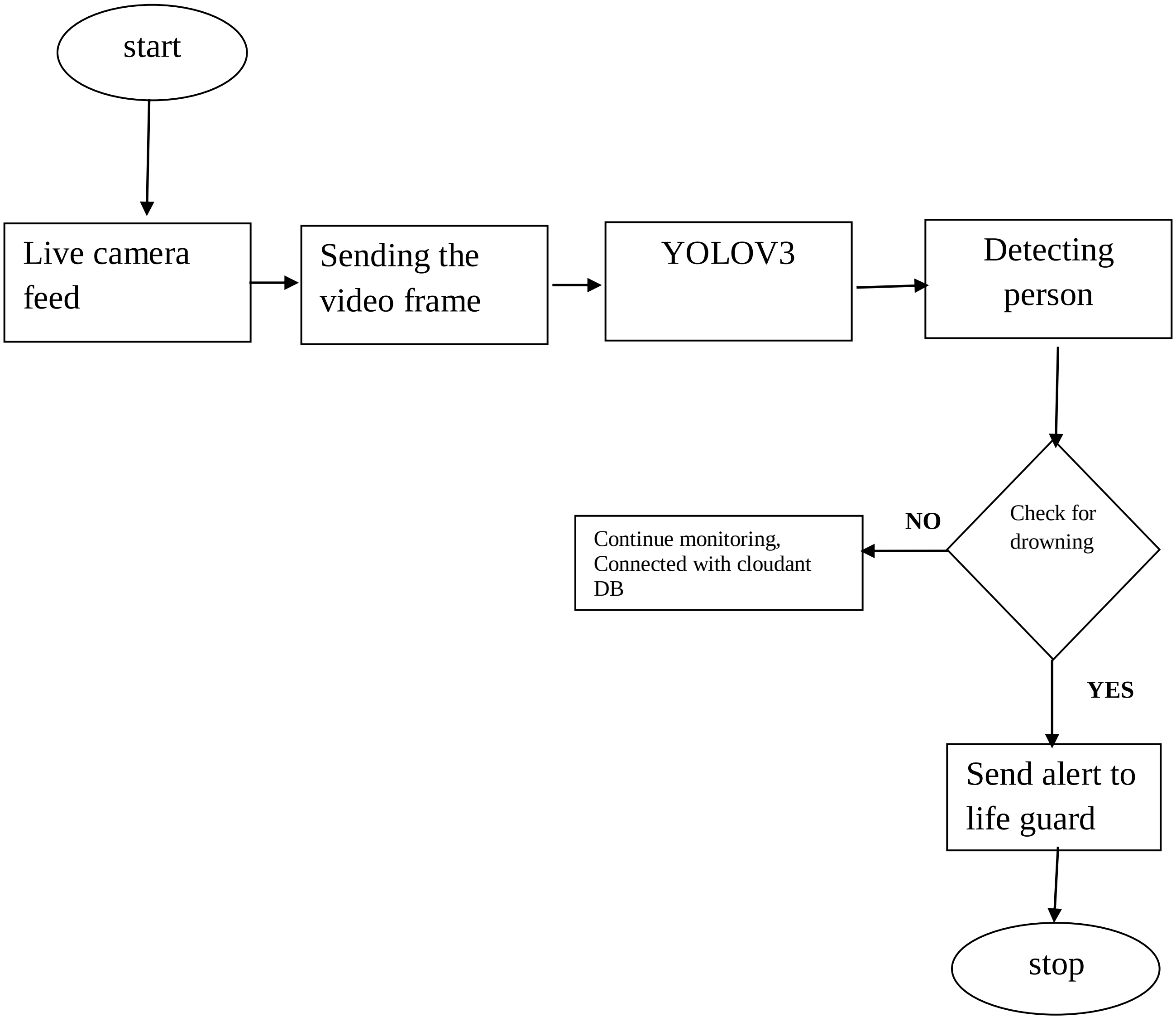


Fig 5.1 DATA FLOW DIAGRAM

5.2Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are ,Finding the best tech solution to solve existing business problems.Describe the structure, characteristics, behavior, and other aspects of the software .

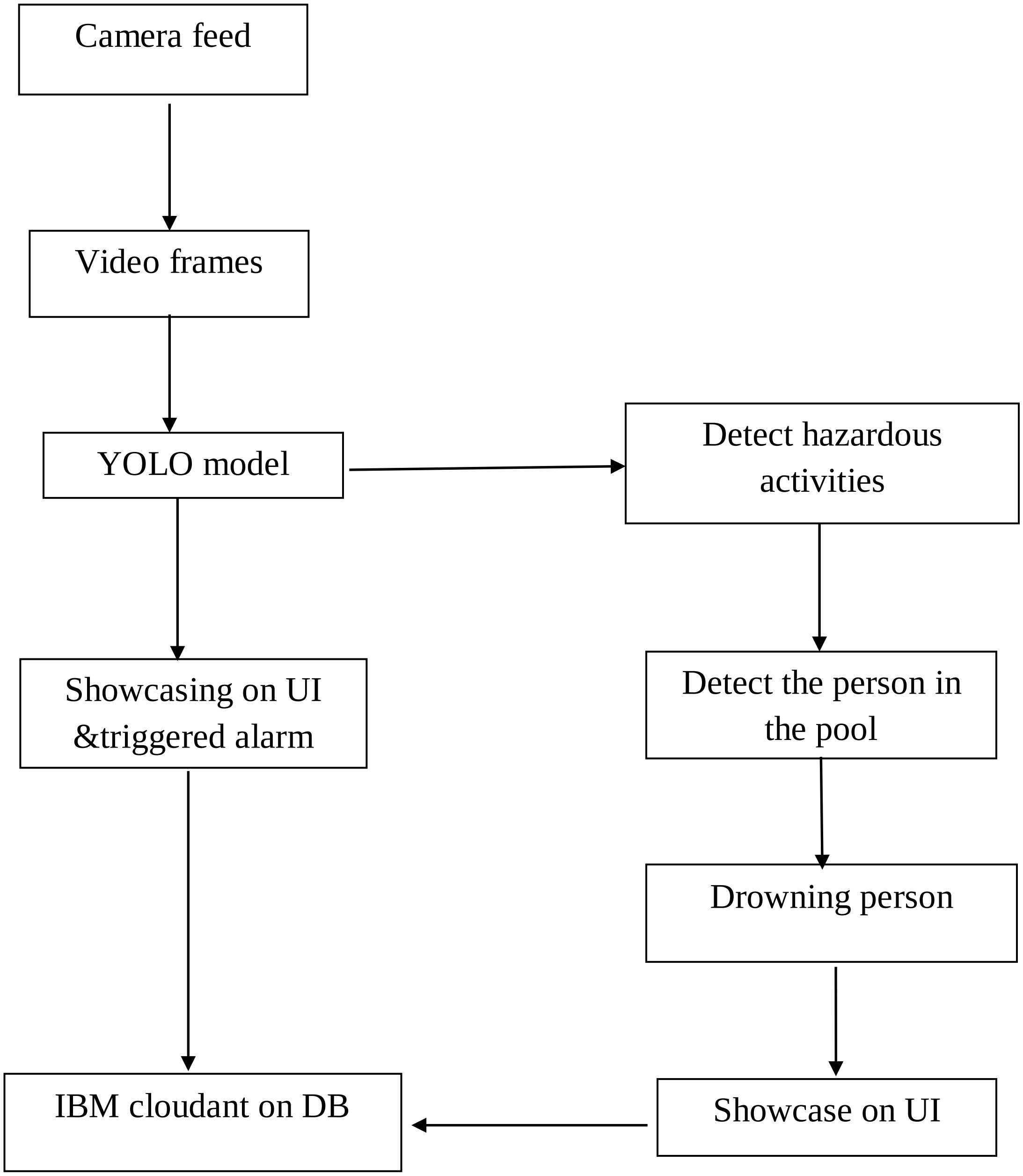

Camera feed
Video frames
YOLO model
Detect hazardous activities
Showcasing on UI &triggered alarm
Detect the person in the pool
Drowning person
IBM cloudant on DB
Showcase on UI

Fig 5.2 SOLUTION AND TECHNICAL ARCHITECTURE

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Pool owner)	Installation	USN-1	As a pool owner, I can install the cameras and set up the drowning detection system	I can connect the cameras to the cloud-hosted software	High	Sprint-1
User	Register and Login	USN-2	As a user, register an username and password on website.	Registered persons only allow on a swimming pool	medium	Sprint-1
Developers	Training and Testing	USN-3	We implement on a code to train and test the model.	Testing will be done.	High	Sprint-2
Detection (Camera)	Detecting the drowning persons	USN-4	As a user, I can find the drowning persons by using the drowning detection system.	I would receive an alert if a person is drowning.	High	Sprint-3

	Notify the lifeguard	USN-5	As a user, I can notify the lifeguard when the system detects a drowning person	can set up an alarm that would notify the lifeguard	High	Sprint-3
Customer (Lifeguard)	Rescue people	USN-6	As a user, I can rescue the drowning persons from the pool	I can save the drowning person	High	Sprint-3
Customer (Swimmers)	Safety	USN-7	As a user, I can swim without the fear of drowning	I can swim safely with the help of the system and the lifeguard	medium	Sprint-3
Customer Care Executive	Contact	USN-8	resolve technical issues	I can contact the customer care executive	Medium	Sprint-4

				to resolve any issues		
Administrator	Dashboard	USN-9	Management of the drowning detection system,database management.	I can access the system’s logs and any other data instantly	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

CHAPTER 6

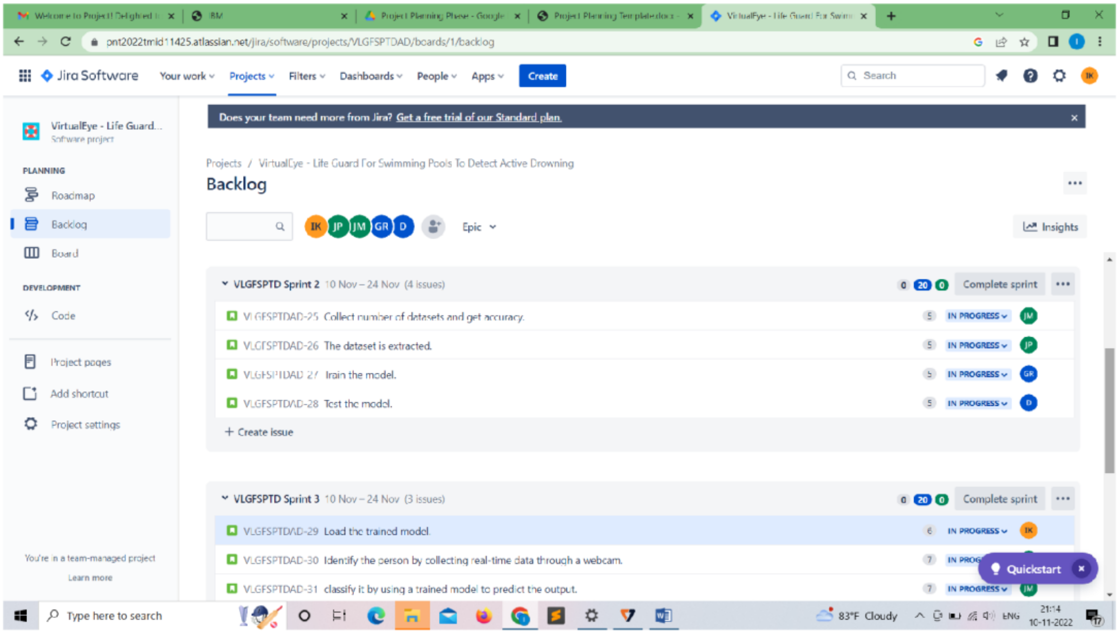
PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional requirement(epic)	User story number	User story/task	Story points	Priority	Team members
Sprint-1	Registration form	USN -1	All users will register an application form will be on website.	4	high	Ishwarya G K
Sprint-1	Registration form	USN -2	First, user can register an application form by entering a user name, mail id and password in website.	4	high	Jeyadarshini P
Sprint-1	Registration form	USN -3	Registered user will receive a confirmation mail once they have registered for the application.	4	high	Jeevapriya G M
Sprint-1	Login	USN -4	As a user, login an application by registered mail id and	4	high	Gopika R

			password.			
Sprint-1	Prediction	USN -5	All users will see a prediction demo about swimming pool.	4	Low	Divya S
Sprint-2	Dataset collection	USN -6	Collect number of datasets and get accuracy.	5	medium	Jeevapriya G M
Sprint-2	Pre – processing	USN -7	The dataset is extracted.	5	High	Jeyadarshini P
Sprint-2	Train the model	USN -8	Train the model.	5	High	Gopika R
Sprint-2	Test the model	USN -9	Test the model	5	High	Divya S
Sprint-3	Detection	USN -10	Load the trained model.	6	High	Ishwarya G K
Sprint-3	Detection	USN -11	Identify the person by collecting real-time data through a webcam.	7	Medium	Jeyadarshini P
Sprint-3	Detection	USN -12	classify it by using a trained model to predict the output	7	High	Jeevapriya G M
Sprint-4	Detection	USN -13	If person is drowning, the system will ring an alarm to give signal.	5	High	Divya S

Sprint-4	Detection	USN - 14	User will detect the drowning person.	10	Medium	Ishwarya G K
Sprint-4	logout	USN - 15	User can logout the application.	5	low	Gopika R



pnt2022tmid11425.atlassian.net/jira/software/projects/VLGFSPTDAD/boards/1/backlog

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

VirtualEye - Life Guard...
Software project

Projects / VirtualEye - Life Guard For Swimming Pools To Detect Active Drowning

Backlog

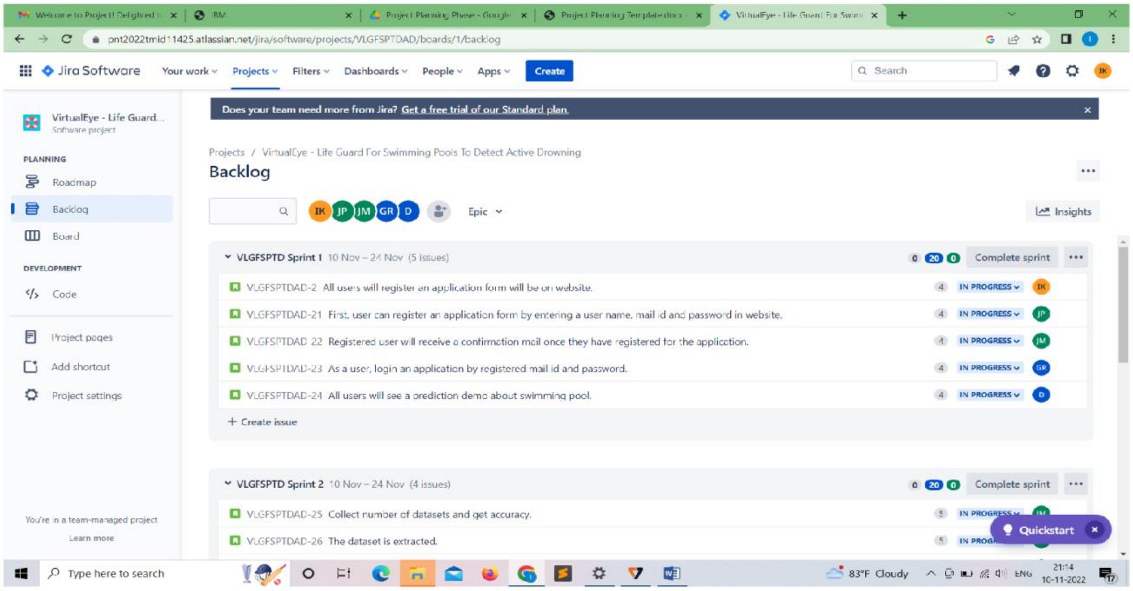
VLGFSPTD Sprint 2 10 Nov – 24 Nov (4 issues)

VLGFSPTDAD-25	Collect number of datasets and get accuracy.	5	IN PROGRESS	JM
VLGFSPTDAD-26	The dataset is extracted.	5	IN PROGRESS	JP
VLGFSPTDAD 27	Train the model.	5	IN PROGRESS	GR
VLGFSPTDAD-28	Test the model.	5	IN PROGRESS	D

VLGFSPTD Sprint 3 10 Nov – 24 Nov (3 issues)

VLGFSPTDAD-29	Load the trained model.	6	IN PROGRESS	IK
VLGFSPTDAD-30	Identify the person by collecting real-time data through a webcam.	7	IN PROG	
VLGFSPTDAD-31	classify it by using a trained model to predict the output.	7	IN PROGRESS	JM

You're in a team-managed project
Learn more



pnt2022tmid11425.atlassian.net/jira/software/projects/VLGFSPTDAD/boards/1/backlog

Projects / VirtualEye - Life Guard For Swimming Pools To Detect Active Drowning

Backlog

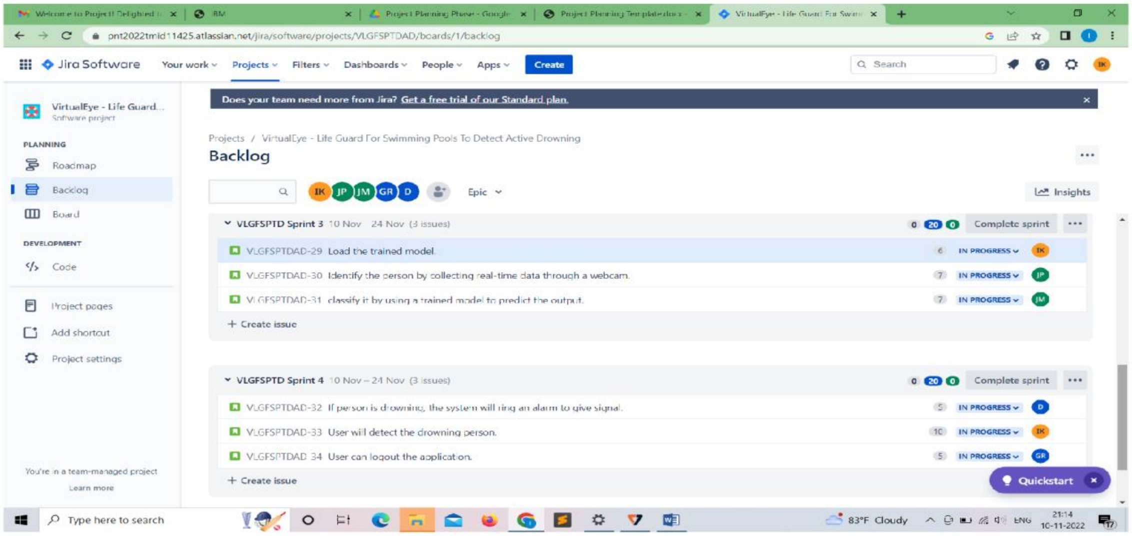
VLGFSPTD Sprint 1 10 Nov – 24 Nov (5 issues) 0 20 0

VLGFSPTDAD-2	All users will register an application form will be on website.	4	IN PROGRESS	IK
VLGFSPTDAD-21	First, user can register an application form by entering a user name, mail id and password in website.	4	IN PROGRESS	JP
VLGFSPTDAD 22	Registered user will receive a confirmation mail once they have registered for the application.	4	IN PROGRESS	JM
VLGFSPTDAD-23	As a user, login an application by registered mail id and password.	4	IN PROGRESS	GR
VLGFSPTDAD-24	All users will see a prediction demo about swimming pool.	4	IN PROGRESS	D

+ Create issue

VLGFSPTD Sprint 2 10 Nov – 24 Nov (4 issues) 0 20 0

VLGFSPTDAD-25	Collect number of datasets and get accuracy.	5	IN PROGRESS	JM
VLGFSPTDAD-26	The dataset is extracted.	5	IN PROG	



pnt2022tmid11425.atlassian.net/jira/software/projects/VLGFSPTDAD/boards/1/backlog

Projects / VirtualEye - Life Guard For Swimming Pools To Detect Active Drowning

Backlog

VLGFSPTD Sprint 3 10 Nov 24 Nov (3 issues) 0 20 0

VLGFSPTDAD-29	Load the trained model.	6	IN PROGRESS	IK
VLGFSPTDAD-30	Identify the person by collecting real-time data through a webcam.	7	IN PROGRESS	JP
VLGFSPTDAD-31	classify it by using a trained model to predict the output.	7	IN PROGRESS	JM

+ Create issue

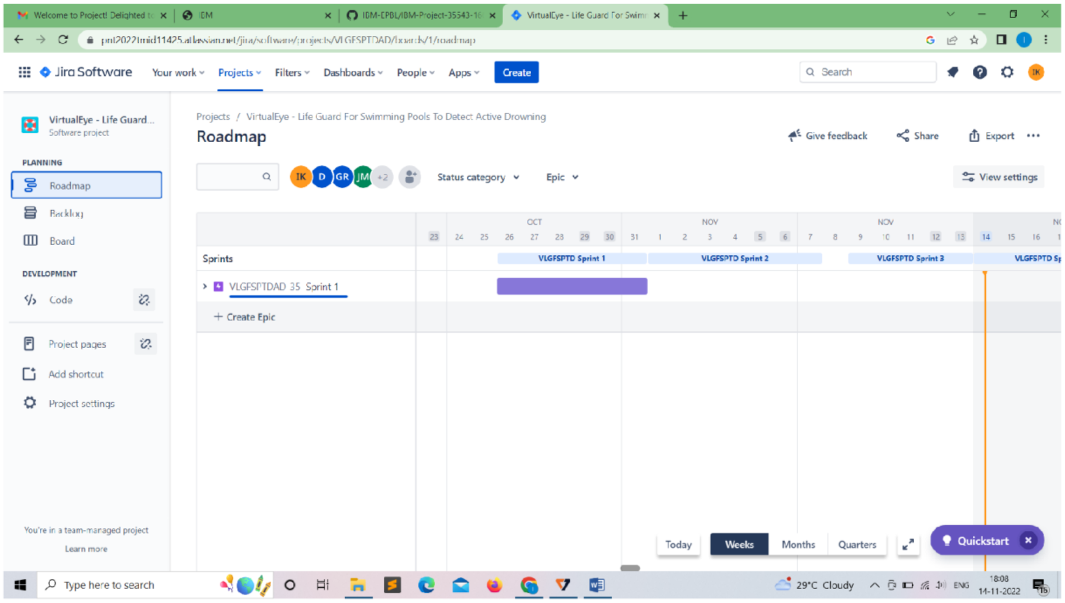
VLGFSPTD Sprint 4 10 Nov – 24 Nov (3 issues) 0 20 0

VLGFSPTDAD-32	If person is drowning, the system will ring an alarm to give signal.	5	IN PROGRESS	D
VLGFSPTDAD-33	User will detect the drowning person.	10	IN PROGRESS	IK
VLGFSPTDAD 34	User can logout the application.	5	IN PROGRESS	GR

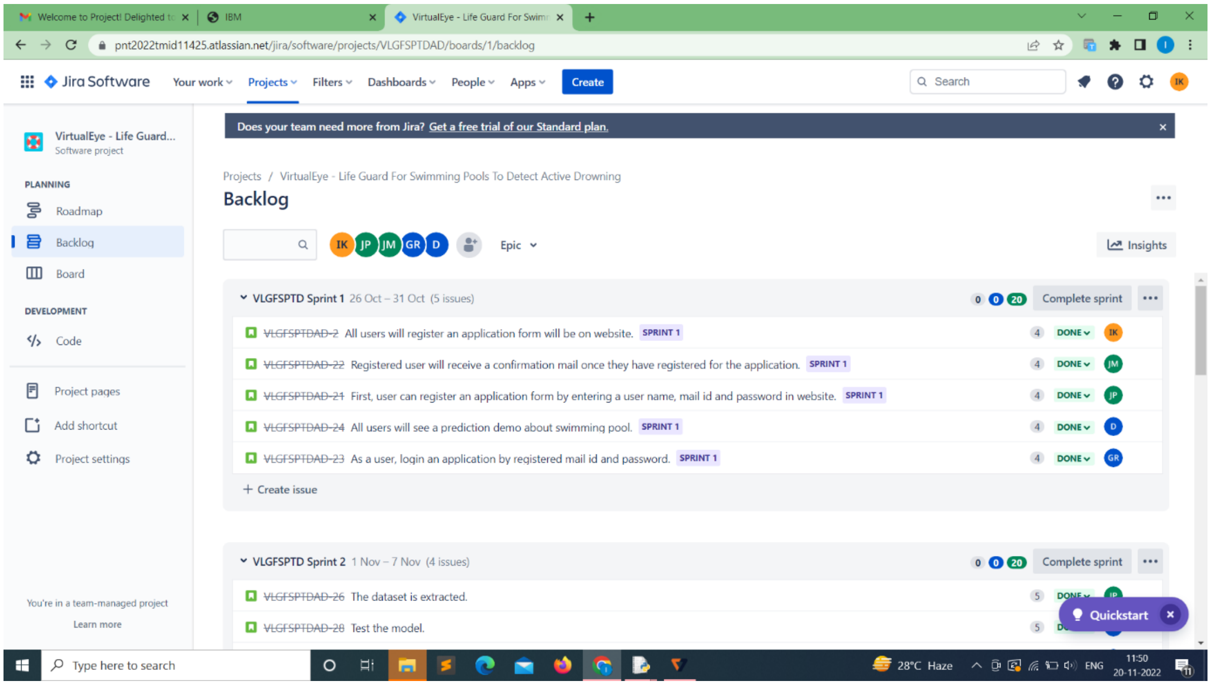
+ Create issue

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total story points	Duration	Sprint start date	Sprint end date	Story points completed (as on planned end date)	Sprint release date (actual)
Sprint-1	20	6 days	26 oct 2022	31 oct 2022	20	31 oct 2022
Sprint-2	20	6 days	01 nov 2022	06 nov 2022	20	06 nov 2022
Sprint-3	20	6 days	07 nov 2022	12 nov 2022	20	12 nov 2022
Sprint-4	20	6 days	13 nov 2022	19 nov 2022	20	19 nov 2022



6.3 REPORTS FROM JIRA



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

Projects / VirtualEye - Life Guard For Swimming Pools To Detect Active Drowning

Backlog

VLGFSPTD Sprint 1 26 Oct – 31 Oct (5 issues) 0 0 20

~~VLGFSPTDAD-2~~ All users will register an application form will be on website. SPRINT 1 4 DONE IK

~~VLGFSPTDAD-22~~ Registered user will receive a confirmation mail once they have registered for the application. SPRINT 1 4 DONE JM

~~VLGFSPTDAD-21~~ First, user can register an application form by entering a user name, mail id and password in website. SPRINT 1 4 DONE JP

~~VLGFSPTDAD-24~~ All users will see a prediction demo about swimming pool. SPRINT 1 4 DONE D

~~VLGFSPTDAD-23~~ As a user, login an application by registered mail id and password. SPRINT 1 4 DONE GR

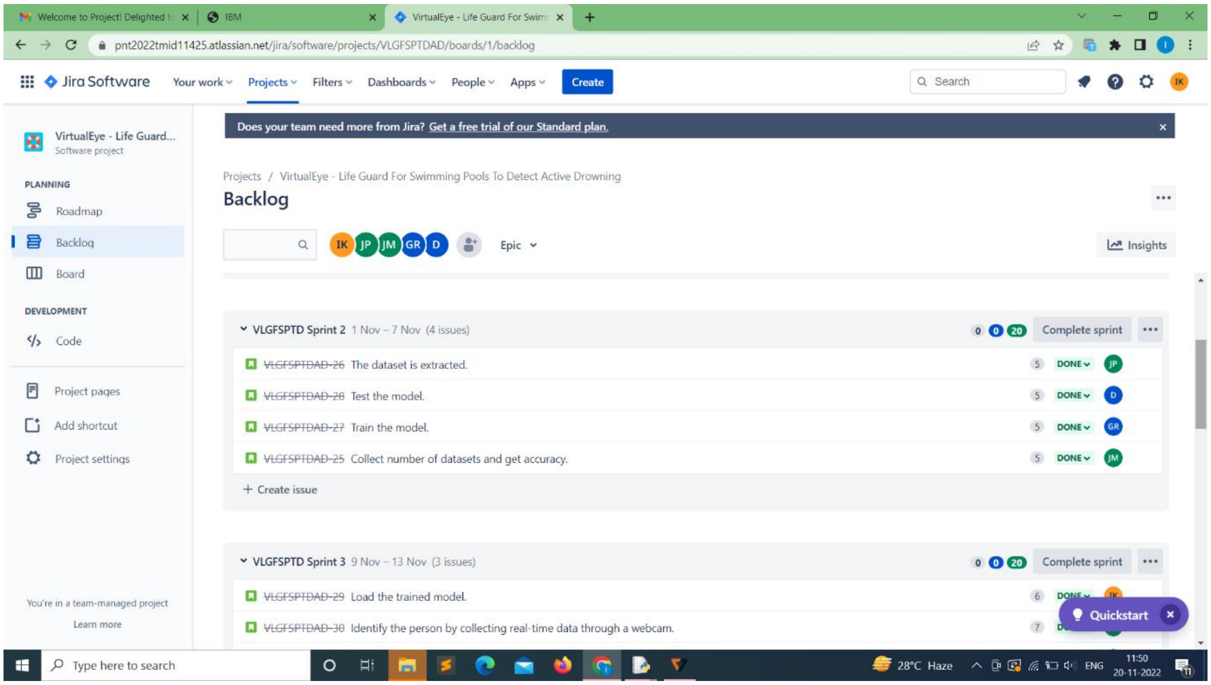
+ Create issue

VLGFSPTD Sprint 2 1 Nov – 7 Nov (4 issues) 0 0 20

~~VLGFSPTDAD-26~~ The dataset is extracted. 5 DONE JP

~~VLGFSPTDAD-28~~ Test the model. 5

Fig 6.3.1 backlog(sprint 1)



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

Projects / VirtualEye - Life Guard For Swimming Pools To Detect Active Drowning

Backlog

VLGFSPTD Sprint 2 1 Nov – 7 Nov (4 issues) 0 0 20

~~VLGFSPTDAD-26~~ The dataset is extracted. 5 DONE JP

~~VLGFSPTDAD-28~~ Test the model. 5 DONE D

~~VLGFSPTDAD-27~~ Train the model. 5 DONE GR

~~VLGFSPTDAD-25~~ Collect number of datasets and get accuracy. 5 DONE JM

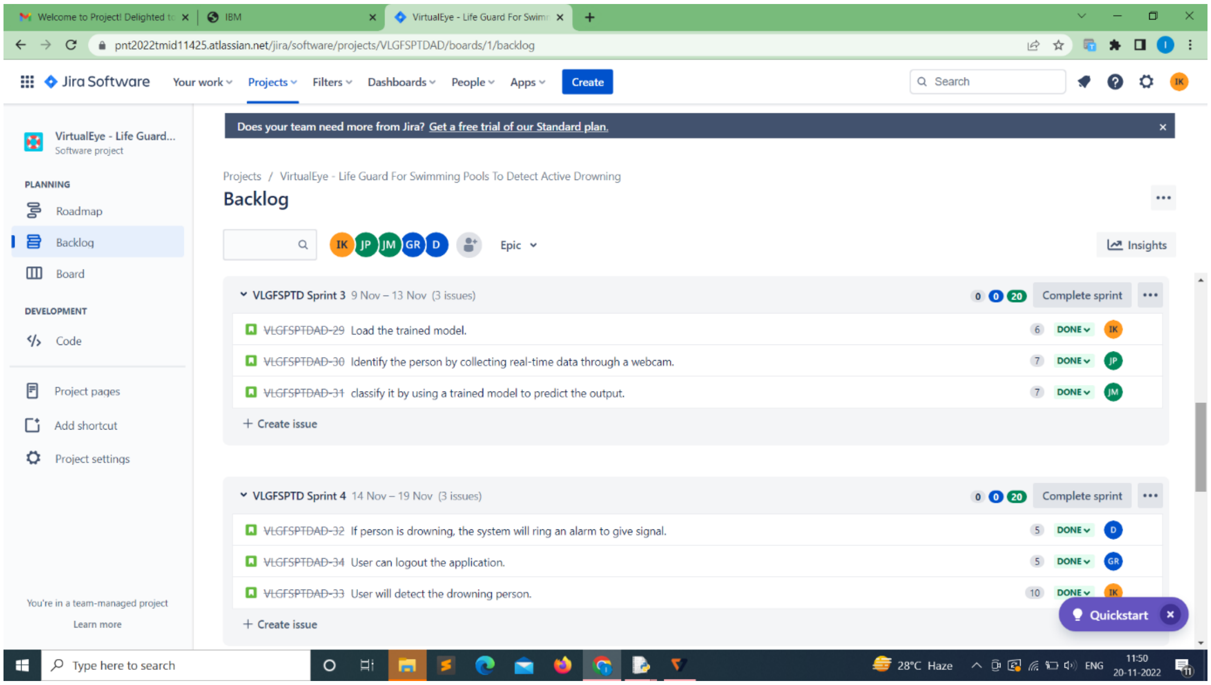
+ Create issue

VLGFSPTD Sprint 3 9 Nov – 13 Nov (3 issues) 0 0 20

~~VLGFSPTDAD-29~~ Load the trained model. 6 DONE IK

~~VLGFSPTDAD-30~~ Identify the person by collecting real-time data through a webcam. 7

Fig 6.3.2 backlog(sprint 2



Projects / VirtualEye - Life Guard For Swimming Pools To Detect Active Drowning

Backlog

VLGFSPTD Sprint 3 9 Nov – 13 Nov (3 issues) 0 0 20 Complete sprint

~~VLGFSPTDAD-29~~ Load the trained model.	6	DONE	IK
~~VLGFSPTDAD-30~~ Identify the person by collecting real-time data through a webcam.	7	DONE	JP
~~VLGFSPTDAD-31~~ classify it by using a trained model to predict the output.	7	DONE	JM

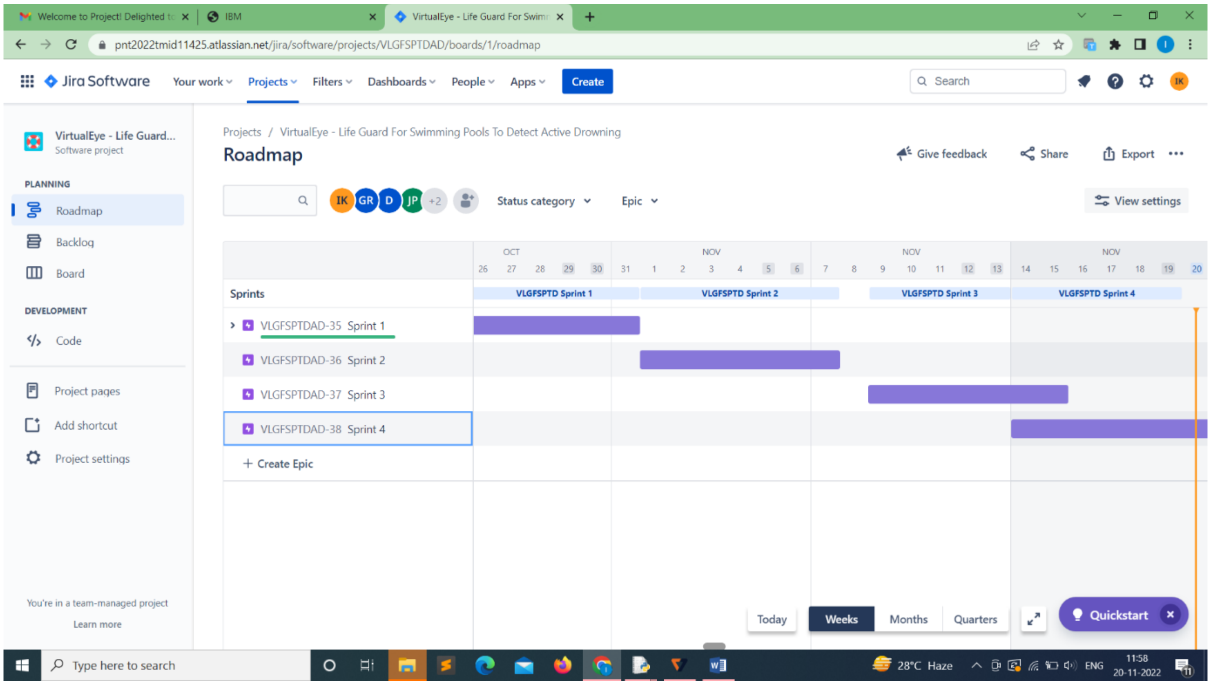
+ Create issue

VLGFSPTD Sprint 4 14 Nov – 19 Nov (3 issues) 0 0 20 Complete sprint

~~VLGFSPTDAD-32~~ If person is drowning, the system will ring an alarm to give signal.	5	DONE	D
~~VLGFSPTDAD-34~~ User can logout the application.	5	DONE	GR
~~VLGFSPTDAD-33~~ User will detect the drowning person.	10	DONE	IK

+ Create issue

Fig 6.3.3 backlog(sprint 3,4)



Projects / VirtualEye - Life Guard For Swimming Pools To Detect Active Drowning

Roadmap

Sprints	VLGFSPTD Sprint 1	VLGFSPTD Sprint 2	VLGFSPTD Sprint 3	VLGFSPTD Sprint 4
VLGFSPTDAD-35 Sprint 1				
VLGFSPTDAD-36 Sprint 2				
VLGFSPTDAD-37 Sprint 3				
VLGFSPTDAD-38 Sprint 4				

+ Create Epic

Fig 6.3.4 roadmap

7. CODING & SOLUTIONING

CHAPTER 7

CODING & SOLUTIONING

7.1 FEATURE 1

DROWNING PERSON DETECTION:

In order to quickly help lifesavers judge whether people are drowning in the swimming pool. First, by analyzing the spatial distribution of swimming pool when swimmers are normally swimming, the data labeling and swimmer detection methods are determined. Second, a behavior recognition framework of swimmers on the basis of YOLOv3 algorithm (BR-YOLOv3). The spatial relationship between the location information of the target and swimming/drowning area of swimming pool is analyzed to further determine the swimmer’s drowning or swimming behavior.

Coding:

```
import time
from absl import app, flags, logging
from absl.flags import FLAGS
import cv2
import numpy as np
import tensorflow as tf
from yolov3_tf2.models import (
YoloV3, YoloV3Tiny)
from yolov3_tf2.dataset import transform_images, load_tfrecord_dataset
from yolov3_tf2.utils import draw_outputs
flags.DEFINE_string('classes', './data/labels/coco.names', 'path to classes file')
flags.DEFINE_string('weights', './weights/yolov3.tf',
'path to weights file')
flags.DEFINE_boolean('tiny', False, 'yolov3 or yolov3-tiny')
flags.DEFINE_integer('size', 416, 'resize images to')
flags.DEFINE_list('images', '/data/images/dog.jpg', 'list with paths to input
images')
flags.DEFINE_string('tfrecord', None, 'tfrecord instead of image')
flags.DEFINE_string('output', './detections/', 'path to output folder')
flags.DEFINE_integer('num_classes', 80, 'number of classes in the model')
def main(_argv):
physical_devices = tf.config.experimental.list_physical_devices('GPU')
if len(physical_devices) > 0:
tf.config.experimental.set_memory_growth(physical_devices[0], True)
if FLAGS.tiny:
yolo = YoloV3Tiny(classes=FLAGS.num_classes)
else:
yolo = YoloV3(classes=FLAGS.num_classes)
yolo.load_weights(FLAGS.weights).expect_partial()
print('weights loaded')
class_names = [c.strip() for c in open(FLAGS.classes).readlines()]
print('classes loaded')
if FLAGS.tfrecord:
dataset = load_tfrecord_dataset(
FLAGS.tfrecord, FLAGS.classes, FLAGS.size)
dataset = dataset.shuffle(512)
img_raw, _label = next(iter(dataset.take(1)))
else:
raw_images = []
```

```
images = FLAGS.images
for image in images:
img_raw = tf.image.decode_image(
open(image, 'rb').read(), channels=3)
raw_images.append(img_raw)
num = 0
for raw_img in raw_images:
num+=1
img = tf.expand_dims(raw_img, 0)
img = transform_images(img, FLAGS.size)
t1 = time.time()
boxes, scores, classes, nums = yolo(img)
t2 = time.time()
logging.info('time: {}'.format(t2 - t1))
print('detections:')
for i in range(nums[0]):
print('\t{}, {}, {}'.format(class_names[int(classes[0][i])],
np.array(scores[0][i]), np.array(boxes[0][i])))
img = cv2.cvtColor(raw_img.numpy(), cv2.COLOR_RGB2BGR)
img = draw_outputs(img, (boxes, scores, classes, nums), class_names)
cv2.imwrite(FLAGS.output + 'detection' + str(num) + '.jpg', img)
print('output saved to: {}'.format(FLAGS.output + 'detection' + str(num) + '.jpg'))
if __name__ == '__main__':
try:
app.run(main)
except SystemExit:
pass
```

7.2 FEATURE 2

DROWNING DETECTION ALARM

If any abnormal activities, breathing suffocation or drowning in the swimming pool, alarm will ring. So that the lifeguard can help the drowning person from the pool and can save the life.

Coding:

```
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 пр
from playsound import playsound
import requests
from flask import Flask, request, render_template, redirect, url_for
#Loading the model
from cloudant.client import Cloudant
client=Cloudant.iam('e80322c6-5b15-4385-ba6a-c587c3471e4b-
bluemix','Kf6tBtrDrpQZtYfredJ-rkYky1lX39giPycwe0lhCmyj',connect=True)
# Create a database using an initialized client
my_database = client['my_db']
#my_database = client.create_database('my_db')
app=Flask(__name__,template_folder='template')
```

```
@app.route('/')
def index():
return render_template("index.html")
@app.route('/register', methods=['POST', 'GET'])
def register():
return render_template('register.html')
@app.route('/afterreg', methods=['POST', 'GET'])
def afterreg():
#x = [x for x in request.form.values()]
#print(x)
uname = request.args.get('name')
username = request.args.get('email')
password = request.args.get('psw')
print(list(request.form.values()))
#_id=(request.form.get("_id",False))
#name=(request.form.get("name",False))
#psw=(request.form.get("psw",False))
'''data = {
'_id': _id, # Setting _id is optional
'name': name,
'psw':psw}'''
#data = {'_id': x[1],'name': x[0],'psw': x[2]}
data = {
'name': uname,
'email': username,
'psw': password
}
```

```
print (data)
query = {'email': {'$eq': data['email']}}
#query = {'_id': {'$eq': data['_id']}}
docs = my_database.get_query_result(query)
print(docs)
print(len(docs.all()))
if(len(docs.all())==0):
url = my_database.create_document(data)
return render_template('register.html',prediction="Registration Successful, please
login using your details")
else:
return render_template('register.html',prediction="You are already a member,
please login using your details")
#login page
@app.route('/login')
def login():
return render_template('register.html')
@app.route('/afterlogin',methods=['POST', 'GET'])
def afterlogin():
user = request.args.get('email')
passw = request.args.get('psw')
print (user, passw)
query = {'email': {'$eq': user}}
docs = my_database.get_query_result(query)
print(docs)
my_database.get_query_result(query)
print(len(docs.all()))
```

```
if(len(docs.all())==0):
return render_template("register.html", prediction="The username is not found.")
else:
if((user==docs[0][0]['email'] and passw==docs[0][0]['psw'])):
return redirect(url_for('prediction'))
else:
print('Invalid User')
@app.route('/Logout')
def logout ():
return render_template('Logout.html')
@app.route('/prediction',methods=["GET","POST"])
def prediction():
return render_template('prediction.html')
@app.route('/result',methods=["GET","POST"])
def result():
webcam = cv2.VideoCapture('drowning.mp4')
if not webcam. isOpened():
print("Could not open webcam")
exit()
t0 = time.time() #gives time in seconds after 1970
#variabledcount stands for how many seconds the person has been standing still for
centre0 = np.zeros(2)
isDrowning = False
#this loop happens approximately every 1 second, so if a person doesn't move,
#or moves very little for 10seconds, we can say they are drowning
#loop through frames
while webcam.isOpened():
```

```
status, frame = webcam.read()
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])
#this threshold is for checking how much the centre has moved
x=time.time()
threshold = 10
if (hmov>threshold or vmov>threshold):
print(x-to, 's')
t0=time.time()
isDrowning = False
else:
print(x-t, 's')
if( (time.time() - t0) > 10):
isDrowning = True
#print('bounding box: bbox, 'label: label, 'confidence: conf[0], 'centre: ', centre)
#print (bbox, label,conf, centre)
```

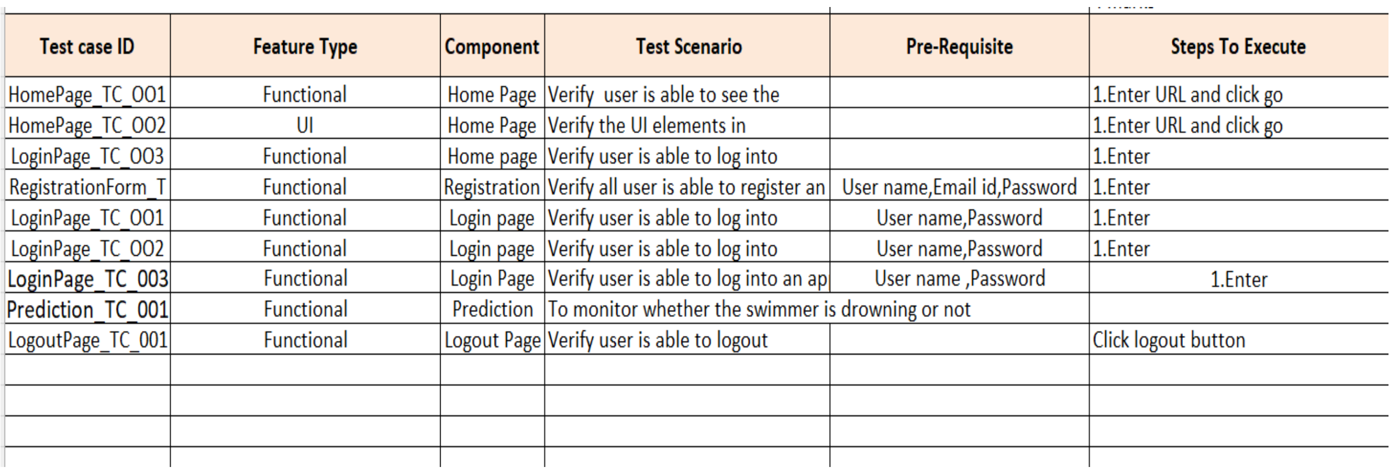
```
print('bbox: ', bbox, 'centre:', centre, 'centre:', centre0)
print('Is he drowning: ', isDrowning)
centre0 = centre
#draw bounding box over detected objects
out = draw_bbox (frame, bbox, label, conf,isDrowning)
#print('Seconds since last epoch: ', time.time()-to)
# display output
cv2.imshow("Real-time object detection", out)
if(isDrowning == True):
playsound('alarm.mp3')
webcam.release()
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")
#returnrender_template('base.html')
if cv2.waitkey (1) & 0xFF == ord('q'):
system.exit()
webcam.release()
cv2.destroyAllWindows()
#returnrender_template('prediction.html',)
if __name__ == "__main__":
app.run(port=4000,debug=True)
```

8. TESTING

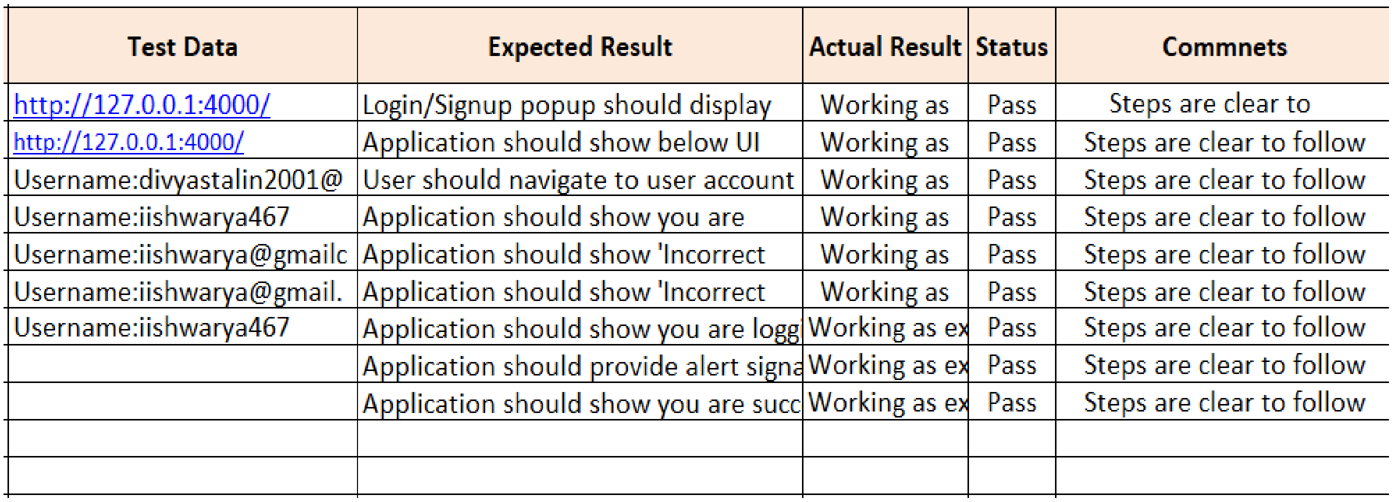
CHAPTER 8

TESTING

8.1 Test Cases



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute
HomePage_TC_001	Functional	Home Page	Verify user is able to see the		1.Enter URL and click go
HomePage_TC_002	UI	Home Page	Verify the UI elements in		1.Enter URL and click go
LoginPage_TC_003	Functional	Home page	Verify user is able to log into		1.Enter
RegistrationForm_T	Functional	Registration	Verify all user is able to register an	User name,Email id,Password	1.Enter
LoginPage_TC_001	Functional	Login page	Verify user is able to log into	User name,Password	1.Enter
LoginPage_TC_002	Functional	Login page	Verify user is able to log into	User name,Password	1.Enter
LoginPage_TC_003	Functional	Login Page	Verify user is able to log into an ap	User name ,Password	1.Enter
Prediction_TC_001	Functional	Prediction	To monitor whether the swimmer is drowning or not		
LogoutPage_TC_001	Functional	Logout Page	Verify user is able to logout		Click logout button



Test Data	Expected Result	Actual Result	Status	Comnmets
http://127.0.0.1:4000/	Login/Signup popup should display	Working as	Pass	Steps are clear to
http://127.0.0.1:4000/	Application should show below UI	Working as	Pass	Steps are clear to follow
Username:divyastalin2001@	User should navigate to user account	Working as	Pass	Steps are clear to follow
Username:iishwarya467	Application should show you are	Working as	Pass	Steps are clear to follow
Username:iishwarya@gmailc	Application should show 'Incorrect	Working as	Pass	Steps are clear to follow
Username:iishwarya@gmail.	Application should show 'Incorrect	Working as	Pass	Steps are clear to follow
Username:iishwarya467	Application should show you are logg	Working as ex	Pass	Steps are clear to follow
	Application should provide alert signa	Working as ex	Pass	Steps are clear to follow
	Application should show you are succ	Working as ex	Pass	Steps are clear to follow

8.2 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).

2. 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	8	6	5	6	25
Duplicate	3	2	3	2	10
External	5	3	2	1	11
Fixed	11	4	4	20	39
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	27	20	18	31	96

3. Test Case Analysis

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

Section	**Total**	**Not**	**Fail**	**Pass**

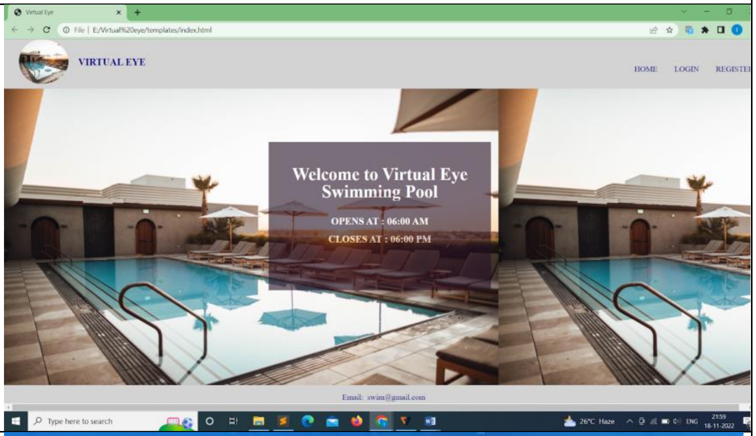
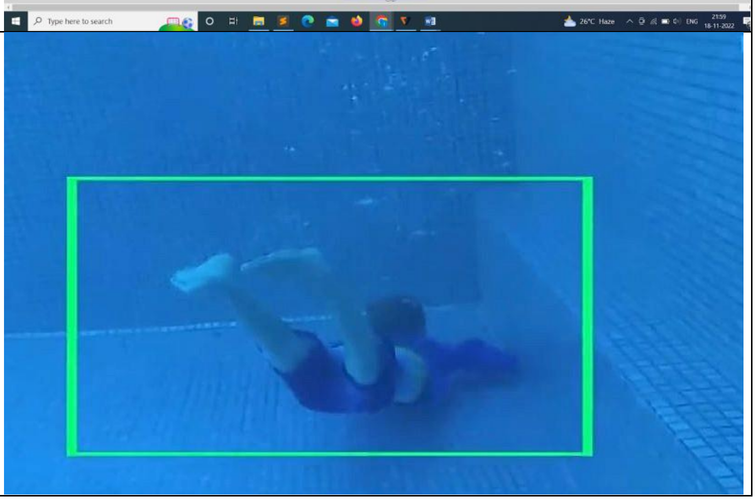
	Cases	**Tested**		
Print Engine	6	0	0	6
Client Application	45	0	0	45
Security	3	0	0	3
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

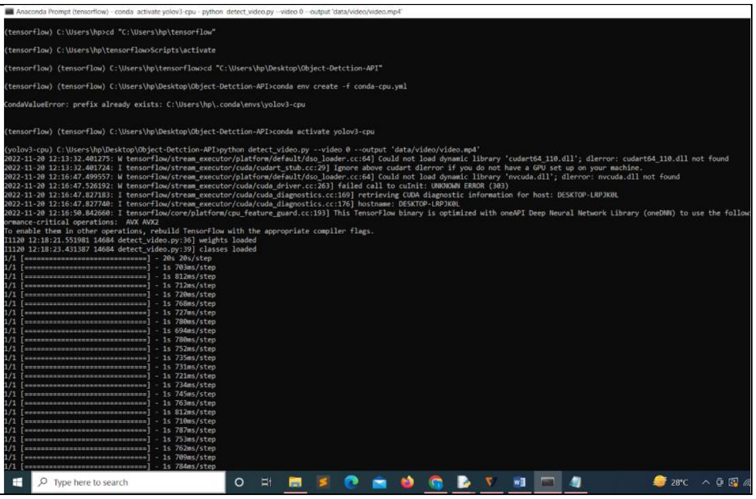
9. RESULTS

CHAPTER 9

RESULTS

9.1 Performance Metrics

S.No.	**Parameter**	**Values**	**Screenshot**
1.	Model Summary	100%	 Virtual Eye File | E:/Virtual%20eye/templates/index.html VIRTUAL EYE HOME LOGIN REGISTE Welcome to Virtual Eye Swimming Pool OPENS AT : 06:00 AM CLOSES AT : 06:00 PM Email: swim@gmail.com Type here to search 26°C Haze ENG 21:59 18-11-2022
2.	Accuracy	Training Accuracy – 95% Validation Accuracy – 92%	

3.	Confidence Score (Only Yolo Projects)	Class Detected – 96% Confidence Score –98%	

```
Anaconda Prompt (tensorflow) - conda  activate yolov3-cpu - python  detect_video.py --video 0 --output 'data/video/video.mp4'

(tensorflow) C:\Users\hp>cd "C:\Users\hp\tensorflow"

(tensorflow) C:\Users\hp\tensorflow>Scripts\activate

(tensorflow) (tensorflow) C:\Users\hp\tensorflow>cd "C:\Users\hp\Desktop\Object-Detction-API"

(tensorflow) (tensorflow) C:\Users\hp\Desktop\Object-Detction-API>conda env create -f conda-cpu.yml

CondaValueError: prefix already exists: C:\Users\hp\.conda\envs\yolov3-cpu


(tensorflow) (tensorflow) C:\Users\hp\Desktop\Object-Detction-API>conda activate yolov3-cpu

(yolov3-cpu) C:\Users\hp\Desktop\Object-Detction-API>python detect_video.py --video 0 --output 'data/video/video.mp4'
2022-11-20 12:13:32.401275: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-11-20 12:13:32.401724: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2022-11-20 12:16:47.499557: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2022-11-20 12:16:47.526192: W tensorflow/stream_executor/cuda/cuda_driver.cc:263] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-20 12:16:47.827183: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: DESKTOP-LRPJK0L
2022-11-20 12:16:47.827740: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: DESKTOP-LRPJK0L
2022-11-20 12:16:50.842660: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the follow
ormance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
I1120 12:18:21.551981 14684 detect_video.py:36] weights loaded
I1120 12:18:23.431387 14684 detect_video.py:39] classes loaded
1/1 [==============================] - 20s 20s/step
1/1 [==============================] - 1s 703ms/step
1/1 [==============================] - 1s 812ms/step
1/1 [==============================] - 1s 712ms/step
1/1 [==============================] - 1s 720ms/step
1/1 [==============================] - 1s 768ms/step
1/1 [==============================] - 1s 727ms/step
1/1 [==============================] - 1s 780ms/step
1/1 [==============================] - 1s 694ms/step
1/1 [==============================] - 1s 780ms/step
1/1 [==============================] - 1s 752ms/step
1/1 [==============================] - 1s 735ms/step
1/1 [==============================] - 1s 731ms/step
1/1 [==============================] - 1s 721ms/step
1/1 [==============================] - 1s 734ms/step
1/1 [==============================] - 1s 745ms/step
1/1 [==============================] - 1s 763ms/step
1/1 [==============================] - 1s 812ms/step
1/1 [==============================] - 1s 710ms/step
1/1 [==============================] - 1s 787ms/step
1/1 [==============================] - 1s 753ms/step
1/1 [==============================] - 1s 762ms/step
1/1 [==============================] - 1s 709ms/step
1/1 [==============================] - 1s 784ms/step
```

10. ADVANTAGES & DISADVANTAGES

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- The monitoring system can help to reduce drowning and assure pool safety effectively.
- This system ability to save a drowning victim in under a minute has been explained in experiments.
- This system don’t have to wait until life guard comes to rescue because it has uplifting mesh.
- More effective and cost Efficient than previous other models.
- The full security system promotes the development of waterlifesaving services, which is also the objective requirement for thecurrent development of swimming lifeguards.
- Video surveillance can be used as a tool for monitoring and security.

Disadvantages

- Early on, failure to recognize a drowning scene could result in a longer rescue time, which is a significant issue to consider in a timecritical emergency.
- The wearable-based system is the discomfort of use, which may lead to younger children seeking to alleviate the discomfort by removing the device, which is an unsubstantiated theory.
- Internet connection is necessary to use GPS or sending alert messages. Sometimes to send messages SIM balance may be required.

11. CONCLUSION

CHAPTER 11

CONCLUSION

- Thus the VirtualEye software has worked in close integration with the camerainstalled in the pool to continuously to scan the swimming pool.
- The spatial relationship between the locationinformation of the target and swimming/drowning area of swimming poolis analyzed to determine the swimmer's drowning or swimming behavior.
- Thus using convolutional neural network (CNN) models, it can detect adrowning person.
- Through the live video stream from under water cameras,the swimmers are monitor by "state-of-the-art" object recognition software.
- Thus successfully the alert signals and the alert messageare given to the lifeguards, while a person drowning in the swimming pool.

12. FUTURE SCOPE

CHAPTER 12

FUTURE SCOPE

- 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.
- The single camera 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.
- 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.
- As swimming in extreme weather conditions is not preferred, the system could be further improved to emit a warning signal if a person was to swim in any of the above weather conditions.

13. APPENDIX

Source code

INDEX.HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Virtual Eye</title>
<link rel="stylesheet" type="text/css" href="style.css">
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style type="text/css">
html,body,div,span,object,iframe,h1,h2,h3,h4,h5,h6,p,blockquote,pre,abbr,address,
cite,code,del,dfn,em,img,ins,kbd,q,samp,small,strong,sub,sup,var,b,i,dl,dt,dd,ol,ul,
li,fieldset,form,label,legend,table,caption,tbody,tfoot,thead,tr,th,td,article,aside,can
vas,details,figcaption,figure,footer,header,hgroup,menu,nav,section,summary,time,
mark,audio,
video {
margin: 0;
padding: 0;
border: 0;
outline: 0;
font-size: 100%;
vertical-align: baseline;
background: transparent;
}
body {
line-height: 1;
```

```
}
article,aside,details,figcaption,figure,footer,header,hgroup,menu,nav,
section {
display: block;
}
nav ul {
list-style: none;
}
blockquote,
q {
quotes: none;
}
blockquote:before,
blockquote:after,
q:before,
q:after {
content: ";
content: none;
}
a {
margin: 0;
padding: 0;
font-size: 100%;
vertical-align: baseline;
background: transparent;
}
ins {
```

```
background-color: #ff9;
color: #000;
text-decoration: none;
}
mark {
background-color: #ff9;
color: #000;
font-style: italic;
font-weight: bold;
}
del {
text-decoration: line-through;
}
abbr[title],
dfn[title] {
border-bottom: 1px dotted;
cursor: help;
}
table {
border-collapse: collapse;
border-spacing: 0;
}
/* change border colour to suit your needs */
hr {
display: block;
height: 1px;
border: 0;
```

```
border-top: 1px solid #cccccc;
margin: 1em 0;
padding: 0;
}
input,
select {
vertical-align: middle;
}
/*nav {
float: right;
word-spacing: 30px;
padding: 20px;
}
nav li {
display: inline-block;
line-height: 80px;
}*/
/*---------------------------Main code   INDEX---------------------------------------------
---------------------*/
.wrapper {
height: 600px;
width: 1200px;
/*background-color: red;*/
}
header {
height: 100px;
width: 1536px;
```

```
background-color: lightgrey;
}
section {
height: 600px;
width: 1520px;
background-color: grey;
}
footer {
height: 80px;
width: 1536px;
background-color: lightgrey;
}
.logo {
float: left;
padding-left: 20px;
display: flex;
align-items: center;
justify-content: center
}
.logoimg {
padding-left: 10px;
}
li a {
color: white;
text-decoration: none;
}
.round {
```

```
border-radius: 50%;
}
.pics {
flex-basis: 40%
}
.txt {
font-size: 20px;
padding-left: 20px;
}
/*nav
{
float: right;
word-spacing: 40px;
padding: 20px;
}
nav li
{
display: inline-block;
line-height: 80px;
}*/
section .sec_img {
height: 607px;
margin-top: 0px;
background-image: url("static/img/s1.jpg");
}
.box {
height: 300px;
```

```
width: 450px;
background-color: #251025;
margin: 60px auto;
opacity: .6;
color: white;
}
.btn {
font-family: 'arial black';
color: #FA4BD1 !important;
font-size: 17px;
padding: 17px 29px;
transform: translate(145px);
border-radius: 40px;
border: 1px solid #FFFFFF;
background: #FFFFFF;
}
/*.btn:hover {
color: #FFFFFF !important;
background: #FAFAFA;
background: linear-gradient(to top, #FAFAFA, #EB79FC);
}*/
</style>
</head>
<body>
<div class="wrapper">
<form action="/register" method="post">
<header>
```

```
<div class="logo">
<div class="pics">
<img class="round" src="static/img/s1.jpg" width="100" height="90">
</div>
<div class="txt">
<h1 style="color: darkblue;">VIRTUAL EYE</h1>
</div>
</div>
<!--<nav>
<ul>
<li><a href="#" style="color: darkblue;">HOME</a></li>
<li><a href="#" style="color: darkblue;">LOGIN</a></li>
<li><a href="#" style="color: darkblue;">REGISTER</a></li>
</ul>
</nav>-->
</header>
<section>
<div class="sec_img">
<br><br><br>
<div class="box">
<br><br><br>
<h1 style="text-align: center; font-size: 35px;">Welcome to Virtual Eye
Swimming Pool</h1><br><br>
<h1 style="text-align: center; font-size: 20px;">OPENS AT : 06:00 AM</h1><br>
<h1 style="text-align: center; font-size: 20px;">CLOSES AT : 06:00
PM</h1><br>
<button class="btn" type="submit">WELCOME</button>
```

```
</div>
</div>
</section>
<footer>
<p style="color: darkblue;text-align: center;">
<br>
Email:&nbsp swim@gmail.com <br><br>
mobile:&nbsp 8679598745
</p>
</footer>
</form>
</div>
</body>
</html>
```

PREDICTION.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Virtual Eye</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
html,body,div,span,object,iframe,h1,h2,h3,h4,h5,h6,p,blockquote,pre,abbr,address,
cite,code,del,dfn,em,img,ins,kbd,q,samp,small,strong,sub,sup,var,b,i,dl,dt,dd,ol,ul,
li,fieldset,form,label,legend,table,caption,tbody,tfoot,thead,tr,th,td,article,aside,can
vas,details,figcaption,figure,footer,header,hgroup,menu,nav,section,summary,time,
mark,audio,
video {
margin: 0;
padding: 0;
border: 0;
outline: 0;
font-size: 100%;
vertical-align: baseline;
background: transparent;
}
body {
line-height: 1;
}
article,aside,details,figcaption,figure,footer,header,hgroup,menu,nav,
section {
display: block;
}
nav ul {
list-style: none;
}
blockquote,
q {
quotes: none;
}
blockquote:before,
blockquote:after,
q:before,
q:after {
content: ";
content: none;
}
a {
margin: 0;
padding: 0;
font-size: 100%;
vertical-align: baseline;
background: transparent;
}
/* change colours to suit your needs */
ins {
background-color: #ff9;
color: #000;
text-decoration: none;
}
/* change colours to suit your needs */
mark {
background-color: #ff9;
color: #000;
font-style: italic;
font-weight: bold;
}
del {
```

```
text-decoration: line-through;
}
abbr[title],
dfn[title] {
border-bottom: 1px dotted;
cursor: help;
}
table {
border-collapse: collapse;
border-spacing: 0;
}
/* change border colour to suit your needs */
hr {
display: block;
height: 1px;
border: 0;
border-top: 1px solid #cccccc;
margin: 1em 0;
padding: 0;
}
input,
select {
vertical-align: middle;
}
nav {
float: right;
word-spacing: 30px;
```

```
padding: 20px;
}
nav li {
display: inline-block;
line-height: 80px;
}
body {
font-family: Arial, Helvetica, sans-serif;
margin: 0;
}
/* Style the header */
.header {
padding: 40px;
text-align: center;
background: #1abc9c;
color: white;
}
/* Increase the font size of the h1 element */
.header h1 {
font-size: 40px;
}
/* Style the top navigation bar */
.navbar {
overflow: hidden;
background-color: #333;
}
/* Style the navigation bar links */
```

```
.navbar a {
float: left;
display: block;
color: white;
text-align: center;
padding: 14px 20px;
text-decoration: none;
}
/* Right-aligned link */
.navbara.right {
float: right;
}
/* Change color on hover */
.navbar a:hover {
background-color: #ddd;
color: black;
}
/*-------------------------------prediction---------------------------*/
.pred_img {
height: 500px;
margin-top: 0px;
}
.header {
padding: 60px;
text-align: center;
background: #1abc9c;
color: white;
font-size: 30px;
}
.left {
padding: 20px;
text-indent: 10px;
}
</style>
</head>
<body>
<div class="header">
<h1>VIRTUAL EYE LIFEGUARD FOR SWIMMING POOL TO DETECT
ACTIVE DROWNING</h1>
</div>
<div class="navbar">
<a href="#">Virtual Eye</a>
<a href="{{ url_for('logout') }}" class="right">LOGOUT</a>
<a href="{{ url_for('index') }}" class="right">HOME</a>
</div>
<section id="about">
<div class="body">
<div class="left">
<p style="letter-spacing: 1.5px;font-family: monospace;font-size: 17px;">
```

Swimming is one of the best exercises that helps people toreduce stress in this urban lifestyle. Swimming pools are foundlarger in number in the hotels, weekend tourist spots and barelypeople have in their house backyard. Beginners, especially oftenfeel it difficult to breathe under water and causes breathingtrouble which in turn cause a drowning accident. Worldwide,drowning produces a higher rate of

```
mortality without causinginjury to children. Children under six of their age are
found to besuffering the highest drowning mortality rates worldwide.Suchkinds of
deaths account for the third cause of unplanned deathglobally, with about 1.2
million cases yearly.
</p>
</div>
</div>
<center>
<div class="left">
<div class="prediction-input">
<img class="d-block w-100" src="static/img/swim1.jpg" alt="Second slide"
width="2000" height="500">
<br>
<form method="post" id="form" action="/predict" enctype="multipart/form-
data">
<!--<input type="file" name="file" autocomplete="off" required>-->
<input type="submit" class="submit" value="Click
Me! For a Demo">
</form>
<!--<center>
{% if filename %}
<div>
<imgsrc="{{url_for('response',filename=filename)}}" width="50%"
height="400px"/>
</div>
{% endif %}
</center>-->
```

```
</div>
<div>
<h5 style="text-color:Red">
<b style="text-color:Red">{{prediction}}</b>
</h5>
</div>
</div>
</center>
</section>
<br><br>
<section id="footer">
<p>Copyright Ã,Â© 2022. All Rights Reserved</p>
</section>
</body>
</html>
```

REGISTER.HTML

```
<html>
<head>
<title>
Virtual Eye
</title>
<link rel="stylesheet" type="text/css" href="style.css">
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
```

```
@import
url('https://fonts.googleapis.com/css?family=Poppins:400,500,600,700&display=s
wap');
* {
margin: 0;
padding: 0;
box-sizing: border-box;
font-family: 'Poppins', sans-serif;
}
html,
body {
display: grid;
height: 100%;
width: 100%;
place-items: center;
/*background: -webkit-linear-gradient(left, #003366,#004080,#0059b3
, #0073e6);*/
background-image: url(static/img/swim1.jpg)
}
::selection {
background: #1a75ff;
color: #fff;
}
.wrapper {
overflow: hidden;
max-width: 390px;
background: #fff;
```

```
padding: 30px;
border-radius: 15px;
box-shadow: 0px 15px 20px rgba(0, 0, 0, 0.1);
}
.wrapper .title-text {
display: flex;
width: 200%;
}
.wrapper .title {
width: 50%;
font-size: 35px;
font-weight: 600;
text-align: center;
transition: all 0.6s cubic-bezier(0.68, -0.55, 0.265, 1.55);
}
.wrapper .slide-controls {
position: relative;
display: flex;
height: 50px;
width: 100%;
overflow: hidden;
margin: 30px 0 10px 0;
justify-content: space-between;
border: 1px solid lightgrey;
border-radius: 15px;
}
.slide-controls .slide {
```

```
height: 100%;
width: 100%;
color: #fff;
font-size: 18px;
font-weight: 500;
text-align: center;
line-height: 48px;
cursor: pointer;
z-index: 1;
transition: all 0.6s ease;
}
.slide-controls label.signup {
color: #000;
}
.slide-controls .slider-tab {
position: absolute;
height: 100%;
width: 50%;
left: 0;
z-index: 0;
border-radius: 15px;
background: -webkit-linear-gradient(left, #003366, #004080, #0059b3, #0073e6);
transition: all 0.6s cubic-bezier(0.68, -0.55, 0.265, 1.55);
}
input[type="radio"] {
display: none;
}
```

```
#signup:checked~.slider-tab {
left: 50%;
}
#signup:checked~label.signup {
color: #fff;
cursor: default;
user-select: none;
}
#signup:checked~label.login {
color: #000;
}
#login:checked~label.signup {
color: #000;
}
#login:checked~label.login {
cursor: default;
user-select: none;
}
.wrapper .form-container {
width: 100%;
overflow: hidden;
}
.form-container .form-inner {
display: flex;
width: 200%;
}
.form-container .form-inner form {
```

```
width: 50%;
transition: all 0.6s cubic-bezier(0.68, -0.55, 0.265, 1.55);
}
.form-inner form .field {
height: 60px;
width: 100%;
margin-top: 20px;
}
.form-inner form .field input {
height: 100%;
width: 100%;
outline: none;
padding-left: 15px;
border-radius: 15px;
border: 1px solid lightgrey;
border-bottom-width: 2px;
font-size: 17px;
transition: all 0.3s ease;
}
.form-inner form .field input:focus {
border-color: #1a75ff;
/* box-shadow: inset 0 0 3px #fb6aae; */
}
.form-inner form .field input::placeholder {
color: #999;
transition: all 0.3s ease;
}
```

```
form .fieldinput:focus::placeholder {
color: #1a75ff;
}
.form-inner form .pass-link {
margin-top: 5px;
}
.form-inner form .signup-link {
text-align: center;
margin-top: 30px;
}
.form-inner form .pass-link a,
.form-inner form .signup-link a {
color: #1a75ff;
text-decoration: none;
}
.form-inner form .pass-link a:hover,
.form-inner form .signup-link a:hover {
text-decoration: underline;
}
form .btn {
height: 50px;
width: 100%;
border-radius: 15px;
position: relative;
overflow: hidden;
}
form .btn .btn-layer {
```

```
height: 100%;
width: 300%;
position: absolute;
left: -100%;
background: -webkit-linear-gradient(right, #003366, #004080, #0059b3, #0073e6);
border-radius: 15px;
transition: all 0.4s ease;
;
}
form .btn:hover .btn-layer {
left: 0;
}
form .btn input[type="submit"] {
height: 100%;
width: 100%;
z-index: 1;
position: relative;
background: none;
border: none;
color: #fff;
padding-left: 0;
border-radius: 15px;
font-size: 20px;
font-weight: 500;
cursor: pointer;
}
</style>
```

```
</head>
<body>
<div class="wrapper">
<div class="title-text">
<div class="title login">Login Form</div>
<div class="title signup">Signup Form</div>
</div>
<div class="form-container">
<div class="slide-controls">
<input type="radio" name="slide" id="login" checked>
<input type="radio"name="slide" id="signup">
<label for="login" class="slide login">Login</label>
<label for="signup" class="slide signup">Signup</label>
<div class="slider-tab"></div>
</div>
<div class="form-inner">
<form action="/login" method="post" class="login">
<div class="field">
<input type="text" name="email" placeholder="Email Address" required>
</div>
<div class="field">
<input type="password" name="psw" placeholder="Password" required>
</div>
<div class="pass-link"><a href="#">Forgot password?</a></div>
<div class="field btn">
<div class="btn-layer"></div>
<input type="submit" value="Login">
```

```
{{msg}}
</div>
<div class="signup-link">Not a member? <a href="">Signup now</a></div>
</form>
<form action="/register" method="post" class="signup">
<div class="field">
<input type="text" placeholder="Name" name="name" required>
</div>
<div class="field">
<input type="text" placeholder="Email Address" name="email" required>
</div>
<div class="field">
<input type="password" placeholder="Password" name="psw" required>
</div>
<div class="field">
<input type="password" placeholder="Confirm password" name="psw" required>
</div>
<div class="field btn">
<div class="btn-layer"></div>
<input type="submit" value="Signup">
{{prediction}}
</div>
</form>
</div>
</div>
</div>
<script type="text/javascript">
```

```
const loginText = document.querySelector(".title-text .login");
const loginForm = document.querySelector("form.login");
const loginBtn = document.querySelector("label.login");
const signupBtn = document.querySelector("label.signup");
const signupLink = document.querySelector("form .signup-link a");
signupBtn.onclick = (() => {
loginForm.style.marginLeft = "-50%";
loginText.style.marginLeft = "-50%";
});
loginBtn.onclick = (() => {
loginForm.style.marginLeft = "0%";
loginText.style.marginLeft = "0%";
});
signupLink.onclick = (() => {
signupBtn.click();
return false;
});
</script>
</body>
</html>
```

LOGOUT.HTML

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<title>Logout</title>
<style>
body {
background-image: url(static/img/bg1.jpg);
background-position: center;
background-repeat: no-repeat;
background-size: cover;
}
.container {
background-color: aliceblue;
padding-bottom: 10px;
margin-top: 120px;
margin-left: 250px;
margin-right: 250px;
margin-bottom: 200px;
}
.txt {
padding-top: 100px;
font-size: 50px;
}
.btn {
background-color: cadetblue;
border: 0.5;
color: white;
text-align: center;
font-size: 16px;
}
```

```
</style>
</head>
<body>
<div class="container">
<center>
<h1 class="txt">Successfully logged out!</h1>
<h4>Login for more information</h4>
</center>
<center><a href="{{ url_for('register') }}">
<input class="btn" type="submit" name="submit" value="Login" style="color:
black; width: 70px; height: 30px"></a>
</center>
</body></html>
```

app2.py

```
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 time
from playsound import playsound
import requests
from cloudant.client import Cloudant
from flask import Flask, flash, redirect, render_template, request, url_for,
Response
from werkzeug.utils import secure_filename
```

```
#import detect
UPLOAD_FOLDER = "static/uploads/"
RESULTS_FOLDER = "static/results/"
app=Flask(__name__,template_folder='template')
app.secret_key = "secret-key"
app.config["UPLOAD_FOLDER"] = UPLOAD_FOLDER
from cloudant.client import Cloudant
client=Cloudant.iam('e80322c6-5b15-4385-ba6a-c587c3471e4b-
bluemix','Kf6tBtrDrpQZtYfredJ-rkYky1lX39giPycwe0lhCmyj',connect=True)
@app.route("/")
def index():
return render_template("index.html")
@app.route("/register", methods=["GET", "POST"])
def register():
if request.method == "POST":
# Get the form data
try:
uname = request.args.get('name')
username = request.args.get('email')
psw = request.args.get('psw')
print(list(request.form.values()))
#email = request.form["email"]
#password = request.form["password"]
# Create a database using an initialized client
my_database = client['my_db']
# Check that the database doesn't already exist
if my_database.exists():
```

```
print(f"'{my_database}' successfully created.")
# Create a JSON document
json_document = {
"_id": email,
"name": uname,
"email": email,
"psw": psw,
}
if email in my_database:
return render_template("register.html", msg="Email already exists")
else:
# Create a document using the Database API
new_document = my_database.create_document(json_document)
return render_template("register.html", msg="Account created successfully!")
except Exception as e:
return render_template("register.html", msg="Something went wrong! Please try
again")
if request.method == "GET":
return render_template("register.html")
@app.route("/login", methods=["GET", "POST"])
def login():
if request.method == "POST":
username = request.args.get('email')
psw = request.args.get('psw')
print (username, psw)
# Create a database using an initialized client
my_database = client['my_db']
```

```
query = {'email': {'$eq': username}}
docs = my_database.get_query_result(query)
print(docs)
my_database.get_query_result(query)
print(len(docs.all()))
if(len(docs.all())==0):
return render_template("register.html", prediction="The username is not found.")
else:
if((username==docs[0][0]['email'] and psw==docs[0][0]['psw'])):
return redirect(url_for("predict"))
else:
return render_template("login.html", msg="Invalid credentials!")
if request.method == "GET":
return render_template("login.html")
@app.route("/predict", methods=["GET", "POST"])
def predict():
if request.method == "POST":
webcam = cv2.VideoCapture("drowning.mp4")
if not webcam.isOpened():
print("Could not open webcam")
exit()
t0 = time.time()  # gives time in seconds after 1970
# variabledcount stands for how many seconds the person has been standing still
for
centre0 = np.zeros(2)
isDrowning = False
# this loop happens approximately every 1 second, so if a person doesn't move,
# or moves very little for 10seconds, we can say they are drowning
# loop through frames
t0 = time.time()  # gives time in seconds after 1970
# variabledcount stands for how many seconds the person has been standing still for
centre0 = np.zeros(2)
isDrowning = False
# this loop happens approximately every 1 second, so if a person doesn't move,
# or moves very little for 10seconds, we can say they are drowning
# loop through frames
while webcam.isOpened():
# read frame from webcam
status, frame = webcam.read()
if not status:
print("Could not read frame")
exit()
# apply object detection
bbox, label, conf = cv.detect_common_objects(frame)
# simplifying for only 1 person
# s = (len(bbox), 2)
print(bbox)
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 = 30
if hmov> threshold or vmov> threshold:
print(x - t0, "s")
t0 = time.time()
isDrowning = False
else:
print(x - t0, "s")
if (time.time() - t0) > 5:
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)
```

```
print(isDrowning)
if isDrowning == True:
playsound("alarm.mp3")
# press "Q" to stop
if cv2.waitKey(1) & 0xFF == ord("q"):
break
# release resources
webcam.release()
cv2.destroyAllWindows()
if isDrowning == True:
return render_template("prediction.html",prediction='Emergency!!! The person is drowning')
else:
return render_template("logout.html")
return render_template("logout.html")
if request.method == "GET":
return render_template("prediction.html")
@app.route("/logout", methods=["GET"])
def logout():
return render_template("logout.html")
if __name__ == "__main__":
app.run(port=4000,debug=True)
```

init.py

```
from .object_detection import detect_common_objects
```

object detection

```
#import necessary packages
import cv2
import os
import numpy as np
from cvlib.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 + 'yolov4'
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()
layer_names = [layer_names[i - 1] for i in net.getUnconnectedOutLayers()]
return layer_names
def draw_bbox(img, bbox, labels, confidence, Drowning, write_conf=False):
global COLORS
global classes
if classes is None:
classes = populate_class_labels()
for i, label in enumerate(labels):
#if the person is drowning, the box will be drawn red instead of blue
if label == 'person' and Drowning:
color = COLORS[0]
label = 'DROWNING'
else:
color = COLORS[1]
if write_conf:
label += ' ' + str(format(confidence[i] * 100, '.2f')) + '%'
#you only need to points (the opposite corners) to draw a rectangle. These points
#are stored in the variable bbox
cv2.rectangle(img, (bbox[i][0],bbox[i][1]), (bbox[i][2],bbox[i][3]), color, 2)
cv2.putText(img, label, (bbox[i][0],bbox[i][1]-10),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
return img
def detect_common_objects(image, confidence=0.5, nms_thresh=0.3):
Height, Width = image.shape[:2]
scale = 0.00392
```

```
global classes
global dest_dir
#all the weights and the neural network algorithm are already preconfigured
#as we are using YOLO
#this part of the script just downloads the YOLO files
config_file_name = 'yolov4.cfg'
config_file_abs_path = dest_dir + os.path.sep + config_file_name
weights_file_name = 'yolov4.weights'
weights_file_abs_path = dest_dir + os.path.sep + weights_file_name
url = 'https://github.com/Nico31415/Drowning-Detector/raw/master/yolov3.cfg'
if not os.path.exists(config_file_abs_path):
download_file(url=url, file_name=config_file_name, dest_dir=dest_dir)
url = 'https://pjreddie.com/media/files/yolov3.weights'
if not os.path.exists(weights_file_abs_path):
download_file(url=url, file_name=weights_file_name, dest_dir=dest_dir)
global initialize
global net
if initialize:
classes = populate_class_labels()
net = cv2.dnn.readNet(weights_file_abs_path, config_file_abs_path)
initialize = False
blob = cv2.dnn.blobFromImage(image, scale, (416,416), (0,0,0), True, crop=False)
net.setInput(blob)
outs = net.forward(get_output_layers(net))
class_ids = []
confidences = []
boxes = []
```

```
for out in outs:
for detection in out:
scores = detection[5:]
class_id = np.argmax(scores)
max_conf = scores[class_id]
if max_conf> confidence:
center_x = int(detection[0] * Width)
center_y = int(detection[1] * Height)
w = int(detection[2] * Width)
h = int(detection[3] * Height)
x = center_x - w / 2
y = center_y - h / 2
class_ids.append(class_id)
confidences.append(float(max_conf))
boxes.append([x, y, w, h])
indices = cv2.dnn.NMSBoxes(boxes, confidences, confidence, nms_thresh)
bbox = []
label = []
conf = []
for i in indices:
i = i
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
```

utils

```
import requests
import progressbar as pb
import os
def download_file(url, file_name, dest_dir):
if not os.path.exists(dest_dir):
os.makedirs(dest_dir)
full_path_to_file = dest_dir + os.path.sep + file_name
if os.path.exists(dest_dir + os.path.sep + file_name):
return full_path_to_file
print("Downloading " + file_name + " from " + url)
try:
r = requests.get(url, allow_redirects=True, stream=True)
except:
print("Could not establish connection. Download failed")
return None
file_size = int(r.headers['Content-Length'])
chunk_size = 1024
num_bars = round(file_size / chunk_size)
bar = pb.ProgressBar(maxval=num_bars).start()
if r.status_code != requests.codes.ok:
print("Error occurred while downloading file")
return None
```

```
count = 0
with open(full_path_to_file, 'wb') as file:
for chunk in  r.iter_content(chunk_size=chunk_size):
file.write(chunk)
bar.update(count)
count +=1
return full_path_to_file
```

GITHUB LINK:https://github.com/IBM-EPBL/IBM-Project-35543-1660285895

PROJECT DEMO LINK:https://youtu.be/uBycudpAmqQ