VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning

Team Members:

- Alzeemar Khan K
- Dilliendra N
- Mano M
- Yuvarajan M

Team ID: PNT2022TMID24428

PROJECT REPORT CONTENT

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:

The system is not designed to replace a lifeguard or other human monitor, but to act as an additional tool. “It helps the lifeguard to detect the underwater situation where they can’t easily observe. Life Guard is a drowning detection system that detects every dangerous situation and accident. The software works in close integration with the cameras installed in the pool to continuously scan the pool. Thanks to this combination of hardware, software and profound innovations, today Life Guard represents excellence in drowning detection.

2.LITERATURE SURVEY

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

2.1 Existing problem:

Although computer-aided diagnosis (CAD) is used to improve the quality of diagnosis in various medical fields such as mammography and colonography, it is not used in dermatology, where non-invasive screening tests are performed only with the naked eye, and avoidable inaccuracies may exist.
This study shows that CAD may also be a viable option in dermatology by presenting a novel method to sequentially combine accurate segmentation and classification models. Given an image of the skin, we decompose the image to normalize and extract highlevel features.
Using a neural network-based segmentation model to create a segmented map of the image, we then cluster sections of abnormal skin and pass this information to a classification model.
We classify each cluster into different common skin diseases using another neural network model. Our segmentation model achieves better performance compared to previous studies, and also achieves a near-perfect sensitivity score in unfavourable conditions.
Our classification model is more accurate than a baseline model trained without segmentation, while also being able to classify multiple diseases within a single image. This improved performance may be sufficient to use CAD in the field of dermatology

2.2 References:

[1] AngelEye. (2019). AngelEye – Distributors. Retrieved from: https://www.angeleye.it/news. php?id=28&newscat=10

[2] Aquatics International. (2007). Traumatic Experiences – Should we make our youngest lifeguards come face to face with death? Retrieved from: https://www.aquaticsintl.com/facilities/traumaticexperiences_o

[3] British Standards Institution. (2018). BS EN 15288-1, Swimming pools for public use. Safety requirements for design. Retrieved from: https://shop.bsigroup.com/ProductDetail/?pid=000000000030360254

[4] British Standards Institution 1. (2018). BS EN 15288-2, Swimming pools for public use. Safety requirements for operation. Retrieved from: https://shop.bsigroup.com/ProductDetail/?p id=000000000030360257

[5] Drowning Prevention. (2017). The Need. Retrieved from: https://www.drowningprevention.com.au/

[6] German Institute for Standardization. (2019). German national guideline DGfdB R 94.15 “Test methods for camera-based drowning detection systems under operational conditions” (German Association for Public Swimming Pools).

[7] Haizhou Li, Haizhou Li, Kar-Ann Toh and Liyuan Li. (2012). Advanced Topics in Biometrics, World Scientific Publishing Co. Pte. Ltd., ISBN-13 978-981-4287-84-5

2.3 Problem Statement Definition:

1. Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. Swimming pools are found larger in number in hotels.
2. Applying the CNN algorithm to the dataset.Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident.
3.To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life.

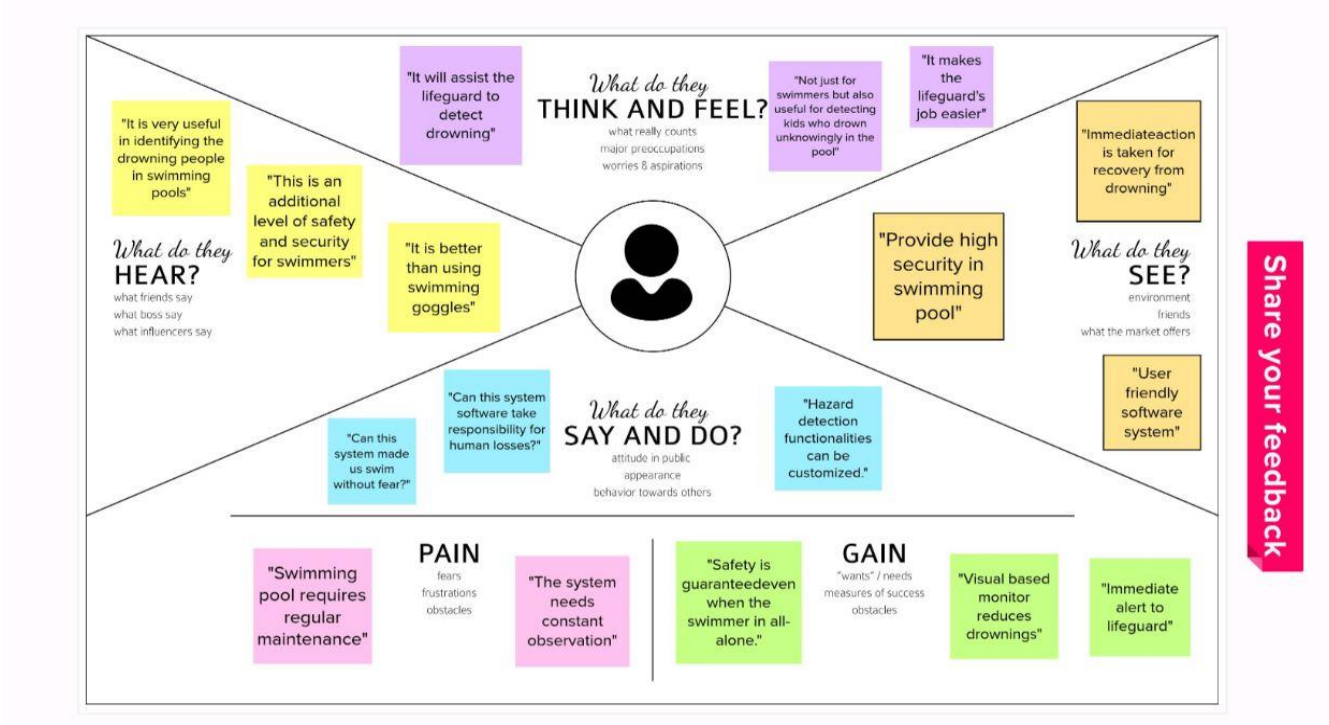
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Swimmer	Practice swimming	Suddenly drowned inside the water	I Lost the breathing control	Panic
PS-2	Lifeguard	Monitor the swimmers pulse rate	It is a difficult task to monitor	I can't able to monitor all the swimmers	Stress
PS-3	Trainer	Teach swimming	I can't able to pay attention to all the swimmers	I can't able to monitor all the learners at the same time.	Burden & stress

3 IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

An empathy map is a collaborative tool teams can use to gaina deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment.

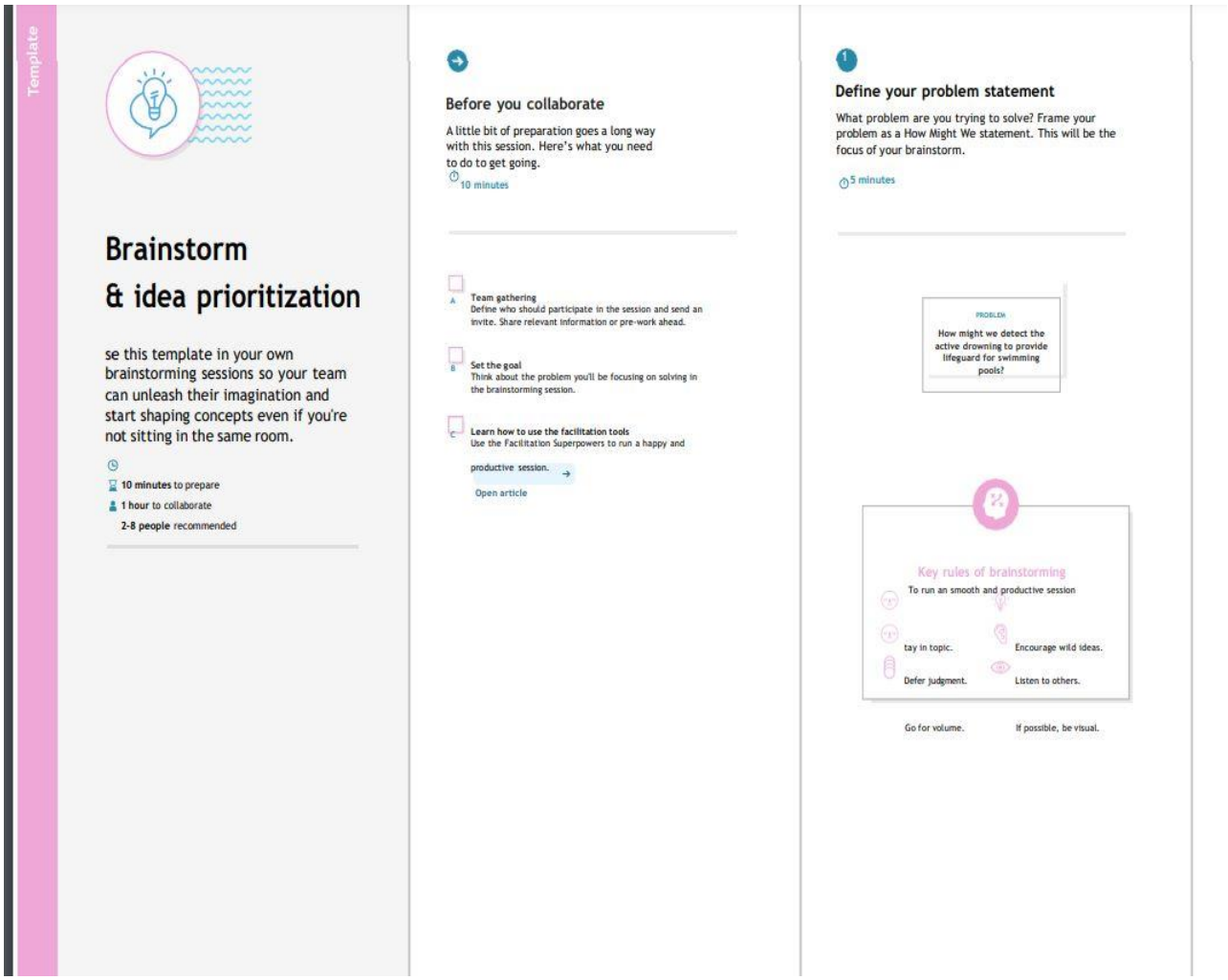
An Empathy Map consists of four quadrants. The four quadrants reflectfour key traits, which the user demonstrated/possessed during the observation/research stage. The four quadrants refer to what the user:



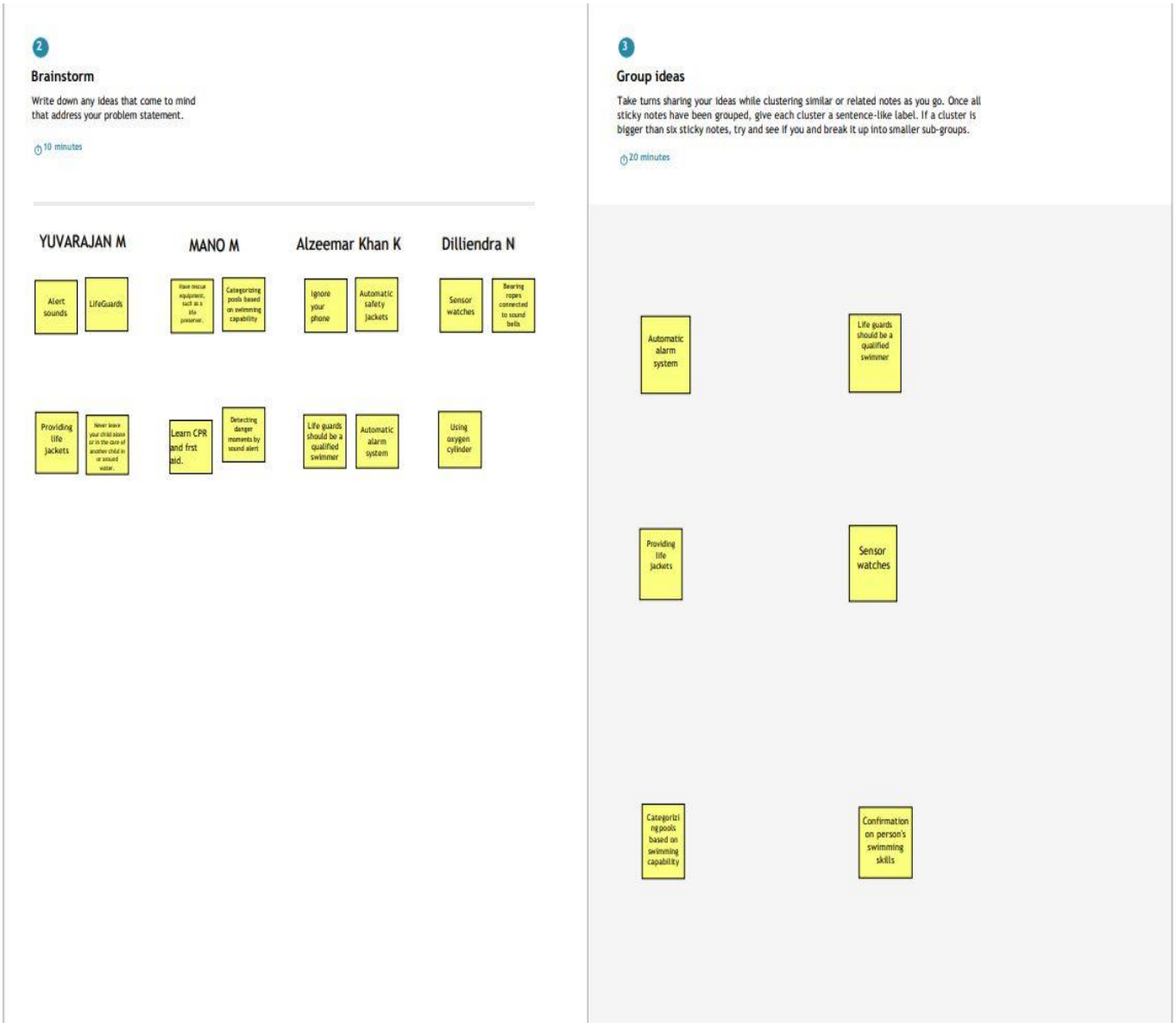
3.2 Ideation & Brainstorming:

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



Step 2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

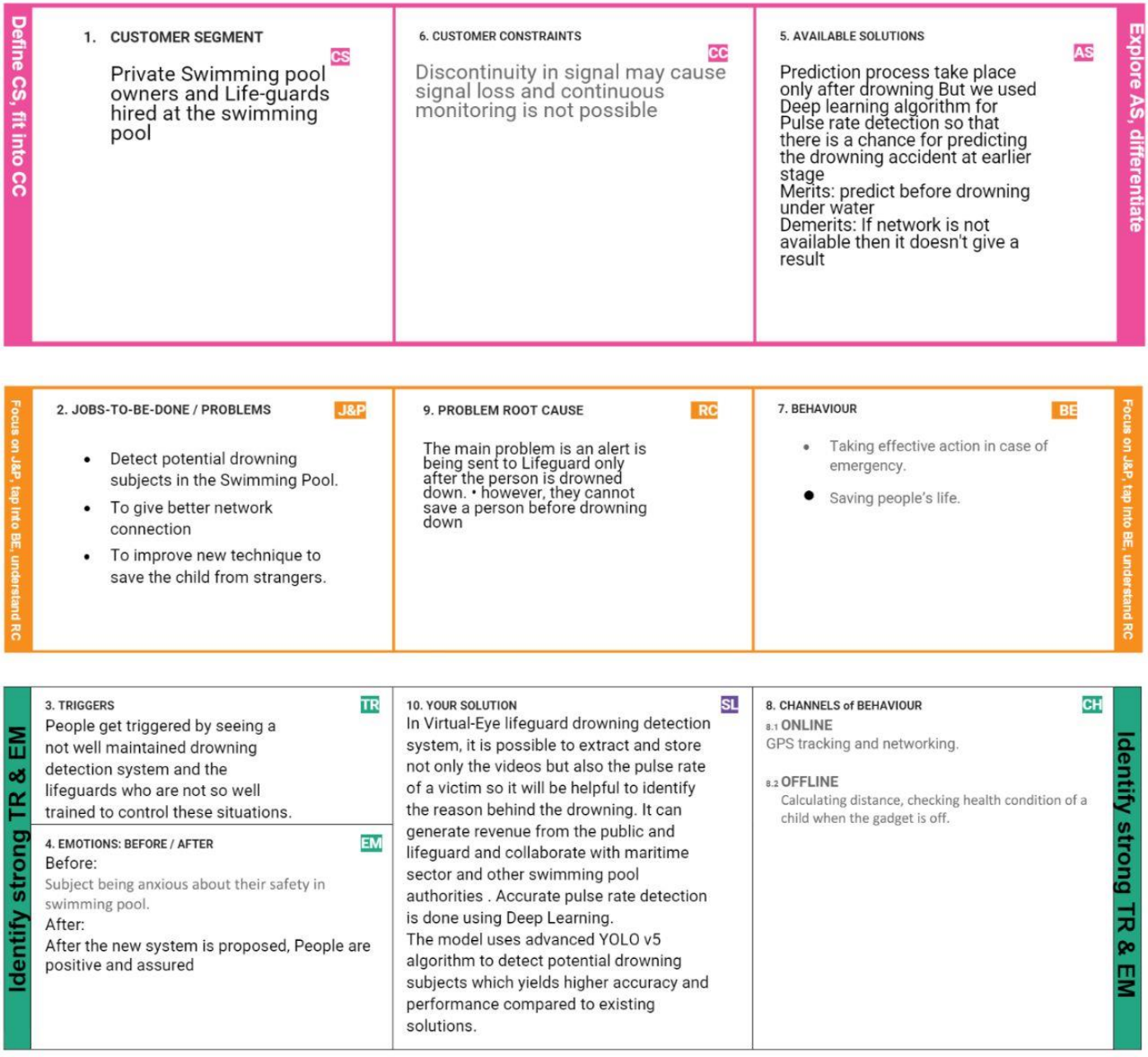
3.3 Proposed Solution:

Proposed Solution means the technical solution to be provided by the Implementation agency in response to the requirements and the objectivesof the Project.
The main goal of presenting a business proposal is to provide solution to a problem faced by a potential buyer. This section should be as comprehensive as possible, and able to address all the needs that you havepointed in the first section.

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Swimming pools are generally places of fun and healthy exercise, but they can be deadly as well. Even with a lifeguard observer on duty, swimmers may still have trouble in underwater or in parts of the pool beyond the lifeguard's field of view. Identifying the drowned person in the crowd will be difficult from above
2.	Idea / Solution description	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. 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 with the help of advanced technology.
4.	Social Impact / Customer Satisfaction	The device will be socialized in no time because of the help of this device. Drowning globally has a higher death rate and is also the third leading cause of unexpected deaths worldwide, especially among children under the age of six. To overcome this conflict our drowning detection system will have an impact on society.
5.	Business Model (Revenue Model)	The number of features makes it attractive for end users to use our software system. It is medium cost.
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:

Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for itactually solves the customer's problem.The Problem-Solution Fit is an important step towards the Product-Market Fit, but often an underestimated one. Problem-Solution canvas is a tool for entrepreneurs, marketers and corporate innovators, which helps them identify solutions with higher chances for solution adoption, reduce time spent on solution testing andget a better overview of current situation.



4 REQUIREMENT ANALYSIS

4.1 Functional requirement:

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in use cases.

Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Support	Take swim tubes or take the help of rescuer
FR-4	Alert	Send alert message to the lifeguard
FR-5	Output	Visual representation Image detection Report generation

4.2 Non-Functional requirements:

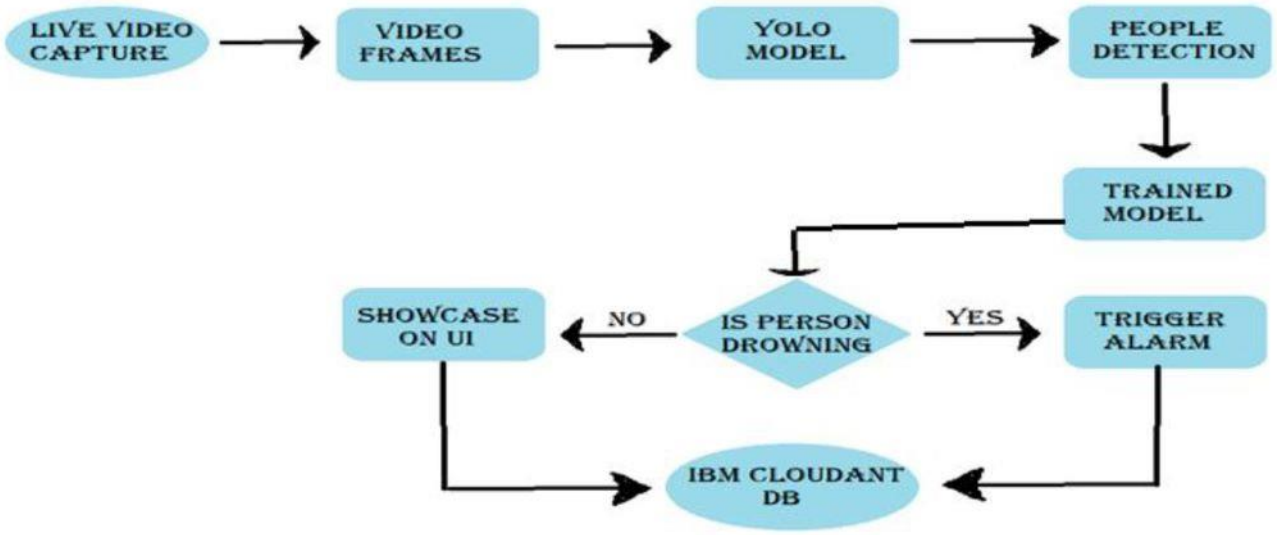
Non-functional requirements are often mistakenly called the "quality attributes" of a system, however there is a distinction between the two. Non-functional requirements are the criteria for evaluating how a softwaresystem should perform and a software system must have certain quality attributes in order to meet non-functional requirements.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Eco – Friendly. To ensure the safety of each and every person present in the pool.
NFR-2	**Security**	Observing each and every body movement of the swimmers. Lifeguards should be aware of the alert message to save the life of the swimmer
NFR-3	**Reliability**	Virtual eye lifeguard triggers an immediate prior alarm if a swimmer is in peril, helping to avoid panic even in critical situations.
NFR-4	**Performance**	Life guard can visually access the developing situation within seconds of the event first occurring and initiate the rescue procedure when necessary.
NFR-5	**Availability**	24/7 monitoring cameras.
NFR-6	**Scalability**	Virtual eye lifeguard detects potential drowning and promptly notifies you. Its comfortable for all swimmers.

5 PROJECT DESIGN

5.1 Data Flow Diagrams:

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputsand inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops.



5.2 Solution & Technical Architecture:

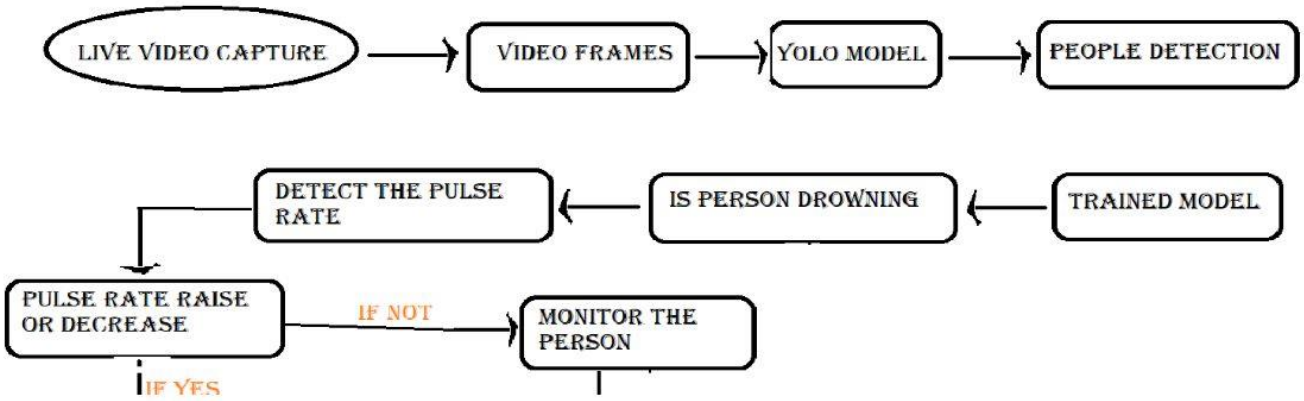
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.Its goals are to:

5.2.1 Find the best tech solution to solve existing business problems.

5.2.2 Describe the structure, characteristics, behavior, and other aspects ofthe software project stakeholders.

5.2.3 Define features, development phases, and solution requirements.

5.2.4 Provide specifications according to which the solution is defined,managed, and delivered.



Technical Architecture (TA) is a form of IT architecture that is used to design computer systems. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

5.3 User Stories:

A user story is an informal, general explanation of a software featurewritten from the perspective of the end user or customer. The purpose of auser story is to articulate how a piece of work will deliver a particular valueback to the customer.

In software development and product management, a user story is an informal, natural language description of features of a software system.

Sprint	Functional Requirement (Epic)	User Story Numb er	User Story / Task	Story Points	Priority	Team member
Sprint-1	Registration	VLGFSP1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Mano.M
Sprint-1	Registration	VLGFSP2	As a user, I will receive confirmation email oncel have registered for the application	1	High	Alzeemar Khan.K
Sprint-1	Registration		3 As a user, I can register for the application through Facebook	2	Low	Diliendra.N
Sprint-1	Registration	p	4 As a user, I can register for plication through Gmail	2	Medium	Mano.M
Sprint-1	Login	VLGFSP -6	As a user, I can log into the application by entering email & password	1	High	Yuvarajan.M
Sprint-2	Dataset Collect	VLGF SP - 11	Collect number of datasets and get accuracy	2	Medium	Alzeemar Khan.K
Sprint-2	Pre-processing	VLGF SP - 12	The dataset is extracted	2	High	Dilliendra.N
Sprint-2	Train the model	VLGF SP - 13	Train the model.	4	High	Yuvarajan.M
Sprint-2	Test the model	VLGFS P - 14	Test the model	6	High	Mano.M
Sprint-3	Detection	VLGFS P - 15	Load the trained model.	3	High	Alzeemar Khan.K

Sprint-3	Detection	VLGFS P - 16	Identify the person by collecting real-time data through a webcam.	5	Medium	Dilliendra.N
Sprint-3	Detection	VLGFS P - 16	classify it by using a trained model to predict the output	8	High	Yuvarajan.M
Sprint-4	Detection	VLGFS P - 17	If person is drowning, the system will ring an alarm to give signal	7	High	Yuvarajan.M
Sprint-4	Detection	VLGFS P - 18	As a User,I can detect the drowning person.	3	Medium	Dilliendra.N
Sprint-4	Logout	VLGFS P - 19	As a User,I can logout the application.	2	Low	Mano.M

6 PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

In Scrum Projects, Estimation is done by the entire team during Sprint Planning Meeting. The objective of the Estimation would be to consider theUser Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.

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

6.2 Sprint Delivery Schedule:

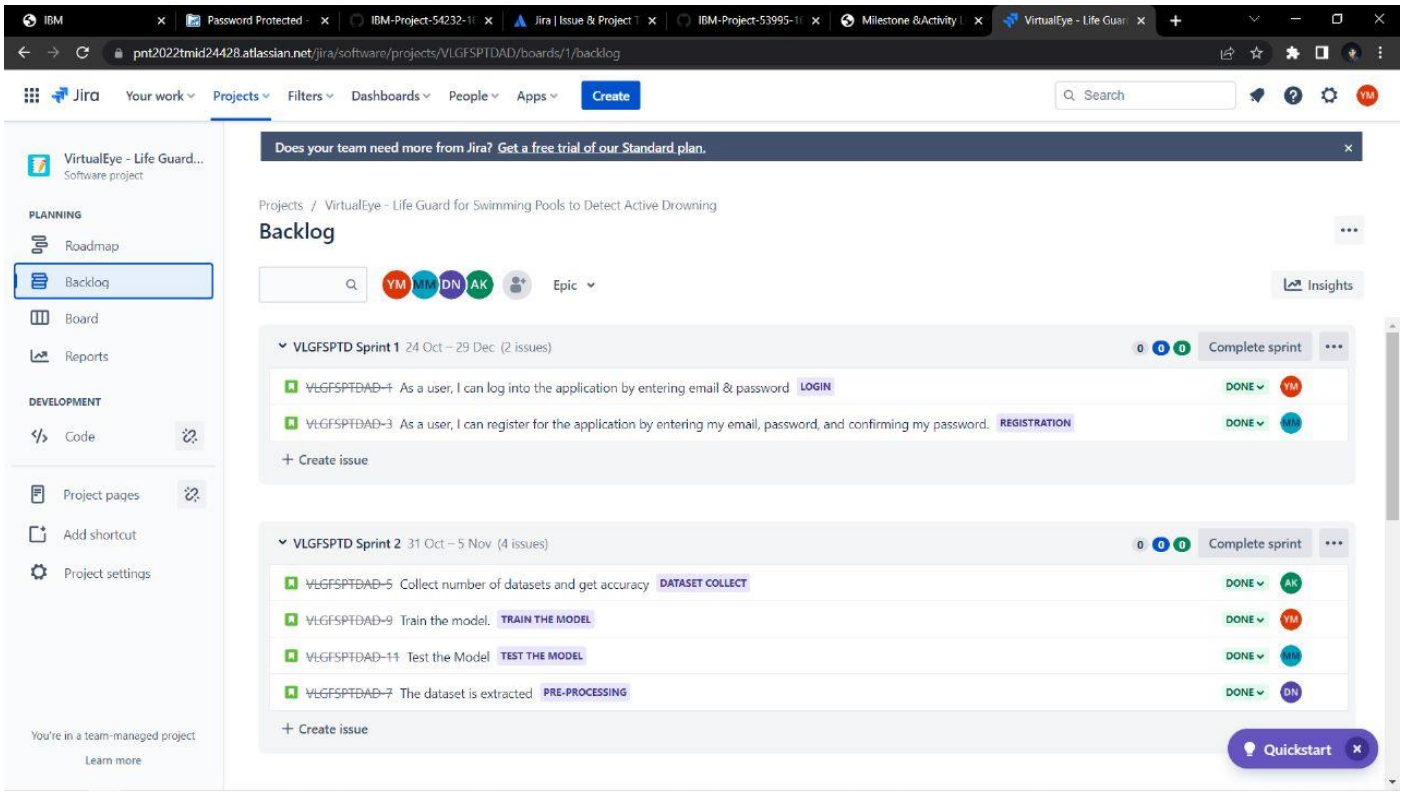
In Scrum project sprint delivery schedule is used to estimate when sprint has started and delivery date of the sprint. Due to estimation of the sprint delivery schedule it helps the developer to complete their project within theestimated time.

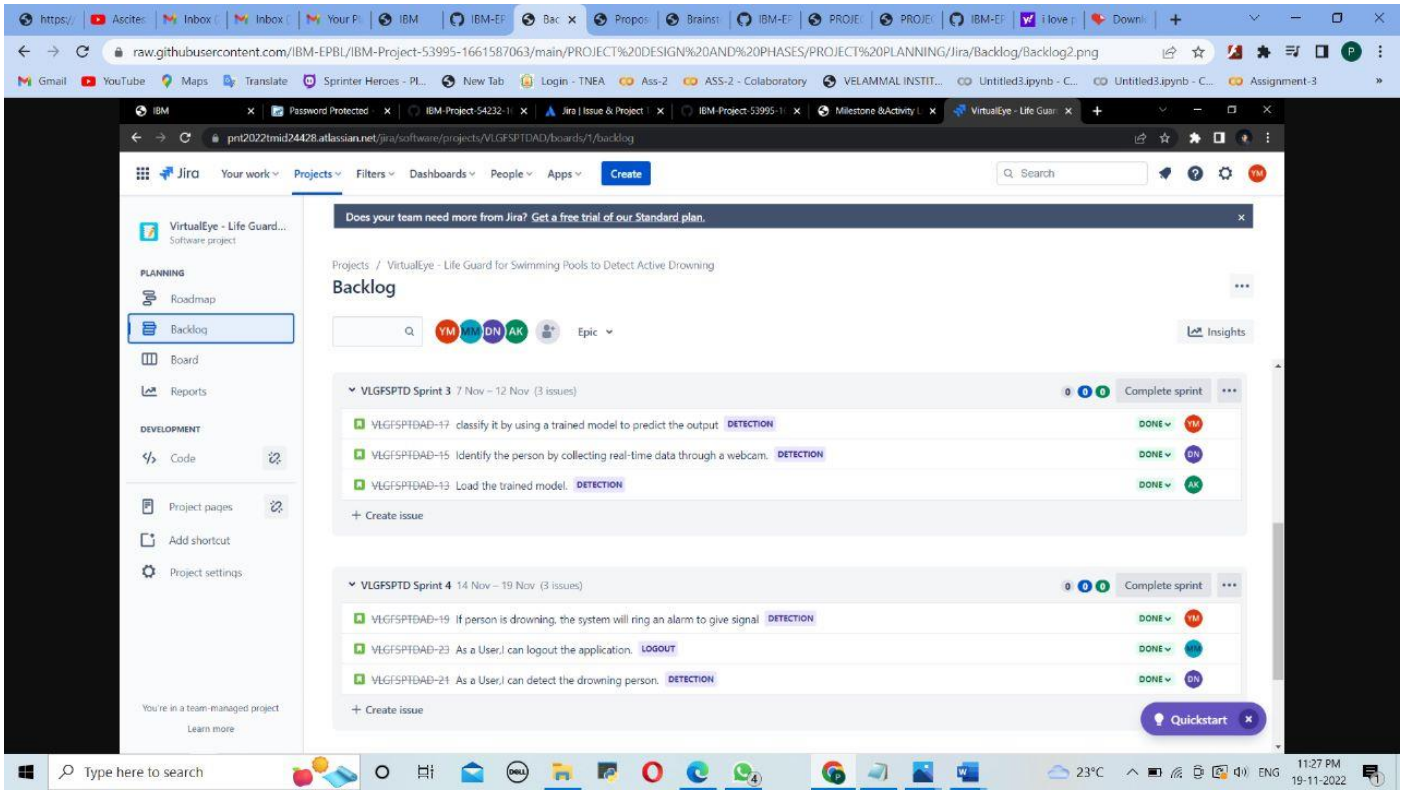
6.3 Reports from JIRA:

The reports in Jira has been denoted below:

BACKLOG:

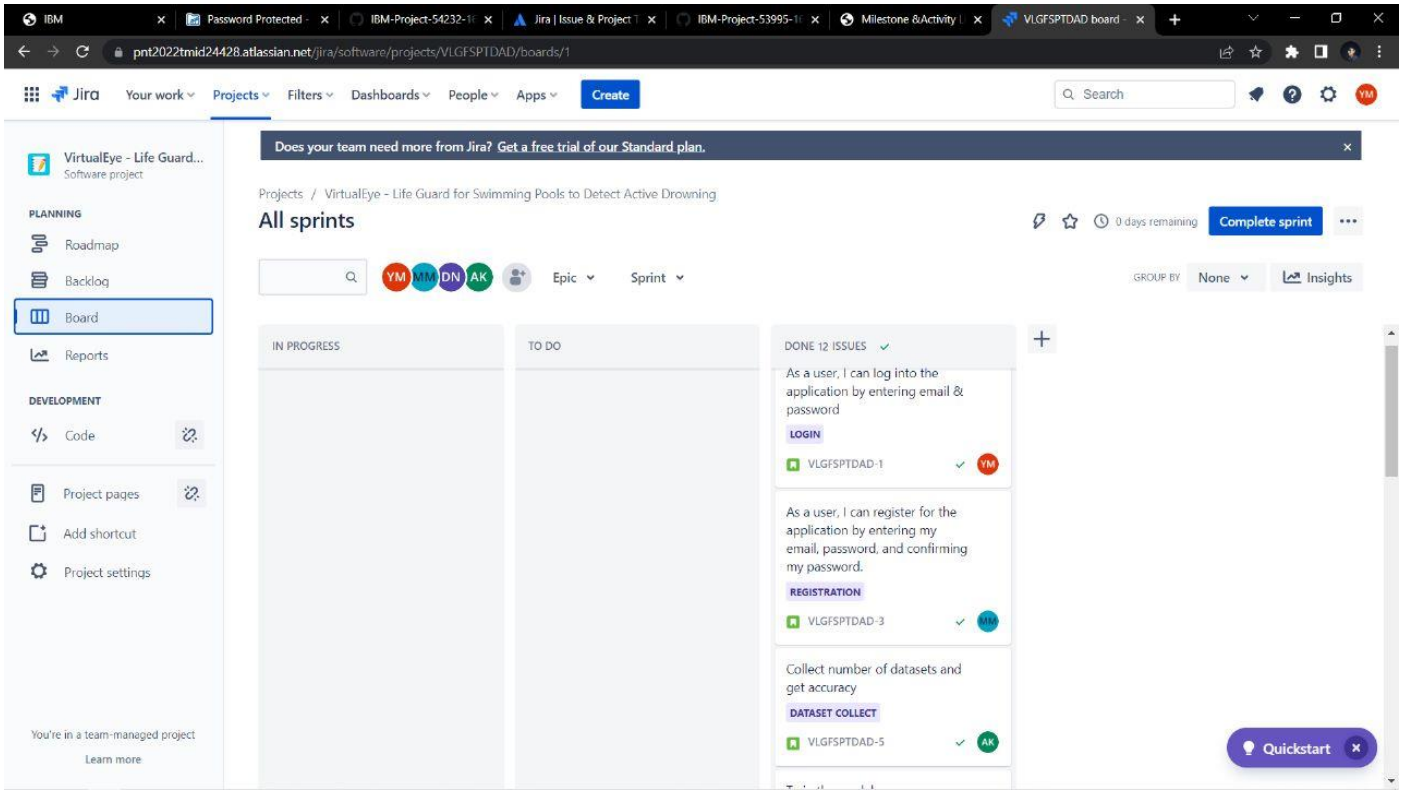
Backlog is usually a list of issues describing what your team is going to do on a project. It's a convenient place for creating, storing, and managing several kinds of issues: issues that you're currently working on (you can also see them on the board and in the current sprint if you're using a Scrumproject).

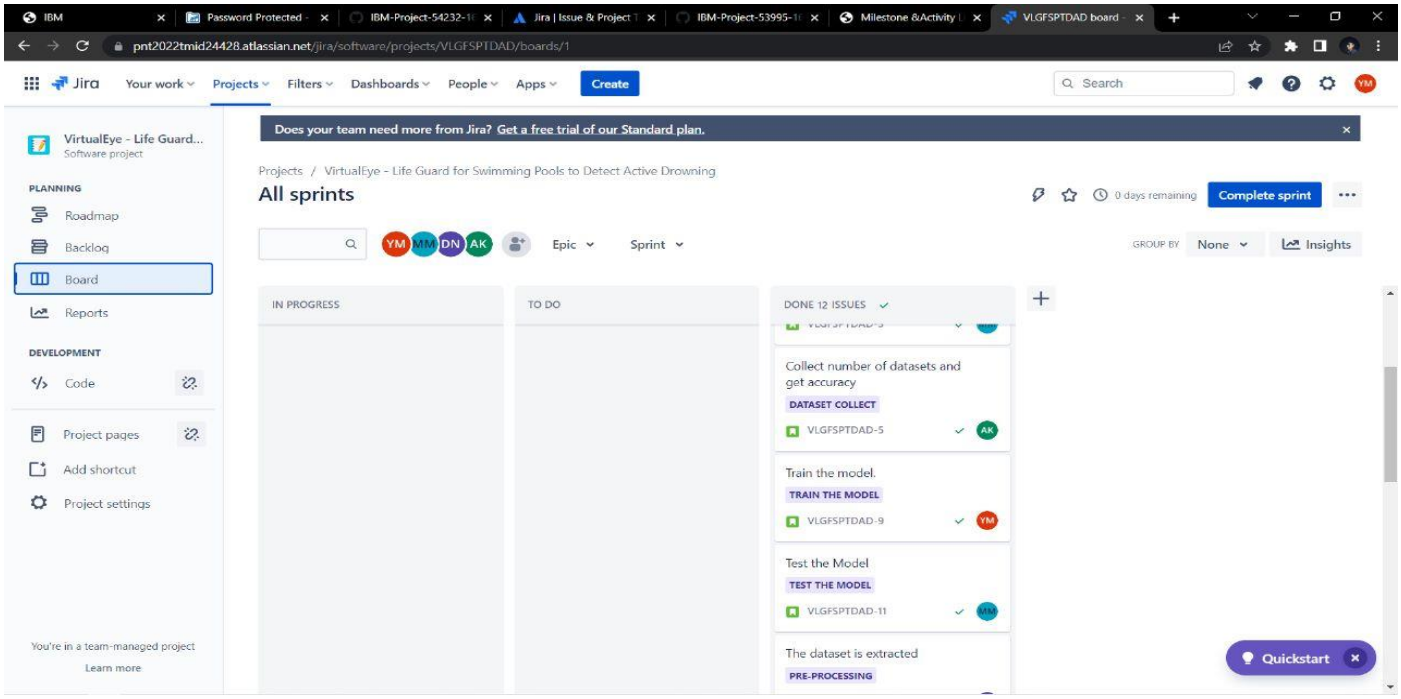

pnt2022tmid24428.atlassian.net/jira/software/projects/VLGFSPTDAD/boards/1/backlog
Jira
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
VirtualEye - Life Guard...
Software project
PLANNING
Roadmap
Backlog
Board
Reports
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
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
Epic
Insights
VLGFSPTD Sprint 1 24 Oct – 29 Dec (2 issues)
Complete sprint
VLGFSPTDAD-1 As a user, I can log into the application by entering email & password LOGIN
VLGFSPTDAD-3 As a user, I can register for the application by entering my email, password, and confirming my password. REGISTRATION
DONE
+ Create issue
VLGFSPTD Sprint 2 31 Oct – 5 Nov (4 issues)
Complete sprint
VLGFSPTDAD-5 Collect number of datasets and get accuracy DATASET COLLECT
VLGFSPTDAD-9 Train the model. TRAIN THE MODEL
VLGFSPTDAD-11 Test the Model TEST THE MODEL
VLGFSPTDAD-7 The dataset is extracted PRE-PROCESSING
+ Create issue
Quickstart

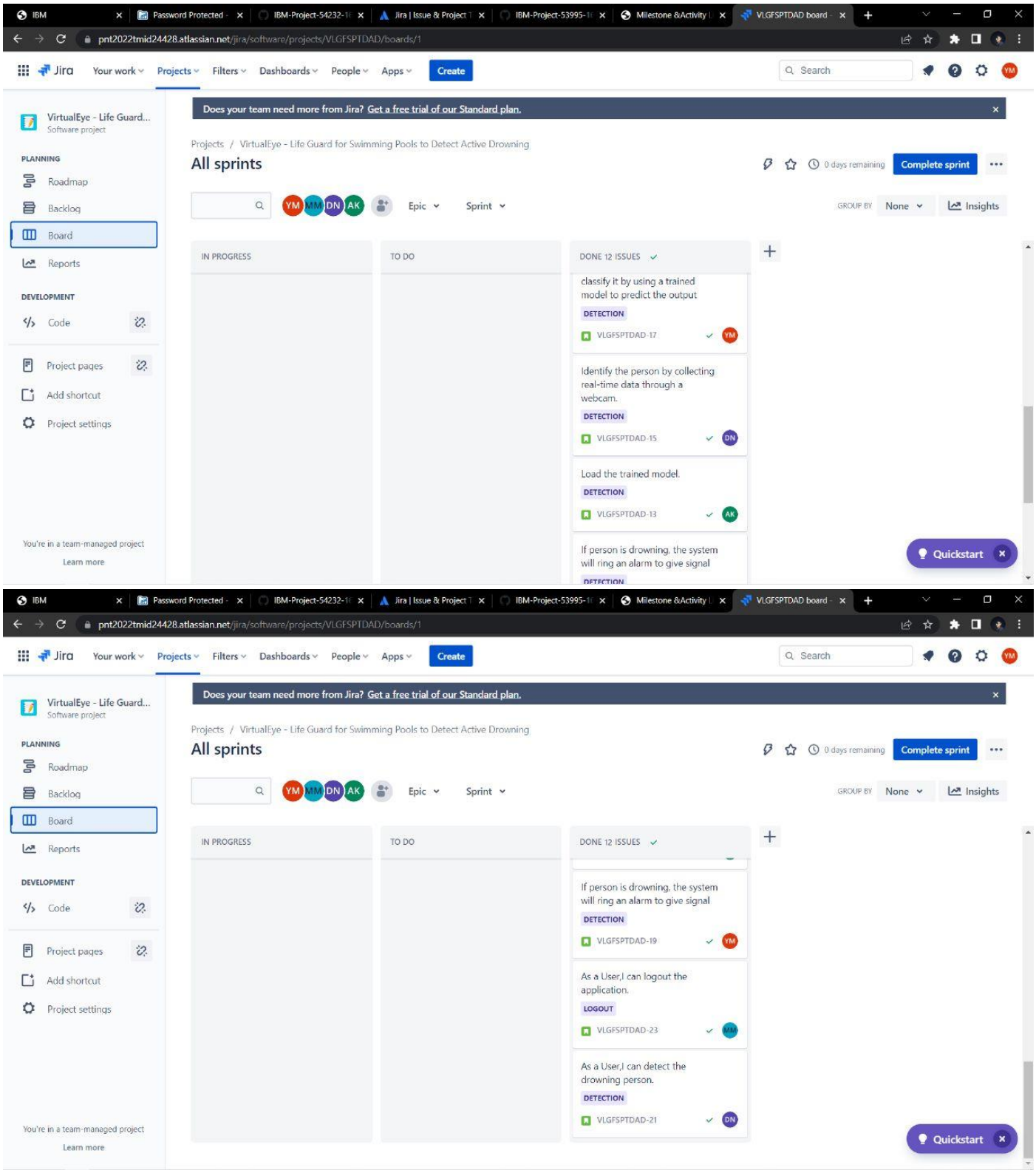

raw.githubusercontent.com/IBM-EPBL/IBM-Project-53995-1661587063/main/PROJECT%20DESIGN%20AND%20PHASES/PROJECT%20PLANNING/Jira/Backlog/Backlog2.png
pnt2022tmid24428.atlassian.net/jira/software/projects/VLGFSPTDAD/boards/1/backlog
Jira
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
VirtualEye - Life Guard...
Software project
PLANNING
Roadmap
Backlog
Board
Reports
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
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
Epic
Insights
VLGFSPTD Sprint 3 7 Nov – 12 Nov (3 issues)
Complete sprint
VLGFSPTDAD-17 classify it by using a trained model to predict the output DETECTION
VLGFSPTDAD-15 Identify the person by collecting real-time data through a webcam. DETECTION
VLGFSPTDAD-13 Load the trained model. DETECTION
DONE
+ Create issue
VLGFSPTD Sprint 4 14 Nov – 19 Nov (3 issues)
Complete sprint
VLGFSPTDAD-19 If person is drowning, the system will ring an alarm to give signal DETECTION
VLGFSPTDAD-23 As a User,I can logout the application. LOGOUT
VLGFSPTDAD-21 As a User,I can detect the drowning person. DETECTION
+ Create issue
Quickstart
Type here to search
23°C
ENG
11:27 PM
19-11-2022

BOARD:

A board displays your team's work as cards you can move between columns. In Jira Software, cards and the tasks they represent are called“issues”. Usually, your board reflects your team's process, tracking the status of work as it makes its way through your team's process.

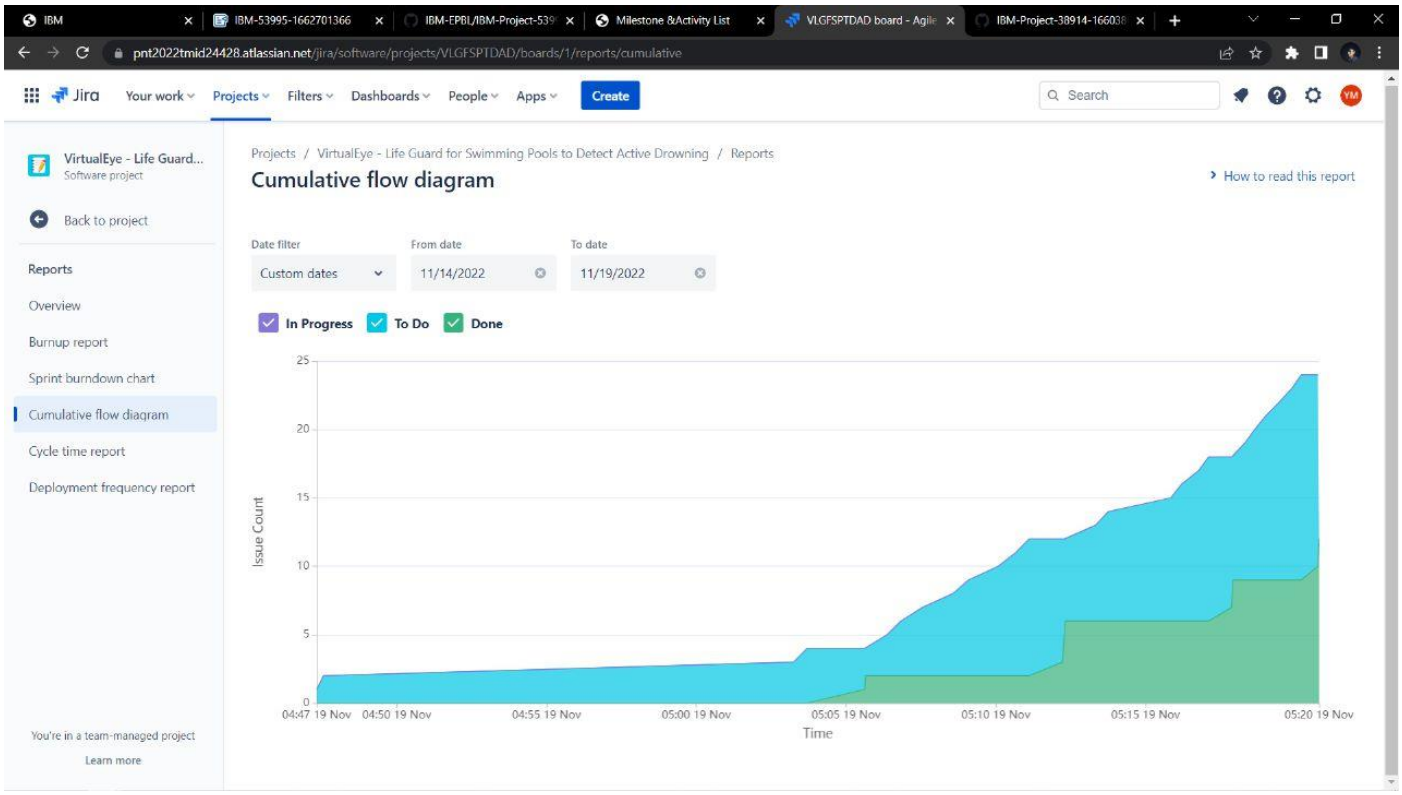





IBM
Password Protected
IBM-Project-54232-1
Jira | Issue & Project
IBM-Project-53995-1
Milestone &Activity
VLGFSPTDAD board
pnt2022tmid24428.atlassian.net/jira/software/projects/VLGFSPTDAD/boards/1
Jira
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
VirtualEye - Life Guard...
Software project
PLANNING
Roadmap
Backlog
Board
Reports
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
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
All sprints
0 days remaining
Complete sprint
Epic
Sprint
GROUP BY
None
Insights
IN PROGRESS
TO DO
DONE 12 ISSUES
classify it by using a trained model to predict the output
DETECTION
VLGFSPTDAD-17
Identify the person by collecting real-time data through a webcam.
DETECTION
VLGFSPTDAD-15
Load the trained model.
DETECTION
VLGFSPTDAD-13
If person is drowning, the system will ring an alarm to give signal
DETECTION
Quickstart
IBM
Password Protected
IBM-Project-54232-1
Jira | Issue & Project
IBM-Project-53995-1
Milestone &Activity
VLGFSPTDAD board
pnt2022tmid24428.atlassian.net/jira/software/projects/VLGFSPTDAD/boards/1
Jira
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
VirtualEye - Life Guard...
Software project
PLANNING
Roadmap
Backlog
Board
Reports
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
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
All sprints
0 days remaining
Complete sprint
Epic
Sprint
GROUP BY
None
Insights
IN PROGRESS
TO DO
DONE 12 ISSUES
If person is drowning, the system will ring an alarm to give signal
DETECTION
VLGFSPTDAD-19
As a User,I can logout the application.
LOGOUT
VLGFSPTDAD-23
As a User,I can detect the drowning person.
DETECTION
VLGFSPTDAD-21
Quickstart

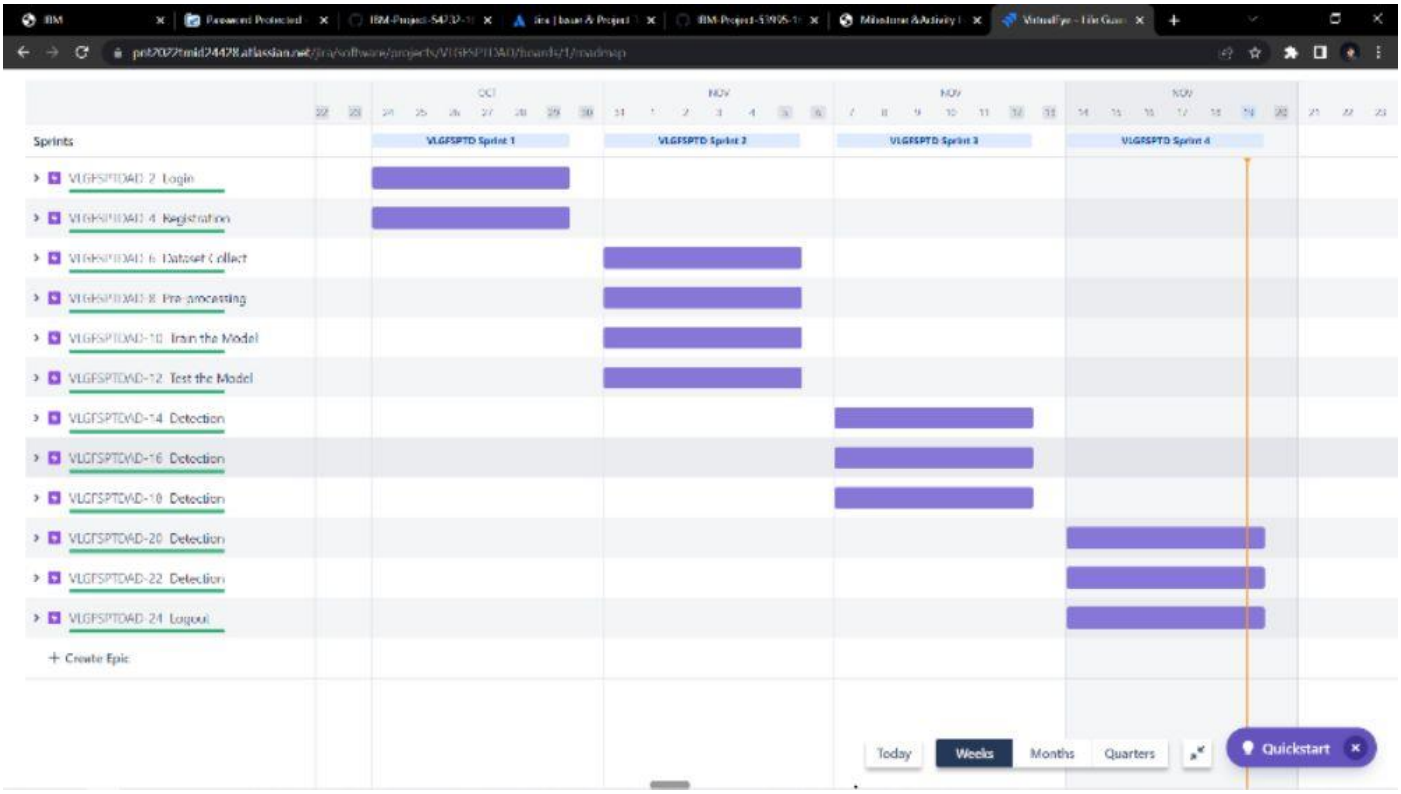
COMMULATIVE FLOW DIAGRAM:

A Cumulative Flow Diagram (CFD) is an area chart that shows the various statuses of work items for an application, version, or sprint. The horizontalx-axis in a CFD indicates time, and the vertical y-axis indicates cards (issues).



ROAD MAP:

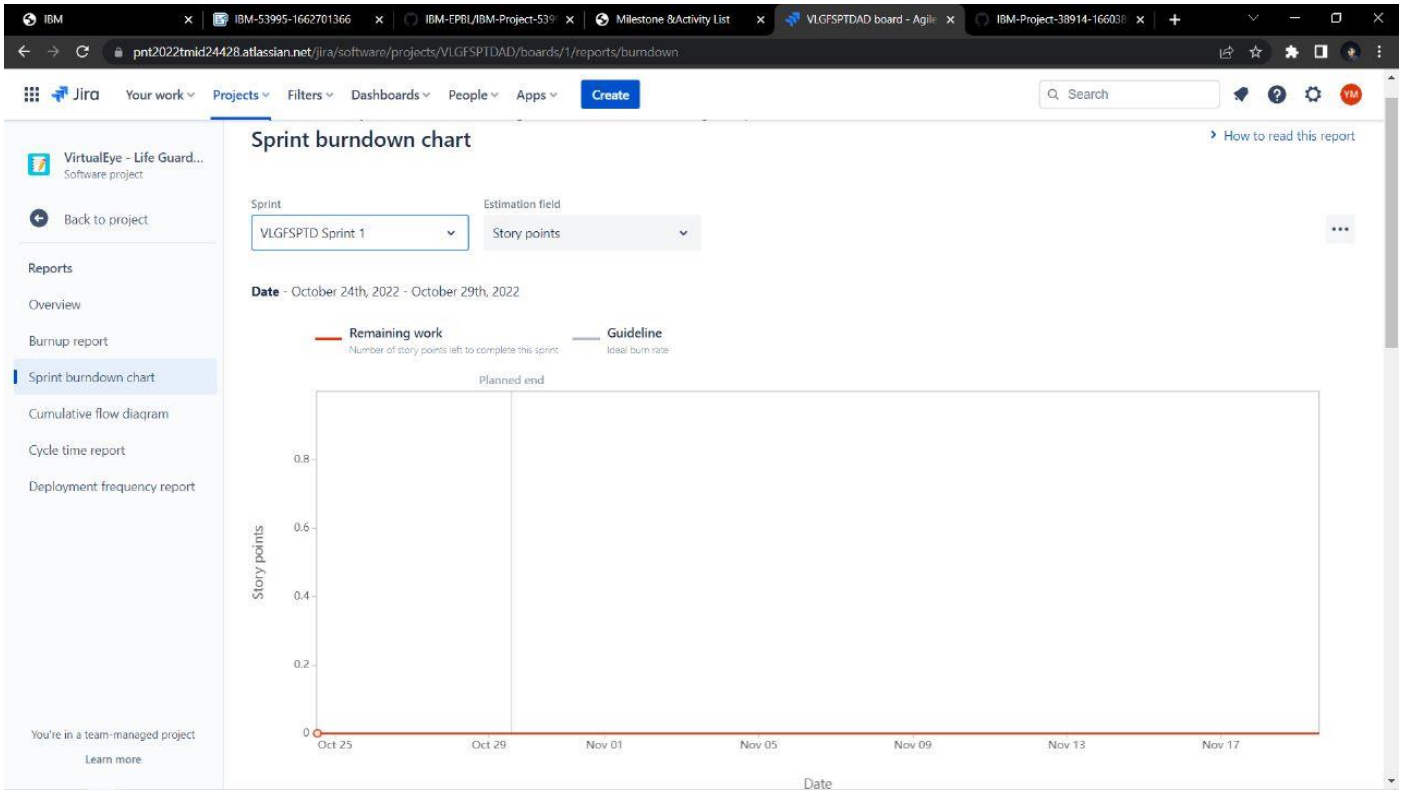
Roadmaps in Jira Software are team-level roadmaps useful for planning large pieces of work several months in advance at the Epic level within a single project. Simple planning and dependency management features helpyour teams visualize and manage work better together.



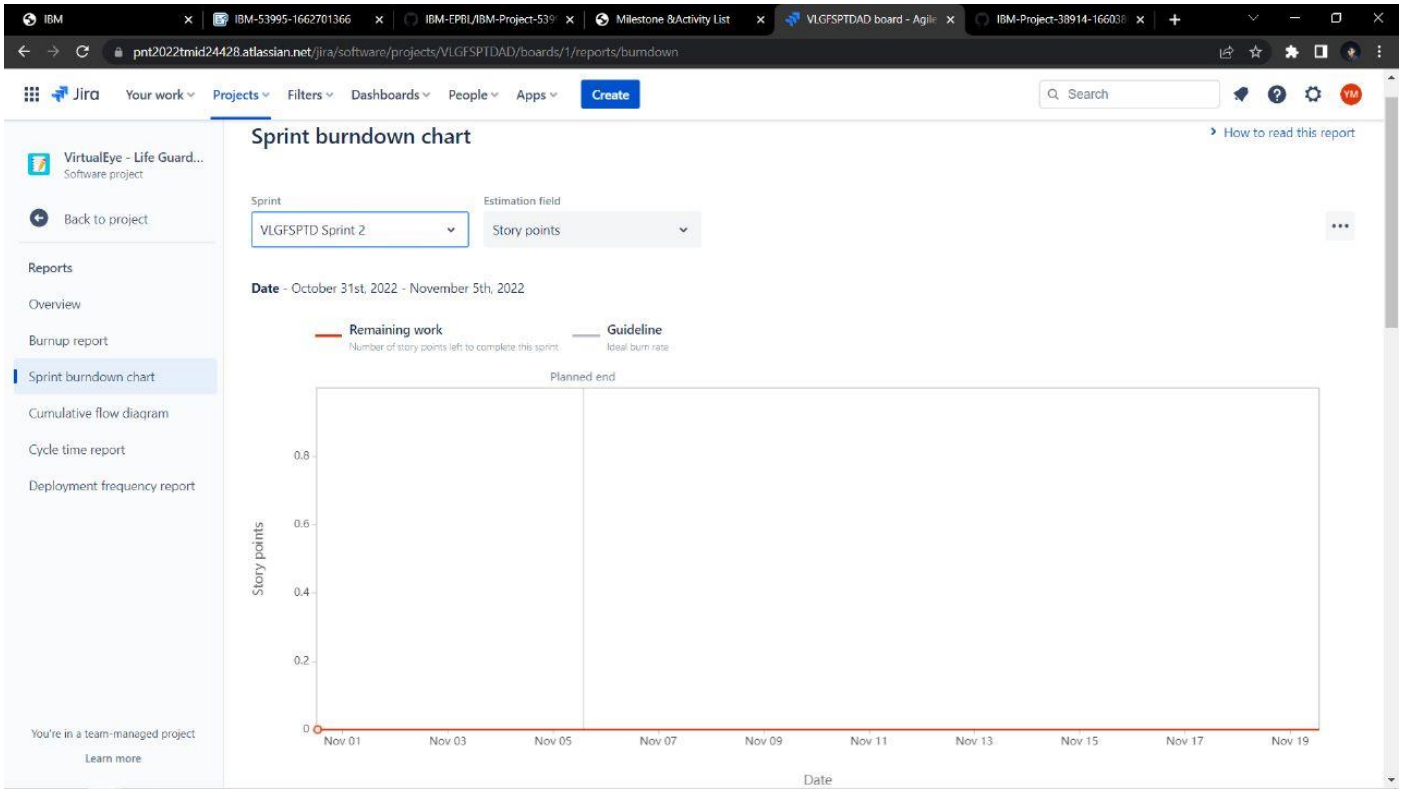
SPRINT BURNDOWN CHART:

A burndown chart shows the amount of work that has been completed in an epic or sprint, and the total work remaining. Burndown charts are usedto predict your team's likelihood of completing their work in the time available.

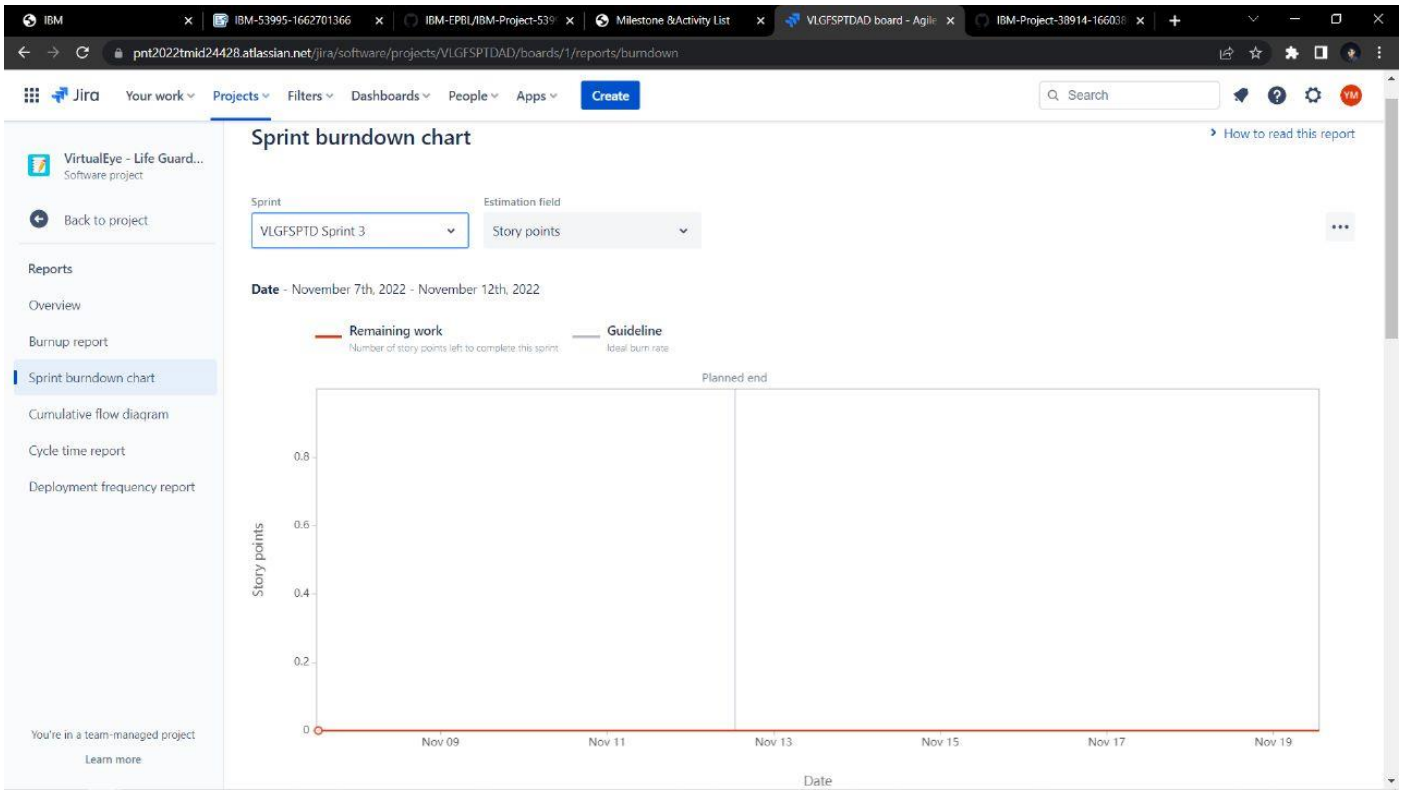
SPRINT1:



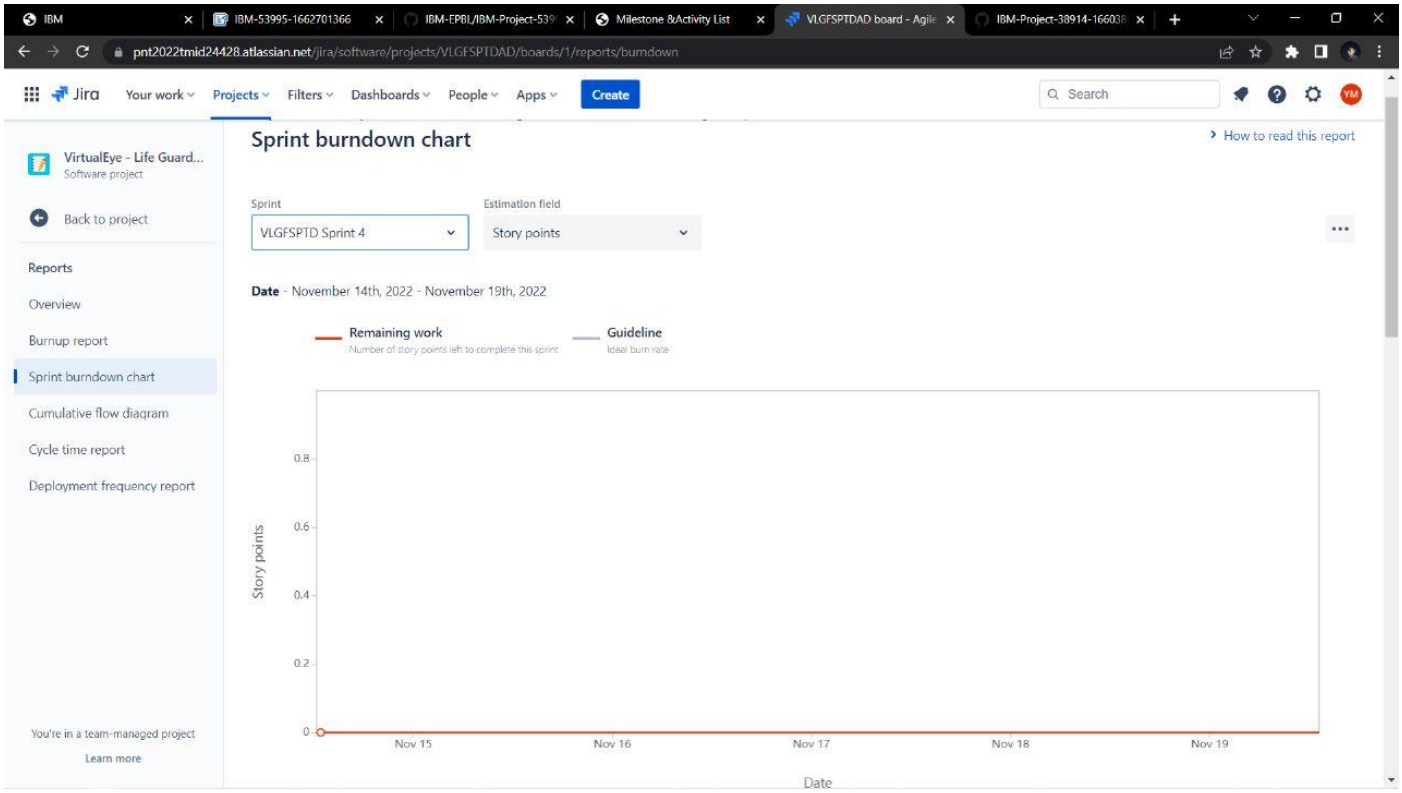
SPRINT2:



SPRINT3:


IBM
pnt2022tmid24428.atlassian.net/jira/software/projects/VLGFSPTDAD/boards/1/reports/burndown
Jira
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
VirtualEye - Life Guard...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Sprint burndown chart
How to read this report
Sprint
VLGFSPTD Sprint 3
Estimation field
Story points
Date - November 7th, 2022 - November 12th, 2022
Remaining work
Number of story points left to complete this sprint
Guideline
Ideal burn rate
Planned end
Story points
0.8
0.6
0.4
0.2
0
Nov 09
Nov 11
Nov 13
Nov 15
Nov 17
Nov 19
Date

SPRINT4



7.CODING AND SOLUTIONING

FEATURE 1:

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

learning_rate=0.01
burn_in=1000 max_batches =
500200policy=steps
steps=400000,450000
scales=.1,.1
[convolutional]
batch_normalize=1
filters=32 size=3
stride=1
pad=1
activation=leaky
# Downsample
[convolutional]
batch_normalize=1
filters=64 size=3
stride=2
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=32 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=64 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear#
Downsample
[convolutional]
batch_normalize=1
filters=128 size=3
stride=2
pad=1

activation=leaky
[convolutional]
batch_normalize=1
filters=64 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=64 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
# Downsample
[convolutional]
batch_normalize=1
filters=256size=3
stride=2 pad=1
activation=leaky
[convolutional]
batch_normalize=1

filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1

```
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
```

activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3

```
activation=linear#
Downsample
[convolutional]
batch_normalize=1
filters=512 size=3
stride=2
pad=1 activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
```

filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1 activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky

```
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
```

```
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear#
Downsample
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=2
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
```

filters=1024 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
######################
[convolutional]

batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=1024
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=1024
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=1024
activation=leaky
[convolutional]size=1
stride=1
pad=1 filters=255
activation=linear
[yolo]
mask = 6,7,8
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,

156,198, 373,326
classes=80
num=9 jitter=.3
ignore_thresh = .7
truth_thresh = 1random=1
[route] layers = -4
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[upsample]
stride=2
[route]
layers = -1, 61
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=512
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=512
activation=leaky
[convolutional]

batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=512
activation=leaky
[convolutional]size=1
stride=1
pad=1 filters=255
activation=linear
[yolo]
mask = 3,4,5
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,
156,198, 373,326
classes=80
num=9 jitter=.3
ignore_thresh = .7
truth_thresh = 1random=1
[route] layers = -4
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[upsample]
stride=2
[route]
layers = -1, 36
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky

```
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=256
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1
pad=1 filters=256
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=256
activation=leaky
[convolutional]size=1
stride=1
pad=1 filters=255
activation=linear
[yolo]
mask = 0,1,2
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,
156,198, 373,326
classes=80
num=9 jitter=.3
ignore_thresh = .7
truth_thresh = 1
```

```
random=1
7.2 FEATURE 2
#import necessary packagesimport
cv2
import os
import numpy as np
from .utils import download_file
initialize = Truenet
= None
dest_dir = os.path.expanduser('~') + os.path.sep + '.cvlib' + os.path.sep + 'object_detection' +os.path.sep +
'yolo' +
os.path.sep + 'yolov3'
classes = None
#colors are BGR instead of RGB in python
COLORS = [0,0,255], [255,0,0]
def populate_class_labels():
#we are using a pre existent classifier which is more reliable and more efficient than one#we could make
using only a laptop
#The classifier should be downloaded automatically when you run this scriptclass_file_name =
'yolov3_classes.txt'
class_file_abs_path = dest_dir + os.path.sep + class_file_name
url = 'https://github.com/Nico31415/Drowning-Detector/raw/master/yolov3.txt'if not
os.path.exists(class_file_abs_path):
download_file(url=url, file_name=class_file_name, dest_dir=dest_dir)f =
open(class_file_abs_path, 'r')
classes = [line.strip() for line in f.readlines()]
return classes
def get_output_layers(net)
#the number of output layers in a neural network is the number of possible#things the network
can detect, such as a person, a dog, a tie, a phone... layer_names = net.getLayerNames()
output_layers = [layer_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]
return output_layers
def draw_bbox(img, bbox, labels, confidence, Drowning, write_conf=False):
global COLORS
global classes
if classes is None:
classes = populate_class_labels()
for i, label in enumerate(labels):
#if the person is drowning, the box will be drawn red instead of blueif label ==
```

```
'person' and Drowning:
color = COLORS[0] label
= 'DROWNING'
else:
color = COLORS[1]
if write_conf:
label += ' ' + str(format(confidence[i] * 100, '.2f')) + '%'
#you only need to points (the opposite corners) to draw a rectangle. These points#are stored in the
variable bbox
cv2.rectangle(img, (bbox[i][0],bbox[i][1]), (bbox[i][2],bbox[i][3]), color, 2)
cv2.putText(img, label, (bbox[i][0],bbox[i][1]-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
return img
def detect_common_objects(image, confidence=0.5, nms_thresh=0.3):
Height, Width = image.shape[:2]scale =
0.00392
global classes
global dest_dir
#all the weights and the neural network algorithm are already preconfigured#as we are using
YOLO
#this part of the script just downloads the YOLO files
config_file_name = 'yolov3.cfg'
config_file_abs_path = dest_dir + os.path.sep + config_file_name
weights_file_name = 'yolov3.weights'
weights_file_abs_path = dest_dir + os.path.sep + weights_file_name
url = 'https://github.com/Nico31415/Drowning-Detector/raw/master/yolov3.cfg'
if not os.path.exists(config_file_abs_path):
download_file(url=url, file_name=config_file_name, dest_dir=dest_dir)
url = 'https://pjreddie.com/media/files/yolov3.weights'
if not os.path.exists(weights_file_abs_path):
download_file(url=url, file_name=weights_file_name, dest_dir=dest_dir)
global initialize
global net
if initialize:
classes = populate_class_labels()
net = cv2.dnn.readNet(weights_file_abs_path, config_file_abs_path)initialize = False
blob = cv2.dnn.blobFromImage(image, scale, (416,416), (0,0,0), True, crop=False)
net.setInput(blob)
outs = net.forward(get_output_layers(net))
class_ids = []
```

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

8 TESTING

8.1 Test Case:

- Verifies whether the user can login if he/she was an registered user.
- Verifies whether an unregistered user cannot proceed with the login.
- Verifies whether an unregistered user can successfully register as answer.
- Verifies whether an register user cannot register them self as an newuser.
- Verifies whether an alert message popup when an unregistered usertries to login .
- Verifies whether an alert message popup when an registered usertries to register again.
- Verifies whether an alert message popup when an registered userenters his/her username or password incorrect.
- Verifies whether an alert message popup when an new userregisters.

- Verifies whether all UI button(signup, login now ,logout , report ,story ,user dashboard) works efficiently.
- Verifies whether username popup on the welcome note.

Test Case No	Action	Expected Output	Actual Output	Result
1	Register for the website	Stores name, email, and password in Database	Stores name, email, and password in Database	Pass
2	Login to the website	Giving the right credentials, results in a successful login.	Giving the right credentials, results in a successful login	Pass
3	Detecting the drowning person	It should assist the lifeguard to detect drowning	It should predict the person	Pass

8.2 User Acceptance Testing:

The purpose of this is to briefly explain the test coverage and open issues of the retail store stock analytics project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis:

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	1	0	0	41
Security	42	0	0	42
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

```
[net]
# Testing#
batch=1
# subdivisions=1#
Training batch=64
subdivisions=16
width=608 height=608
channels=3
momentum=0.9
decay=0.0005 angle=0
saturation = 1.5
exposure = 1.5hue=.1
learning_rate=0.01
burn_in=1000 max_batches =
500200policy=steps
steps=400000,450000
scales=.1,.1
[convolutional]
batch_normalize=1
filters=32 size=3
stride=1
pad=1
activation=leaky
# Downsample
[convolutional]
batch_normalize=1
filters=64 size=3
stride=2
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=32 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=64 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear#
Downsample
```

```
[convolutional]
batch_normalize=1
filters=128 size=3
stride=2
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=64 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=64 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
# Downsample
[convolutional]
batch_normalize=1
filters=256size=3
stride=2 pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
```

stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear

```
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
```

```
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear#
Downsample
[convolutional]
batch_normalize=1
filters=512 size=3
stride=2
pad=1 activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
```

[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1 activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-

3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear#
Downsample
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=2
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=1
pad=1
activation=leaky

```
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
######################
[convolutional]
batch_normalize=1
```

filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=1024
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=1024
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=1024
activation=leaky
[convolutional]size=1
stride=1
pad=1 filters=255
activation=linear
[yolo]
mask = 6,7,8
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,
156,198, 373,326
classes=80
num=9 jitter=.3
ignore_thresh = .7
truth_thresh = 1random=1
[route] layers = -4
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[upsample]

```
stride=2
[route]
layers = -1, 61
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=512
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=512
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=512
activation=leaky
[convolutional]size=1
stride=1
pad=1 filters=255
activation=linear
[yolo]
mask = 3,4,5
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,
156,198, 373,326
classes=80
num=9 jitter=.3
ignore_thresh = .7
truth_thresh = 1random=1
[route] layers = -4
[convolutional]
batch_normalize=1
```

filters=128 size=1
stride=1
pad=1
activation=leaky
[upsample]
stride=2
[route]
layers = -1, 36
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=256
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1
pad=1 filters=256
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=256
activation=leaky
[convolutional]size=1
stride=1
pad=1 filters=255
activation=linear
[yolo]
mask = 0,1,2
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,
156,198, 373,326
classes=80
num=9 jitter=.3

```
ignore_thresh = .7
truth_thresh = 1
random=1
7.2 FEATURE 2
#import necessary packagesimport
cv2
import os
import numpy as np
from .utils import download_file
initialize = Truenet
= None
dest_dir = os.path.expanduser('~') + os.path.sep + '.cvlib' + os.path.sep + 'object_detection' +os.path.sep
+ 'yolo' +
os.path.sep + 'yolov3'
classes = None
#colors are BGR instead of RGB in python
COLORS = [0,0,255], [255,0,0]
def populate_class_labels():
#we are using a pre existent classifier which is more reliable and more efficient than one#we could make
using only a laptop
#The classifier should be downloaded automatically when you run this scriptclass_file_name =
'yolov3_classes.txt'
class_file_abs_path = dest_dir + os.path.sep + class_file_name
url = 'https://github.com/Nico31415/Drowning-Detector/raw/master/yolov3.txt'if not
os.path.exists(class_file_abs_path):
download_file(url=url, file_name=class_file_name, dest_dir=dest_dir)f =
open(class_file_abs_path, 'r')
classes = [line.strip() for line in f.readlines()]
return classes
def get_output_layers(net)
#the number of output layers in a neural network is the number of possible#things the network
can detect, such as a person, a dog, a tie, a phone... layer_names = net.getLayerNames()
output_layers = [layer_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]
return output_layers
def draw_bbox(img, bbox, labels, confidence, Drowning, write_conf=False):
global COLORS
global classes
if classes is None:
classes = populate_class_labels()
for i, label in enumerate(labels):
#if the person is drowning, the box will be drawn red instead of blueif label ==
'person' and Drowning:
color = COLORS[0] label
= 'DROWNING'
else:
color = COLORS[1]
if write_conf:
label += ' ' + str(format(confidence[i] * 100, '.2f')) + '%'
#you only need to points (the opposite corners) to draw a rectangle. These points#are stored in the
variable bbox
cv2.rectangle(img, (bbox[i][0],bbox[i][1]), (bbox[i][2],bbox[i][3]), color, 2)
```

```
cv2.putText(img, label, (bbox[i][0],bbox[i][1]-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
return img
def detect_common_objects(image, confidence=0.5, nms_thresh=0.3):
Height, Width = image.shape[:2]scale =
0.00392
global classes
global dest_dir
#all the weights and the neural network algorithm are already preconfigured#as we are using
YOLO
#this part of the script just downloads the YOLO files
config_file_name = 'yolov3.cfg'
config_file_abs_path = dest_dir + os.path.sep + config_file_name
weights_file_name = 'yolov3.weights'
weights_file_abs_path = dest_dir + os.path.sep + weights_file_name
url = 'https://github.com/Nico31415/Drowning-Detector/raw/master/yolov3.cfg'
if not os.path.exists(config_file_abs_path):
download_file(url=url, file_name=config_file_name, dest_dir=dest_dir)
url = 'https://pjreddie.com/media/files/yolov3.weights'
if not os.path.exists(weights_file_abs_path):
download_file(url=url, file_name=weights_file_name, dest_dir=dest_dir)
global initialize
global net
if initialize:
classes = populate_class_labels()
net = cv2.dnn.readNet(weights_file_abs_path, config_file_abs_path)initialize = False
blob = cv2.dnn.blobFromImage(image, scale, (416,416), (0,0,0), True, crop=False)
net.setInput(blob)
outs = net.forward(get_output_layers(net))
class_ids = []
confidences = []
boxes = []
for out in outs:
for detection in out: scores =
detection[5:]
class_id = np.argmax(scores)
max_conf = scores[class_id] if
max_conf > confidence:
center_x = int(detection[0] * Width) center_y =
int(detection[1] * Height)w = int(detection[2] * Width)
h = int(detection[3] * Height)x = center_x - w / 2
y = center_y - h / 2 class_ids.append(class_id)
confidences.append(float(max_conf))boxes.append([x, y, w, h])
indices = cv2.dnn.NMSBoxes(boxes, confidences, confidence, nms_thresh)
bbox = []label = []conf =
[]
for i in indices:
i = i[0]
box = boxes[i]x = box[0]
y = box[1] w = box[2]h =
box[3]
bbox.append([round(x), round(y), round(x+w), round(y+h)])
```

```
  label.append(str(classes[class_ids[i]])) conf.append(confidences[i])
return bbox, label, conf
```

9 .RESULT

PERFORMANCE METRICS

```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>High Quality Facial Recognition</title>
<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min. css"
rel="stylesheet">
<script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js">
</script>
<script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js">
</script>
<script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js">
</script>
<link href="{{ url_for('static', filename='css/main.css') }}"rel="stylesheet">
<style>
.bg-dark {
background-color: #42678c!important;
}
#result {
color: #0a1c4ed1;
</head>
}
</style>
<body style="background-color:black";>
<header id="head" class="header">
<section id="navbar">
<h1 class="nav-heading"></i>Virtual Eye</h1>
<div class="nav--items">
<ul>
<li><a href="{{ url_for('index')}}">Home</a></li>
<li><a
href="{{ url_for('logout')}}">Logout</a></li>
<!-- <li><a href="#about">About</a></li>
<li><a href="#services">Services</a></li> -->
</ul>
</div>
</section>
</header>
<div class="container">
<div id="content" style="margin-top:2em">
<div class="container">
<div class="row">
```

```
<div class="col-sm-6 bd" >
<h2><em style="color:white;">High Quality Facial
Recognition</em></h2>
<br>
<p><h5><i style="color:white;">Emotion Detection Through
Facial Feature Recognition</i></h5></p>
<img
src="https://130e178e8f8ba617604b8aedd782b7d22cfe0d1146da69a52436.ssl.cf1.rackcdn.com/facial-
recognitionuse-triggers-gdpr-fine-showcase_image-10-a- 12991.jpg"
style="height:240px"class="img-rounded" alt="Gesture">
</div>
<div class="col-sm-6">
<div>
Image Here</h4>
<h4 style="color:white;">Upload
<form action = "http://localhost:5000/" id="upload-file"
method="post" enctype="multipart/form-data">
<label for="imageUpload" class="uploadlabel">
</label>
Choose Image
<input type="file" name="image"
id="imageUpload" accept=".png, .jpg, .jpeg,.pdf">
</form>
<div class="image-section" style="display:none;">
<div class="img-preview">
<div id="imagePreview">
</div>
</div>
<div>
<button type="button" class="btn btn-info btn-lg "
id="btn-predict">Analyse</button>
</div>
</div>
<div class="loader" style="display:none;"></div>
<h3>
</h3>
<span id="result"> </span>
</div>
</div>
</div>
</body>
</div>
</div>
</div>
<footer>
<script src="{{ url_for('static', filename='js/main.js') }}"
type="text/javascript"></script>
</footer>
</html>
```

Index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!--Bootstrap -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/boo tstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGg
FAW/dAiS6JXm" crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpG
FF93hXpG5KkN" crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/
popper.min.js" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPsk
vXusvfa0b4Q" crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootst
rap.min.js" integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5
+76PVCmYl" crossorigin="anonymous"></script>
<script src="https://kit.fontawesome.com/8b9cdc2059.js"
crossorigin="anonymous"></script>
<link href="https://fonts.googleapis.com/css2?family=Akronim&family=
Roboto&display=swap" rel="stylesheet">
<link rel="stylesheet" href="../static/style.css">
<!-- <script defer src="../static/js/main.js"></script> -->
<title>Virtual Eye</title>
</head>
<body>
<header id="head" class="header">
<section id="navbar">
<h1 class="nav-heading"></i>Virtual Eye</h1>
<div class="nav--items">
<ul>
<li><a
href="{{ url_for('index')}}">Home</a></li>
<li><a
href="{{ url_for('login')}}">Login</a></li>
<li><a
href="{{ url_for('register')}}">Register</a></li>
<li><a href="{{ url_for('login')}}">Demo</a></li>
</ul>
</div>
</section>
<section id="slider">
<div id="carouselExampleIndicators" class="carousel" data-ride="carousel">
<ol class="carousel-indicators ">
<li data-target="#carouselExampleIndicators" data-slide- to="0"
class="active "></li>
```

```
<li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
<li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
</ol>
<div class="carousel-inner">
<div class="carousel-item active">
<img class="d-block w-100" src="../static/img/1.png"alt="First slide">
</div>
<div class="carousel-item">
<img class="d-block w-100" src="../static/img/second.jpg"
alt="Second slide">
</div>
<div class="carousel-item">
<img class="d-block w-100" src="../static/img/third.jpg"alt="Third slide">
</div>
</div>
<a class="carousel-control-prev" href="#carouselExampleIndicators"
role="button" data-slide="prev">
<span class="carousel-control-prev-icon" ariahidden="true"></span>
<span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next" href="#carouselExampleIndicators"
role="button" data-slide="next">
<span class="carousel-control-next-icon" ariahidden="true"></span>
<span class="sr-only">Next</span>
</a>
</div>
</section>
</header>
<section id="about">
<div class="top">
<h3 class="title text-muted">
ABOUT PROJECT
</h3>
<div class="line"></div>
</div>
<div class="body">
<div class="left">
<h2>Problem:</h2>
<p>
Swimming is one of the best exercises that helps people to reduce stress in
this urban lifestyle. Swimming pools are found larger in number in the hotels,
weekend tourist spots and barelypeople have in their house backyard. Beginners,
especially oftenfeel it difficult to breathe under water and causes breathing
trouble which in turn cause a drowning accident. Worldwide, drowning produces
a higher rate of mortality without causing injury to children. Children under six
of their age are found to besuffering the highest drowning mortality rates
worldwide..Such kinds of deaths account for the third cause of unplanned death
globally, with about 1.2 million cases yearly.
</p>
</div>
<div class="left">
```

<h2>Solution:</h2>

To overcome the conflict, a meticulous system is to be implemented along the swimming pools to save the human life. Bystudying body movement patterns and connecting cameras to an artificial intelligence (AI)system we can devise an underwater poolsafety system that reduces the risk of drowning. Usually such systems can be developed by installing more than 16 cameras underwater and ceiling and analysing the video feeds to detect any anomalies . but AS a POC we make use of one camera that streams the video underwater and analyses the position of
swimmers to assess the probability of drowning ,if it is higher thanan alert will be generated to attract lifeguards attention.

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

</section>
<section id="footer">
Copyright Â© 2022. All Rights Reserved

</section>
</body>
</html>
Logout.html
<!DOCTYPE html>
<html >
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Virtual Eye</title>
<link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet'
type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>

```
<link href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin Sans'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>
<style>
.header {
top:0; margin:0px;
left: 0px;
right: 0px;
position: fixed;
background-color: #28272c;
color: white;
box-shadow: 0px 8px 4px grey;
overflow: hidden;
padding-left:20px;
font-family: 'Josefin Sans';
font-size: 2vw;
width: 100%;
height:8%;
text-align: center;
}
.topnav {
overflow: hidden;
background-color: #333;
}
.topnav-right a {
float: left; color:
#f2f2f2;
text-align: center;
padding: 14px 16px; textdecoration: none; fontsize: 18px;
}
.topnav-right a:hover {
background-color: #ddd;
color: black;
}
.topnav-right a.active {
background-color: #565961;
color: white;
}
.topnav-right {
float: right;
padding-right:100px;
}
.login{
margin-top:-70px;
}
body {
background-color:#ffffff;
background-repeat: no-repeat;
```

```
background-size:cover;
background-position: 0px 0px;
}
.main{
margin-top:100px;
text-align:center;
}
form { margin-left:400px;margin-right:400px;}
input[type=text], input[type=email],input[type=number],input[type=password] {
width: 100%;
padding: 12px 20px;
display: inline-block;
margin-bottom:18px;
border: 1px solid #ccc;
box-sizing: border-box;
}
button {
background-color: #28272c;
color: white;
padding: 14px 20px;
margin-bottom:8px;
border: none; cursor:
pointer; width: 20%;
}
button:hover {
opacity: 0.8;
}
.cancelbtn {
width: auto;
padding: 10px 18px;
background-color: #f44336;
}
.imgcontainer { textalign: center;
margin: 24px 0 12px 0;
}
img.avatar {
width: 30%;
border-radius: 50%;
}
.container {
padding: 16px;
}
span.psw {
float: right;
padding-top: 16px;
}
/* Change styles for span and cancel button on extra small screens
*/
@media screen and (max-width: 300px) {
span.psw {
display: block;
```

```
float: none;
}
.cancelbtn {
width: 100%;
}
}
</style>
</head>
<body style="font-family:Montserrat;">
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text- align:left;color:white; paddingtop:1%">Virtual eye</div>
<div class="topnav-right" style="padding-top:0.5%;">
<a href="{{ url_for('home')}}">Home</a>
<a href="{{ url_for('login')}}">Login</a>
<a href="{{ url_for('register')}}">Register</a>
</div>
</div>
<div class="main">
<h1>Successfully Logged Out!</h1>
<h3 style="color:#4CAF50">Login for more information<h3>
<a href="{{ url_for('login') }}"><button
type="submit">Login</button></a>
</form>
</div>
</body>
</html>
```

Prediction.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!--Bootstrap -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/boo tstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGg
FAW/dAiS6JXm" crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpG
FF93hXpG5KkN" crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/
popper.min.js" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPsk
vXusvfa0b4Q" crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootst
rap.min.js" integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5
+76PVCmYl" crossorigin="anonymous"></script>
<script src="https://kit.fontawesome.com/8b9cdc2059.js"
```

```
crossorigin="anonymous"></script>
<link href="https://fonts.googleapis.com/css2?family=Akronim&family=
Roboto&display=swap" rel="stylesheet">
<link rel="stylesheet" href="../static/style.css">
<script defer src="../static/js/JScript.js"></script>
<title>Prediction</title>
</head>
<body>
<header id="head" class="header">
<section id="navbar">
<h1 class="nav-heading"></i>Virtual Eye</h1>
<div class="nav--items">
<ul>
<li><a href="{{ url_for('index')}}">Home</a></li>
<li><a
href="{{ url_for('logout')}}">Logout</a></li>
<!-- <li><a href="#about">About</a></li>
<li><a href="#services">Services</a></li> -->
</ul>
</div>
</section>
</header>
<!-- dataset/Training/metal/metal326.jpg -->
</br>
<section id="prediction">
<h2 class="title text-muted">Virtual Eye- Life Guard forSwimming Pools to
Detect Active Drowning</h1>
<div class="line" style="width: 900px;"></div>
</section>
</br>
<section id="about">
<div class="body">
<div class="left">
<p>
Swimming is one of the best exercises that helps people to reduce stress in
this urban lifestyle. Swimming pools are found larger in number in the hotels,
weekend tourist spots and barelypeople have in their house backyard. Beginners,
especially oftenfeel it difficult to breathe under water and causes breathing
trouble which in turn cause a drowning accident. Worldwide, drowning produces
a higher rate of mortality without causing injury to children. Children under six
of their age are found to besuffering the highest drowning mortality rates
worldwide..Such kinds of deaths account for the third cause of unplanned death
globally, with about 1.2 million cases yearly.
</p>
</div>
<div class="left">
<div class="prediction-input">
<img class="d-block w-100" src="../static/img/second.jpg"alt="Second slide">
</br>
<form id="form" action="/result" method="post"
enctype="multipart/form-data">
```

```
 Demo">
 </div>
 </div>
 </section>
 <input type="submit" class="submitbtn" value="ClickMe! For a
 </form>
 </div>
 <h5 style="text-color:Red">
 <b style="text-color:Red">{{prediction}}<b>
 </h5>
 </br></br>
 <section id="footer">
 <p>Copyright Â© 2021. All Rights Reserved</p>
 </section>
 </body>
</html>
```

10. ADVANTAGES & DISADVANTAGES:

Advantage:

(i) user feel comfortable and more secure
(ii) Children, adult, pet animal , old age people are used
(iii) spending more time for family, freedom for safety guards near the Swimming pool
(iv) Swimmers, resort are gain in the financal
(v) drowning should be monitored

Disadvantage:

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

11 CONCLUSION:

This section will draw from three core documents: ISO_20380, HSG179,and the recently published German guideline, DGfdB R 94.15. A summary of each is given, outlining the key messages they disseminate and what this means for those involved with DDS.ISO_20380 This document focuses on the requirements for the installation, operation, maintenance and performance of DDS, the testing methods, and the information required from the supplier in the operating manual. These international standards do not apply to systems used in domestic pools or pools smaller than 150m2 .Prior to the installation of any DDS, ‘a technical study shall be carried out by the supplier in consultation with or based on information provided by the swimming pool’s owner/operator’. This is to establish the quantity and positioning of th

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

12 FUTURE SCOPE:

This lifeguard system consists of three main components, i.e., the drowning detection, the rescuing drone, and the hazardous activity detection. All three components combined will create a system capable of detecting drowning victims, dispatching an inflatable tube using a drone (as depicted in Fig.9) and detecting hazardous activities—eventually becoming an entity that could assist a lifeguard. The system is accessible to its primary user, presumably a pool owner or a lifeguard, in the form of an interface with a sound alarm and an android mobile service that holds the capabilities of receiving Firebase notifications. Confined with a few of the hardware limitations, such as the use of a single camera and the Jetson Nano in the presence of better-quality hardware, the speed and accuracy of the overall system is becoming a state-of-theart. This limitation could be omitted with the use of multiple cameras that could be placed over the premises in several ground coordinates, increasing the accuracy of the computer vision algorithms. Moreover, due to the inability to fly a drone in extreme weather conditions such as rain, strong winds or lightning, the system is limited to be used under few specifications. As swimming in extreme weather conditions is not preferred either, the system could be further improved to emit a warning signal if a person was to swim in any of the above weather conditions, bypassing the need to fly the drone. Additionally, all the processing is done on the client side of the applications on the Jetson Nano board, preventing any security and privacy issues that might arise due to the sensitive information inputted through the cameras. For future developments convenience wise, the system could benefit by having an additional set of cameras to identify and verify a drowning or a hazardous activity on the premises. Accessibility could also be improved by extending the Android service to be an application both in Android and iOS platforms that could hold the details of each premise individually, making a centralised system that watches over the decentralised 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.

APPENDIX:

Source Code:

```
# Testing#
batch=1
# subdivisions=1#
Training batch=64
subdivisions=16
width=608 height=608
channels=3
momentum=0.9
decay=0.0005 angle=0
saturation = 1.5
exposure = 1.5hue=.1
learning_rate=0.01
burn_in=1000 max_batches =
500200policy=steps
steps=400000,450000
scales=.1,.1
[convolutional]
batch_normalize=1
filters=32 size=3
stride=1
pad=1
activation=leaky
# Downsample
[convolutional]
batch_normalize=1
filters=64 size=3
stride=2
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=32 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=64 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear#
Downsample
[convolutional]
batch_normalize=1
filters=128 size=3
```

stride=2
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=64 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=64 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
# Downsample
[convolutional]
batch_normalize=1
filters=256size=3
stride=2 pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-

3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1

```
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=3
stride=1
pad=1
activation=leaky
```

[shortcut]from=-
3
activation=linear#
Downsample
[convolutional]
batch_normalize=1
filters=512 size=3
stride=2
pad=1 activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1 activation=leaky
[shortcut]from=-
3

```
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
```

```
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear#
Downsample
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=2
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
```

```
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=1024 size=3
stride=1
pad=1
activation=leaky
[shortcut]from=-
3
activation=linear
######################
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=1024
activation=leaky
[convolutional]
```

batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=1024
activation=leaky
[convolutional]
batch_normalize=1
filters=512 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=1024
activation=leaky
[convolutional]size=1
stride=1
pad=1 filters=255
activation=linear
[yolo]
mask = 6,7,8
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,
156,198, 373,326
classes=80
num=9 jitter=.3
ignore_thresh = .7
truth_thresh = 1random=1
[route] layers = -4
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[upsample]
stride=2
[route]
layers = -1, 61
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3

stride=1 pad=1
filters=512
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=512
activation=leaky
[convolutional]
batch_normalize=1
filters=256 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=512
activation=leaky
[convolutional]size=1
stride=1
pad=1 filters=255
activation=linear
[yolo]
mask = 3,4,5
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,
156,198, 373,326
classes=80
num=9 jitter=.3
ignore_thresh = .7
truth_thresh = 1random=1
[route] layers = -4
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[upsample]
stride=2
[route]
layers = -1, 36
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1

pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=256
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1
pad=1 filters=256
activation=leaky
[convolutional]
batch_normalize=1
filters=128 size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1size=3
stride=1 pad=1
filters=256
activation=leaky
[convolutional]size=1
stride=1
pad=1 filters=255
activation=linear
[yolo]
mask = 0,1,2
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90,
156,198, 373,326
classes=80
num=9 jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1
Source code(ii)
#import necessary packagesimport
cv2
import os
import numpy as np
from .utils import download_file
initialize = Truenet
= None
dest_dir = os.path.expanduser('~') + os.path.sep + '.cvlib' + os.path.sep + 'object_detection' +os.path.sep +
'yolo' +

```
os.path.sep + 'yolov3'
classes = None
#colors are BGR instead of RGB in python
COLORS = [0,0,255], [255,0,0]
def populate_class_labels():
#we are using a pre existent classifier which is more reliable and more efficient than one#we could make
using only a laptop
#The classifier should be downloaded automatically when you run this scriptclass_file_name =
'yolov3_classes.txt'
class_file_abs_path = dest_dir + os.path.sep + class_file_name
url = 'https://github.com/Nico31415/Drowning-Detector/raw/master/yolov3.txt'if not
os.path.exists(class_file_abs_path):
download_file(url=url, file_name=class_file_name, dest_dir=dest_dir)f =
open(class_file_abs_path, 'r')
classes = [line.strip() for line in f.readlines()]
return classes
def get_output_layers(net)
#the number of output layers in a neural network is the number of possible#things the network
can detect, such as a person, a dog, a tie, a phone... layer_names = net.getLayerNames()
output_layers = [layer_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]
return output_layers
def draw_bbox(img, bbox, labels, confidence, Drowning, write_conf=False):
global COLORS
global classes
if classes is None:
classes = populate_class_labels()
for i, label in enumerate(labels):
#if the person is drowning, the box will be drawn red instead of blueif label ==
'person' and Drowning:
color = COLORS[0] label
= 'DROWNING'
else:
color = COLORS[1]
if write_conf:
label += ' ' + str(format(confidence[i] * 100, '.2f')) + '%'
#you only need to points (the opposite corners) to draw a rectangle. These points#are stored in the
variable bbox
cv2.rectangle(img, (bbox[i][0],bbox[i][1]), (bbox[i][2],bbox[i][3]), color, 2)
cv2.putText(img, label, (bbox[i][0],bbox[i][1]-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
return img
def detect_common_objects(image, confidence=0.5, nms_thresh=0.3):
Height, Width = image.shape[:2]scale =
0.00392
global classes
global dest_dir
#all the weights and the neural network algorithm are already preconfigured#as we are using
YOLO
#this part of the script just downloads the YOLO files
config_file_name = 'yolov3.cfg'
config_file_abs_path = dest_dir + os.path.sep + config_file_name
weights_file_name = 'yolov3.weights'
weights_file_abs_path = dest_dir + os.path.sep + weights_file_name
```

```
url = 'https://github.com/Nico31415/Drowning-Detector/raw/master/yolov3.cfg'
if not os.path.exists(config_file_abs_path):
download_file(url=url, file_name=config_file_name, dest_dir=dest_dir)
url = 'https://pjreddie.com/media/files/yolov3.weights'
if not os.path.exists(weights_file_abs_path):
download_file(url=url, file_name=weights_file_name, dest_dir=dest_dir)
global initialize
global net
if initialize:
classes = populate_class_labels()
net = cv2.dnn.readNet(weights_file_abs_path, config_file_abs_path)initialize = False
blob = cv2.dnn.blobFromImage(image, scale, (416,416), (0,0,0), True, crop=False)
net.setInput(blob)
outs = net.forward(get_output_layers(net))
class_ids = []
confidences = []
boxes = []
for out in outs:
for detection in out: scores =
detection[5:]
class_id = np.argmax(scores)
max_conf = scores[class_id] if
max_conf > confidence:
center_x = int(detection[0] * Width) center_y =
int(detection[1] * Height)w = int(detection[2] *
Width)
h = int(detection[3] * Height)x =
center_x - w / 2
y = center_y - h / 2 class_ids.append(class_id)
confidences.append(float(max_conf))boxes.append([x, y,
w, h])
indices = cv2.dnn.NMSBoxes(boxes, confidences, confidence, nms_thresh)
bbox = []
label = []
conf = []
for i in indices:
i = i[0]
box = boxes[i]x =
box[0]
y = box[1] w =
box[2] h =
box[3]
bbox.append([round(x), round(y), round(x+w), round(y+h)])
label.append(str(classes[class_ids[i]])) conf.append(confidences[i])
  return bbox, label, confreturn bbox, label, conf
```

GitHub & Project Demo Link:

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-53995-1661587063

Project Demo Link:

https://drive.google.com/file/d/1VKqYitc_G28DdCt2rFIwAV4QvNONaVRX/view?usp=share_link