VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning

PROJECT REPORT

Submitted by

KEVINKERUBAKAR M (19EUCS063)

KOWSHIK S (19EUCS067)

MELLWYN JUBEN S (19EUCS083)

KUMARAGURU T (19EUCS071)

in partial fulfillment of the requirements for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

COIMBATORE

(An Autonomous Institution)

ANNA UNIVERSITY: CHENNAI

MAY 2022

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution)
(Approved by AICTE and Affiliated to Anna University, Chennai)
ACCREDITED BY NAAC WITH "A" GRADE

BONAFIDE CERTIFICATE

Certified that this project report titled **"VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning"** is the bonafide work of **KEVINKERUBAKAR M (19EUCS063), KOWSHIK S (19EUCS067), MELLWYN JUBEN S(19EUCS083),KUMARAGURU T(19EUCS071)** who carried out the project work under my supervision.

SIGNATURE	SIGNATURE
Dr.K. SASI KALA RANI, M.E., Ph.D.,	**Ms.K.M.MAJIDHA FATHIMA**
HEAD OF THE DEPARTMENT	**SUPERVISOR**

Department of Computer Science and Engineering
Sri Krishna College of Engineering and Technology
Kuniamuthur,
Coimbatore

This project report is submitted for the Autonomous Project Viva-Voce examination held on 15.11.2022

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our sincere thanks to the management and **Dr.J.JANET, M.E.,Ph.D.,** Principal, Sri Krishna College of Engineering and Technology, Coimbatore for providing us the facilities to carry out this project work.

We are highly indebt to **Dr.K. SASIKALA RANI, M.E.,Ph.D.,** Head of Computer Science and Engineering for her continuous evaluation, valuable suggestions and comments given during the course of the project work.

We are thankful to **Dr.P.MOHAN,** Project Co-ordinator**,** Department of Computer Science and Engineering for her continuous evaluation, valuable suggestions and comments given during the course of the project work.

We express our deep sense of gratitude to our guide**, Ms.K.M.MAJIDHA FATHIMA,** Professor in the department of Computer science and Engineering for her valuable advice, guidance and support during the course of our project work.

By this, we express our heartfelt sense of gratitude and thanks to our beloved parents, family and friends who have all helped in collecting the resources and materials needed for this project and for their support during the study and implementation this project.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Life Guard have an important role in saving person if in any trouble on water bodies. To make their role a little less complex,a drowning person detector can be used to alert them if any in trouble.Life Guard saves that drowning person .This project focused on how LifeGuard can detect suspicious drowning activities over sea or swimming places

1.2 PURPOSE

-To overcome this lacking ability a camera detector can be made and fit in some areas
-These camera can detect a drowning person and alert the guard
-Help in easing Life Guard job and save many more life
-Life Guard plays important role in security of person in water bodies
-With proper equipment he saves people from danger
-Life Guard should be trained well to help people
-Life Guard lacks ability to see every single area of water body

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Drowning is the process of experiencing respiratory impairment from submersion/immersion in liquid. Drowning outcomes are classified as death, morbidity and no morbidity. Agreed terminology is essential to describe the problem and to allow effective comparisons of drowning trends. Thus, this definition of drowning adopted by the 2002 World Congress on Drowning should be widely used. Drowning is a leading killer. The latest WHO Global Health Estimates indicate that almost 236 000 people lost their lives to drowning in 2019. Just over 50% of these deaths occur among those aged under 30 years, and drowning is the sixth leading cause of death worldwide for children aged 5-14 years. Over 90% of drowning deaths occur in low- and middle-income countries. Drowning prevention interventions range from community-based solutions, such as day care for children and barriers controlling access to water, to effective national policies and legislation around water safety, including setting and enforcing boating, shipping and ferry regulations. Much more needs to be done to prevent drowning, and achieving commitments made under the Sustainable Development Goals will not be possible without addressing drowning prevention.

2.2 REFERENCES

[1] (Omer & Abdullah, 2013)Omer, E., & Abdullah, M. F. A. (2013). GPS and SMS-Based Child Tracking System Using Smart Phone. Internasionala Journal of Electrical, Computer, Electronic and Communication Engineering, 7(2), 171–174.

[2] (Pawade & Gaikwad, 2015)Pawade, R. H., & Gaikwad, A. N. (2015). Android Based Children Tracking System, 4(6), 2088–2092.

[3] (Pham, Drieberg, & Nguyen, 2013)Pham, H. D., Drieberg, M., & Nguyen, C. C. (2013). Development of vehicle tracking system using GPS and GSM modem. In 2013 IEEE Conference on Open Systems, ICOS 2013 (pp. 89– 94). https://doi.org/10.1109/ICOS.2013.6735054

[4] (Rycroft, 1997)Rycroft, M. J. (1997). Understanding GPS. Principles and applications. Journal of Atmospheric and Solar-Terrestrial Physics, 59(5), 598–599. https://doi.org/10.1016/S1364-6826(97)83337-8

[5] (Sarjana & Ii, 2012)Sarjana, P., & Ii, M. (2012). GSM & GPS BASED SCHOOL KIDS TRACKING SYSTEM NG WOON CEA This Report Is Submitted In Partial Fulfilment of Requirements for the Award of Bachelor Degree of Electronic Engineering (Industrial Electronic) With Honours Faculty of Electronic Engineering.

[6] (Salihoglu & Widom, 2013)Salihoglu, S., & Widom, J. (2013). Gps. Proceedings the 25th International Conference on Scientific and Statistical Database Management SSDBM, 1. https://doi.org/10.1145/2484838.2484843

2.3 PROBLEM STATEMENT

Life Guard have an important role in saving person if in any trouble on water bodies. To make their role a little less complex,a drowning person detector can be used to alert them if any in trouble.Life Guard saves that drowning person.

There are many actions to prevent drowning. Installing barriers (e.g. covering wells, using doorway barriers and playpens, fencing swimming pools etc.) to control access to water hazards, or removing water hazards entirely greatly reduces water hazard exposure and risk. Community-based, supervised child care for pre-school children can reduce drowning risk and has other proven health benefits. Teaching school-age children basic swimming, water safety and safe rescue skills is another approach. But these efforts must be undertaken with an emphasis on safety, and an overall risk management that includes a safety-tested curricula, a safe training area, screening and student selection, and student-instructor ratios established for safety. Effective policies and legislation are also important for drowning prevention. Setting and enforcing safe boating, shipping and ferry regulations is an important part of improving safety on the water and preventing drowning. Building resilience to flooding and managing flood risks through better disaster preparedness planning, land use planning, and early warning systems can prevent drowning during flood disasters. Developing a national water safety strategy can raise awareness of safety around water, build consensus around solutions, provide strategic direction and a framework to guide multisectoral action and allow for monitoring and evaluation of efforts.

WHO Response:

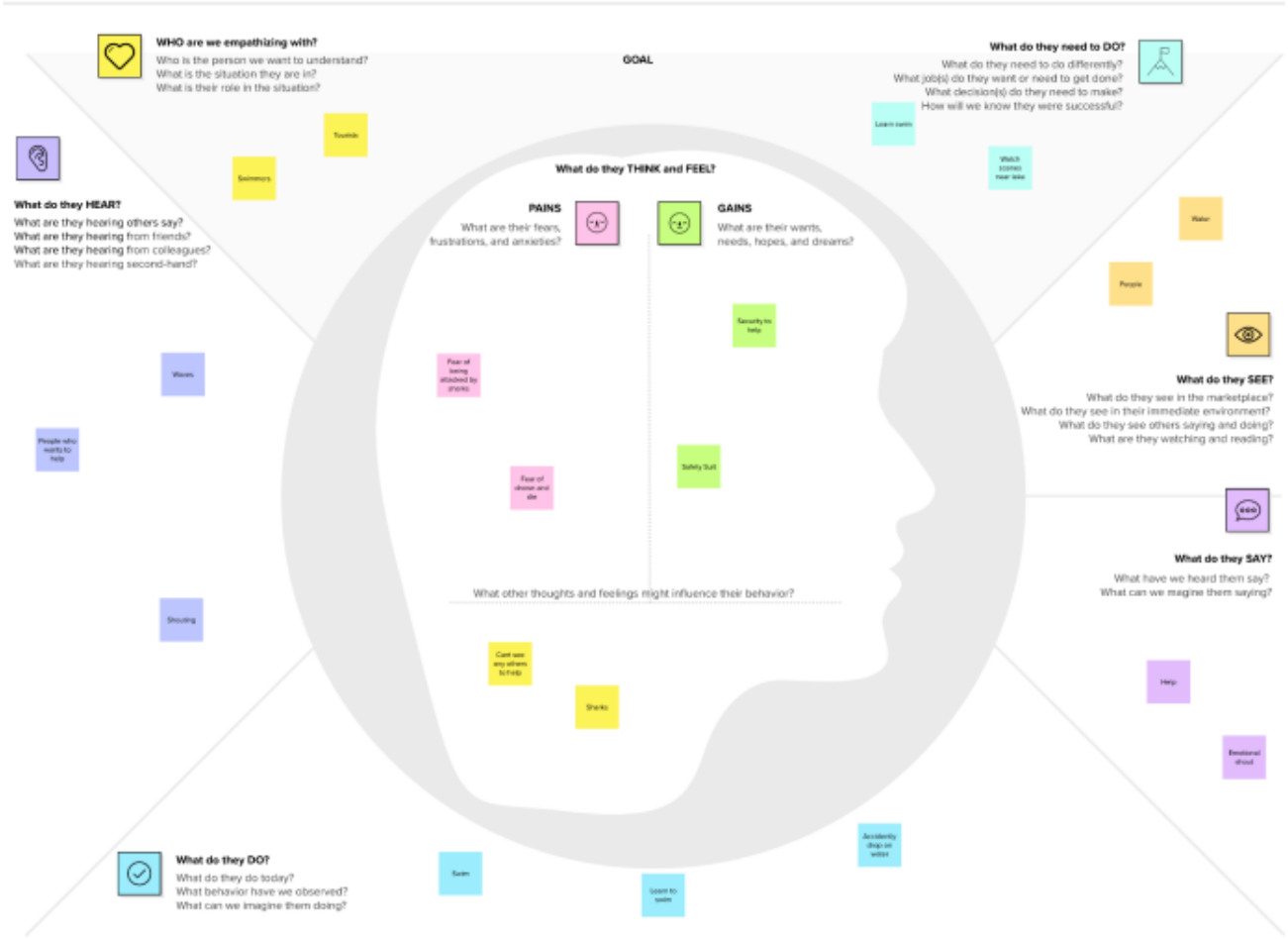
The Global report on drowning provides recommendations to governments to tailor and implement effective drowning prevention programmes to their settings, improve data about drowning, and develop national water safety plans. The report also points out the multisectoral nature of drowning and calls for greater coordination and collaboration among UN agencies, governments, key NGOs and academic institutions to prevent drowning. In May 2017, WHO released Preventing drowning: an implementation guide. This publication builds on the Global report on drowning and provides concrete guidance for drowning prevention practitioners on how to implement drowning prevention interventions. At country level, WHO has worked with Ministries of Health in some low- and middle-income countries to prevent drowning through the use of barriers controlling access to water and the establishment of day care centres for pre-school children. In addition, WHO has also funded research in low-income countries exploring priority questions related to drowning prevention.

At a regional level, WHO organizes training programmes and convenes workshops to draw together representatives of governments, NGOs and UN agencies working on drowning prevention. Life Guard: A lifeguard is responsible for the safety of people in an area of water, and usually a defined area immediately surrounding or adjacent to it, such as a beach next to an ocean or lake. The priority is to ensure no harm comes to users of the area for which they are responsible. Lifeguards often take on this responsibility upon employment, although they can also be volunteers. The conditions resulting in drowning are summarized by the 'drowning chain' in which each link can lead directly to an incident, or contribute to a succession of links.[2] It consists of lack of education about water safety or local conditions, a lack of safety advice (for example, about rip currents at a beach) a lack of protection (like no flotation device for a weak swimmer), lack of safety supervision, or an inability to cope with conditions (strong surf with a weak swimmer).

CHAPTER 3

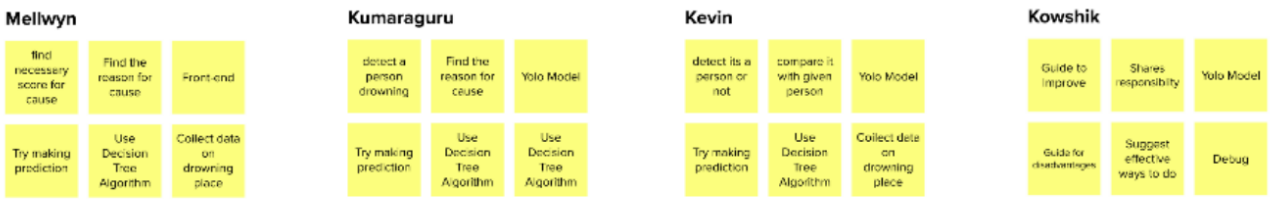
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

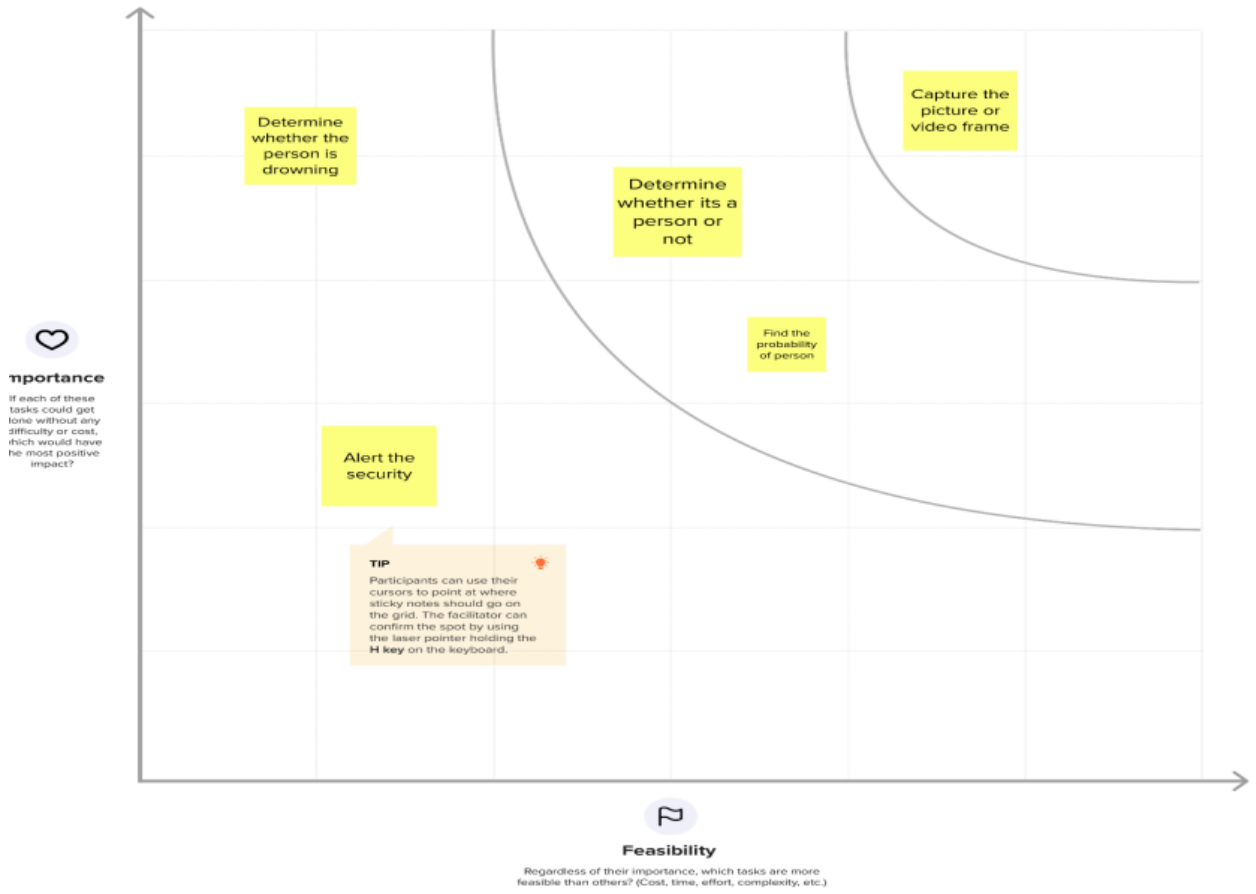


3.2 IDEATION AND BRAINSTORMING

3.2.1 BRAIN STORMING



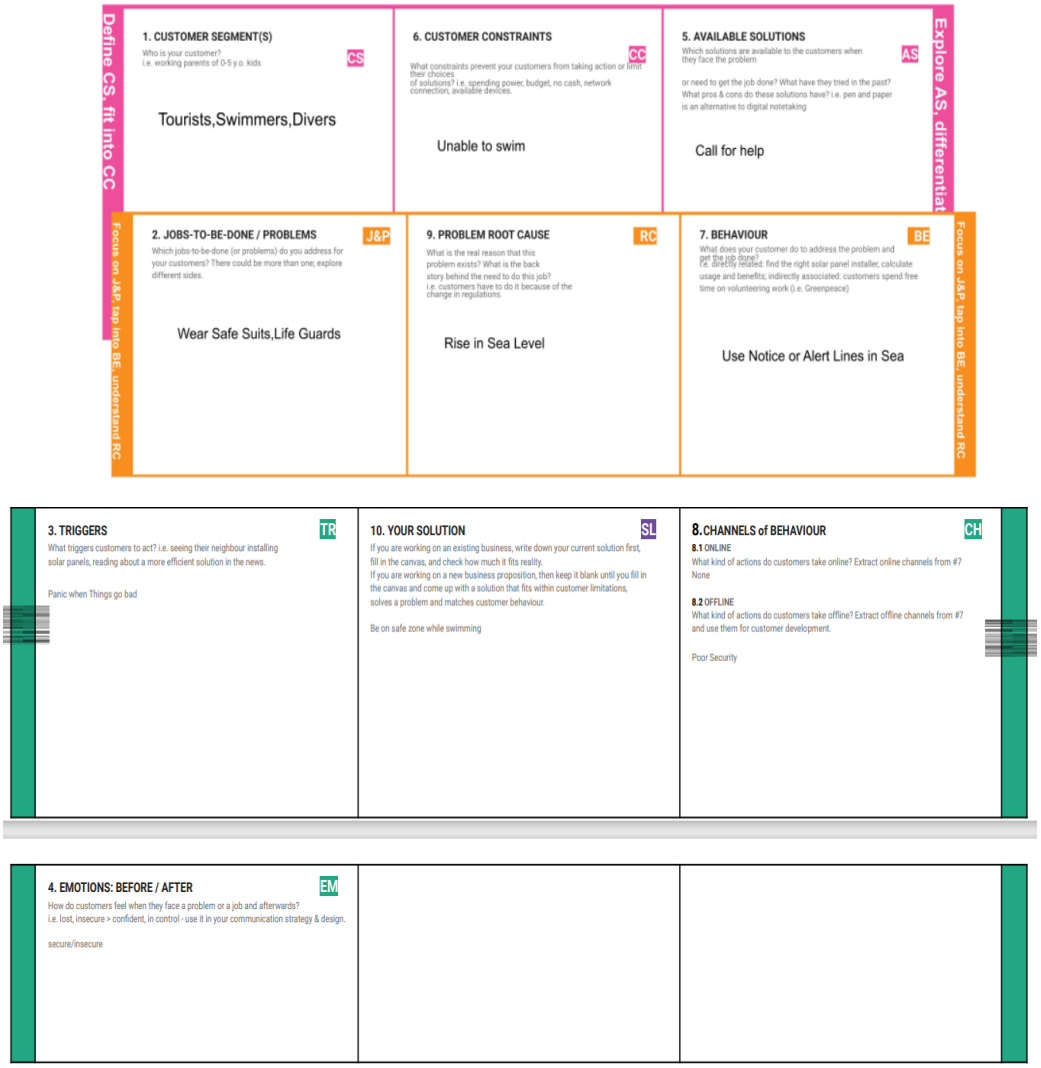
3.2.2 IDEA PRIORITIZATION



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Detect Drowning Person
2.	Idea / Solution description	Scan each frame and find drowning by comparing given model
3.	Novelty / Uniqueness	Help Alert by finding person in danger
4.	Social Impact / Customer Satisfaction	Makes them feel safe
5.	Business Model (Revenue Model)	Yolo-TensorFlow
6.	Scalability of the Solution	Very High

3.4 PROBLEM SOLUTION FIT



CHAPTER-4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

- The system shall allow the user to register their id
- The system should provide all the sensed data from each frame send by text message.
- The system shall check the person probability with the threshold value of each input.
- The system shall notify the user while if person exceed or become below the threshold value.

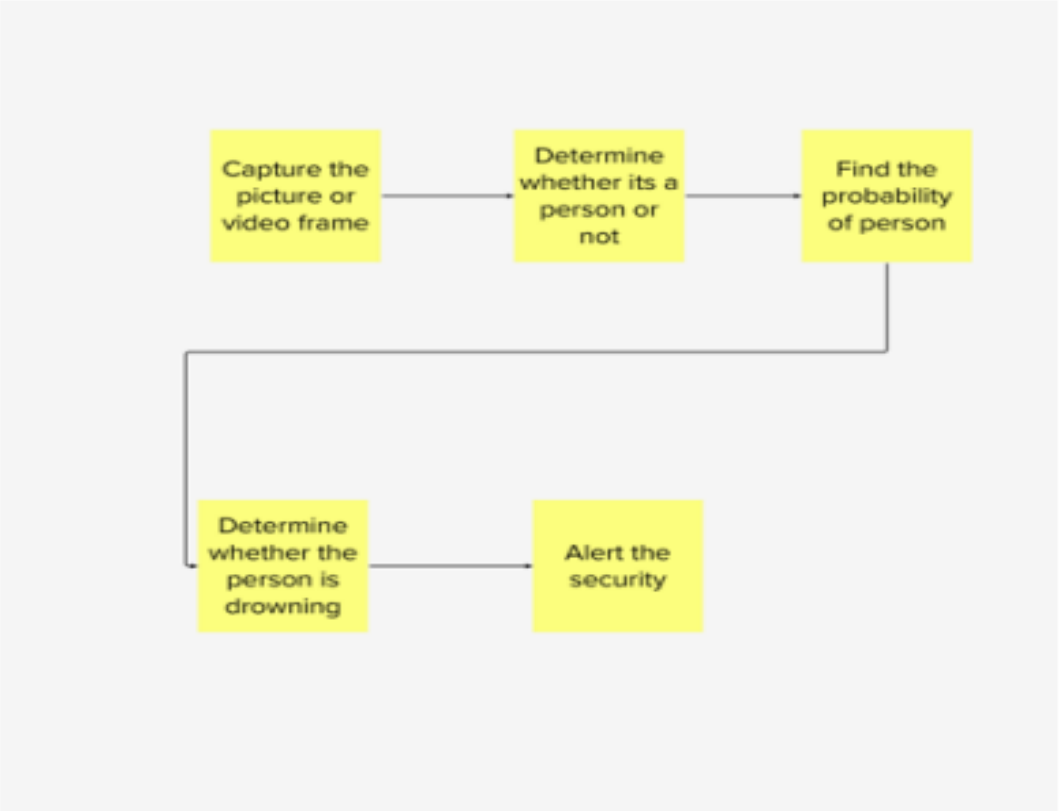
4.2 NON-FUNCTIONAL REQUIREMENTS

- The system shall give the result for different factors using YOLO model as a result their will not be any damage.
- The system shall be maintainable whenever faller occurs.
- The system is cost effective comparing to the features it provides.
- The system shall be usable within a few minutes training.

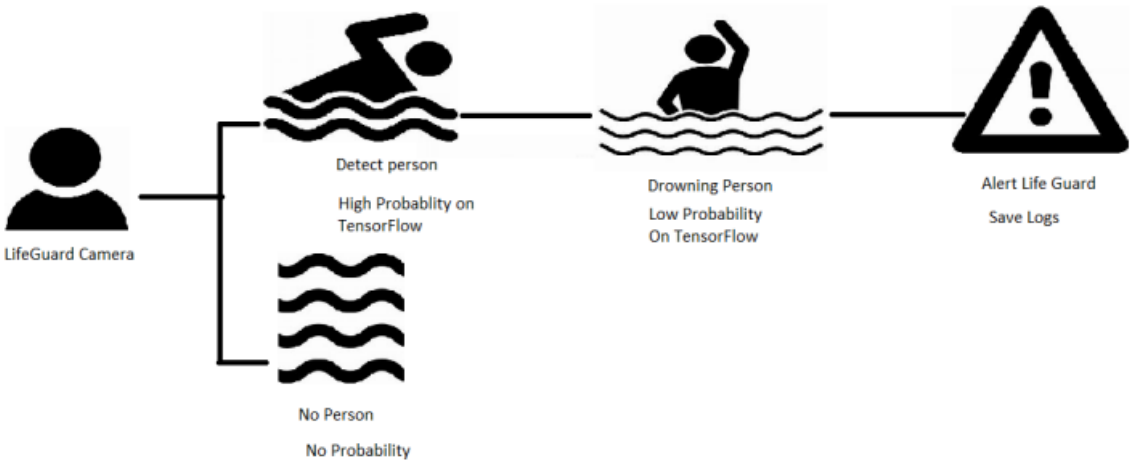
CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



IoT Device :

IoT devices are the non standard computing devices that connect wirelessly to a network and have the ability to transmit data. Here we use IoT devices like microcontroller , sensors for input and buzzers , LED light for output.

IBM Watson IoT Platform :

Clean and simple UI where you can simply add and manage the devices and control access to the IoT devices and monitor the usage of the devices. Here we add up all the devices that our application need , monitor them and connect with api calls to the node - red platform.

Node - RED :

Node - RED is a programming tool for wiring the hardware , API and online services in new and interesting ways. It is a low code platform where you can easily create the application by just drag and drop . We can keep on adding features to our application without worrying about the internal operations. In our application , node red helps to interact with the db (Cloudant db) , store and manage data in the cloud db.

Geofence :

A geofence is a virtual perimeter of a real world - geographical area . A geo world can be dynamically created or match a predefined set of boundaries. In our application , the geo fence is created by the parents using node - red.

Cloudant DB :

Cloudant is an IBM software product, which is primarily delivered as a cloud-based service. Cloudant is a non-relational, distributed database service of the same name. Cloudant db is the datastore where we store every data of this application.

Web UI :

Web is a place where every device in the internet can communicate with each other. Web UI is the application layer for our application where users can interact with their devices , monitor their child , check the location history and check the future and current weather conditions of the child location.

User :

LifeGuard to monitor swimming places

5.3 USER STORIES

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2
	Login	USN-2	As a user, I can log into the application by entering email & password	1
Sprint-2	Camera	USN-3	Gain access to camera	2
Sprint-3	Detect Person	USN-4	Person or not	2
Sprint-4	Detect Drowning Person	USN-5	Drowning or not	1

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022
Sprint-5	20	6 Days	21 Nov 2022	27 Nov 2022	20	27 Nov 2022

CHAPTER 7

CODING & SOLUTIONING

7.1.1 FEATURE 1 : LOGIN

Algorithm :

1. Enter the credentials and hit enter (email and password).
2. If already logged in user is taken to home page
3. If wrong credentials entered , notification displayed to user and user stays in login page.
4. On correct credentials , user is taken to home page.

7.1.2 FEATURE 2 : SIGNUP

Algorithm :

1. Enter the signup form fields (name , email , password , re-enter password , date of birth) and hit enter.
2. All credentials are validated at client side.
3. Email is checked if already registered or not in the database.
4. If already registered , notification displayed. Or else, the user is taken to the successful signup page.

Code:

```
from tkinter import *
import tkinter.messagebox as tkMessageBox
import sqlite3
import subprocess
root = Tk()
root.title("Python: Simple Inventory System")

width = 640
height = 480
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()
x = (screen_width/2) - (width/2)
y = (screen_height/2) - (height/2)
root.geometry("%dx%d+%d+%d" % (width, height, x, y))
```

```
root.resizable(0, 0)


#=====================================VARIABLES=================
====================
USERNAME = StringVar()
PASSWORD = StringVar()
FIRSTNAME = StringVar()
LASTNAME = StringVar()

#=====================================METHODS==================
====================
def Database():
    global conn, cursor
    conn = sqlite3.connect("db_member.db")
    cursor = conn.cursor()
    cursor.execute("CREATE TABLE IF NOT EXISTS `member` (mem_id INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL, username TEXT, password TEXT,
firstname TEXT, lastname TEXT)")


def Exit():
    result = tkMessageBox.askquestion('System', 'Are you sure you want to exit?',
icon="warning")
    if result == 'yes':
        root.destroy()
        exit()


def LoginForm():
    global LoginFrame, lbl_result1
    LoginFrame = Frame(root)
    LoginFrame.pack(side=TOP, pady=80)
```

```
    lbl_username = Label(LoginFrame, text="Username:", font=('arial', 25), bd=18)
    lbl_username.grid(row=1)
    lbl_password = Label(LoginFrame, text="Password:", font=('arial', 25), bd=18)
    lbl_password.grid(row=2)
    lbl_result1 = Label(LoginFrame, text="", font=('arial', 18))
    lbl_result1.grid(row=3, columnspan=2)
    username = Entry(LoginFrame, font=('arial', 20), textvariable=USERNAME, width=15)
    username.grid(row=1, column=1)
    password = Entry(LoginFrame, font=('arial', 20), textvariable=PASSWORD, width=15,
show="*")
    password.grid(row=2, column=1)
    btn_login = Button(LoginFrame, text="Login", font=('arial', 18), width=35,
command=Login)
    btn_login.grid(row=4, columnspan=2, pady=20)
    lbl_register = Label(LoginFrame, text="Register", fg="Blue", font=('arial', 12))
    lbl_register.grid(row=0, sticky=W)
    lbl_register.bind('<Button-1>', ToggleToRegister)

def RegisterForm():
    global RegisterFrame, lbl_result2
    RegisterFrame = Frame(root)
    RegisterFrame.pack(side=TOP, pady=40)
    lbl_username = Label(RegisterFrame, text="Username:", font=('arial', 18), bd=18)
    lbl_username.grid(row=1)
    lbl_password = Label(RegisterFrame, text="Password:", font=('arial', 18), bd=18)
    lbl_password.grid(row=2)
    lbl_firstname = Label(RegisterFrame, text="Firstname:", font=('arial', 18), bd=18)
    lbl_firstname.grid(row=3)
    lbl_lastname = Label(RegisterFrame, text="Lastname:", font=('arial', 18), bd=18)
    lbl_lastname.grid(row=4)
    lbl_result2 = Label(RegisterFrame, text="", font=('arial', 18))
    lbl_result2.grid(row=5, columnspan=2)
    username = Entry(RegisterFrame, font=('arial', 20), textvariable=USERNAME, width=15)
```

```
    username.grid(row=1, column=1)
    password = Entry(RegisterFrame, font=('arial', 20), textvariable=PASSWORD, width=15,
show="*")
    password.grid(row=2, column=1)
    firstname = Entry(RegisterFrame, font=('arial', 20), textvariable=FIRSTNAME, width=15)
    firstname.grid(row=3, column=1)
    lastname = Entry(RegisterFrame, font=('arial', 20), textvariable=LASTNAME, width=15)
    lastname.grid(row=4, column=1)
    btn_login = Button(RegisterFrame, text="Register", font=('arial', 18), width=35,
command=Register)
    btn_login.grid(row=6, columnspan=2, pady=20)
    lbl_login = Label(RegisterFrame, text="Login", fg="Blue", font=('arial', 12))
    lbl_login.grid(row=0, sticky=W)
    lbl_login.bind('<Button-1>', ToggleToLogin)

def ToggleToLogin(event=None):
    RegisterFrame.destroy()
    LoginForm()

def ToggleToRegister(event=None):
    LoginFrame.destroy()
    RegisterForm()

def Register():
    Database()
    if USERNAME.get == "" or PASSWORD.get() == "" or FIRSTNAME.get() == "" or
LASTNAME.get == "":
        lbl_result2.config(text="Please complete the required field!", fg="orange")
    else:
        cursor.execute("SELECT * FROM `member` WHERE `username` = ?",
(USERNAME.get(),))
        if cursor.fetchone() is not None:
            lbl_result2.config(text="Username is already taken", fg="red")
```

```
        else:
            cursor.execute("INSERT INTO `member` (username, password, firstname, lastname)
VALUES(?, ?, ?, ?)", (str(USERNAME.get()), str(PASSWORD.get()),
str(FIRSTNAME.get()), str(LASTNAME.get())))
            conn.commit()
            USERNAME.set("")
            PASSWORD.set("")
            FIRSTNAME.set("")
            LASTNAME.set("")
            lbl_result2.config(text="Successfully Created!", fg="black")
        cursor.close()
        conn.close()
def Login():
    Database()
    if USERNAME.get == "" or PASSWORD.get() == "":
        lbl_result1.config(text="Please complete the required field!", fg="orange")
    else:
        cursor.execute("SELECT * FROM `member` WHERE `username` = ? and `password` =
?", (USERNAME.get(), PASSWORD.get()))
        if cursor.fetchone() is not None:
            lbl_result1.config(text="You Successfully Login", fg="blue")
            subprocess.Popen('python DrownDetect.py')
        else:
            lbl_result1.config(text="Invalid Username or password", fg="red")
LoginForm()

#========================================MENUBAR
WIDGETS==================================
menubar = Menu(root)
filemenu = Menu(menubar, tearoff=0)
filemenu.add_command(label="Exit", command=Exit)
menubar.add_cascade(label="File", menu=filemenu)
root.config(menu=menubar)
```

```
#=======================================INITIALIZATION============
======================
if __name__ == '__main__':
    root.mainloop()
```

7.2 FEATURE 3 : DETECT DROWNING

Algorithm:

1. Detect a object using yolo model
2. Check whether its person or not
3. If its a person,check if person moving or not
4. If not moving for 10 secs,alert the life guard

Code:

```
import cvlib as cv
from cvlib.object_detection import draw_bbox
import cv2
import time
import numpy as np
#for PiCamera
#from picamera Import PiCamera
#camera = PiCamera
#camera.start_preview()
# open webcam
webcam = cv2.VideoCapture(0)

if not webcam.isOpened():
    print("Could not open webcam")
    exit()




t0 = time.time() #gives time in seconds after 1970

#variable dcount stands for how many seconds the person has been standing still for
centre0 = np.zeros(2)
isDrowning = False
```

```
#this loop happens approximately every 1 second, so if a person doesn't move,
#or moves very little for 10seconds, we can say they are drowning

#loop through frames
while webcam.isOpened():

    # read frame from webcam
    status, frame = webcam.read()

    if not status:
        print("Could not read frame")
        exit()

    # apply object detection
    bbox, label, conf = cv.detect_common_objects(frame)
    #simplifying for only 1 person

    #s = (len(bbox), 2)

    if(len(bbox)>0):
            bbox0 = bbox[0]
            #centre = np.zeros(s)
            centre = [0,0]

            centre =[(bbox0[0]+bbox0[2])/2,(bbox0[1]+bbox0[3])/2 ]

            #make vertical and horizontal movement variables
            hmov = abs(centre[0]-centre0[0])
            vmov = abs(centre[1]-centre0[1])

            #there is still need to tweek the threshold
            #this threshold is for checking how much the centre has moved
```

```
        x=time.time()

        threshold = 10
        if(hmov>threshold or vmov>threshold):
            print(x-t0, 's')
            t0 = time.time()
            isDrowning = False

        else:

            print(x-t0, 's')
            if((time.time() - t0) > 10):
                isDrowning = True




        #print('bounding box: ', bbox, 'label: ' label ,'confidence: ' conf[0], 'centre: ', centre)
        #print(bbox,label ,conf, centre)
        print('bbox: ', bbox, 'centre:', centre, 'centre0:', centre0)
        print('Is he drowning: ', isDrowning)

        centre0 = centre
        # draw bounding box over detected objects

    out = draw_bbox(frame, bbox, label, conf,isDrowning)

    #print('Seconds since last epoch: ', time.time()-t0)

    # display output
    cv2.imshow("Real-time object detection", out)
```

```
    # press "Q" to stop
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# release resources
webcam.release()
cv2.destroyAllWindows()
```

CHAPTER 8

TESTING

8.1 TEST CASES

1. Login button click with wrong credentials entered.
2. Signup with already registered mail ID.
3. Signup with wrong form data entered.
4. Entering home page with logged out session.
5. Clicking home page buttons with logged out session.
6. Invalid data entered in change password page and requested for change in password.

7.

8.2 USER ACCEPTANCE TESTING

S.NO	TEST CASE	REQUIRED OUTPUT	RESULT OUTPUT	STATUS
1	Login button click with wrong credentials	Wrong credentials entered notification	Wrong credentials entered notification	ACCEPTED
2	Signup with already registered mail ID.	Email already registered notification	Email already registered notification	ACCEPTED
3	Signup with wrong form data entered.	Wrong credentials entered notification	Wrong credentials entered notification	ACCEPTED
4	Entering home page with logged out session.	Take user to login page	Take user to login page	ACCEPTED
5	Clicking home page buttons with logged out session.	Take user to login page	Take user to login page	ACCEPTED
6	Invalid data entered in change password page and requested for change in password.	Wrong form data entered notification	Wrong form data entered notification	ACCEPTED

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS

1. Planned value : Rs.4000
2. Actual value : Rs.1300
3. Hours worked : 50 hours
4. Stick to Timelines : 100%
5. Stay within budget : 100%
6. Consistency of the product : 75%
7. Efficiency of the product : 80%
8. Quality of the product : 80%

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES :

1. Low cost.
2. Simple UI.
3. Faster response
4. Capability of adding many features with ease and less cost.

DISADVANTAGES :

1. Lack of UI
2. Consistency of the product is not 100%.
3. Detecting person who is drowning is not accurate

CHAPTER 11
CONCLUSION

To overcome this lacking ability a camera detector can be made and fit in some areas.These camera can detect a drowning person and alert the guard.Help in easing Life Guard job and save many more lif

CHAPTER 12
FUTURE SCOPE

The product can include many other additional features like checking the weather forecast of the child location, interacting with the child etc. If we improve the efficiency of the code and reduce the size of our product , the market will be able to find a new child tracker gadget with low cost and high quality.

CHAPTER 13
APPENDIX

INTERNET OF THINGS

The **Internet of things** (**IoT**) describes physical objects (or groups of such objects) with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks. Internet of things has been considered a misnomer because devices do not need to be connected to the public internet, they only need to be connected to a network and be individually addressable. The field has evolved due to the convergence of multiple technologies, including ubiquitous computing, commodity sensors, increasingly powerful embedded systems, and machine learning. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), independently and collectively enable the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices

and appliances (such as lighting fixtures, thermostats, home security systems, cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. IoT is also used in healthcare systems.[10]

There are a number of concerns about the risks in the growth of IoT technologies and products, especially in the areas of privacy and security, and consequently, industry and governmental moves to address these concerns have begun, including the development of international and local standards, guidelines, and regulatory frameworks.

MQTT

MQTT is an OASIS standard messaging protocol for the Internet of Things (IoT). It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth. MQTT today is used in a wide variety of industries, such as automotive, manufacturing, telecommunications, oil and gas, etc.

NODE RED

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range of nodes in the palette. Flows can be then deployed to the runtime in a single-click. JavaScript functions can be created within the editor using a rich text editor. A built-in library allows to save useful functions, templates or flows for re-use.

IBM WATSON IOT PLATFORM

IBM Watson IoT Platform for Bluemix provides a versatile toolkit that includes gateway devices, device management, and powerful application access. By using Watson IoT Platform, you can collect connected device data and perform analytics on real-time data. The IBM Watson IoT Platform is a fully managed, Cloud-hosted service that provides device management capabilities as well as data

collection and management in a time series format. As part of IBM's Platform as a Service offering, IBM Bluemix, you can use the IBM Watson IoT Platform to rapidly build IoT apps from the catalog of services available in IBM Bluemix. You can choose from such IoT app options as storage services, rules, analytics services, stream analytics, machine learning, visualization, and user apps (Web or mobile). You also can embed cognitive capabilities in your IoT apps by using IBM Watson services available in IBM Bluemix

CLOUDANT

A fully managed, distributed database optimized for heavy workloads and fast-growing web and mobile apps, IBM Cloudant is available as an IBM Cloud® service with a 99.99% SLA. Cloudant elastically scales throughput and storage, and its API and replication protocols are compatible with Apache CouchDB for hybrid or multicloud architecture.

SOURCE CODE LINK : https://github.com/IBM-EPBL/IBM-Project-17392-1659665995
