A

Mini Project

On

# DetectDUI: AN IN-CAR DRINK DRIVE DETECTION SYSTEM

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING (AI&ML)

By

V.VIKRAM(207R1A66J0)

G.BHEMESHWAR (207R1A66E0)

A.ROHITH (207R1A66C5)

Under the Guidance of

**Mr.M.RAVINDRAN**

(Associate Professor)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)**

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

**2020-2024**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)**

# CERTIFICATE

This is to certify that the project entitled **"DetectDUI: AN IN-CAR DRINK DRIVE DETECTION SYSTEM"** being submitted by **V.VIKRAM(207R1A66J0), G.BHEMESHWAR (207R1A66E0) & A.ROHITH(207R1A66C5)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering (AI&ML) to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

**Mr.M.RAVINDRAN**
(Associate Professor)
INTERNAL GUIDE

**Dr. S Rao Chintalapudi**
HOD CSE(AI&ML)

**EXTERNAL EXAMINER**

**Submitted for viva voice Examination held on __________________**

# ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Mr.M.RAVINDRAN,** Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. G. Vinoda Reddy, Dr. K. Mahesh, Md. Hafeena & D. Babu Rao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. S Rao Chintalapudi,** Head, Department of Computer Science and Engineering (AI&ML) for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

**V.VIKRAM (207R1A66J0)**

**G. BHEMESHWAR (207R1A66E0)**

**A.ROHITH (207R1A66C5)**

# ABSTRACT

As one of the biggest contributors to road accidents and fatalities, drink driving is worthy of significant research attention. However, most existing systems on detecting or preventing drink driving either require special hardware or require much effort from the user, making these systems inapplicable to continuous drink driving monitoring in a real driving environment. In this paper, we present DetectDUI, a contactless, non-invasive, real-time system that yields a relatively highly accurate drink driving monitoring by combining vital signs (heart rate and respiration rate) extracted from in-car WiFi system and driver's psychomotor coordination through steering wheel operations.

The framework consists of a series of signal processing algorithms for extracting clean and informative vital signs and psychomotor coordination, and integrate the two data streams using a self-attention convolutional neural network (i.e., C-Attention). In safe laboratory experiments with 15 participants, DetectDUI achieves drink driving detection accuracy of 96.6% and BAC predictions with an average mean error of $2 \sim 5$mg/dl. These promising results provide a highly encouraging case for continued development.

# LIST OF FIGURES/TABLES

# LIST OF SCREENSHOTS

# TABLE OF CONTENTS

# TABLE OF CONTENTS

# 1. INTRODUCTION

# 1. INTRODUCTION

## 1.1 PROJECT SCOPE

This project is titled "DetectDUI: AN IN-CAR DRINK DRIVE DETECTION SYSTEM" can be supported by in-car IMU and WiFi systems.It allows continuous monitoring of drunkenness during driving without interfering the driving process.It is also non-invasive, real-time system that yields a relatively highly accurate drink driving monitoring by combining vital signs (heart rate and respiration rate).

## 1.2 PROJECT PURPOSE

In this Project , We develop a WiFi-based drunk driving detection system, which contactless vital sign monitoring, which is more suitable for the in-car environment. Compared with existing works on drunkenness detection, our system does not require extra devices, is privacy-preserving and can perform continuous detection without interfering the driving process. Wi-Fi is being deployed in many automobiles, such as Chevrolet Cruze, Chrysler 200, Dodge Dart, Audi A3, Ford Escape, and Mercedes-Benz GLS. Different from existing works on WiFi-based breath or heartbeat monitoring, we have developed a series of signal processing techniques to remove interference in the received signals to cater to the complicated in-car environment.

## 1.3 PROJECT FEATURES

The main features of this project are that this model classifies the It addresses the problem of learning hierarchical representations with a single algorithm or a few algorithms and We have proposed a series of signal processing algorithms for extracting human vital signs from WiFi signals given chest motions with high levels of accuracy. We have proposed to use C-Attention to combine the information of vital signs and psychomotor coordination to reach a well-round drink driving prediction.There are different signal processing approaches like Sudden Change Removal (SCR), conventional VMD algorithm and Adaptive Variational Mode Decomposition(AVMD). We compare the performance of various lightweight regression models, including Multilayer Perceptron (MLP), Support Vector Machine (SVM), and Random Forest (RF).

# 2. SYSTEM ANALYSIS

# 2. SYSTEM ANALYSIS

## SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, "what must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst hasa firm understanding of what is to be done.

## 2.1 PROBLEM DEFINITION

A general statement of Drink Drive Detection can be formulated as the given information of a driver,To identify whether the driver is drink or not .And also calculates Blood Alcohol Content (BAC) of a Driver.

## 2.2 EXISTING SYSTEM

### Drunkenness Detection

- **Hardware-Based Detection**: First used in the United Kingdom in the 1970s , breathalyzers are the world's most commonly used tools for testing inebriated drivers. Over its years of usage, researchers have connected breathalyzers, as well as other types of breath alcohol sensors, to smartphones via Bluetooth to improve BAC tracking, especially for self monitoring by drivers themselves. Example systems include: BACtrack Mobile Pro, Breathmeter . One major disadvantage of breathalyzers is that the results are highly susceptible to the oral environment and certain diseases (e.g., diabetes, liver and kidney diseases ), which may lead to false detection. Alternatives to breathalyzers include SCRAM, a transdermal sensor that measures the wearer's BAC through their sweat every 30 minutes . The same kind of system is available in a tight wristband that fits closely to the skin . However, SCRAM-based systems require a close contact between the skin and the sensor. Any space or anything between the skin and the sensor will affect the detection accuracy. Moreover, these systems require users to purchase extra devices or sensors, which may be expensive.

- **Camera-Based Detection:** Camera-based drunk driving systems have also been developed . Facial landmarks and motions are recognized in images to detect whether the driver is drunk driving or not. An audiovisual database is utilized to realize bimodal intoxication detection. However, camera-based approaches are sensitive to lighting conditions and there is potential risk of privacy violation .
- **Behavior-Based Detection:** The side effects of alcohol consumption include arrhythmia , slowed respiratory rates , impaired psychomotor performance , and unsteady gait. This abnormality in vital signs and behaviors can be leveraged to detect whether the user is under the influence of alcohol. Bae et al.developed a smart phone based system to track the drinking episodes of users based on built-in sensors (e.g., accelerometer) and the smartphone status (e.g., battery and network usage). Leveraging alcohol's influence on motor coordination and cognition, Markakis et al.designed five human-computer interactions to detect BACs (such as swiping or touching the screen in particular ways), akin to the finger-to-nose DUI tests. However, these works require users to interact with their phones (swipe the phone or engage in games), which interrupts the driving task and cannot offer a continuous drunk driving detection.

### 2.2.1 DISADVANTAGES OF EXISTING SYSTEM

Following are the disadvantages of existing system:

- An existing methodology doesn’t implement variational mode decomposition method.
- DetectDUI can't measure a person’s vital signs through WiFi signals and their psychomotor coordination through steering wheel operations.

## 2.3 PROPOSED SYSTEM

- As far as we are concerned, DetectDUI is the first contactless method of detecting drink driving, including measuring the driver’s BAC that can be administered while driving.
- We have proposed a series of signal processing algorithms for extracting human vital signs from WiFi signals given chest motions with high levels of accuracy.
- We have proposed to use C-Attention to combine the information of vital signs and psychomotor coordination to reach a well-round drink driving prediction.
- Extensive experiments on 15 individuals show DetectDUI is able to distinguish normal driving from drink driving in real-time with a 96.6%-accurate estimation and the driver’s BAC to within an MAE of 0.002% to 0.005%.

### 2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- The proposed system DetectDUI detects drink driving and predicts BAC through a driver’s vital signs and psychomotor coordination.The system shows the architecture of DetectDUI. In DetectDUI, vital signs are tracked through a WiFi sensing system and writing as datasets.
- The system proposes a novel adaptive variational mode decomposition (AVMD) method to separate the mixed signal into multiple modes, and then keep the modes that relate to breathing and heartbeat respectively.

## 2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposalis put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This isto ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis:

- EconomicFeasibility
- TechnicalFeasibility
- SocialFeasibility

### 2.4.1 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased. The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication that the system is economically possible for development.

### 2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### 2.4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

## 2.5 HARDWARE & SOFTWARE REQUIREMENTS

### 2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel Dual Core I5 and above
- Hard disk : 8GB and above
- RAM : 8GB and above
- Inputdevices:Keyboard,mouse.

### 2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system : Windows 8 and above.
- Languages : Python.
- Front-End : Python.
- Back-End : Django-ORM.
- Designing : Html, css, javascript.
- Data Base : MySQL (WAMP Server).

# 3. ARCHITECTURE

# 3. ARCHITECTURE

## 3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

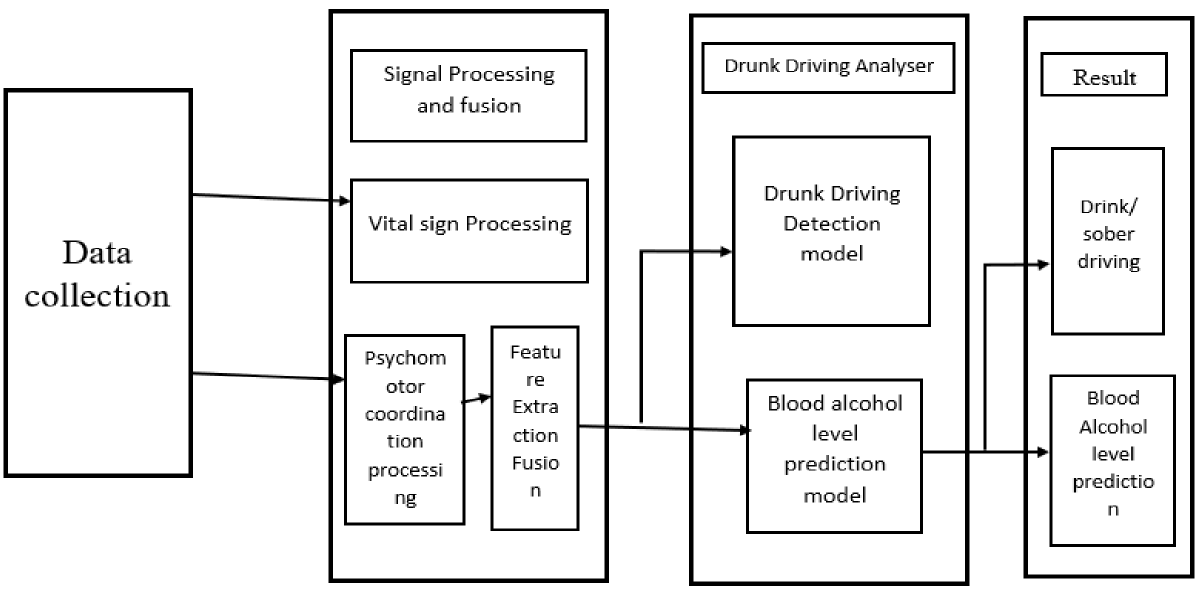



Figure 3.1: Project Architecture of DetectDUI: An In-Car Drink Drive Detection System.

## 3.2 DESCRIPTION

This project is totally based upon detecting whether Driver is Drunk or not.The model is built to combine Vital Signals(heart rate and respiration rate).The model is built with libraries like , django.db.models,django.shortcuts,sklearn, opencv, pandas, numpy etc. Each library is used for a specific purpose. Data is collected from dataset and performed some algorithms to combine vital signs and to get result.

### 3.3 USE CASE DIAGRAM

In the use case diagram, we have basically service provider and Remote user.

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of usersthe system has. The use cases are represented by either circles or ellipses. The provider and users are often shown as stick figures.

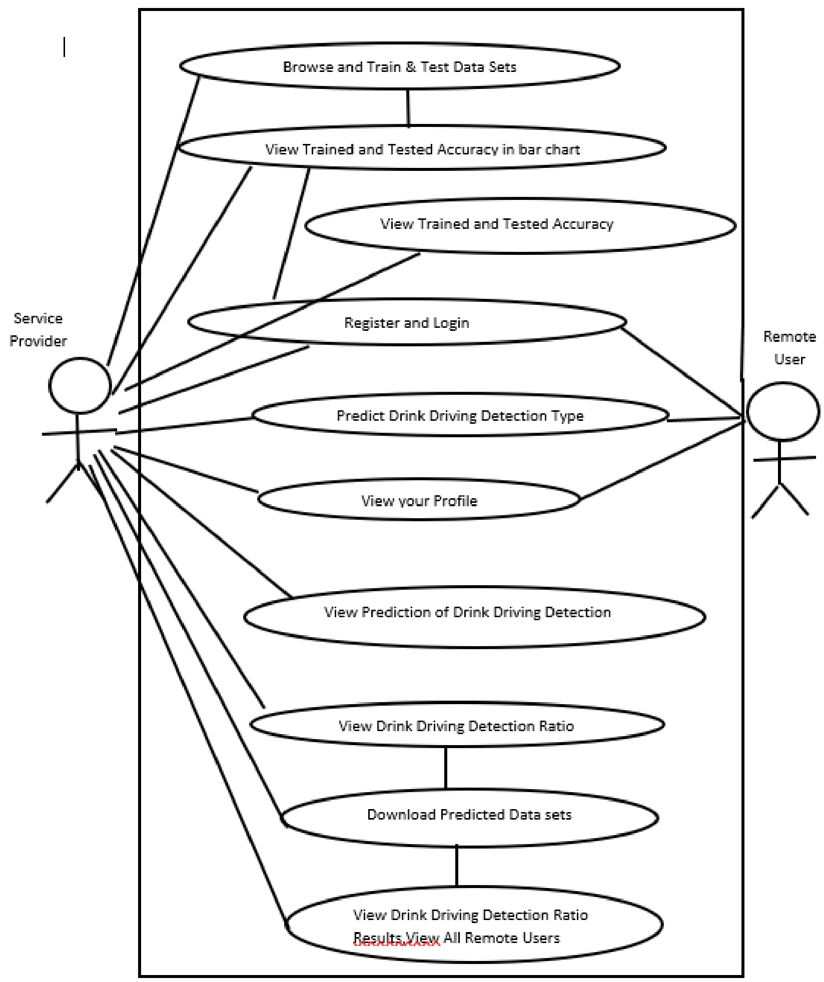



Figure 3.2: Use Case Diagram for DetectDUI: An In-Car Drink Drive Detection System.

## 3.4 CLASS DIAGRAM

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations(or methods), and the relationships among objects.

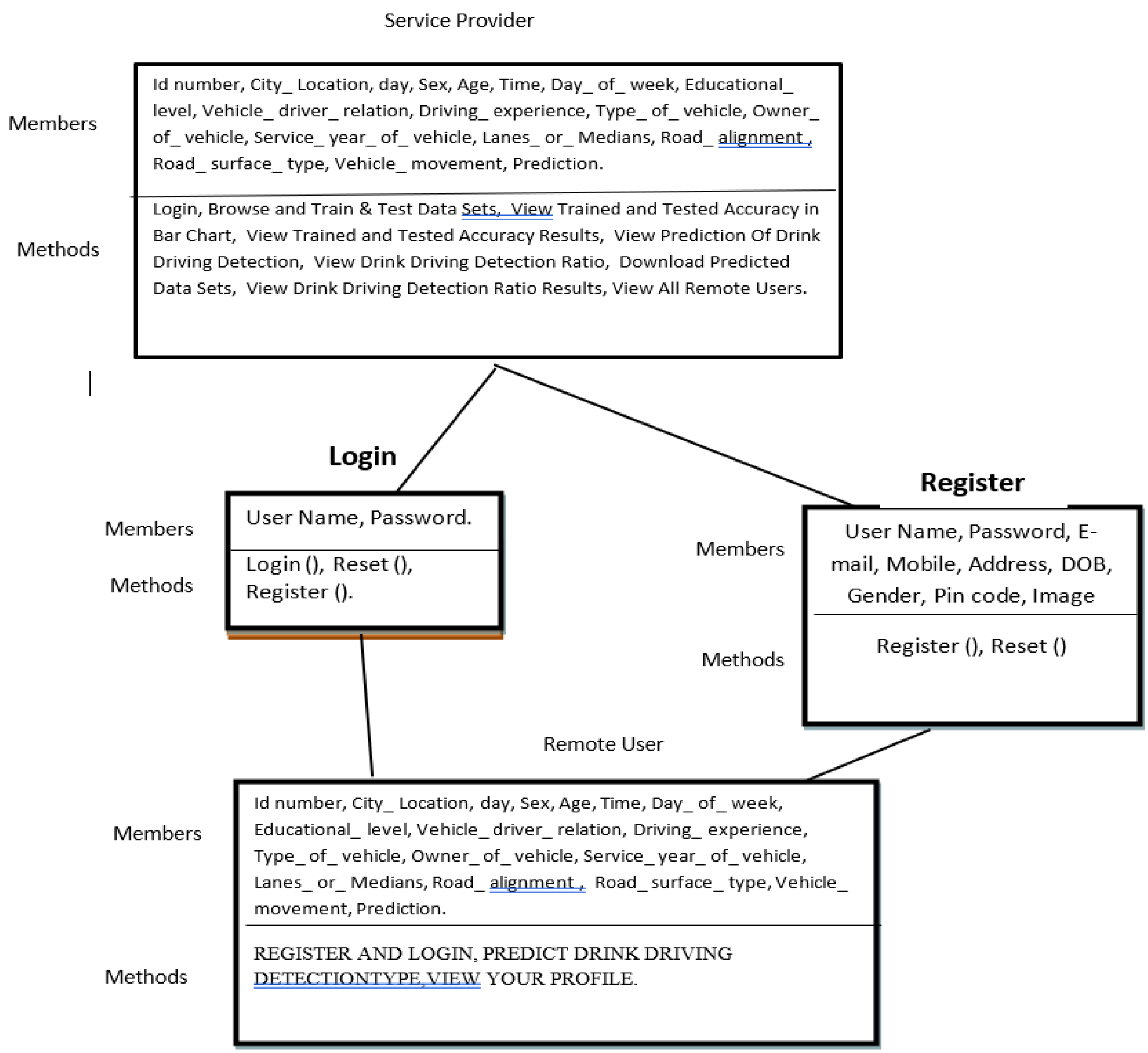



Figure 3.3: Class Diagram for DetectDUI: An In-Car Drink Drive Detection System.

## 3.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

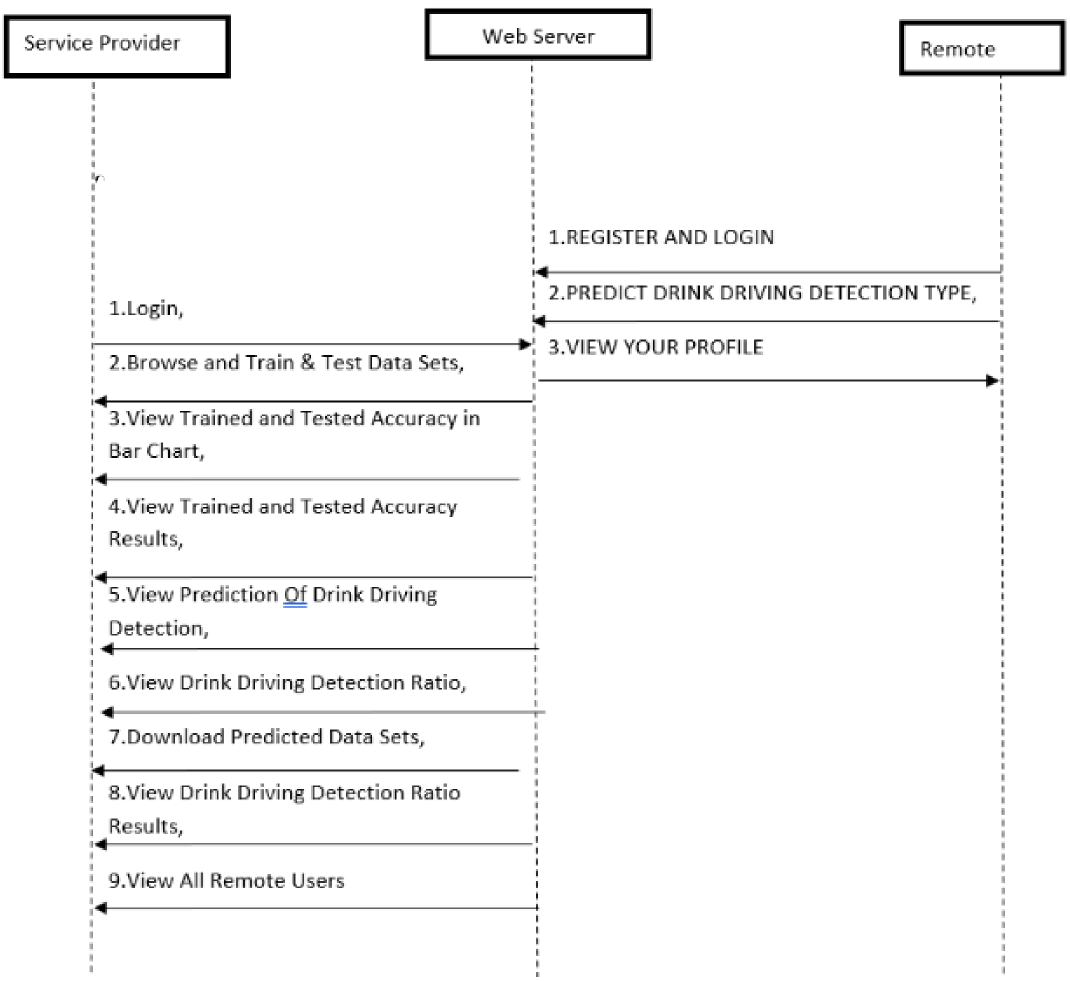



Figure 3.4: Sequence Diagram for DetectDUI: An In-Car Drink Drive Detection System.

## 3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data stores.

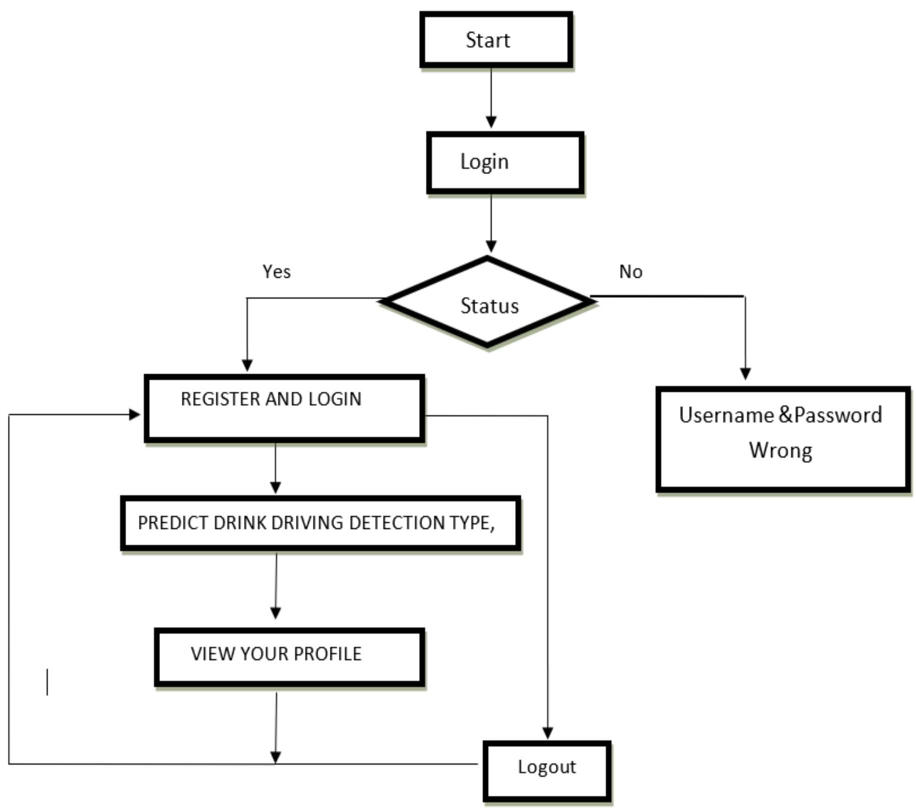



Figure 3.5: Activity Diagram for An In-Car Drink Drive Detection System

# 4. IMPLEMENTATION

## 4.1 SAMPLE CODE

```
from django.db.models import Count
from django.db.models import Q
from django.shortcuts import render, redirect, get_object_or_404

import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import VotingClassifier
# Create your views here.
from Remote_User.models import
ClientRegister_Model,drink_driving_detection,detection_ratio,detection_accuracy

def login(request):
    if request.method == "POST" and 'submit1' in request.POST:

        username = request.POST.get('username')
        password = request.POST.get('password')
        try:
        enter = ClientRegister_Model.objects.get(username=username,password=password)
            request.session["userid"] = enter.id

            return redirect('ViewYourProfile')
        except:
            pass
    return render(request,'RUser/login.html')
             def index(request):
    return render(request, 'RUser/index.html')

def Add_DataSet_Details(request):

    return render(request, 'RUser/Add_DataSet_Details.html', {"excel_data": ''})
```

```
def Register1(request):

    if request.method == "POST":
        username = request.POST.get('username')
        email = request.POST.get('email')
        password = request.POST.get('password')
        phoneno = request.POST.get('phoneno')
        country = request.POST.get('country')
        state = request.POST.get('state')
        city = request.POST.get('city')
        address = request.POST.get('address')
        gender = request.POST.get('gender')

ClientRegister_Model.objects.create(username=username,email=email,password=passw
ord,phoneno=phoneno,country=country,state=state,city=city,address=address,gender=ge
nder)

        obj = "Registered Successfully"
        return render(request, 'RUser/Register1.html',{'object':obj})
    else:
        return render(request,'RUser/Register1.html')

def ViewYourProfile(request):
    userid = request.session['userid']
    obj = ClientRegister_Model.objects.get(id= userid)
    return render(request,'RUser/ViewYourProfile.html',{'object':obj})



def Predict_Drink_Driving_Detection(request):
    if request.method == "POST":

        if request.method == "POST":

idnumber= request.POST.get('idnumber')
City_Location= request.POST.get('City_Location')
```

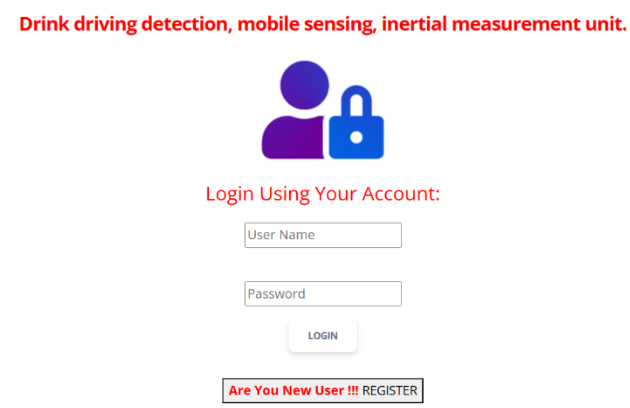
```
    day= request.POST.get('day')
    Sex= request.POST.get('Sex')
    Age= request.POST.get('Age')
    Time= request.POST.get('Time')
    Day_of_week= request.POST.get('Day_of_week')
    Educational_level= request.POST.get('Educational_level')
    Vehicle_driver_relation= request.POST.get('Vehicle_driver_relation')
    Driving_experience= request.POST.get('Driving_experience')
    Type_of_vehicle= request.POST.get('Type_of_vehicle')
    Owner_of_vehicle= request.POST.get('Owner_of_vehicle')
    Service_year_of_vehicle= request.POST.get('Service_year_of_vehicle')
    Lanes_or_Medians= request.POST.get('Lanes_or_Medians')
    Road_allignment= request.POST.get('Road_allignment')
    Road_surface_type= request.POST.get('Road_surface_type')
    Vehicle_movement= request.POST.get('Vehicle_movement')

     df = pd.read_csv('Driving_Datasets.csv', encoding='latin-1')

def apply_response(Label):
    if (Label == 0):
        return 0  # Not Detected
    elif (Label == 1):
        return 1  # Detected

df['results'] = df['Label'].apply(apply_response)

# cv = CountVectorizer()
x = df['idnumber'].apply(str)
y = df['results']

print("Review")
print(x)
print("Results")
print(y)

cv = CountVectorizer(lowercase=False, strip_accents='unicode', ngram_range=(1, 1))
```

```
# X = cv.fit_transform(df['Vehicle_movement'].apply(lambda x: np.str_(x)))
X = cv.fit_transform(x)
models = []
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
X_train.shape, X_test.shape, y_train.shape

print("Naive Bayes")

from sklearn.naive_bayes import MultinomialNB

NB = MultinomialNB()
NB.fit(X_train, y_train)
predict_nb = NB.predict(X_test)
naivebayes = accuracy_score(y_test, predict_nb) * 100
print("ACCURACY")
print(naivebayes)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, predict_nb))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, predict_nb))
models.append(('naive_bayes', NB))

# SVM Model
print("SVM")
from sklearn import svm

lin_clf = svm.LinearSVC()
lin_clf.fit(X_train, y_train)
predict_svm = lin_clf.predict(X_test)
svm_acc = accuracy_score(y_test, predict_svm) * 100
print("ACCURACY")
print(svm_acc)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, predict_svm))
```

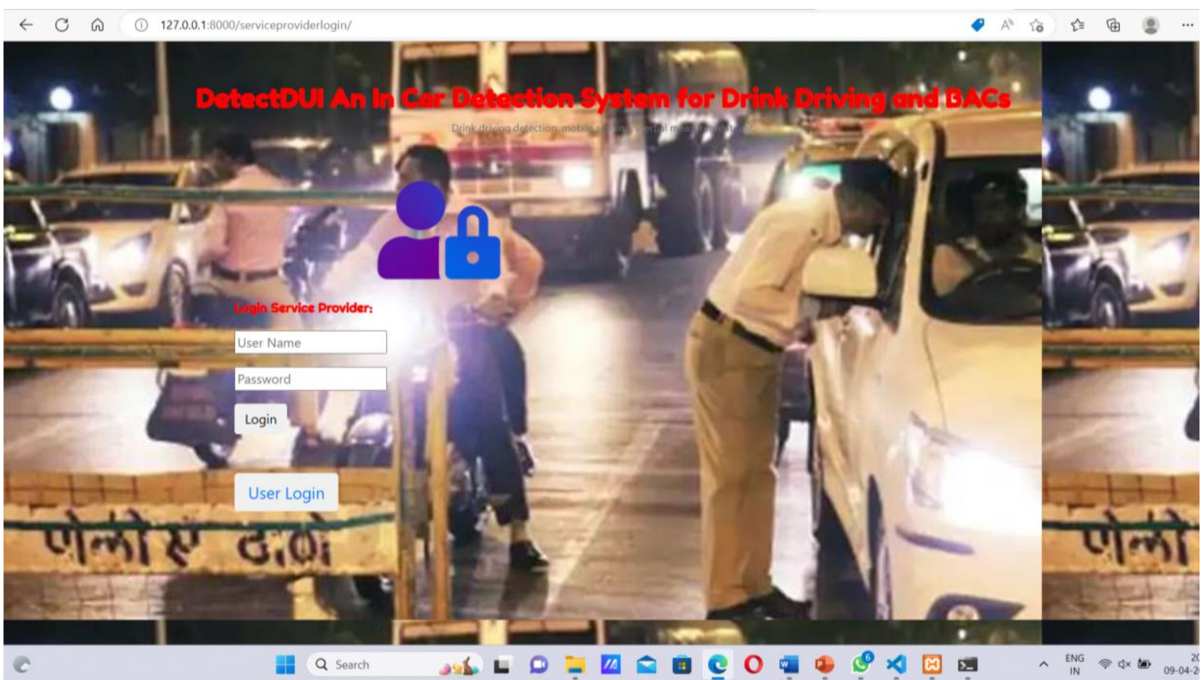
```
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, predict_svm))
models.append(('svm', lin_clf))
print("Logistic Regression")
from sklearn.linear_model import LogisticRegression
reg = LogisticRegression(random_state=0, solver='lbfgs').fit(X_train, y_train)
y_pred = reg.predict(X_test)
print("ACCURACY")
print(accuracy_score(y_test, y_pred) * 100)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, y_pred))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, y_pred))
models.append(('logistic', reg))

print("Decision Tree Classifier")
dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)
dtcpredict = dtc.predict(X_test)
print("ACCURACY")
print(accuracy_score(y_test, dtcpredict) * 100)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, dtcpredict))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, dtcpredict))
models.append(('DecisionTreeClassifier', dtc))

classifier = VotingClassifier(models)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)

idnumber1 = [idnumber]
vector1 = cv.transform(idnumber1).toarray()

predict_text = classifier.predict(vector1)
pred = str(predict_text).replace("[", "")
```

```
        pred1 = pred.replace("]", "")
        prediction = int(pred1)
        if (prediction == 0):
            val = 'No Drink Driving Detection'
        elif (prediction == 1):
            val = 'Drink Driving Detection'

        print(val)
        print(pred1)
        drink_driving_detection.objects.create(idnumber=idnumber,
        City_Location=City_Location,
        day=day,
        Sex=Sex,
        Age=Age,
        Time=Time,
        Day_of_week=Day_of_week,
        Educational_level=Educational_level,
        Vehicle_driver_relation=Vehicle_driver_relation,
        Driving_experience=Driving_experience,
        Type_of_vehicle=Type_of_vehicle,
        Owner_of_vehicle=Owner_of_vehicle,
        Ser_year_of_veh=Service_year_of_vehicle,
        Lanes_or_Medians=Lanes_or_Medians,
        Road_allignment=Road_allignment,
        Road_surface_type=Road_surface_type,
        Vehicle_movement=Vehicle_movement,
        Prediction=val)
        return render(request, 'RUser/Predict_Drink_Driving_Detection.html',{'objs': val})
    return render(request, 'RUser/Predict_Drink_Driving_Detection.html')
```

# 5. SCREENSHOTS




Screenshot 5.1: Remote User.




Screenshot 5.2: Service Provider.

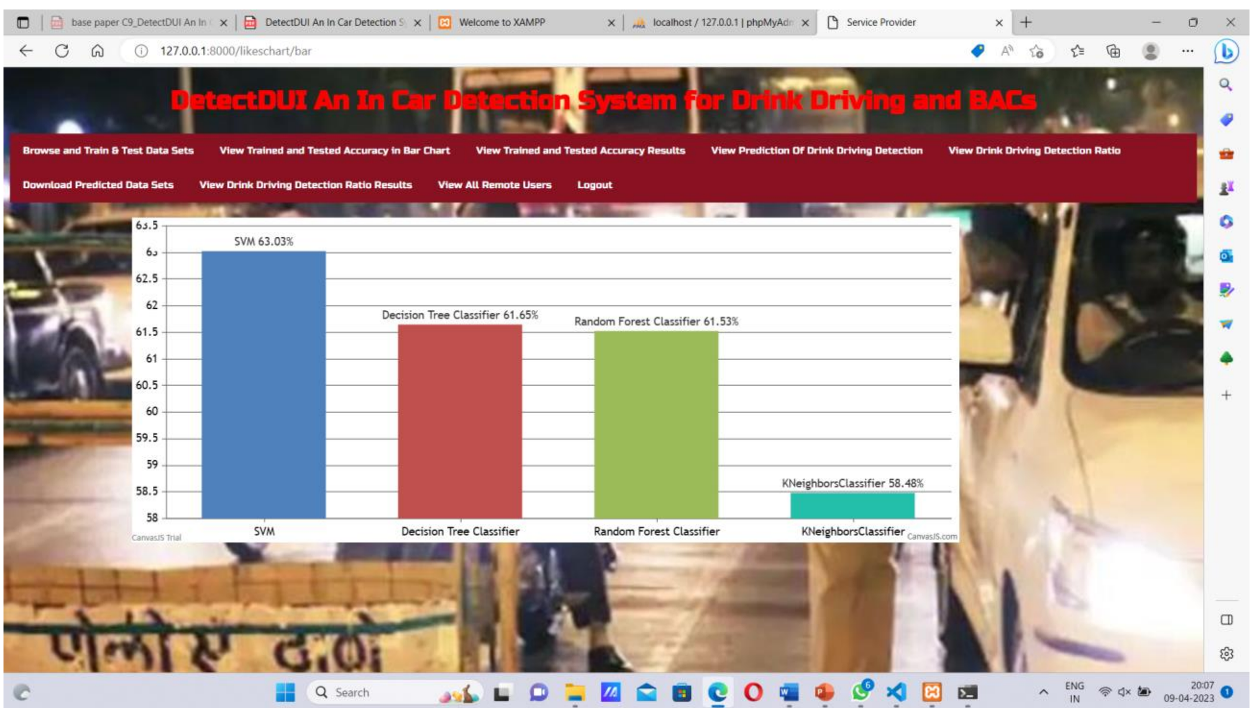
View Drink Deriving Detection Type Details !!!

| Idnumber | City_Location | day | Sex | Age | Time | Day_of_week | Educational_level | Vehicle_driver_relation | Driving_experience | Type_of_ve |
|---|---|---|---|---|---|---|---|---|---|---|
| y15g1yak | Bengaluru | 04-01-20 | M | 56 | 23:25:00 | Wednesday | Elementary school | Owner | 2-5yr | Stationwa |
| n026zwp8 | Singapore | 19-12-19 | M | 52 | 13:45:00 | Wednesday | Junior high school | Employee | 2-5yr | Lorry (41?1 |
| rn8mtrvc | New Delhi | 02-01-20 | M | 63 | 15:55:00 | Sunday | Above high school | Owner | 2-5yr | Pick up up 100 |
| ge9cuqz5 | Delhi | 25-02-21 | M | 63 | 8:10:00 | Wednesday | Elementary school | Employee | Below 1yr | Public (> 4 seats) |

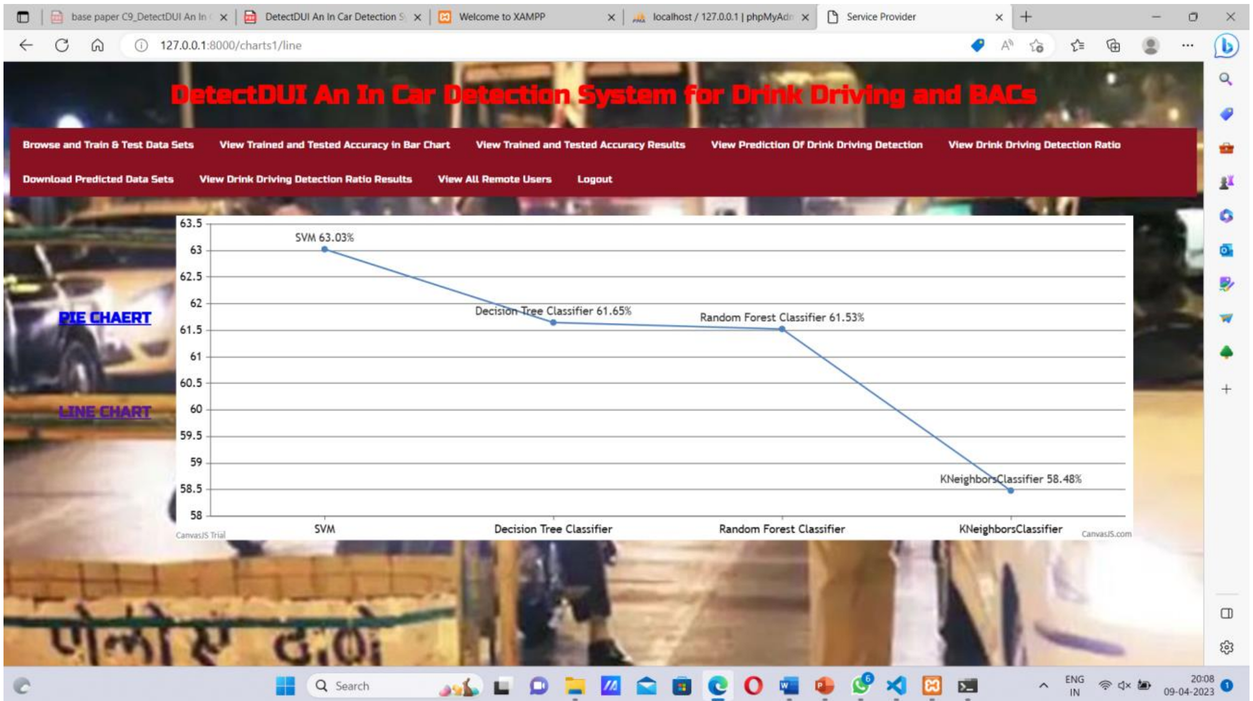
View Drink Deriving Detection Type Details !!!

| ehicle | Owner_of_vehicle | Service_year_of_vehicle | Lanes_or_Medians | Road_allignment | Road_surface_type | Vehicle_movement | Predictio |
|---|---|---|---|---|---|---|---|
| gen | Owner | NA | other | Tangent road with flat terrain | Asphalt roads | Going straight | No Drink Driving Detection |
| 100Q) | Owner | 2-5yrs | Two-way (divided with broken lines road marking) | Escarpments | Gravel roads | Going straight | Drink Driving Detection |
| ito | Owner | Unknown | Undivided Two way | Tangent road with flat terrain | Asphalt roads | Going straight | No Drink Driving Detection |
| 15 | Governmental | 1-2yr | Two-way (divided with broken lines road marking) | Tangent road with flat terrain | Asphalt roads | Moving Backward | Drink Driving Detection |

Screenshot 5.3: Details of Drivers and Prediction.




Screenshot 5.4: Trained and Tested Accuracy in Bar Chart.




Screenshot 5.5: View Trained and Tested
Accuracy Results.

# 6. TESTING

# 6. TESTING

## 6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

## 6.2 TYPES OF TESTING

### 6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### 6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### 6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:

| | |
|---|---|
| Valid Input | : identified classes of valid input must be accepted. |
| Invalid Input | : identified classes of invalid input must be rejected. |
| Functions | : identified functions must be exercised. |
| Output | : identified classes of application outputs must be exercised. |

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

## 6.3 TEST CASES

### 6.3.1 CLASSIFICATION

| Test case ID | Test case name | Purpose | Input | Output |
|---|---|---|---|---|
| 1 | Drink Drive Detection | To detect Whether the Driver is Drink or not. | The user gives the input as Vital signals. | An output is Whether the Driver is Drink or not. |
| 2 | Drink Drive Detection | To detect Whether the Driver is Drink or not. | The user gives input as psychomotor Coordination and vital signals. | An output is Whether the Driver is Drink or not. |

# 7. CONCLUSION

# 7. CONCLUSION & FUTURE SCOPE

## 7.1 PROJECT CONCLUSION

In this paper, we presented Detect DUI, a non-intrusive, contactless, and continuous system of measuring and monitoring the side effects of alcohol on drivers. To develop Detect DUI to this stage, we have overcome two main challenges. The first is to eliminate interference in the WiFi signals caused by the motions of a moving vehicle. This problem was solved with a series of signal processing algorithms. The second is determining which specific features of alcohol's side effects best reflect driving under the influence of alcohol. We have addressed this challenge with a C-Attention network. The results of extensive experiments confirm that Detect *DUI* provides highly accurate drink driving detection and BAC prediction.

Apart from drinking alcohols, other factors may also affect vital signs and psychomotor coordination, e.g., catching a cold or other respiratory diseases. Respiratory diseases will change breathing patterns, which are expected to be different from the breathing patterns of drinking. However, it is difficult to collect training samples to help differentiate the breathing patterns under the two conditions. In the future, we intend to refine our drink driving detection model by considering other impact factors.

## 7.2 FUTURE SCOPE

The potential idea behind the idea of the project is the Integration with autonomous vehicles to ensure the passengers are not drink and driving, the future enhancement would be integrating with mobile devices. From this project the advancement to the Sensor technology can be a great use. The model can also integrate with Ride-sharing services and can Expansion to other countries.

# 8. BIBLIOGRAPHY

# 8. BIBLIOGRAPHY

## 8.1 REFERENCES

[1]C.-W. You et al., “Enabling personal alcohol tracking using transdermal sensing wristbands: Benefits and challenges,” in Proc. 21st Int. Conf. Hum.-Comput. Interact. Mobile Devices Services, Oct. 2019, pp. 1–6.

[2] Y. Jung, J. Kim, O. Awofeso, H. Kim, F. Regnier, and E. Bae, “Smartphone-based colorimetric analysis for detection of saliva alcohol concentration,” Appl. Opt., vol. 54, no. 31, pp. 9183–9189, 2015

[3] H.-L. Kao, B.-J. Ho, A. C. Lin, and H.-H. Chu, “Phone-based gait analysis to detect alcohol usage,” in Proc. ACM Conf. Ubiquitous Comput. (UbiComp), 2012, pp. 661–662.

[4] S. Bae et al., “Detecting drinking episodes in young adults using smartphone-based sensors,” Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol., vol. 1, no. 2, pp. 1–36, Jun. 2017.

[5]A. Mariakakis, S. Parsi, S. N. Patel, and J. O. Wobbrock, “Drunk user interfaces: Determining blood alcohol level through everyday smartphone tasks,” in Proc. CHI Conf. Hum. Factors Comput. Syst., Apr. 2018, pp. 1–13.

[6] H.-L. Kao, B.-J. Ho, A. C. Lin, and H.-H. Chu, “Phone-based gait analysis to detect alcohol usage,” in Proc. ACM Conf. Ubiquitous Comput. (UbiComp), 2012, pp. 661–662.

[7] C. Wu, K. Tsang, H. Chi, and F. Hung, “A precise drunk driving detection using weighted kernel based on electrocardiogram,” Sensors, vol. 16, no. 5, p. 659, May 2016.

[8] S. Brunner et al., “Alcohol consumption, sinus tachycardia, and cardiac arrhythmias at the munich octoberfest: Results from the munich beer related electrocardiogram workup study (MunichBREW),” Eur. Heart J., vol. 38, no. 27, pp. 2100–2106, Jul. 2017.

## 8.2 GITHUB LINK

vupparapallivikram/detectDUI (github.com)