DDD

Drivers Drowsiness Detection Project

Supervisor: Dr. Azza Abd-Elrahman

Members : (AI subgroup)

-Ahmed Mohamed Gaber. (leader) -Kareem Gamal Mahmoud Mohamed.

Members : (Backend subgroup)

- Ghareeb Sayed Ghareeb - Abd Elrahman Salah

Member : (Frontend subgroup)- Yehia Ali

Acknowledgement:

First, we would like to express our gratitude to all our families, professors, and colleagues.

We would like to thank Dr/ Azza Abd Elrahman. If it wasn’t for her endless support, and patience we wouldn’t have been here today, just few weeks away from graduation.

Contents

Chapter 1: Introduction

- Drivers Drowsiness Detection Project It is an idea that works to reduce traffic accidents by knowing the driver’s drowsiness, using mobile and other mistakes by giving him a warning message to avoid it.
- And also tracking system to know where and when these errors occur to make it easier for organizations to take actions and track the car to know where those mistakes are happened.

- <u>Benefits of DDD app in the real life :</u>
    - Using this application saves the lives of the drivers who use this application and also preserves the lives of the passengers and the lives of drivers who walk the same way with the drivers who use the application.
    - Because sometimes the accident caused by one of the drivers causes accidents to other drivers.
    - This application reduces the spread of accidents on the roads and limits drivers' recklessness of some driving rules.

Technology on which software project carried out and Used Tools Software:

- **Back-End:**
  - **Django-framework :** is a web application framework with expressive, elegant syntax.
  - **Python :** is a programming language that allows web developers to create dynamic content that interacts with databases.
- **Front-End:**
  - **HTML**: HTML is the most basic building block of the Web. It defines the meaning and structure of web content.
  - **CSS**: CSS is a simple mechanism for adding style (e.g., fonts, colors, spacing) to web documents.
  - **JS** : is the programming language of the Web, to program the behavior of web pages.
- **AI Models :**
  - **Python :** is a programming language that allows web developers to create dynamic content that interacts with databases.
  - **Dlib :** is an open source C++ library implementing a variety of machine learning algorithms, including classification, regression, clustering, data transformation, and structured prediction.
  - **Mediapipe :** is a framework for building cross platform multimodal applied ML pipelines that consist of fast ML inference, classic computer vision, and media processing (e.g. video decoding)

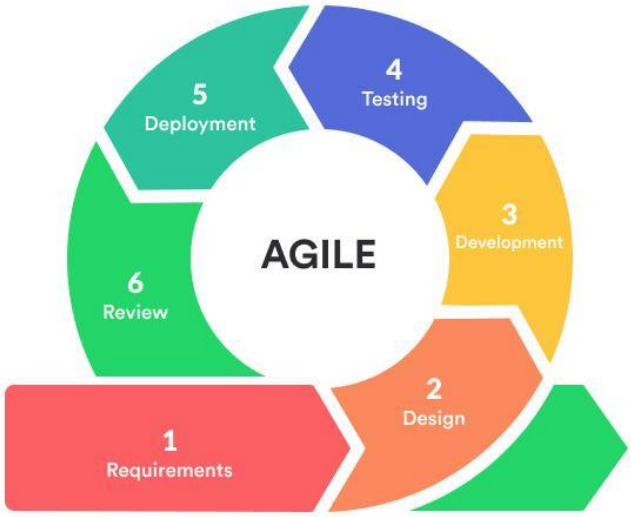
- **Open Cv :** is a library of **Python** bindings designed to solve computer vision problems.
- **Tensorflow :** is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.
- **Keras :** is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs.
- **Playsound :** aims to be a "pure Python, cross-platform, single function module with no dependencies for playing sounds." In other words, if all you need to do is play a sound file, this will get it done. In fact, you can play a sound with a single line of code.
- **Visual studio code:** is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).
- **Draw.io:** is free online diagram software. You can use it as a flowchart maker, network diagram software, to create UML online, as an ER diagram

tool, to design database schema, to build BPMN online, as a circuit diagram maker, and more.

- **ClickUp:** is a project management tool that aims to make your business life easier.

Process model adapted for software project development

our project follows the Agile Model, and this is shown below:



- Agile project management is an iterative development methodology that values human communication and feedback, adapting to change, and producing working results.
- Agile is iterative, meaning that it is done in pieces (sprints), with each sprint building and improving off the lessons from the previous sprint.

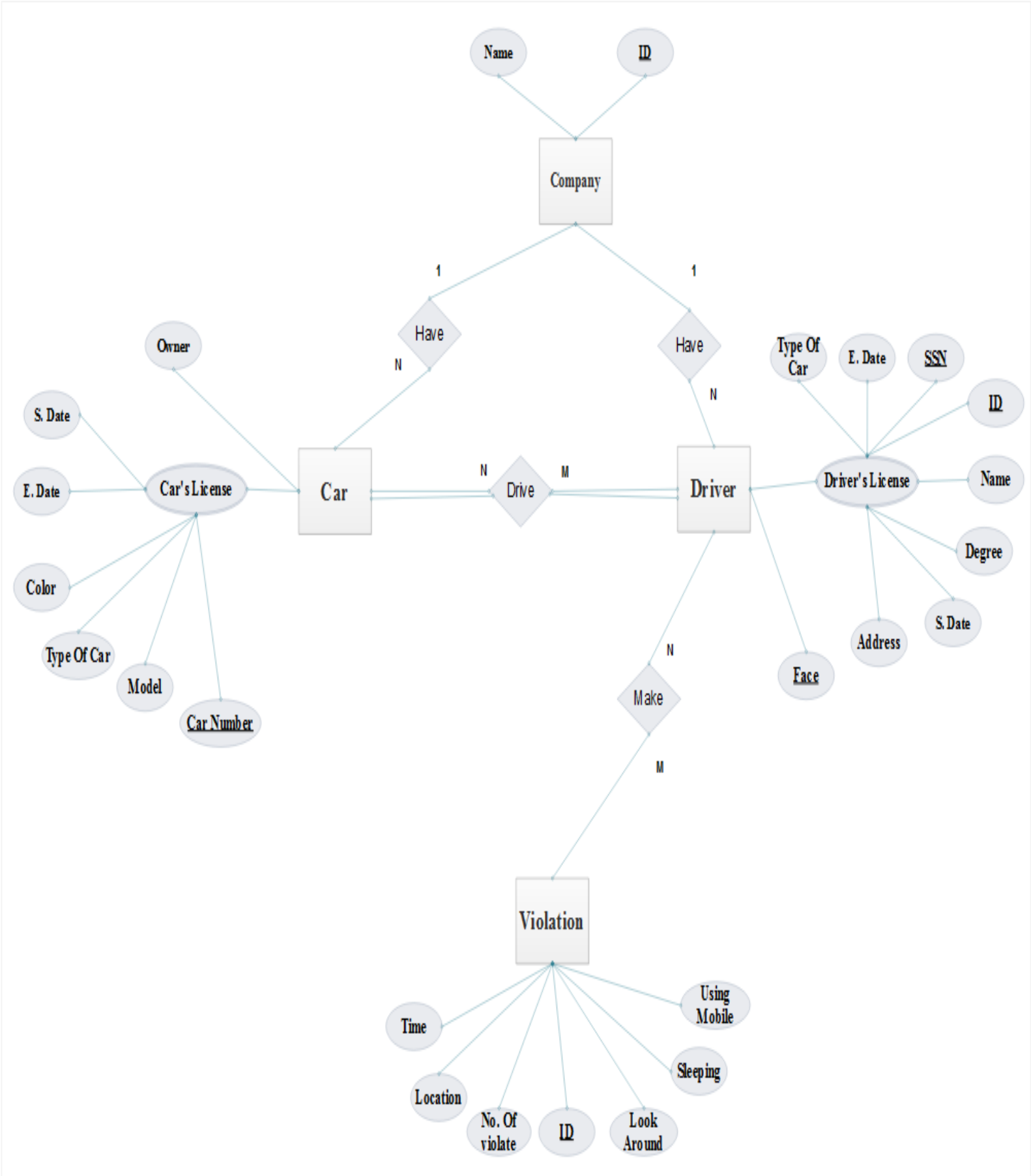
Chapter 2: Project Analysis and Design

Database Analysis :

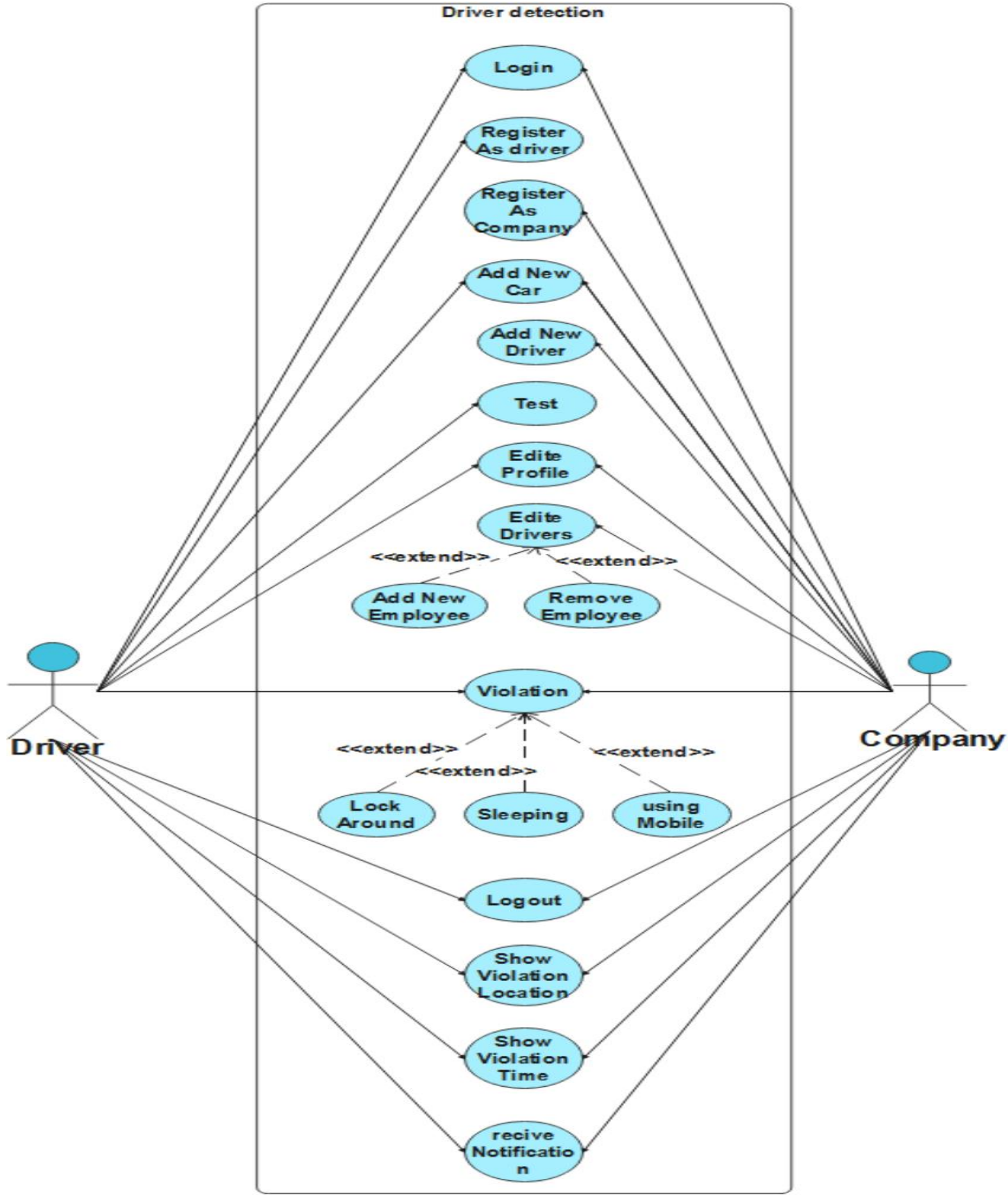
Our relational database consists of 4 entity described as follows:

1) **The driver entity :** have some attributes first attribute for the driver's license contains Ssn , id of license , name of driver , address , type of car , start and the end date of license of driver , and the second attribute for face of driver to know that he is the license owner.
2) **The car entity :** have some attributes first attribute for the car's license contains model of car , number of cars , color of car , types of car as primary key , start and the end date of license of car , and the second attribute for owner of the car which may be a person or a company.
3) **The violations entity :** have some attributes the time of mistake , location of mistake , number of violate , id as primary key for the types of mistakes to know differentiate between them , we have three types of violations
   1. Sleeping
   2. Using mobile
   3. Looking around
4) The company entity , have two attributes which are the name of the company , 'ID' of the company and 'ID' as primary key for the company and have two relations. The first one is the relation between company and the car which each company have many cars.
   The second relation between the company and the Driver which each company have many Drivers.

- **EER Diagram:**

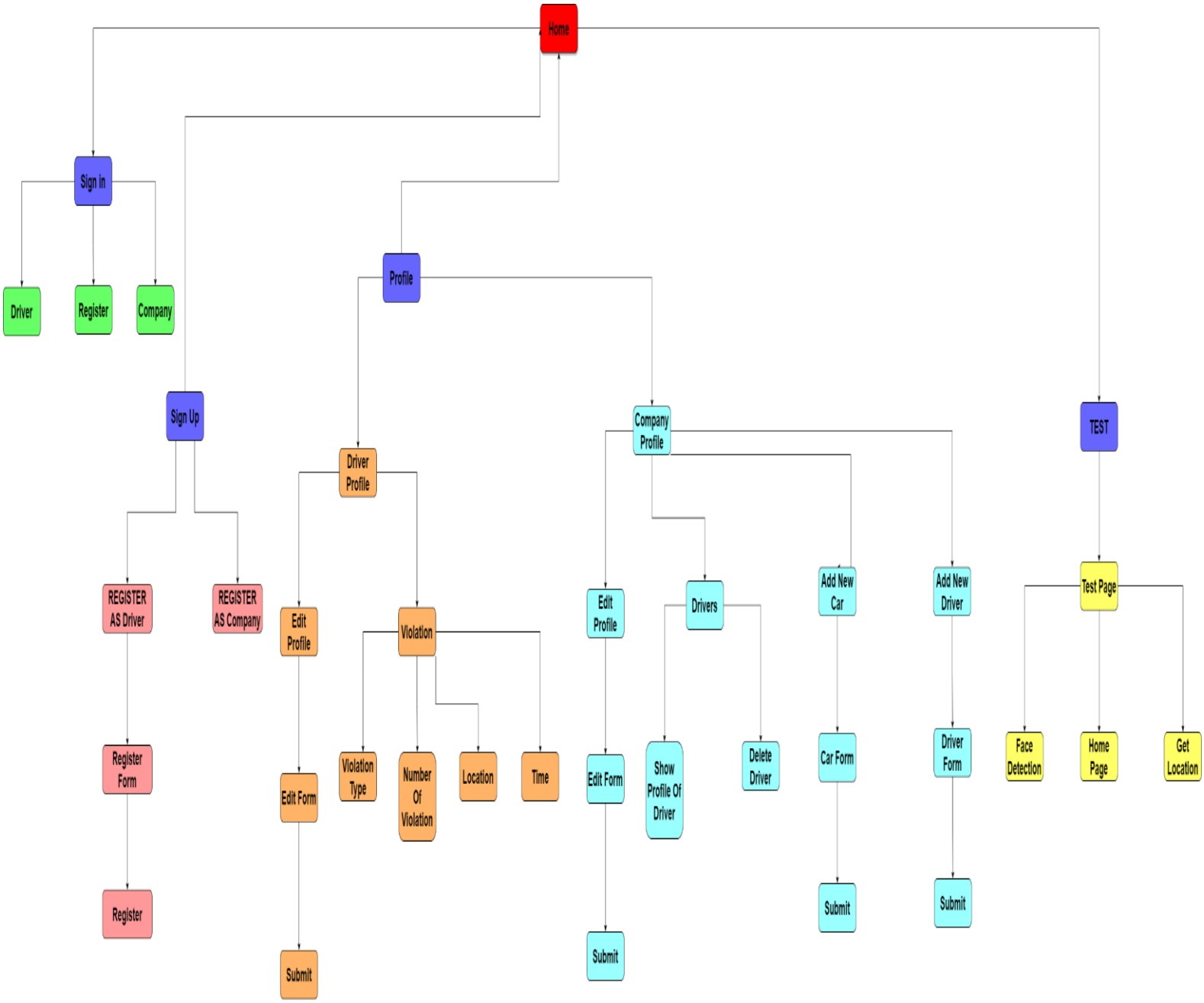

Name
ID
Company
1
1
Have
Have
N
N
Owner
S. Date
E. Date
Car's License
Car
N
Drive
M
Driver
Type Of Car
E. Date
SSN
ID
Driver's License
Name
Degree
S. Date
Address
Face
Color
Type Of Car
Model
Car Number
N
Make
M
Violation
Time
Location
No. Of violate
ID
Look Around
Sleeping
Using Mobile

- **Use Case:**

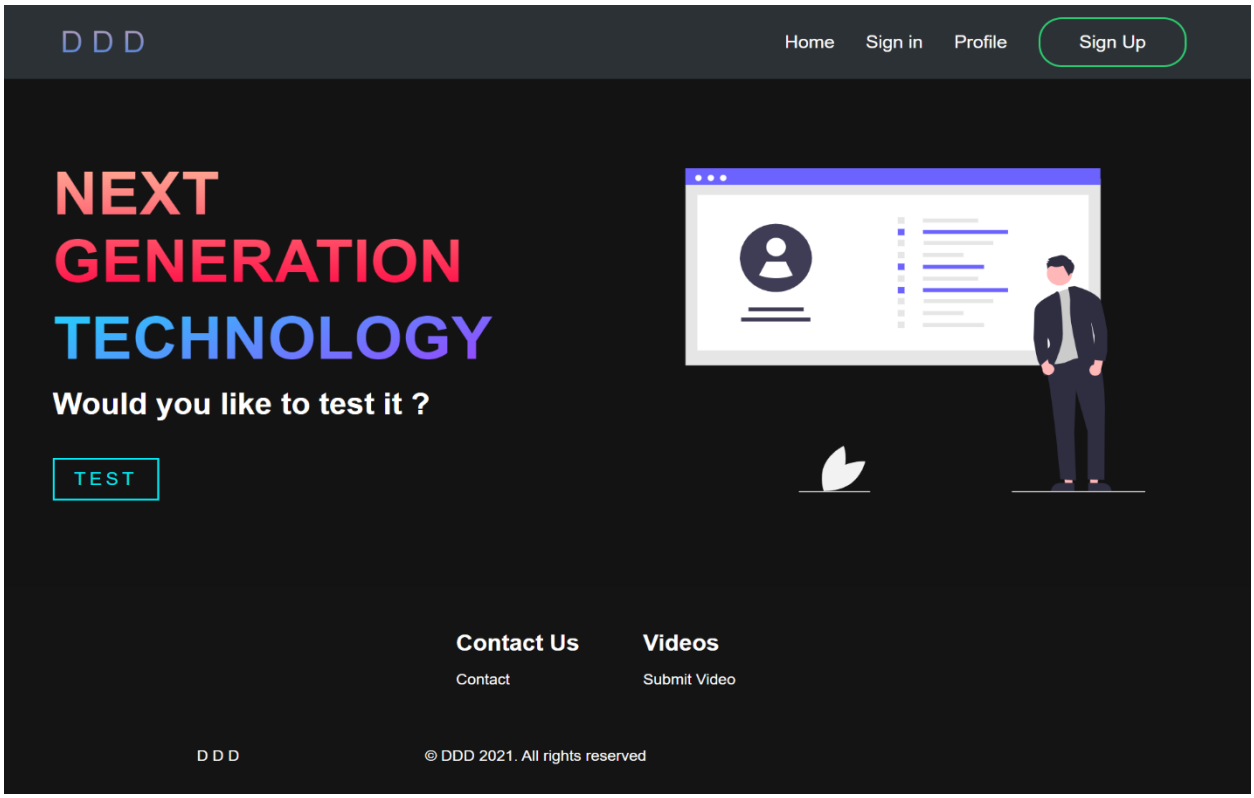

Driver detection
Login
Register As driver
Register As Company
Add New Car
Add New Driver
Test
Edite Profile
Edite Drivers
<<extend>>
<<extend>>
Add New Employee
Remove Employee
Violation
<<extend>>
<<extend>>
<<extend>>
Lock Around
Sleeping
using Mobile
Logout
Show Violation Location
Show Violation Time
recive Notificatio n
Driver
Company

Chapter3: User Interface and Screen Layouts

➢ **First, we will show the Web Map Design of the system.**

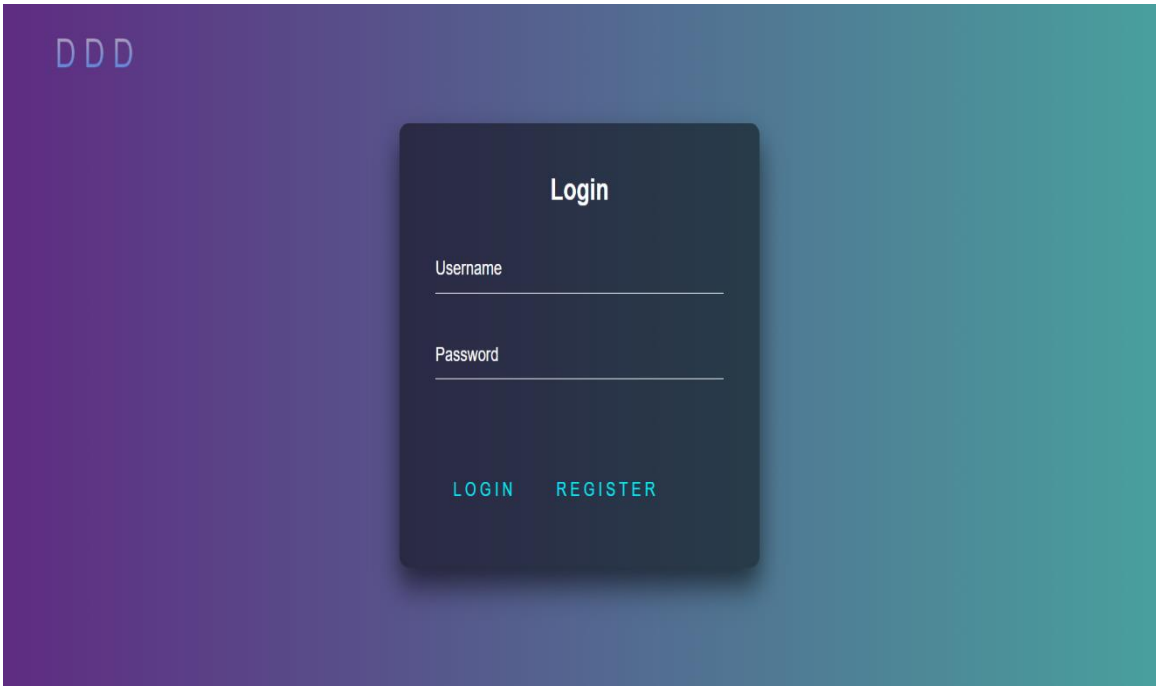


- In index.html we make a Three Sections are called Nave bar , Home and footer. (check source code)
- At the beginning of the system, it shows this Screen.

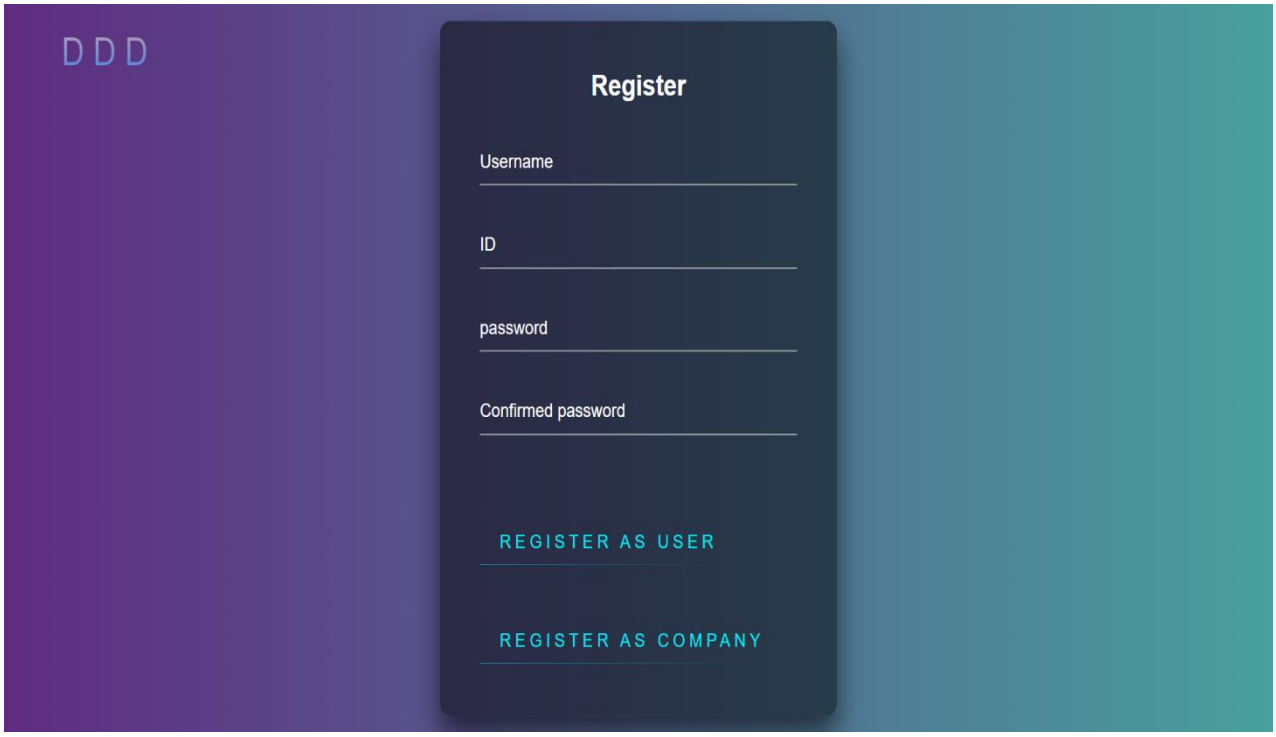


- Then click login to sign in the web application,
- Or sign up to web application if you don't have account in app.

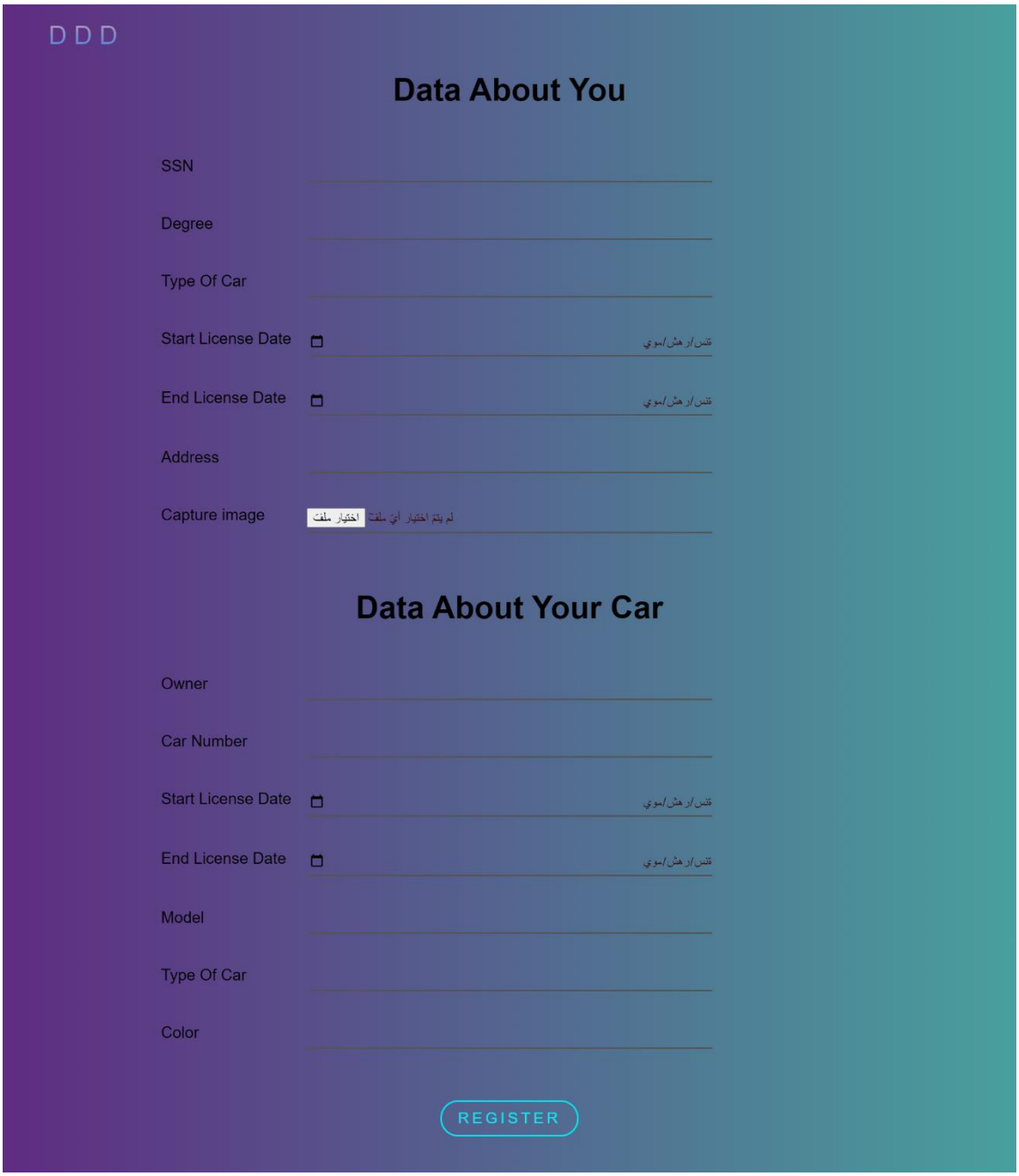
- **In Signin.html made a form for driver login. (check source code)**



- **In Signup.html made a form for driver Register and company Register. (check source code)**

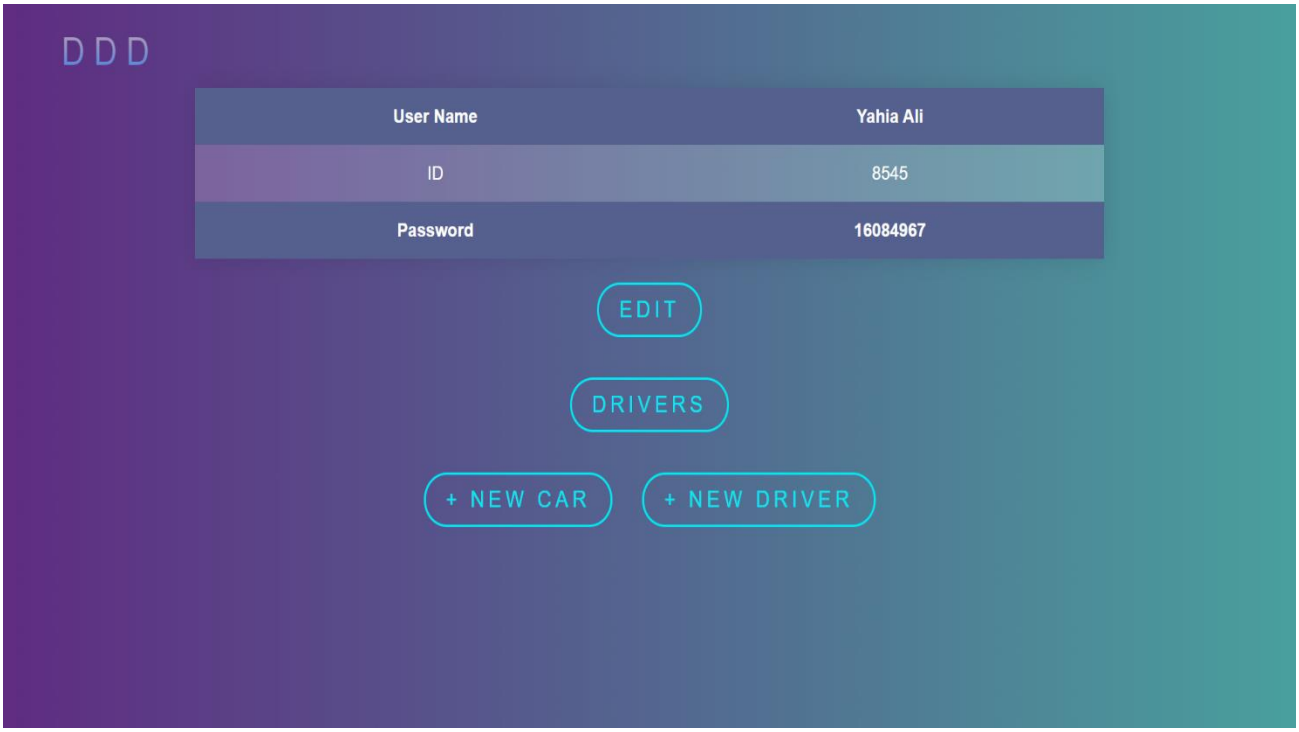


- **register as user ui**



- **In profile.html made a table for drivers and four button for company control. (check source code)**

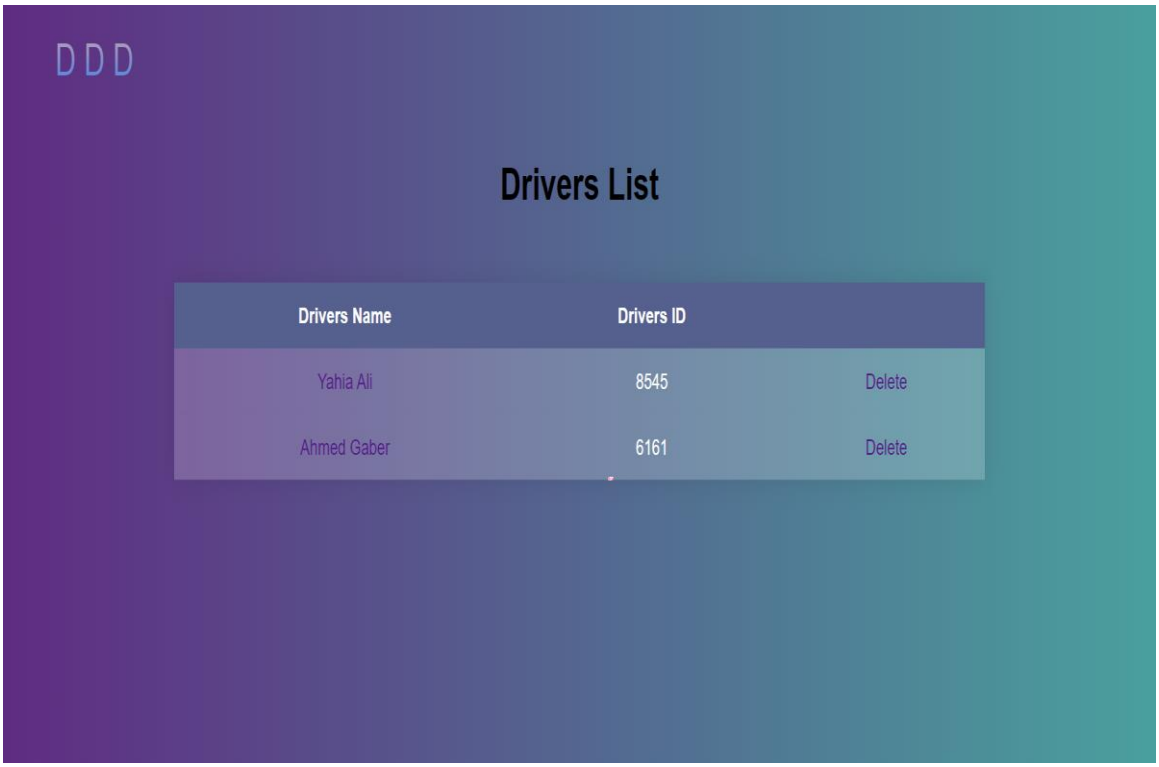
- **Profile Of Company UI**



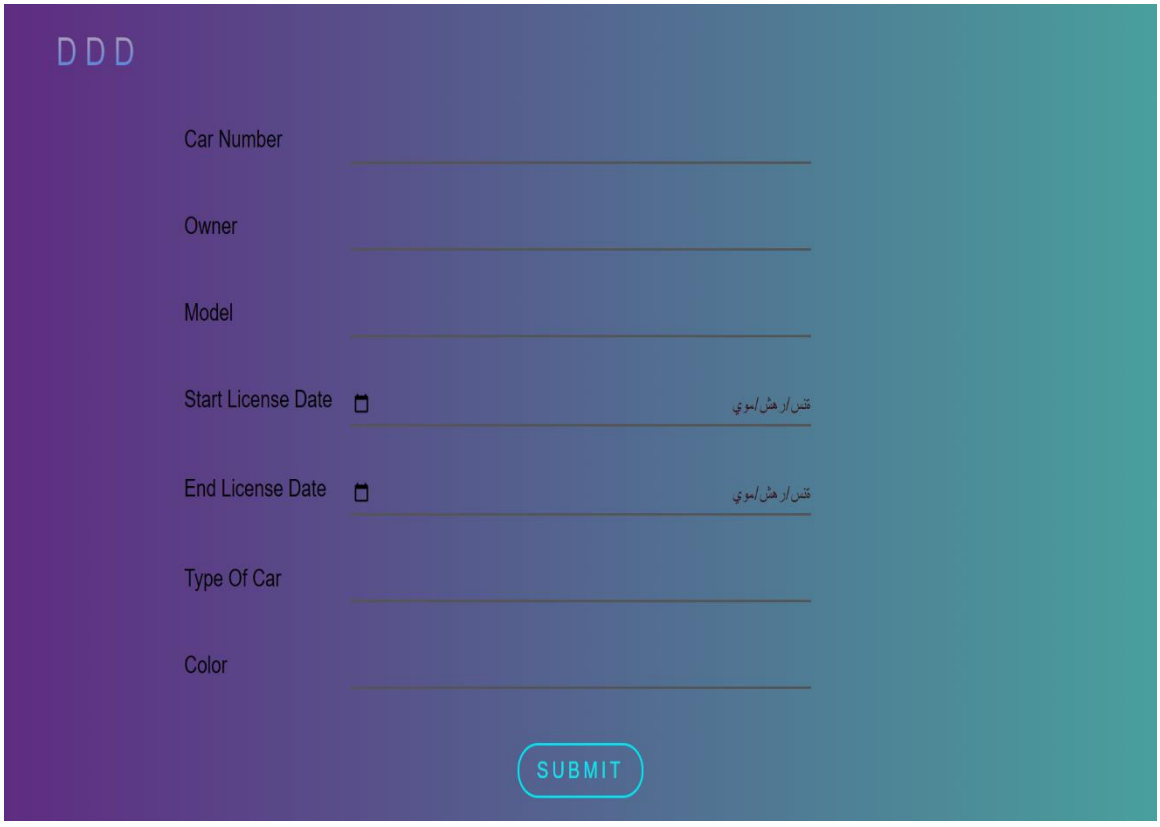
In profile of company there is a display of company data and there is button for edit profile , button for show drivers work in this company , button to add new car and button to add new driver for this company.

Edit company profile UI

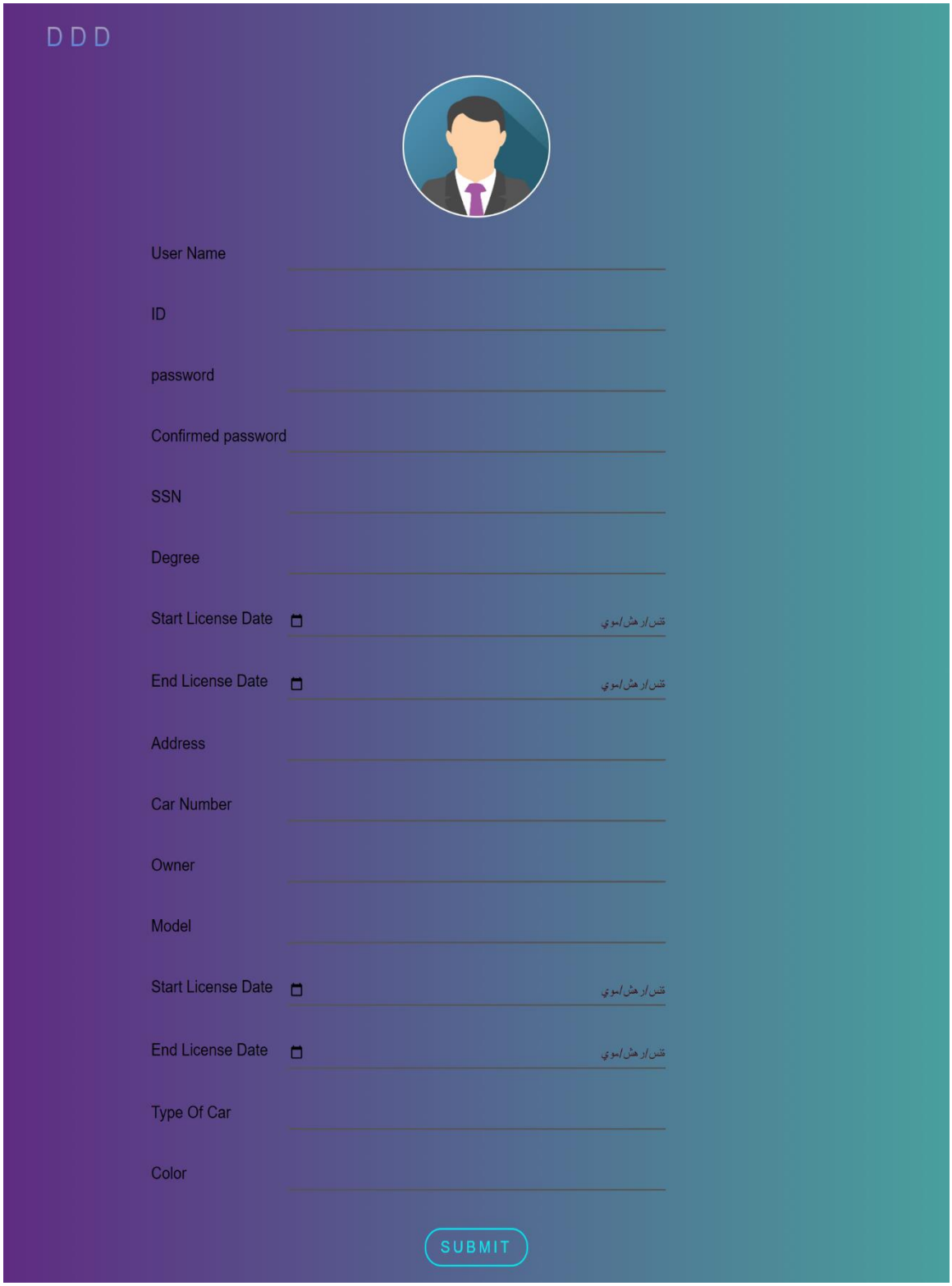
- **List of the Drivers work in the company UI**



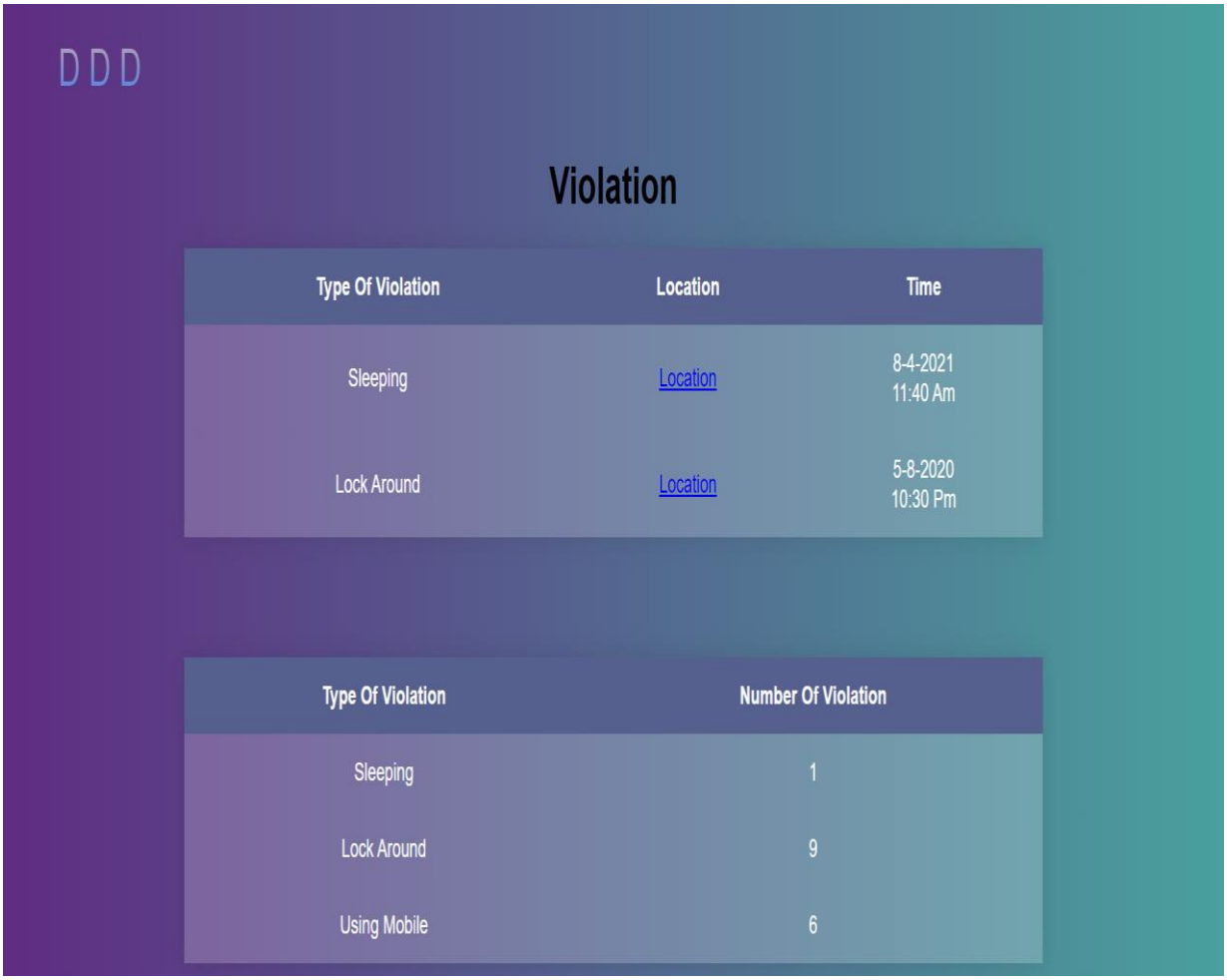
- **When the company need to add new car go to add new car page to add the car to work in company**
- **Add_Car page (UI)**



- **When the company need to add new driver with the car go to add new driver page to add the driver and car to work in company**
- **Add_driver page (UI)**



- **To know the types of violation of the driver and the time of violation and location of the violation and number of repeat this violation.**
- **Violation page UI**



- **The profile of the driver to display the data about the driver and his car.**
- **Profile of the driver UI**

D D D

User Name	Yahia Ali
SSN	29905081675
ID	8545
Degree	1
Start License Date	25-5-2020
End License Date	25-5-2021
Address	22 ST. Fatma El Zahraa Cairo.
Owner	Yahia Ali
Car Number	1293-ABC
Model	Audi A3
Start License Date	5-8-2020
End License Date	5-8-2021
Type Of Car	Personal Car
Color	Silver

EDIT PROFILE

VIOLATION

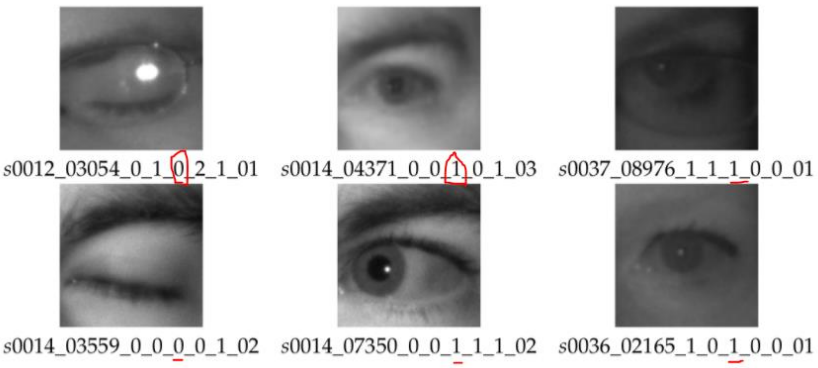
Chapter 4: AI Models : The 1st model will determine if the eye is closed or open using CNN (convolution neural network):

The idea of this model : Is to identify the person's face, then determine the eyes from the face, and then determine whether it is open or closed.

The implementation For this model using: (Tensorflow / keras) , (Opencv for computer vision) .

Note: the input stream will be converted into grayscale images.

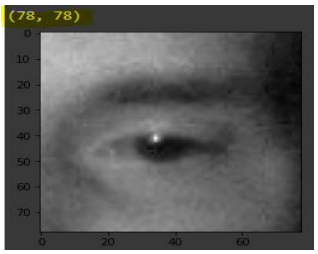
Method for 1st model: We applied customized deep learning model based on eyes data set which is (MRL eye dataset) , This data was organized by using a simple automated python (organize_mrl_data.py) file , Eye state [0-closed, 1- open] = index 16

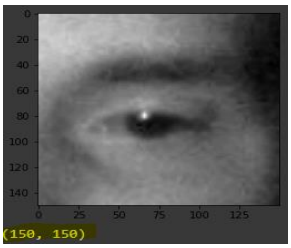


Data processing : Before we build the model, we must first prepare the data. we noticed that the size of the images in the training was 78 * 78 pixels, which is small, so we increased it to 150*150 pixels , by using (resize) function.

After that, we created an automated code to organize training , validation, and test data,which are as follows:

```
total training close images: 15000
total training open images: 15000
total validation close images: 3000
total validation open images: 3000
total test close images: 2000
total test open images: 2000
```





Model architecture : We are using (keras) library to build a CNN model, As we can see we started the first layer with 32 kernels(filters) and use (relu) activation function.

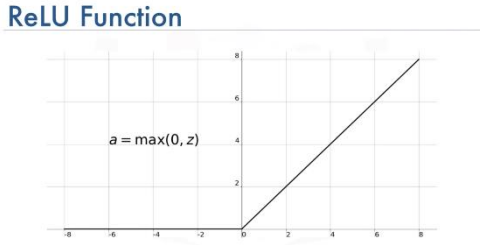
And t he second layer using (2,2) maxpooling

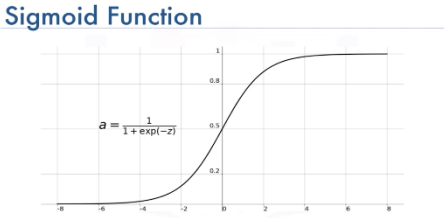
Which mean this layer will divide the output from the previse layer into (4)

```
model = models.Sequential()

model.add(layers.Conv2D(32, (3, 3), activation='relu',input_shape=(150, 150, 3)))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Flatten())

model.add(layers.Dropout(0.5))

model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))

model.summary()

from keras import optimizers
model.compile(loss='binary_crossentropy',
              optimizer=optimizers.RMSprop(lr=1e-4),
              metrics=['acc'])
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 148, 148, 32)      896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 74, 74, 32)        0
```

This is an approximation of building a CNN that shows how (conv , relu) and (polling) are done before (the fully connected layer) , As we can see in this photo in the last layer this model use (softmax) as an activationfunction ,but in our model , we using (sigmoid) Why ? Because this is a binary classification .





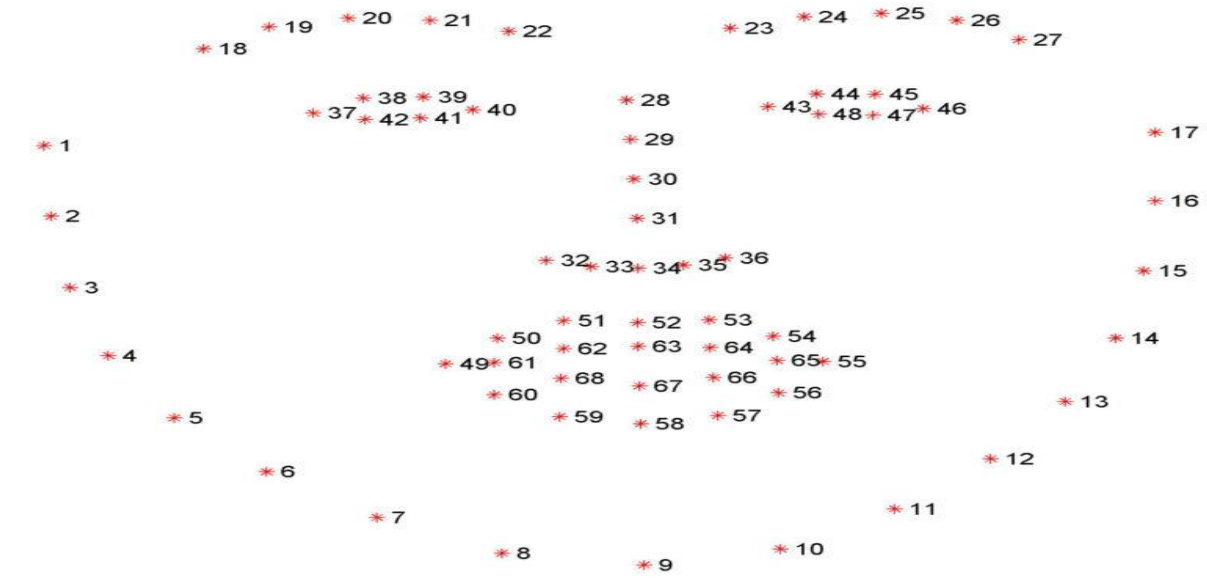
The run & some issues: This is some outputs from the model

As we can see when eyes open the result above (0.8 which we Set it as a threshold or you can make it 0.5) and when eyes is close the result become (<0.8 or <0.5) But we face some issues with openCV for detecting the eye , as when the eye close the model for detecting the eye does not work we searched for this problem on Stack OverFlow and we figured this . So, we decided to use dlib to see which one was better

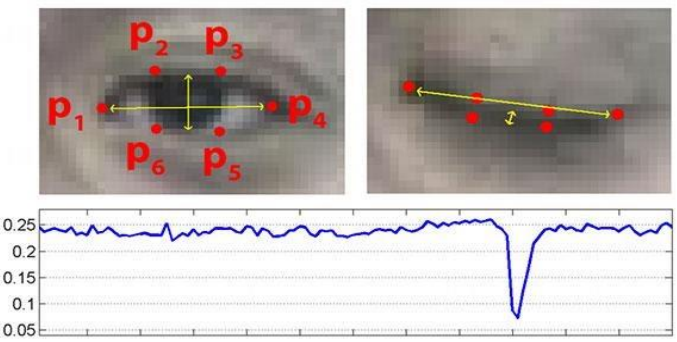
The second model for drowsy with dlib:

We will try to remedy the defect of the previous model by using dlib library which is a modern C++ toolkit containing machine learning algorithms. As we can see this is face landmarks. Which can detect with (face landmarks)

```
hog_face_detector = dlib.get_frontal_face_detector()
dlib_facelandmark = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
```



Algorithm for 2nd model: Each eye is represented by 6 (x, y)-coordinates, starting at the left-corner of the eye (as if you were looking at the person), and then working clockwise around the eye . Condition: It checks 20 consecutive frames and if the Eye Aspect ratio is less than 0.25,Alert is g enerated. So , we will use this in our project.



$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

The third model is pose estimation using DNN(Deep Neural Network):

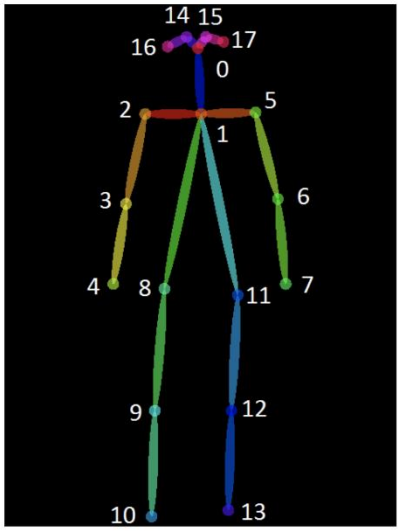
We also know that looking to the right or left or directly using the phone is wrong while driving .

So why ? if he needs to talk on the phone directly, it prevents his eyes from looking into the blind spot, causing some accidents.

If you want to use phone, you can hang the phone on the dashboard or the car windshield and speak from the speaker. We decided to use position estimation to determine whether the driver is looking left or right, and also to see if the driver's hand, whether the right hand or the left hand, is raised from shoulder level or not.

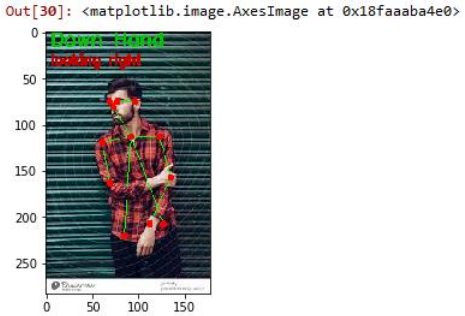
Method of the 3rd model: To implement this model we will use Tensorflow frozen model Which is in an open source. There is another good paper for more understanding ,we use (winsound library for giving a sound when there is a mistake) and then we load the model using Tensorflow module .

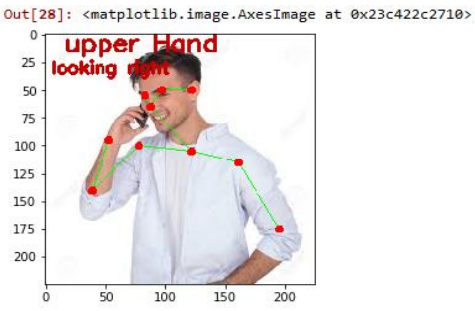
-As we see there are numbers for each part of the body .
-From this we can know every part that falls on the screen.
-We can notice that if one of the ears (16 or 17) disappears from the screen, the person is looking right or left.
-And also, we located one of the wrists, (4 or 7)and found that it was above the shoulders, this is considered wrong, because that distracts the eyes.



The cases :

- If the human is looking ahead .
- If he looks left or right .
- If the wrist is down or up





Fourth Model face recognition system:

There is some steps to perform facial recognition .

Step 1: locate and extract faces from each image.

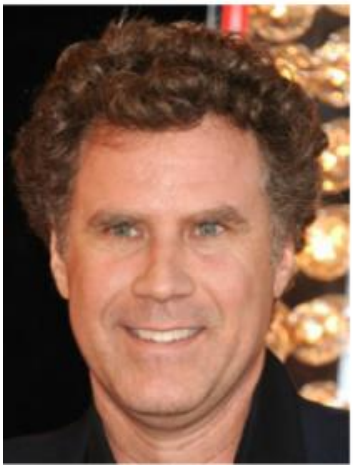
Step 2: identify facial features in each image.

Step 3: align faces to match pose template.

Step 4: encode faces using a trained neural network

Step 5: check Euclidean distance between face encodings

What is Hog ? It is histogram of oriented gradients.



Implementation idea for the 4th model: We will put all our data which is persons images in a file (images)

The next step we will encode the faces using function (fing_encodings) which take the images(list) and convert every single image on it into (RGB) ,because of (opencv) read image(BGR) ,so we need to convert it , after that encode the face with (face_recognition.face_encodings) which is a function in (face_recognition library)to return a list contains encoded numbers .

Then append this encode into the list which we will return it.

The next step we will call this function to encode all images in (images list).

```
path = 'images'
images = []
class_names = []

my_list = os.listdir(path)
print(my_list)
# next use thes names import the images one by one
for cl in my_list:
    cur_img = cv2.imread(f'{path}/{cl}')
    images.append(cur_img)
    class_names.append(os.path.splitext(cl)[0]) #to get
```

```
def find_encodings(images):
    encode_list =  []
    for img in images:
        img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encode_list.append(encode)

    return encode_list
```

The next step we will make capture video with (openCV) to detect the faces using webcam .

Using : (compare_faces) function which

take a list of encoding images and compare it with a single encoded image after that return a (Boolean list)contain only true , false values.

while , (face_distance fun) the same parameters ,but return a list with a float values between (0-1) which mean the num. close to 1 is more different between this two images.

```
for encode_face , face_loc in zip(encodes_cur_frame , faces_loc_ofFrame):
    matches = face_recognition.compare_faces(encode_list_known , encode_face)
    face_dis = face_recognition.face_distance(encode_list_known,encode_face)
```

There is another fun. Which is less important.

(make_attendance fun): which takes the name of person and save it in an excel file with time he attended.

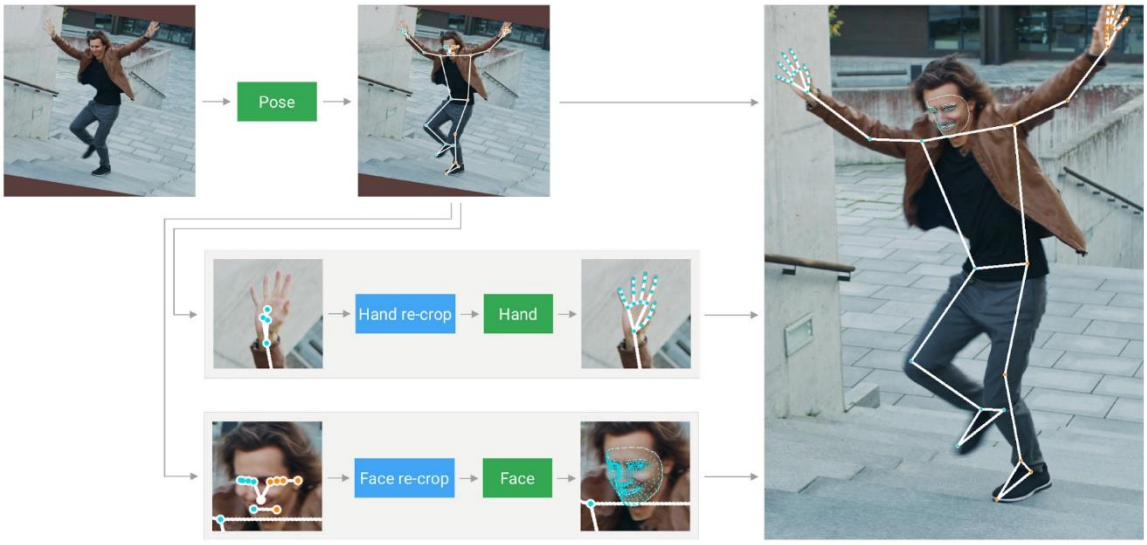
So why ? It is act like a simple data base but we will use (postgresql)

```
def mark_attendance(name):
    with open('attendance.csv','r+') as f:
        my_data_list = f.readlines()
        # print(my_data_list)

        name_list = []
        for line in my_data_list:
            entry = line.split(',')
            name_list.append(entry[0])
        if name not in name_list:
            now = datetime.now()
            dt_string = now.strftime('%H:%M:%S')
            f.writelines(f'\n{name},{dt_string}')
```

Fifth Model Holistic using Mediapipe :

Mediapipe is a framework for building cross platform multimodal applied ML pipelines that consist of fast ML inference, classic computer vision, and media processing (e.g. video decoding).

The MediaPipe Holistic pipeline integrates separate models for pose, face and hand components, each of which are optimized for their particular domain. However, because of their different specializations, the input to one component is not well-suited for the others.

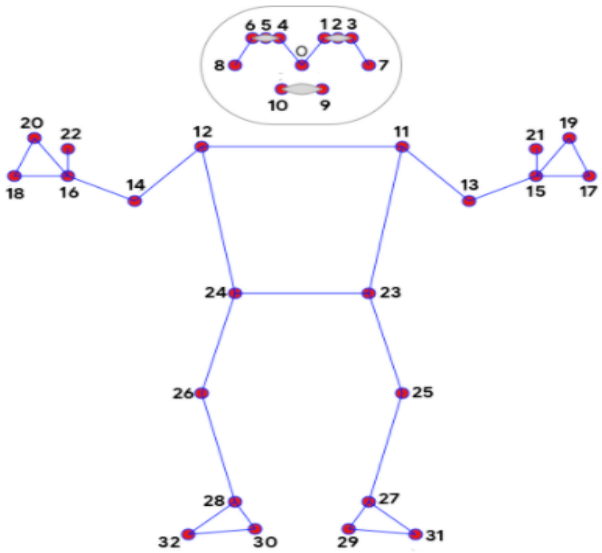


We used this model to find out if the driver is asleep or not, and also to find out if his hand is close to his eye or not.

Note: The driver will be alerted if he makes one of these mistakes in Arabic.

Sixth Model Pose estimation using Mediapipe :

MediaPipe Pose is a ML solution for high-fidelity body pose tracking, inferring 33 3D landmarks on the whole body from RGB video frames utilizing our BlazePose research that also powers the ML Kit Pose Detection API.

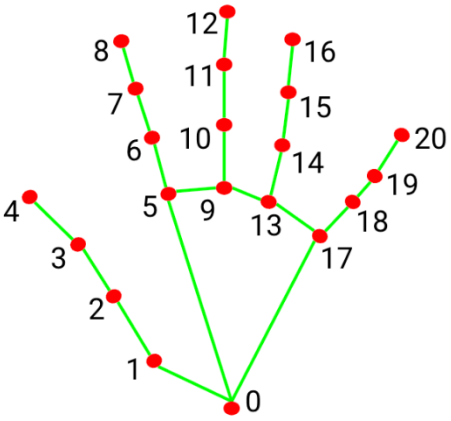


0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

We used this model to find out if the driver is looking to the right or left and is not paying attention to the road, and it is also audibly alerted in Arabic.

Seventh Model For Hands using Mediapipe :

The ability to perceive the shape and motion of hands can be a vital component in improving the user experience across a variety of technological domains and platforms. For example, it can form the basis for sign language understanding and hand gesture control, and can also enable the overlay of digital content and information on top of the physical world in augmented reality.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

We used this model to allow the driver to know some voice aids by making some movements using one or both hands.

Like :

- Raise your first (index) finger in the left hand for voice assistance.
- Raise your index and middle fingers together on the left hand To change the volume by moving your thumb and forefinger on the right hand.

References :

- **For Design :**
  - **www.figma.com**
  - **www.draw.io.com**
  - **www.w3schools.com**
  - **https://codepen.io**
  - **https://www.flaticon.com/**
- **For code:**
  - **https://code.visualstudio.com/**
  - **https://stackoverflow.com/**
- **For AI:**
  - **https://www.kaggle.com/**
  - **https://www.tensorflow.org**
  - **https://opencv.org/**
  - **http://dlib.net/**
  - **https://mediapipe.dev/**
- **For Back-end:**
  -
- **For learning:**
  - **https://www.youtube.com/c/HeshamAsem**
  - **https://www.coursera.org**