Software Design Description Specification

CENG 407

Drowsy Driver Detection System

Project Group 1

Advisor
Serdar Arslan

Names of members

Mert Alp Kuvandık 201911038
Volkan Mazlum 201811045
Bahu TONGAL 201811058

28/12/2022

Table of Contents

List of Figures

1. Introduction

This Software Design Description Report provides understandable information on what kind of system software should include in our project. This document includes the design of the system in the project we will develop, and a general description of the methods used. In this project, we will use software techniques while developing the system. For this software part, we will take help from python language. In general, our system includes a camera system, detection system, classification system, and alert system. We will take help from the OpenCV library and the Dlib library for the Camera and Detection system. OpenCV will be used for both object detection and face detection. The dlib library will act as an auxiliary library for face detection. For the Classification system, it is possible to use the CNN-Inception v3 algorithm included in the TensorFlow library. It can be replaced with a better algorithm in future development stages. For the alert system, help will be taken from the libraries in python. The Flask framework will be used for a web application. It is considered to use this framework as it will be easier to integrate with Python. These techniques will be explained in detail in later sections.

1.1. Purpose and Scope

This document outlines what needs to be done and how the system can be built. As a result, the paper includes all the interfaces, diagrams, and interactions that we used during the project's development.

Our project scope is to create an application that analyzes the user's behavior and status to identify their current level of sleep. The use of computer vision will not be the only factor considered in making this decision. Naturally, instantaneous face, mouth, and eye detection will also be possible, but the goal is to apply machine learning for eye detection. The number of yawning will be immediately counted in mouth detection utilizing face points gathered from various libraries, such as dlib, and a warning will be given to the user after exceeding a specified threshold. The user's status cannot be determined by keeping their eyes closed for an extended period. In this area, numerous initiatives are underway. Instead of just focusing on eye issues, we want to create a system that works with the car. Additionally, it aims to enhance sleepiness detection considering various environmental factors. Our system will mostly consist of:

- Face Detection for checking drowsiness
- Eye, Mouth Detection for checking yawning and drowsiness
- Classification with Machine Learning Model
- Alert System for avoiding drowsiness

1.2. Glossary, Definition and Acronyms

Term	Definition
Actor	An actor can be a user like driver. It can be also another software system that interacts with the system.
Python	Python is high-level programming language which is preferable when Machine Learning, Deep Learning algorithms are used.
Software Requirement Specification (SRS)	A document that contains a detailed explanation of the system's operations, needs, restrictions, and operating conditions. The following document is an SRS document.
User	Users can be drivers who will drive and provide us with the dataset taken from their videos and make our application good.
Object Detection	The detection of objects is a common computer science technology. The computer decides and recognizes what the thing is using visual methods.
Convolutional Neural Network (CNN)	A particular type of deep neural network is the convolutional network, sometimes known as CNN or ConvNet. It's a feed-forward artificial neural network with a high level of complexity.
Artificial Neural Network (ANN)	An artificial neural network is a computer model made up of many processing components that receive inputs and output results depending on predetermined activation functions.
Inception v3	It is used for classification of images and object detection. It is thought of as a database of classified objects.
SDD	A document that offers a detailed description of the system's design, requirements, and conditions for method execution.
Flask	Flask is used for making web application to integrate with backend. Also, Flask is a micro web framework which is written in Python.
Jupyter Notebook	Jupyter Notebook is an opensource development platform for Python.

Object Detection	Object detection is used for applying computer vision algorithms to perform detection of object to image.
OpenCV	This is a library that is used to get video streaming, and provides Computer Vision library.
Scikit-learn	This is a library that is used for applying machine learning algorithm. It includes algorithms like SVM, clustering etc.
Tensorflow	This is a library that is used for applying machine learning algorithms. It includes algorithms like ANN etc.
Pandas	This is a pyhton library for offering different operations for manipulating data.

2. References

- IEEE Computer Society. (2009). IEEE Standard for Information Technology—Systems Design—Software Design Descriptions. IEEE. New York: IEEE. IEEE Std 1016
- Sommerville, I. (2016). Software engineering. Boston: Pearson Education Limited.
- Öztürk, M., KüçükmaniSa, A., & Urhan, O. (2022). Drowsiness detection system based on machine learning using eye state. Balkan journal of electrical and computer engineering, 10(3), 258-263.
- Park, S., Pan, F., Kang, S., & Yoo, C. D. (2017). Driver drowsiness detection system based on feature representation learning using various deep networks. In Asian Conference on Computer Vision (pp. 154-164). Springer, Cham.
- Santoshi, M. G. (2020). AUTOMATED DRIVER DROWSINESS DETECTION FOR NON 2 WHEELERS (Doctoral dissertation, Andhra University).
- BAHARU, B. (2013). DRIVER DROWSINESS DETECTION BY USING WEBCAM.

3. Overview of This Document

We attempted to present a detailed design description of Drowsy Driver Detection System in this SDD. As a result, we expressed our system's design assertions. In the first section, we present a brief overview of our system and discuss its essential characteristics. We classified the first part as 1.1. Purpose and Scope. Glossary 1.2, Stakeholders and Their Concerns 1.3. In the 1.1. Purpose of this Document section, we indicated what this document seeks to convey, as well as the scope of this project. All special terminology used in this report are defined in 1.3. Glossary. Also, in the 2. Reference part, we indicated references that we used while creating this document.

Following that, in the 4. System Design part, we revealed our system's full design architecture. It is divided into three sections: 4.1. Architectural Design and 4.2. User Interface Design, 4.3. Requirement Matrix. In 4.1. Architectural Design, we presented many schematics of our system and explained the context of the design idea behind the system. The User Interface Design section in 4.2 outlines the overall structure of our system. These designs are presented in the form of figures with brief descriptions. We supplied a table structure in the 5. Requirement Matrix section that demonstrates the link between components and requirements contained in the SRS document. We created a link between system components and requirements for this purpose by describing which requirement is achieved by which component. We verified that each need is linked to the required component. As a result, this paper will serve as a guideline for drivers to comprehend the Drowsy Driver Detection System's overall design architecture.

4. System Design

This part of system design includes the Architectural Design of the system, problem description, and it includes which technologies will be used in our project, User Interface Design, and Hardware Design. Also, it contains some diagrams such as Sequence Diagram, Activity Diagram, Data Flow Diagram, Class Diagram.

4.1. Architectural Design

To offer a clearer understanding of our architectural design, we explained the challenge, technologies employed, and many diagrams in this part. As we discussed in earlier parts, our system will perform three major functions. As a result, we will create an architecture that includes these duties. The primary construction of the Drowsy Driver Detection System is shown in Figure 1.

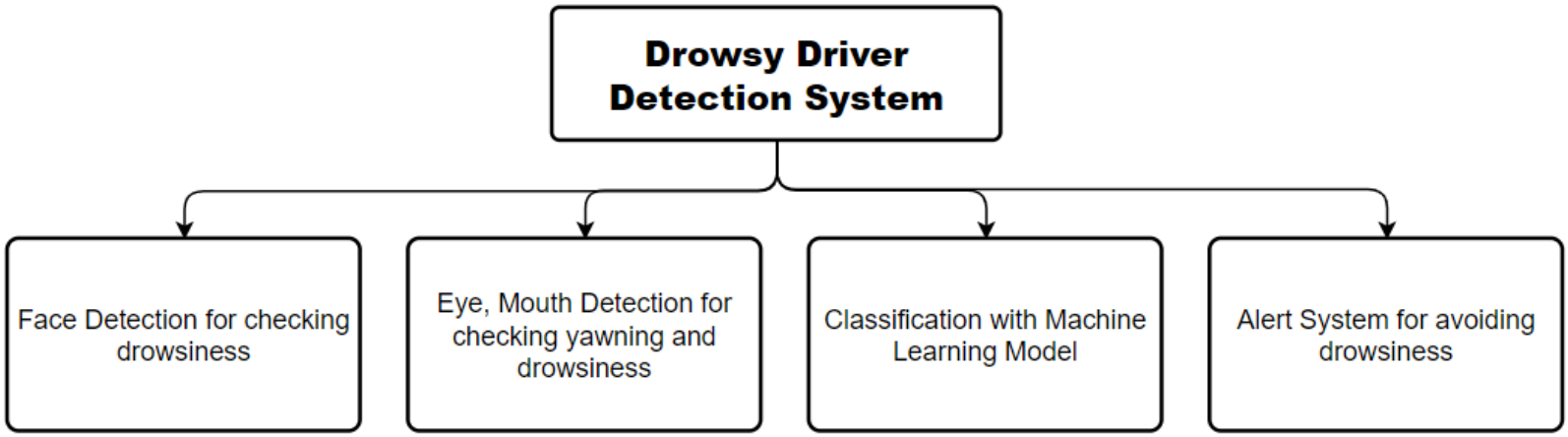

Figure 1: Brief of System Architectural Design

4.1.1. Problem Description

It is a project that has the capacity to provide a certain amount of improvement to car accidents, which are a problem that cannot be ignored around the world. We will try to find solutions to some problems in this project, which we set out to obtain a better result than other projects in the literature. Our system, which can provide classification under different conditions, will try to keep the driver away from situations that may endanger the driver's life with its alarm system. Our project, which also focuses on different features, is expected to provide solutions to technical problems such as data set problems. This project, integrated with machine learning, can work more autonomously than other detection systems. From the moment the driver starts the application, our project will do its best to ensure the driver's safety.

4.1.2. Technologies Used

There are all the software tools we will use in our system below:

- Python
  We use Python for many operations in our system such as Image Processing, Object Detection and Machine Learning.
- Jupyter Notebook
  Jupyter Notebook is an opensource development platform for Python.

- Image Processing
  In field of computer engineering, image processing is used for applying computer algorithms to perform image processing on images.

- Object Detection
  In field of computer vision, object detection is used for applying computer vision algorithms to perform detection of object to image.

- Machine Learning
  We will use Machine Learning Techniques for classification of drowsiness and detection of closed, open eyes in our system. We will use CNN-Inception v3 algorithms. It is best option for classification of images and object detection. It is thought of as a database of classified objects.

- Web Application
  We will use Flask for making web application in our system to integrate with backend of our system. Flask is a micro web framework which is written in Python.

- NoSQL Technology

  We use MongoDB for storing and back-up driver information to login and register to the system.

4.1.2.1. Software Requirements Specification

- Python
  - Python 3 or higher version
  - Flask: Flask is a micro web framework which is written in Python. It is used for creating web applications.

- Libraries
  - NumPy: This library is used for working with arrays.
  - Dlib: This library is used for finding human face and pose of face via facial landmarks.
  - OpenCV: This library is used to get video streaming via Camera.
  - Scikit-learn: This library is used for applying machine learning algorithm. (SVM, Clustering etc.)
  - TensorFlow: This library is used for applying machine learning algorithms. (ANN etc.)
  - Pandas: This library offers different operations for manipulating data.
  - Matplotlib: This library is used for visualization, etc.

    - Winsound: This library is used for alarm system.
    - Pymongo: This library is used for database process.
    - Pillow: This library is used for image process.
- Operating System
    - Windows or Linux

4.1.2.2. Hardware Requirements Specification

- Laptop with basic hardware
- Camera

4.1.3. Data Flow Diagram

In figure 2,3,4,5,6 of this section, data flow diagrams are stated.

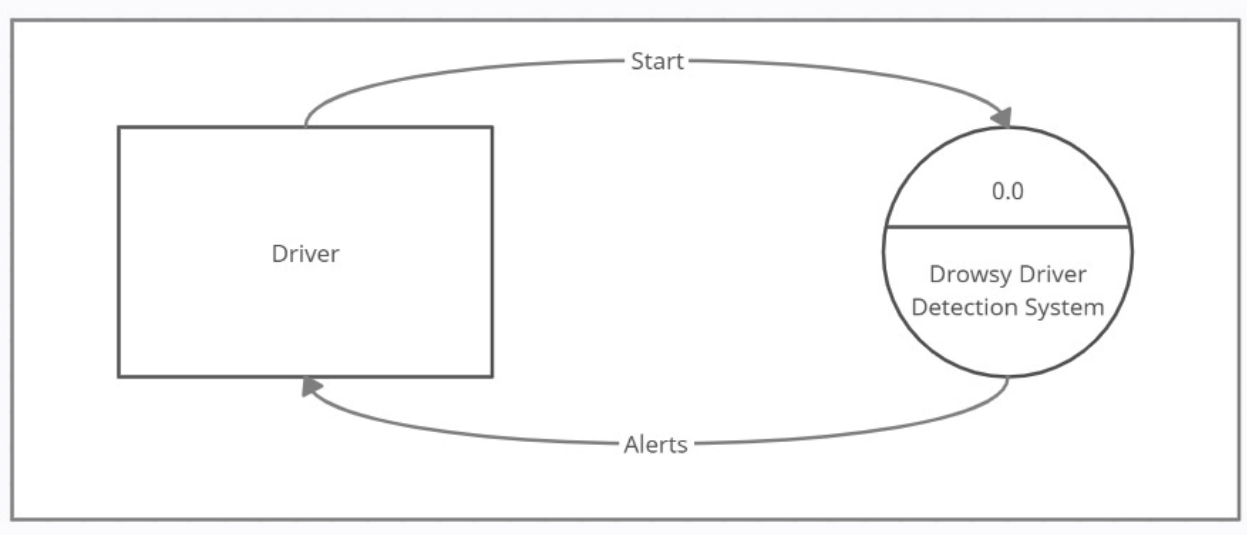

Figure 2: Context Diagram

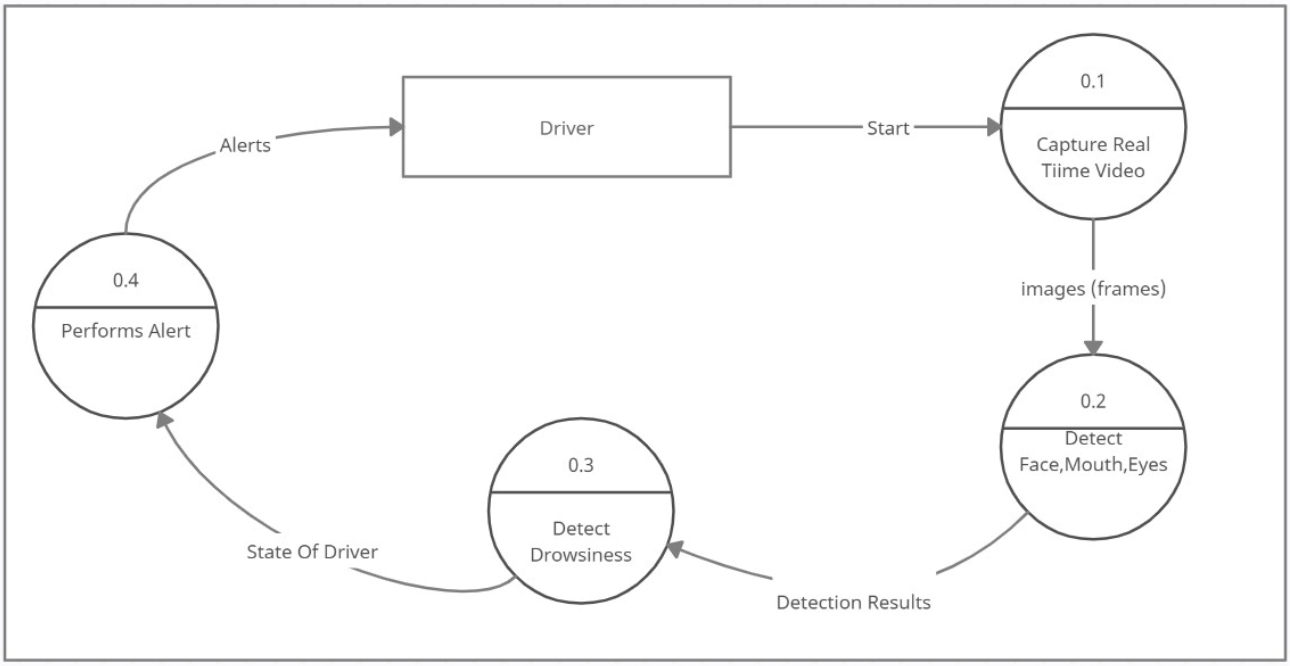

Figure 3: DFD Level -1

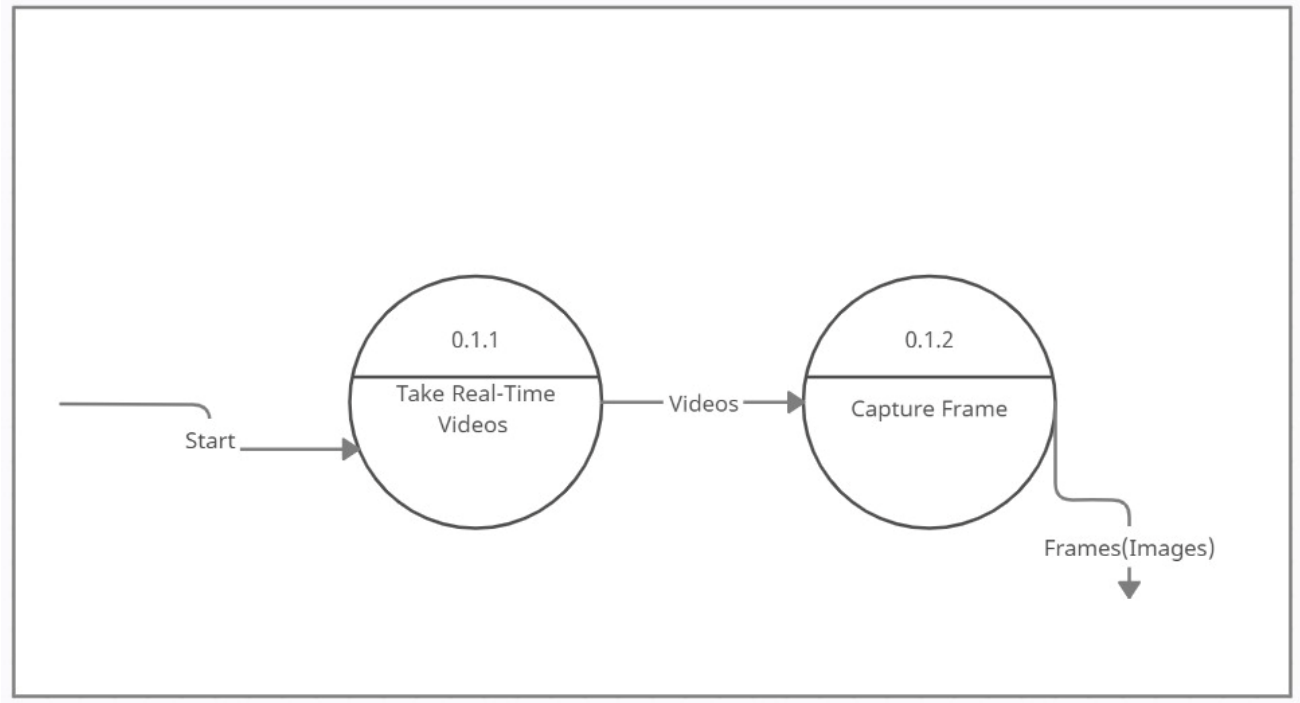

Figure 4: DFD Level-2 for Capture Real-time Video

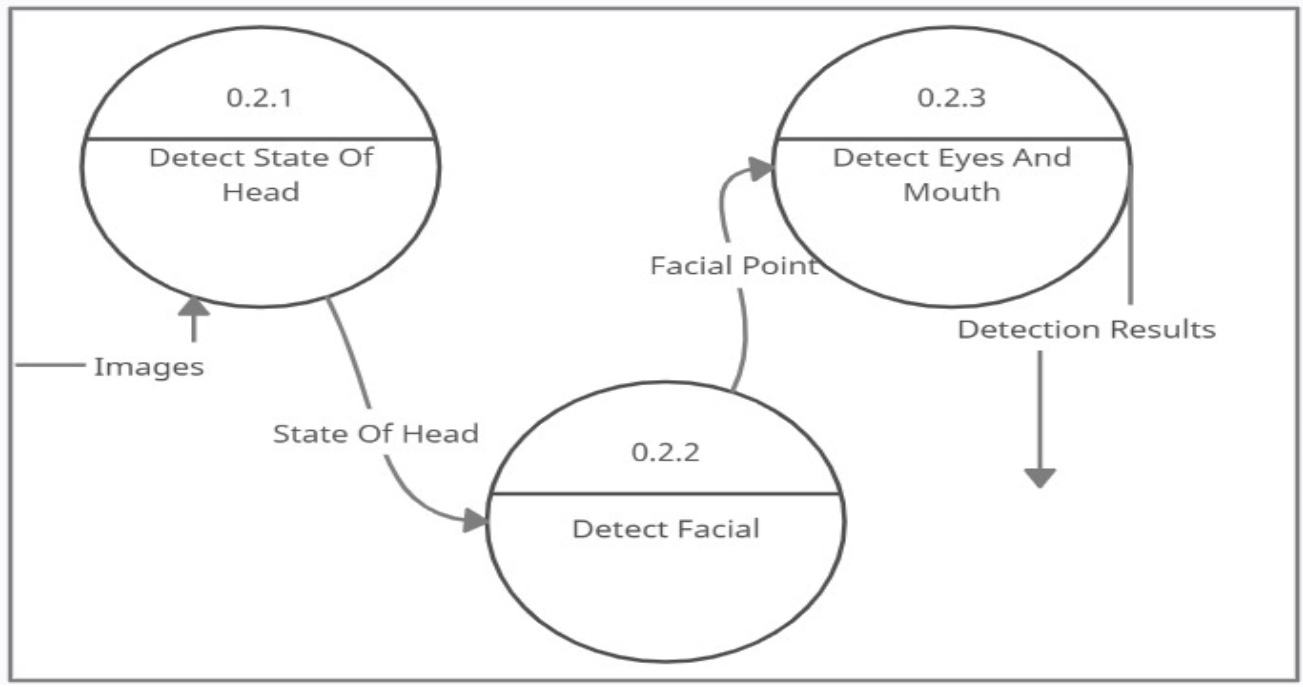

Figure 5: DFD Level-2 for Detect Face, Eyes, and Mouth

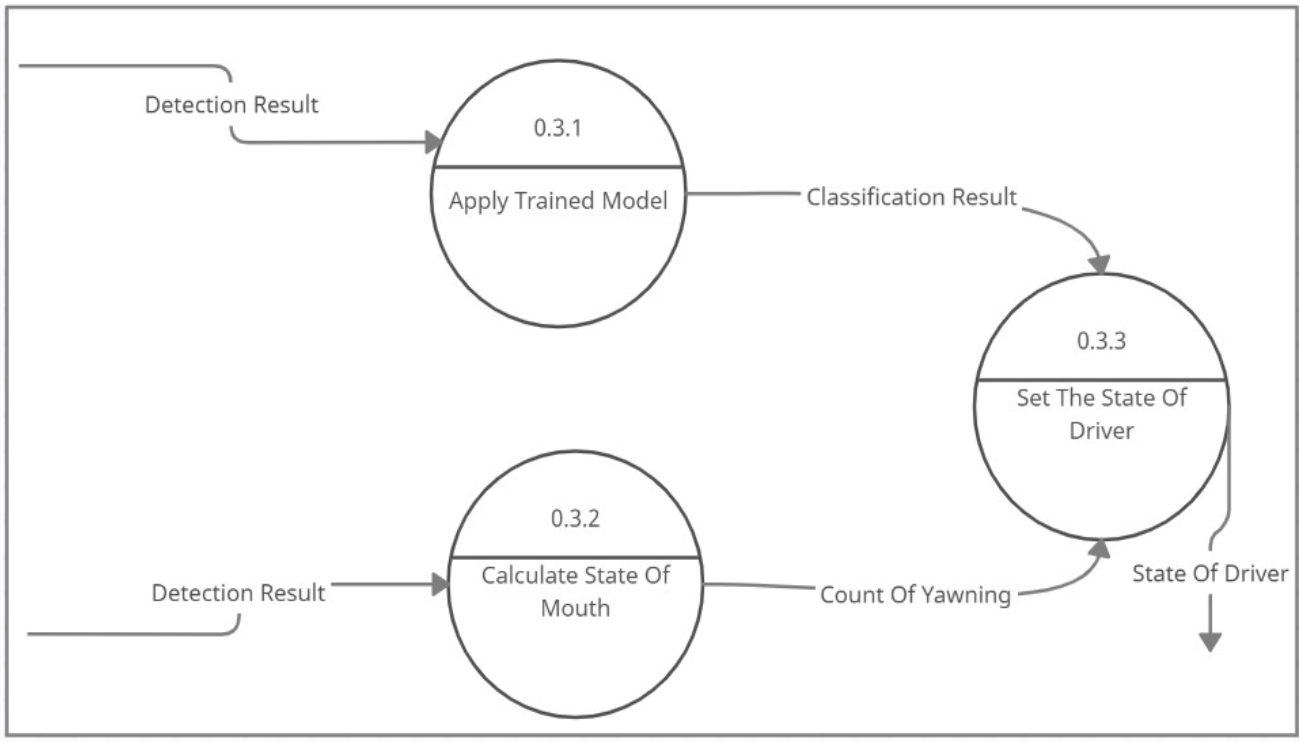

Figure 6: DFD Level-2 for Detect Drowsiness

4.1.4. Activity Diagram

In figure 7, activity diagram is shown.

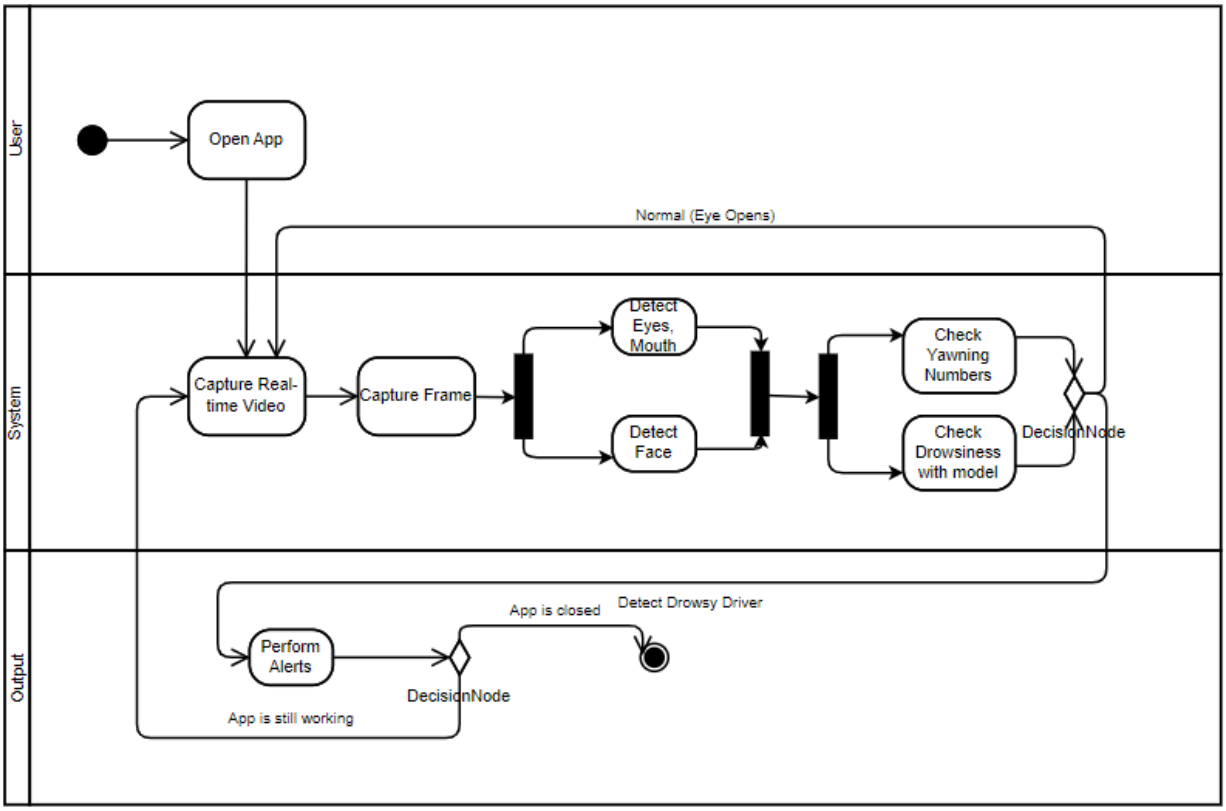

Figure 7: Activity Diagram for Drowsy Driver Detection System

4.1.5. Class Diagram

In figure 8, class diagram is shown.

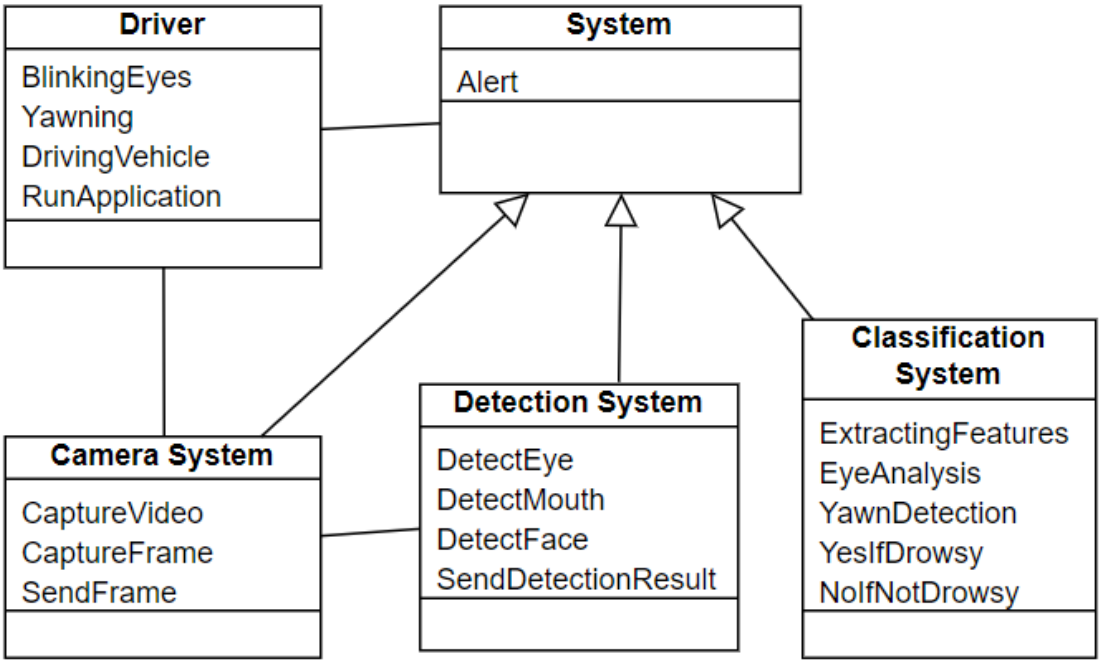

Figure 8: Class Diagram for Drowsy Driver Detection System

4.1.6. Sequence Diagram

In figure 9, activity diagram is stated.

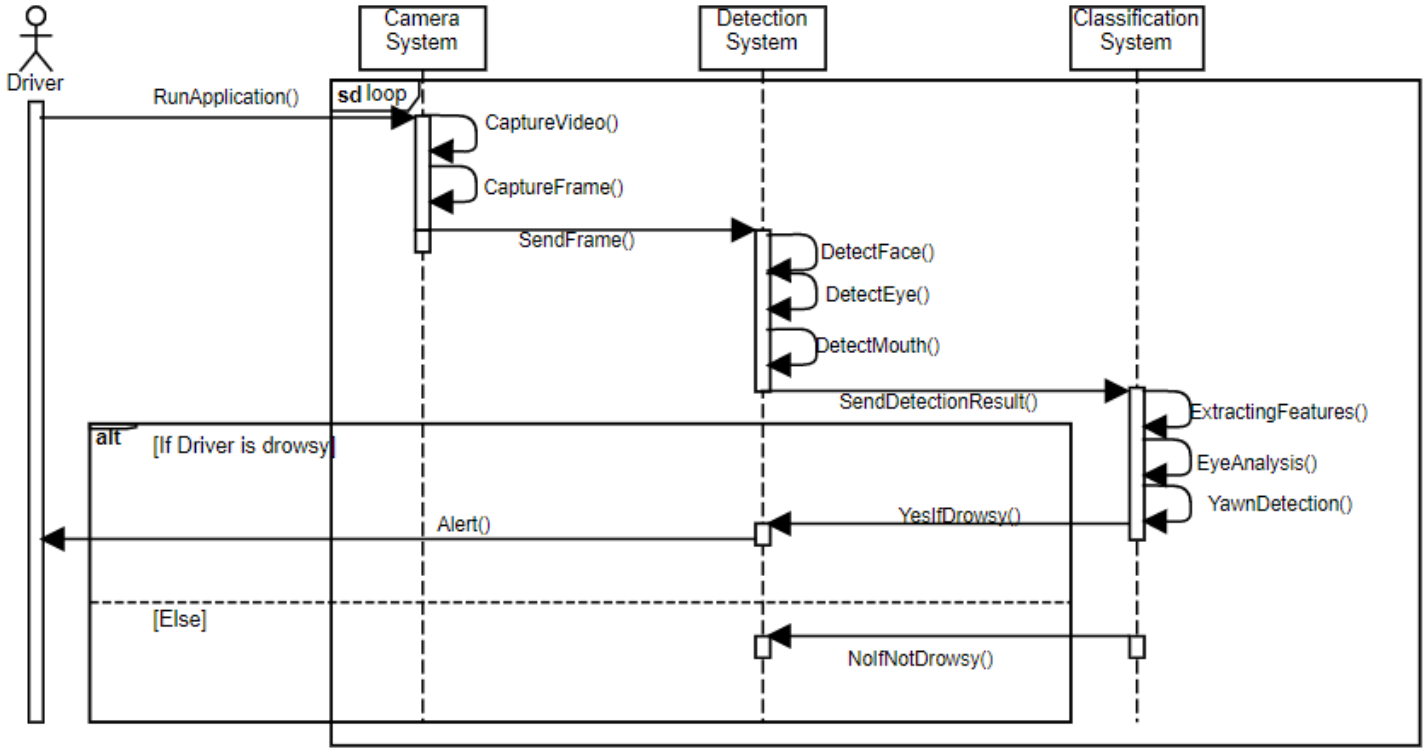

Figure 9: Sequence Diagram for Drowsy Driver Detection System

4.1.7. Project Life-Time Review

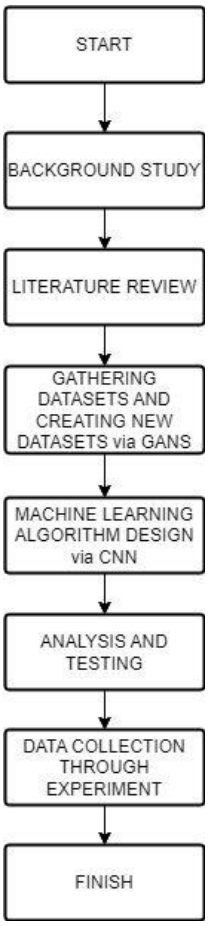

Figure 10: Diagram for Project Lifetime

4.2. User Interface Design

4.2.1. Login Page

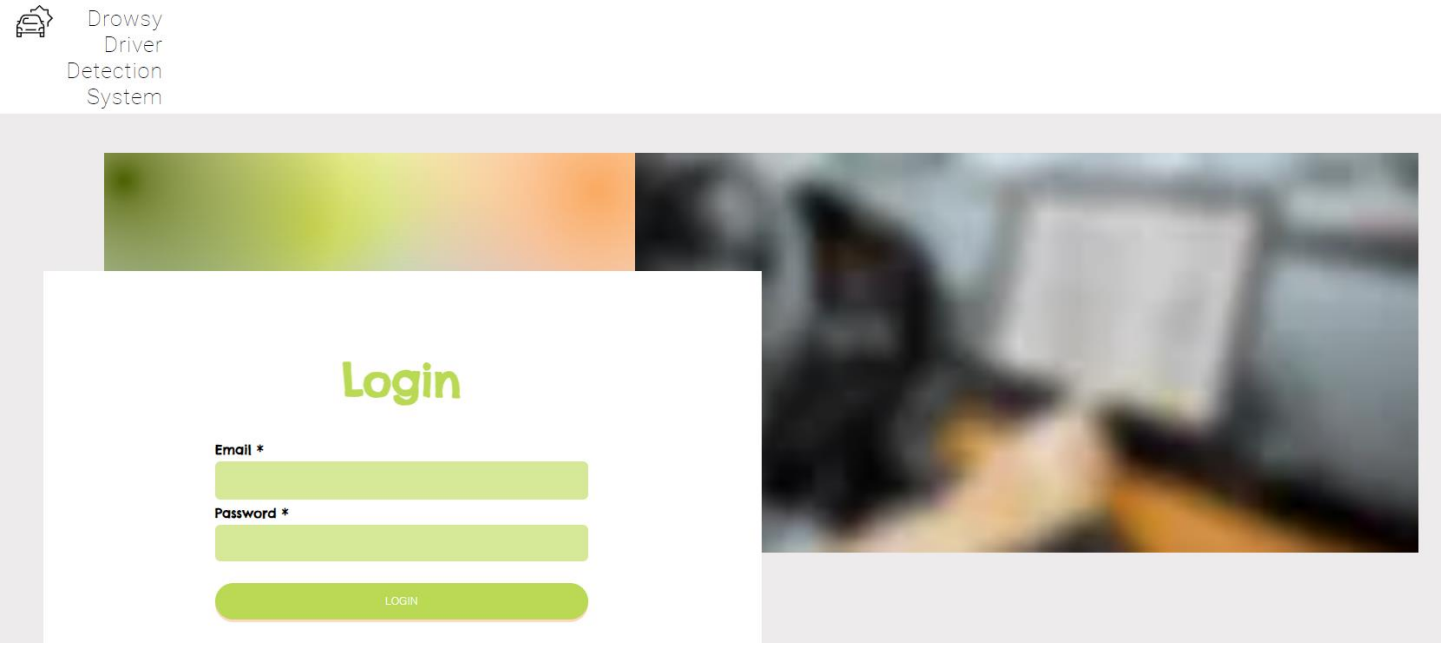

Figure-10 UI for Login Page

In the figure above, there is the login page of our application. Driver must login before using our app. After this entry, the processes will start automatically. It is designed so that the user can use it easily and get used to it immediately. Thanks to this design, it is aimed to attract the attention of the user to the system and to increase the efficiency.

4.2.2. Password Page

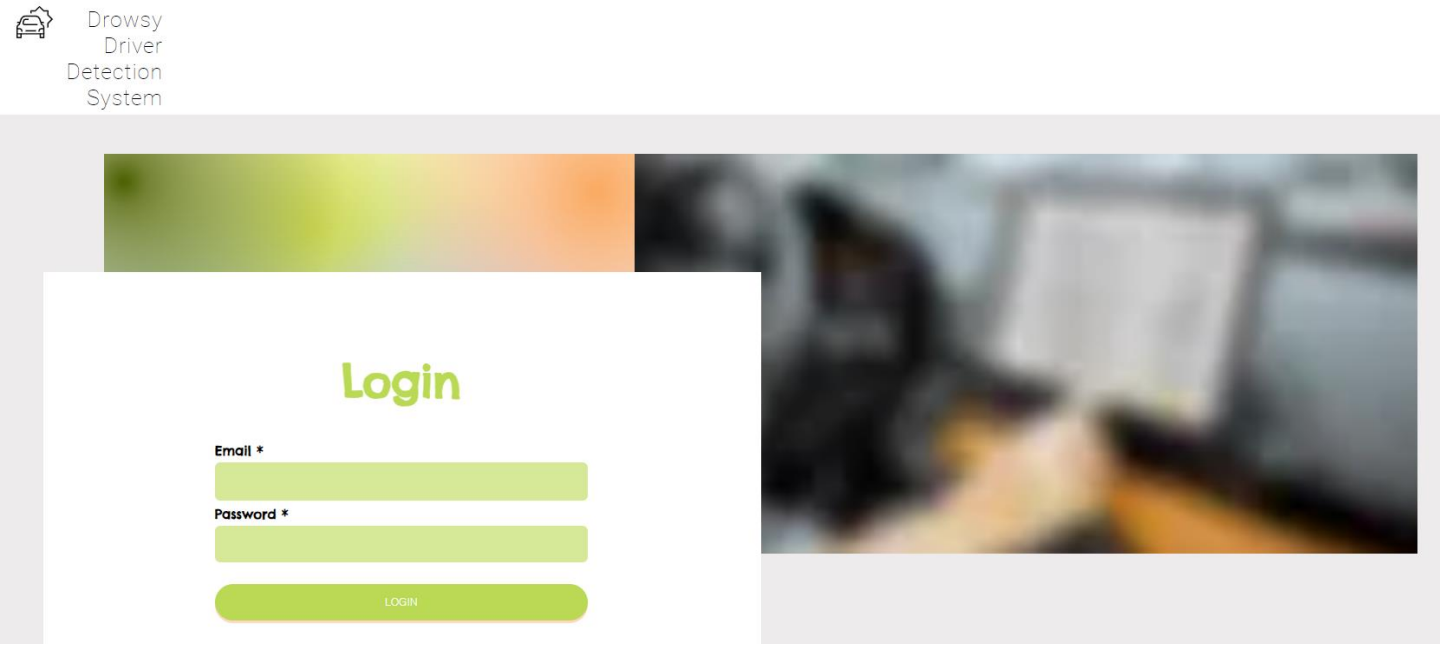

Figure-11 UI for Password Page

In this figure, the system is the interface to be used to enable the driver to log in again when an error occurs in any input.

4.2.3. Detection Page

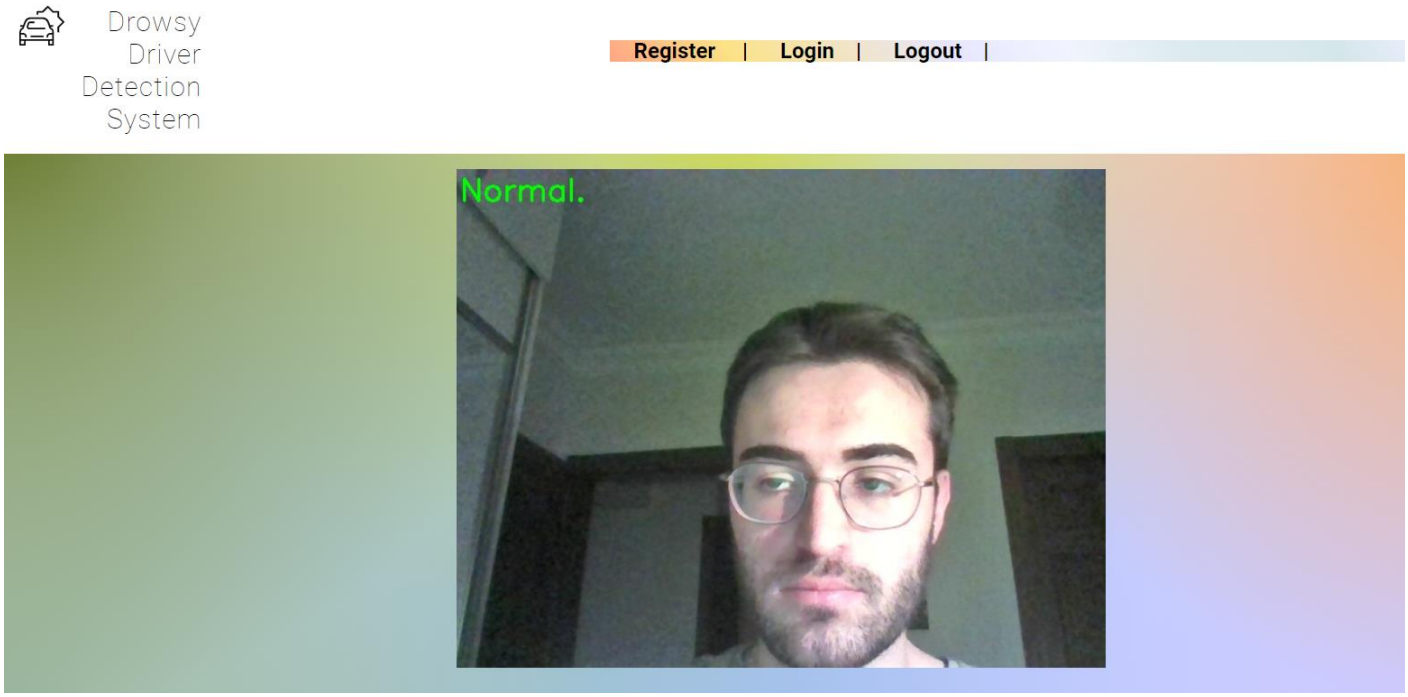

Figure-12 UI for Detection Page

This part is the main part of the application. There is a section where the classification and detection results are shown, and even the values and warning system according to these results. It is a very important interface for the system. It is designed simple so that the user can easily perceive the alert system in any situation.

4.2.4. Create Account Page

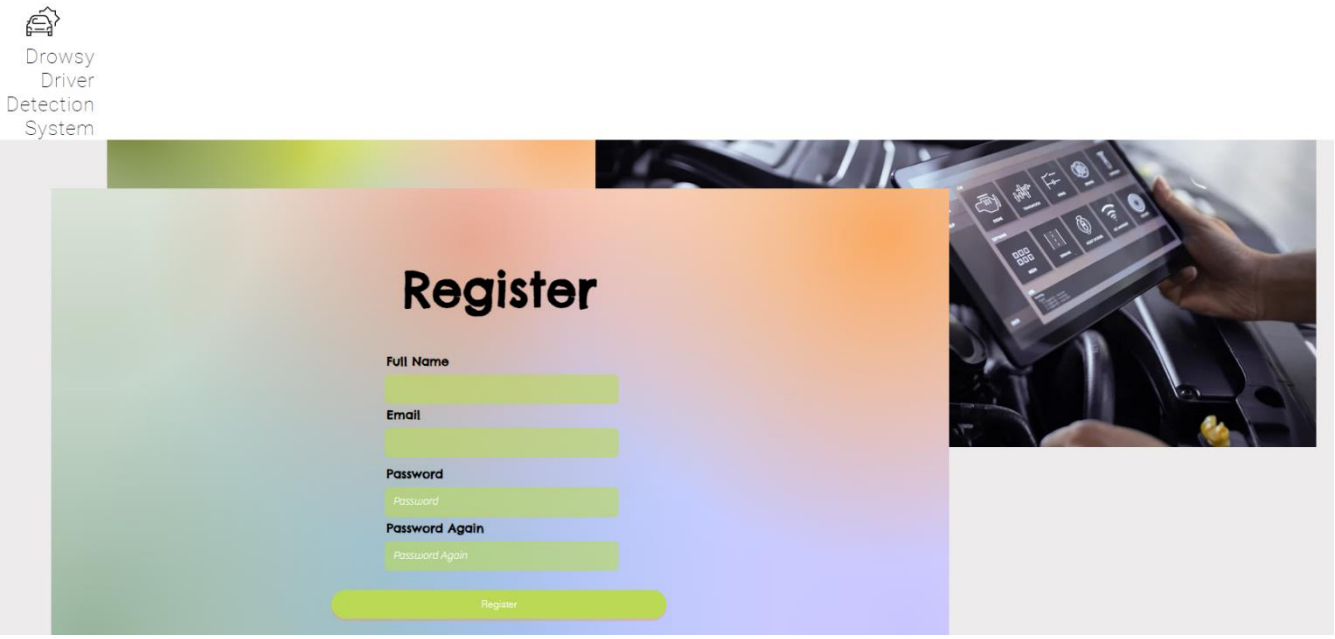

Figure-13 UI for Creating Account Page

In the figure above, there is the create account page of our application. Driver must create an account before loginning our app. After this entry, the system will direct user to login page automatically. It is designed so that the user can use it easily and get used to it immediately. Thanks to this design, it is aimed to attract the attention of the user to the system and to increase the efficiency.

4.3. Requirements Matrix

In Figure 14, requirement matrix of our system is stated.

Component / Requirements		Comp-01	Comp-02	Comp-03	Comp-04
Capture Frame		**X**			
Capture Real-time Video		**X**			
Detect Mouth		**X**	**X**		
Detect Eye		**X**	**X**	**X**	
Detect Face		**X**			
Detect Status of Head		**X**			
Performs Alert					**X**
Detect Drowsiness				**X**	

Comp-01 = Face detection for checking drowsiness
Comp-02 = Eye-Mouth Detection for Checking Yawning and Drowsiness
Comp-03 = Classification with Machine Learning Model
Comp-04 = Alert System for Avoiding Drowsiness

Figure-14 Requirements Matrix

4.4. References

- IEEE Computer Society. (2009). IEEE Standard for Information Technology—Systems Design—Software Design Descriptions. IEEE. New York: IEEE. IEEE Std 1016
- Sommerville, I. (2016). Software engineering. Boston: Pearson Education Limited.
- Öztürk, M., KüçükmaniSa, A., & Urhan, O. (2022). Drowsiness detection system based on machine learning using eye state. Balkan journal of electrical and computer engineering, 10(3), 258-263.
- Park, S., Pan, F., Kang, S., & Yoo, C. D. (2017). Driver drowsiness detection system based on feature representation learning using various deep networks. In Asian Conference on Computer Vision (pp. 154-164). Springer, Cham.
- Santoshi, M. G. (2020). AUTOMATED DRIVER DROWSINESS DETECTION FOR NON 2 WHEELERS (Doctoral dissertation, Andhra University).
- BAHARU, B. (2013). DRIVER DROWSINESS DETECTION BY USING WEBCAM.