FINAL PROJECT REPORT

Driver Safety and Drowsiness Detection Simulation

Module	:	**Hot Topic in Software**
Supervisor	:	**Fadi Fayez**
Date of submission	:	**25th June 2018**
Team Members	:	**Wijekoon Somasiri (170001510)** **Herath Premarathne (170001825)**
Emails	:	soma48@manukaumail.com prem26@manukaumail.com

TABLE OF CONTENTS

LIST OF TABLES

TABLE OF FIGURES

1. INTRODUCTION

1.1 Problem Description

Automobile has become a major part in our lives. They are used for transportation of people, items and any many other from one place to another. Whatever the automobile type (car, van, bike etc.) is being used for transportation, the safety should be a number one priority. While transporting people or any other item we should think about the safety of the passengers of the vehicle, public and private and public properties. According to Ministry of Transport, New Zealand (2018) number of road deaths has been increased from 284 to 393 within the period of 2011 to 2018. They also state that the number of reported injuries in transportation has amplified from 11,000 to 13,000 within a period of 5 years until 2017. Above statistics confirms that there should be a lot of improvements in the automobile and transportation industry for safety and the wellbeing of humans as well as the safety of private and public property.

There are various factors which contributes to road injuries and deaths. Some of them are lost control of the vehicle, speeding, alcohol, driver drowsiness, weather conditions, vehicle conditions and so on (Ministry of Transport New Zealand, 2018). Most of the time drivers do not care about the state of themselves prior to driving a vehicle. Even though the driver is tired, sleepy or has consumed alcohol he/she tries to drive a vehicle without any anxiety. As a result, it causes road accidents, injuries, loss of lives and damage to property. There aren't many systems which could predict the conditions of the driver prior to a trivial incident which could assist the driver immensely. Therefore, this project mainly focuses on the driver drowsiness and how it can be identified using computer vision and facial features in order to provide necessary alerts to the driver when required as a warning, so that the driver can decide whether to continue driving or not and take a proactive approach to such incidents rather than a reactive approach.

1.2 Solution and Research Question

Since the road safety is the main priority, most of the high-end vehicle manufacturers implement safety systems in their vehicles. Unfortunately, these vehicles are expensive and most of the people can't afford to buy them. Therefore, our research questions focus on improving the automobile safety practices using computer vision and low-cost hardware and improving the automobile safety practices by monitoring facial gestures of human.

The proposed solution will be implemented using image processing, computer vision and facial recognition techniques to increase the efficiency and the accuracy of the system. A camera will be the main hardware device to capture images of the eyes and this will reduce the cost of other expensive hardware devices such as embedded sensors and chips.

1.3 Scope

Proposed driver safety and drowsiness detection system will be a desktop application at its first stage as a prototype. In this system, a web camera will be used, and it will be placed in front of the driver (user) to capture images of the driver's face and eyes. Simultaneously, the camera will provide live stream data (image frames) to the application to process them with face recognition and image processing techniques. By this process of execution, the system will detect the status of the eye (open eye or close eye) and alert the driver according to the eye status. The application will monitor image frames per second and identify the state of the eye. If the number of frames within the given period of time has more closed eye states compared to opened eye state, the system will provide warning alerts to the driver via a sound.

1.3.1 Aim of the project

The aim of this proposed system is to develop user-friendly, efficient, accurate and low-cost application using image and video processing algorithms to detect driver drowsiness to reduce road accidents and increase driver safety.

1.3.2 Objectives of the project

1. Research on the identified areas relevant to the project and come up with the literature review. The literature review of the project consists of existing driver safety systems and its functionalities along with the comparison table of the existing systems.
2. Design the system architecture according to the gathered information from the research.
3. Implement the finalized design of the proposed system.
4. Test and evaluate the implemented system.
5. Submit the deliverables on time.

1.3.3 Project Deliverables

- Research project proposal template
- Final research project proposal
- Final project report
- Prototype of the system
- Generate driver statistics and save it in a text file so that this data can be used in the future for machine learning.

1.4 Measurable Organizational Value

Proposed solution of this project covers three main impacts; social, customer and financial aspects. As mentioned above in the solution and research question section majority of the population cannot afford to buy a high-end vehicle just for road safety purposes. Therefore, by implementing a proposed system with low cost hardware will open up the opportunity for a vast number of customers. Customers could integrate the proposed system to their vehicles at a low cost compared to systems with sensors and chips. We expect the proposed solution to reduce the number of road accidents by 20 percent in the first year of installation to vehicles in New Zealand which will impact the society immensely. Team expects to take at least 500 customers aboard to the proposed system within the first year and growing the customer base by 10 percent each year. Customers will be attracted to the proposed solution since it is low-cost and the team can increase revenue and make a profit by new installations and maintenance of the proposed system. Since this application is a safety application for transportation it will have a great impact on our society. Not just the driver but also the passengers, public and property will be safe during transportation.

2. LITERATURE REVIEW

2.1 Overview

Studying existing similar systems is a key aspect of this project. It's because we can learn those existing solutions and figure out what are the pros and cons of various systems. This will also help us improve some functionality of a system when added to this proposed project or may be add a whole new feature to our system which doesn't exist in other similar systems. This will be an added value of the proposed system. Each member of the team has analyzed three similar systems. Since these systems are mostly hardware systems and not software systems the exact functionalities couldn't be tested. But relevant information was extracted from internet sources related those products to come up with the literature review of the existing systems. We have explained our findings in the below section and have done a comparison and contrast of the chosen systems to better analyze the current situation in the automobile industry and driver safety & drowsiness detection systems.

2.2 Existing Systems

2.2.1 StopSleep - Electronic Anti-Sleep Alarm

Figure 1: StopSleep Alarm (StopSleep Pty Ltd, 2017)

According to StopSleep Pty Ltd (2017) the aim of the unit is to prevent drivers falling asleep. Being a portable device, it is made to wear on two fingers like a ring. The device alerts through vibration, light and sound. All the device elements are made of cosmetics which are unlikely to cause any allergies including durable plastic. The StopSleep unit has 8 built-in measuring sensors to monitor the heat change of the skin. This system uses the same technology used in lie detector polygraphs. This system generates two types of warnings. They are attention and danger levels. The unit will vibrate in attention warning type while providing vibrations and loud sound in danger situations.

Use of StopSleep – electronic anti-sleep alarm is not hard. First the user has to power on the unit and wait until the sound and light appears. Once this is completed the user can wear the device in two fingers of a hand. Then the sensors will start monitoring for active, micro-sleep and sleep states of a human body and issues the warnings or alarms. The device needs to be charged and it can hold up-to 15 hours within a single charging session (StopSleep Pty Ltd, 2017). Stop sleep wearable device can be purchased at around 200 US dollars (StopSleep Pty Ltd, 2018).

2.2.2 Bosch Driver Drowsiness Detection

Figure 2: Bosch Driver Drowsiness Detection (Bosch, 2012)

Bosch driver drowsiness detection system monitors the steering patterns of a driver and predicts the fatigue and micro sleepiness of a driver. It evaluates and processes about 70 signals to understand the level of drowsiness of a driver. Bosch uses a steering-angle sensor to determine the steering angle and its velocity. The algorithm behinds the Bosch system starts monitoring steering patterns when a trip begins. It monitors for different parameters such as unexpected steering movements, use of signals and many others to identify the fatigue condition of the driver and when it reaches a certain level it informs the driver that he/she requires a rest by flashing a coffee cup sign in the system. This unit is also a part of a car's anti-skid system which help the wheels of the vehicle to keep tractive contact with the road surface (Bosch, 2012).

2.2.3 Anti Sleep Pilot

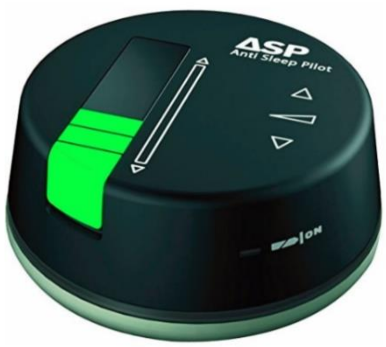


Figure 3: Anti Sleep Pilot (Coxworth, 2011)

Anti sleep pilot is a device which was developed by ASP Technology Ltd, Denmark. It is usually placed on the dashboard of a vehicle to continuously monitor the driver and his/her driving conditions. The device has a light sensor, a sound sensor and a touch sensor and operated using battery power. Initially the driver has to complete an assessment in order to create his/her risk profile. Once this is completed the driver can start using the device. It will automatically start monitoring the driver's fatigue level using 26 parameters including the risk profile. The device conducts random tests so that the driver needs to tap on the device to response to those tests. If the response time is slow the driver will have a slow response time and continuous low response time will warn the driver to take at least a ten-minute break before driving the vehicle again. Anti sleep pilot device also monitors the time and speed of the vehicle, so it can determine the time that the driver actually had the break. This device costs around 250 US dollars (Coxworth, 2011; Asp Technology Ltd, 2018).

2.2.4 Lane Departure Warning System

Figure 4: Standard Lane Departure warning Light on Dashboard (Cottingham, 2018)

In early 2000, Mercedes Benz has invented first lane departure warning system for their heavy duty Actros trucks. After that most of the leading vehicle manufacturers implemented these kinds of systems for their vehicles.

Basically, lane departure warning systems alert the driver when the vehicle drift out of the lane. Nowadays, most of these lane departure warning systems are coming from different technologies of alerting the driver but with the same scenario. Most of the vehicle manufacturers have used low cost cameras mounted in the rear-view mirror which continuously keeps track of the solid lane marks ahead. If the vehicle drifts away from the lane most of the systems alerts the driver by giving visual warning with a beep sound. In advance, some systems vibrate the steering wheel and apply slight steering to the steering wheel to keep the vehicle on track (Cottingham, 2018).

Most of the high-end vehicle manufacturers have implemented these systems with new technologies such as adaptive cruise control and forward collision warning to project the vehicle in front, lane departure warning to project the vehicle by the side and blind spot detection to project the vehicle behind. These systems cost more than 1000 US dollars and only implemented in high-end vehicles. (Howard, 2017)

2.2.5 Drowsiness Detection system using EEG Helmets

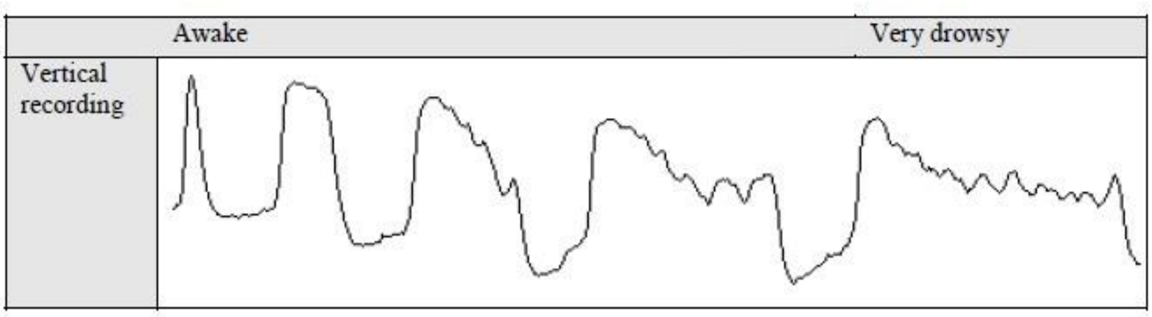


Figure 5: Vertical EEG recording from awake and drowsy condition (Thorslund, 2003)

Physiological measures are most common and effective detectors when it comes to drowsiness detection. Electroencephalogram (EEG) is considered as the most reliable physiological method to detect drowsiness. EEG consists of a method to measure and record the electrical activity of the brain using electrodes placed on the scalp.

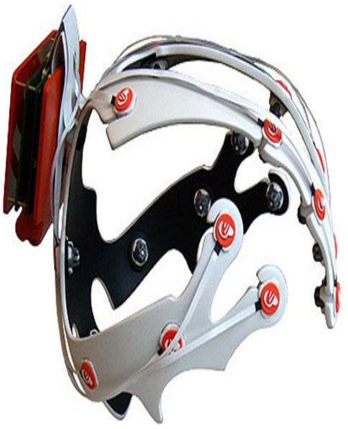


Figure 6: EEG Helmet (Jae-young, 2008)

In this kind of driver safety and drowsiness detection system driver has to wear an EEG electrode mounted helmet to detect the drowsiness. These EEG electrodes detect the rays of the brain. Basically, if these electrodes detect alpha activity of the brain that will become the first indicator of the drowsiness. So, the system alerts the driver by making a sound or vibrating the steering wheel. Above figure 5 shows the changes between awake to drowsiness mode (Awais, Drieberg, & Badruddin, 2014).

Some people do not show any alpha activity even thou he/she in drowsiness mode. To detect this situation EEG helmets should detect theta activity of those people. Therefore, these helmets are considered as high cost driver safety system (Awais, Drieberg, & Badruddin, 2014).

2.2.6 Tesla Auto Pilot

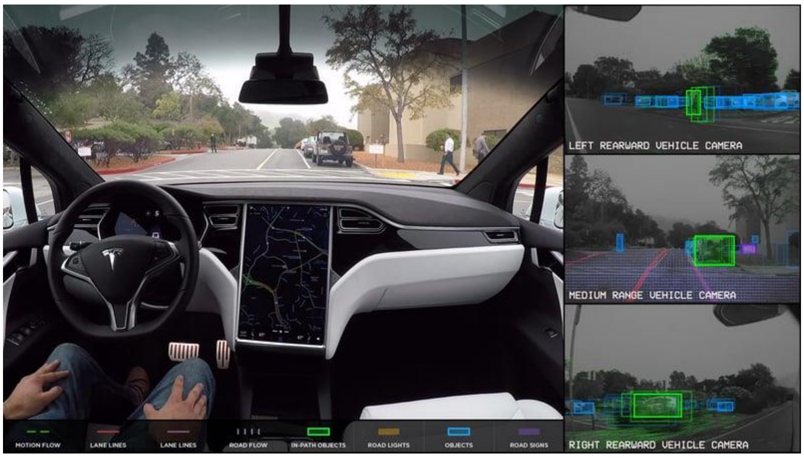

LEFT REARWARD VEHICLE CAMERA
MEDIUM RANGE VEHICLE CAMERA
RIGHT REARWARD VEHICLE CAMERA
MOTION FLOW
LANE LINES
LANE LINES
ROAD FLOW
IN-PATH OBJECTS
ROAD LIGHTS
OBJECTS
ROAD SIGNS

Figure 7: Tesla Auto Pilot (Tesla Inc, 2018)

Tesla auto pilot driver safety system was implemented by Tesla Motors for their vehicles in 2014. This system consists of eight cameras mounted around the vehicle to get 360 degrees visibility up to 250 meters of range. Also, twelve updated ultrasonic sensors for detection of both hard and soft obstacles around the 250-meter range. In addition, a forward-facing radar is providing other data around the car such as rain, fog and dust. Computer process in the vehicle controls the car without human involvement if there is any mistake done by the driver with the assistance of the collected data from the cameras and sensors. This system is an advanced system compared to most of the driver safety systems. Tesla auto pilot model S and X are available in the market for customers and it costs around 80,000 US dollars (Tesla Inc, 2018).

2.3 Comparison of Existing Systems

Table 1: Comparison of existing systems

Existing Systems	***StopSleep - Electronic Anti-Sleep Alarm***	***Bosch Driver Drowsiness Detection***	***Anti-Sleep Pilot***	***Lane Departure Warning System***	***Drowsiness Detection system using EEG Helmets***	***Tesla Auto Pilot***
Developed by	StopSleep Anti-Sleep Alarm PLC	Bosch Germany	ASP Technology Ltd, Denmark	Mercedes Benz	Ford Motor company	Tesla Inc.
Detection method	Skin heat	Steering patterns	Reactions by driver to random tests	Lane boundaries	Alpha activity of human brain	Monitor obstacles around the vehicle
Hardware devices	Eight sensors	Steering angle sensors	Light, sound and touch sensors	Camera	EEG electrode mounted helmet	Eight cameras and twelve ultrasonic sensors
Device Type	Portable	Not Portable	Not Portable	Not Portable	Portable	Not Portable
Alerting Method	Vibrating the device, output light and sound	Flashing a coffee cup sign in the system	Sound	Flashing a warning light with a beep sound	Vibrating the steering wheel with a sound	Control the car without human involvement
Market Price	200 USD	-	250 USD	1000 USD and above	500 – 1000 USD	80,000 USD

2.4 Discussion

In the Literature review the team has focused on some of the existing systems in the market and most of the systems do share common functionalities. Among those, literature review was organized based on six different existing driver warning systems which are listed out in the section 2.2.

StopSleep electronic anti-alarm system seems reasonable since it's a portable device and clients can afford the device for 200 US dollars. This system uses the heat of the skin of the driver to detect the driver's status which is a different approach compared to other systems. Considering its functionalities, driver must wear this device on two of his/her fingers. Then the sensors will start monitoring for active, micro-sleep and sleep states of the driver and issue warnings according to the detected heat signatures. Even though this system can be considered as a good system, deeper investigations on this device show the existence of some issues in it. Human body temperature depends on various factors such as health, weather and climate conditions. In that case there could be doubts to someone on how the system actually detects the state of the driver whether or not he/she can drive the vehicle. Since this device is portable, and driver must wear the device in two fingers, it can annoy and distract the driver as well. In long distance driving, these sensors can demise due to low battery power. Perspiration (sweat) of the fingers after wearing the device for a long time could also allow the sensors not to work properly. Therefore, the team presume that this system cannot be accurate all the time to use as a driver safety system.

Bosch driver drowsiness detection system is another existing system which monitors steering wheel patterns to detect driver drowsiness. Bosch has used steering-angle sensor to determine the steering angle and its velocity. If there are any unusual steering wheel movements, the system will alert the driver by flashing a coffee cup sign lit in the vehicle dashboard. Rather than analyzing only the steering movements, it could have been better to analyze the steering movements with the lane boundaries. Unfortunately, this system doesn't consist of any cameras or sensors to detect lane marks on the road. Moreover, just by flashing a light on the dashboard cannot be considered as a good solution to alert the driver because a driver cannot see any lights when he/she is in drowsy state. Hence drivers cannot rely on this system too much.

Anti pilot is another driver safety system which consists of light, sound and touch sensors. This system monitors the driver and pops random tests to the driver and he/she needs to tap the device within a certain time period as a response. If the response time is too long the system alerts the driver by making a sound. Generally, seems to be a fine solution since the system tries to keep the driver awake. But tapping the device while driving can be a risk and it could distract the driver. Therefore, this system could increase road accidents instead of reducing them.

Mercedes Benz invented the lane departure warning system as a driver safety system in early 2000. In this system a mounted low-cost camera keeps track on the road boundaries by tracking the road lines and alerts the driver if the vehicle drifts out of the lane. But the system can get confused when there are multiple lines drawn on the road. Also, this system monitors the boundaries of the road by tracking the road lines, so it doesn't have any capability of monitoring the roads without lines such as gravel roads. Additionally, this system doesn't have any way of monitoring the road lines in rain, fog and snowing circumstances. In this case, driver cannot depend on the system for safety.

EEG helmets are the most effective way of detecting drowsiness compared to other methods. In this type of system, driver needs to wear the EEG helmet while driving a vehicle. This helmet consists of EEG electrodes and these electrodes detect the alpha rays of the human brain and alerts the driver accordingly. Therefore, this system is one of the best systems to detect driver drowsiness. But wearing a helmet while driving makes the driver uncomfortable and annoying. In addition, long time driving would result in perspiration on the EEG electrodes and reduce the ability of monitoring drowsiness. Also, these helmets are very expensive and can be hard to afford. Moreover, people who doesn't show any alpha activity in the brain must buy a helmet with EEG theta ray detect electrodes. Therefore, these helmets are considered as high-cost driver safety systems.

Tesla auto pilot driver safety system consists of eight cameras and twelve ultrasonic sensors to detect all the hard and soft obstacles around the vehicle. The system will collect all the data and control the car without any human involvement. Therefore, this system is more powerful and effective considering the driver safety. This system can be classified as the most expensive driver safety system among all the existing systems which costs around 80,000 US dollars. As a result of this cost factor, low-end vehicle users cannot afford this type of driver safety systems.

Taking all the facts about the existing systems above into considering, our proposed driver safety and drowsiness detection system targets all the low-end vehicle users who cannot afford expensive systems. Our proposed system will use a camera to get the live video stream of the driver. This camera will be mounted inside the vehicle and the user doesn't have to be worried about rain, fog or any other climate/weather changes. This system will detect drowsiness though human eye open and closure events because a driver must keep his/her eyes open to drive a vehicle. The system will alert the driver by providing a sound if the driver has closed eyes within a certain period of time while driving. Our team believes that this system is simple and affordable to any user compared to all existing systems since our main target is to provide safety in automobile industry using computer vision and low-cost resources such as a camera.

3. SOFTWARE DEVELOPMENT METHODOLOGY

The software development methodology we have chosen to progress with this proposed project is Agile. According to cPRIME (2018) "Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams." In simple words Agile supports to rapid changes in software development unlike other software development methodologies such as waterfall, prototyping etc. The user-requirements can always change and Agile accepts these changes from clients after promoting them via a disciplined project management process which is also called as Agile methods.

According to Beck, et al. (2001) Agile methodology values;

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change rather than following a plan

Since the proposed driver safety and drowsiness detection system is solely based on research, it could stream continuous requirements into the solution. Also, the technologies used in the proposed system are new to the team members. This will allow the members to have more interactions and collaborations in order to come up with a working software since a working prototype is a requirement of this project. Nevertheless, coming up with small deliverable features and continuous improvements as mentioned in Agile methodology will be a good idea over providing a whole complete functionality at once. Taking all the above factors into consideration and the project closely aligns with the above mentioned Agile values, "Agile" was chosen as the software methodology for the proposed project.

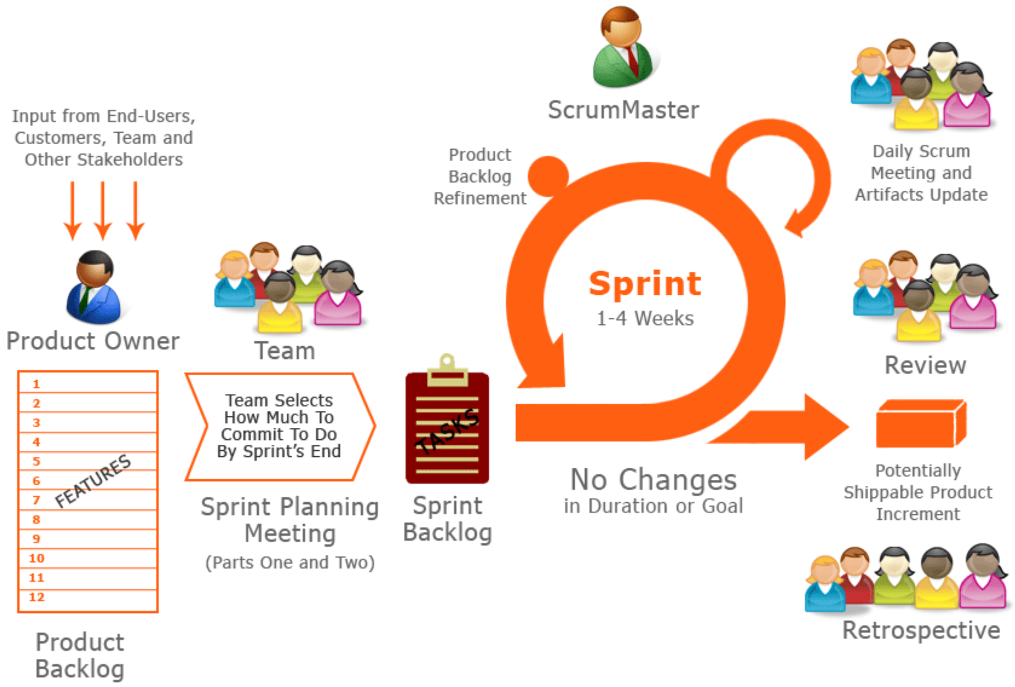

Input from End-Users, Customers, Team and Other Stakeholders
Product Owner
1
2
3
4
5
6
7
8
9
10
11
12
FEATURES
Product Backlog
Team
Team Selects How Much To Commit To Do By Sprint's End
Sprint Planning Meeting
(Parts One and Two)
TASKS
Sprint Backlog
ScrumMaster
Product Backlog Refinement
Sprint
1-4 Weeks
No Changes
in Duration or Goal
Daily Scrum Meeting and Artifacts Update
Review
Potentially Shippable Product Increment
Retrospective

Figure 8: Agile Methodology (Verma, 2015)

According to sinnaps (2018) there are 6 phases in the agile methodology. They are concept, inception, iteration/construction, release, production and retirement. Below is a list of activities done in each phase of the proposed project and we will be ignoring the last two phases (product and retirement) since the proposed system will not be production ready at the end of the project.

1. **Concept phase**
   - Identifying the actual problem and the reasons behind the existing problem in the Automobile industry
   - Identify the research question(s)
   - Carry out research on existing solutions/applications/systems
   - Project risk identification
   - Identification of project risk treatment
   - Define the scope of the project
2. **Inception phase**
   - Form the team
   - Identify the strengths and weaknesses of individual team members
   - Discussion of initial user requirements
   - Estimate the time periods for the overall project
3. **Iteration/Construction phase**
   - Identify the features to be added to the sprint cycle (Requirements gathering)
   - Prioritize the tasks in the sprint
   - Analyze and design feature
   - Implement feature
   - Parallel testing
   - Review code
   - Feedback
4. **Release phase**
   - Quality testing
   - Deliver incremental features
   - Finalize the prototype of the proposed driver warning and drowsiness detection system

The Agile method which will be used in the proposed project is Extreme Programming (XP). This is mainly because it has features such as pre-planning, planning game (spikes), pair programming, testing and in cooperate new features or correct bugs soon compared to other software methodologies. Pre-planning is assisted via the concept of sprint planning. In the Sprint planning the team can decide on which tasks will the team members work before actual tasks. Since the team only has two members pair-programming concept in XP helps this project to move forward without any problems. The team can also perform testing on the developments tasks and always have a progress at the end of sprint cycles. At the end if there are bugs, errors or new client requirements team can include them in the next sprint cycle or in the same sprint cycle if pre-planned.

4. DESIGN AND ANALYSIS

4.1 High Level System Architecture of the Proposed System

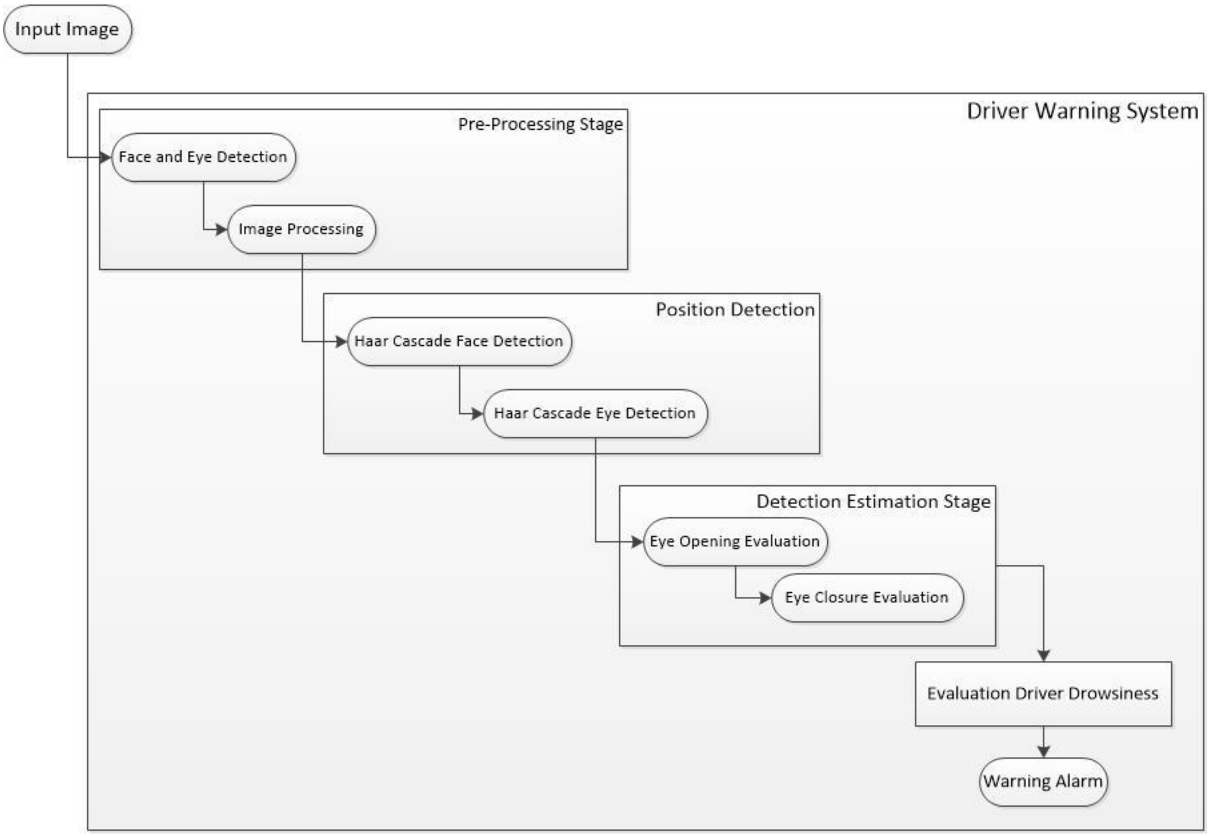


Figure 9: High Level System Architecture

4.1.1 Pre-Processing Stage

Pre-processing stage is the primary stage of the proposed driver safety and drowsiness detection system which consists of two different processes. They are;

- Face and Eye Detection stage
- Image processing

The camera will capture images of the driver and send it through the pre-processing stage to detect the face and eyes of the driver. In this stage image processing algorithms are used to convert the live video stream to a sequence of digital images. Once converted, the system will pass those images to position detection stage.

4.1.2 Position Detection Stage

In this stage, driver safety and drowsiness detection system will use Haar Cascade Classifier to detect the exact positions of the face and eye. Haar Cascade Classifier is a trained data set which is used for object and feature recognition. This stage consists of two different processes to detect face and eye. They are;

- Haar cascade face detection
- Haar cascade eye detection

After these processes those images will be transferred to the next stage of the system.

4.1.3 Detection Estimation Stage

In this stage, the proposed system will determine the state of the eyes. Basically, the system will keep track on eye lids and detect whether both the eyes are in open or closed state. If the eye lids are closed the iris (dark region of the eye) will be invisible and this will be recognized as an eye closure state.

Considering the scenario system will alert the driver by providing a sound if the eye closure state percentage is higher compared to detection percentage supplied by the driver.

4.2 Prototype User Interface wireframe

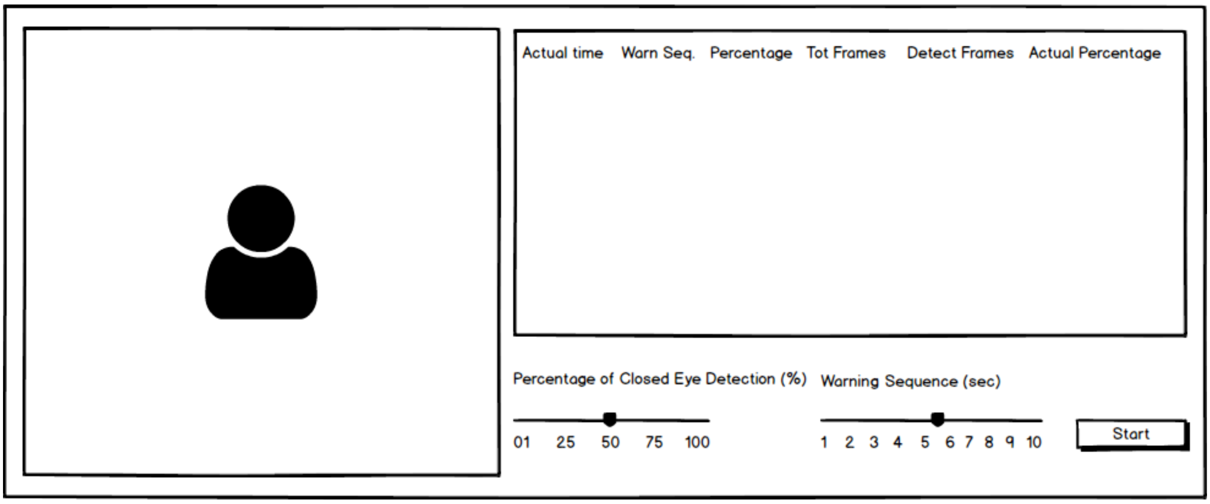
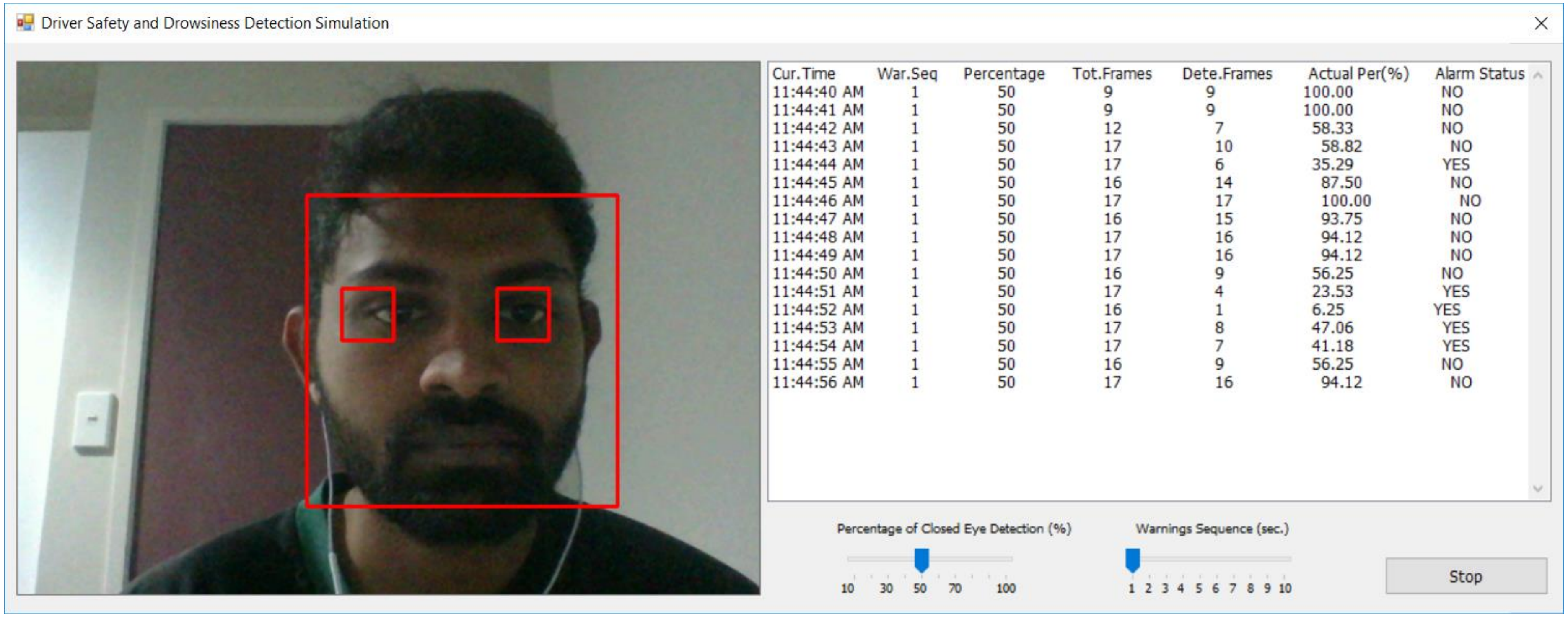


Figure 10: User Interface Wireframe

The proposed driver safety and drowsiness detection system user interface has designed as shown in figure 10. The proposed system will provide the opportunity to the driver to select warning sequence (the time duration that the driver requires the system to process images and provide warnings) and the percentage of detection (the percentage of the detected images with closed eyes within the specified warning sequence) according to his/her preference. Once the warning sequence and detection percentage are set driver can start the system by clicking the start button at the bottom of the system interface.

Also, the system will provide number of statistics in the user interface such as Actual Time, Warning Sequence, detection Percentage, Total Frames counted within the time, Detected frames (images with closed eyes) and Actual Percentage. These statistics will provide the overview of the driver conditions.

4.3 Graphical User Interface



Driver Safety and Drowsiness Detection Simulation
Percentage of Closed Eye Detection (%)
10 30 50 70 100
Warnings Sequence (sec.)
1 2 3 4 5 6 7 8 9 10
Stop

Cur.Time	War.Seq	Percentage	Tot.Frames	Dete.Frames	Actual Per(%)	Alarm Status
11:44:40 AM	1	50	9	9	100.00	NO
11:44:41 AM	1	50	9	9	100.00	NO
11:44:42 AM	1	50	12	7	58.33	NO
11:44:43 AM	1	50	17	10	58.82	NO
11:44:44 AM	1	50	17	6	35.29	YES
11:44:45 AM	1	50	16	14	87.50	NO
11:44:46 AM	1	50	17	17	100.00	NO
11:44:47 AM	1	50	16	15	93.75	NO
11:44:48 AM	1	50	17	16	94.12	NO
11:44:49 AM	1	50	17	16	94.12	NO
11:44:50 AM	1	50	16	9	56.25	NO
11:44:51 AM	1	50	17	4	23.53	YES
11:44:52 AM	1	50	16	1	6.25	YES
11:44:53 AM	1	50	17	8	47.06	YES
11:44:54 AM	1	50	17	7	41.18	YES
11:44:55 AM	1	50	16	9	56.25	NO
11:44:56 AM	1	50	17	16	94.12	NO

Figure 11: Graphical User Interface of Driver Safety & Drowsiness Detection Simulation

Above shown is the implemented Graphical User Interface of the wireframe of the proposed solution mentioned in section 4.2. Below are the descriptions of the driver statistics shown on the right-hand side of the GUI.

Cur.Time = Current Time, War.Seq = Warning Sequence, Percentage = Percentage of Closed Eye Detection, Tot.Frames = Total number of frames captured by the web camera within the selected warning sequence, Dete.Frames = Number of image frames identified as frames with opened eyes, Actual Per = Actual closed eye percentage, Alarm Status = This denotes whether the alarm was activated or not.

4.4 Use Case Diagram

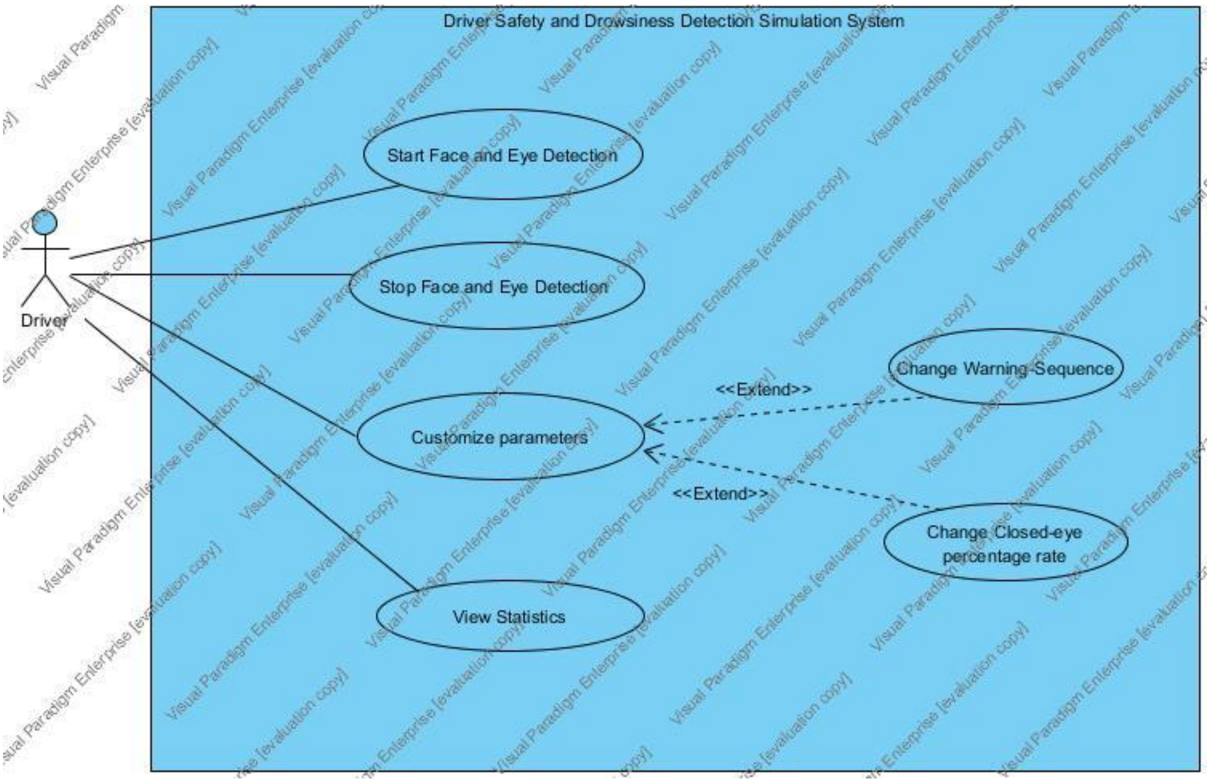

Driver Safety and Drowsiness Detection Simulation System
Driver
Start Face and Eye Detection
Stop Face and Eye Detection
Customize parameters
View Statistics
Change Warning Sequence
Change Closed-eye percentage rate
<<Extend>>
<<Extend>>

Figure 12: Use Case Diagram

In the Driver Safety and Drowsiness Detection Simulation system we have identified a single actor being a driver. Driver can perform the below four use cases in the proposed simulation system. They are;

1. Start the simulation to perform face and eye detection
2. Stop the simulation to terminate face and eye detection
3. Drive can customize parameters such as changing the warning sequence or change the closed-eye percentage rate
4. Driver can view the statistics generated by analyzing the live video feed fetched via the web camera.

4.5 Activity Diagram

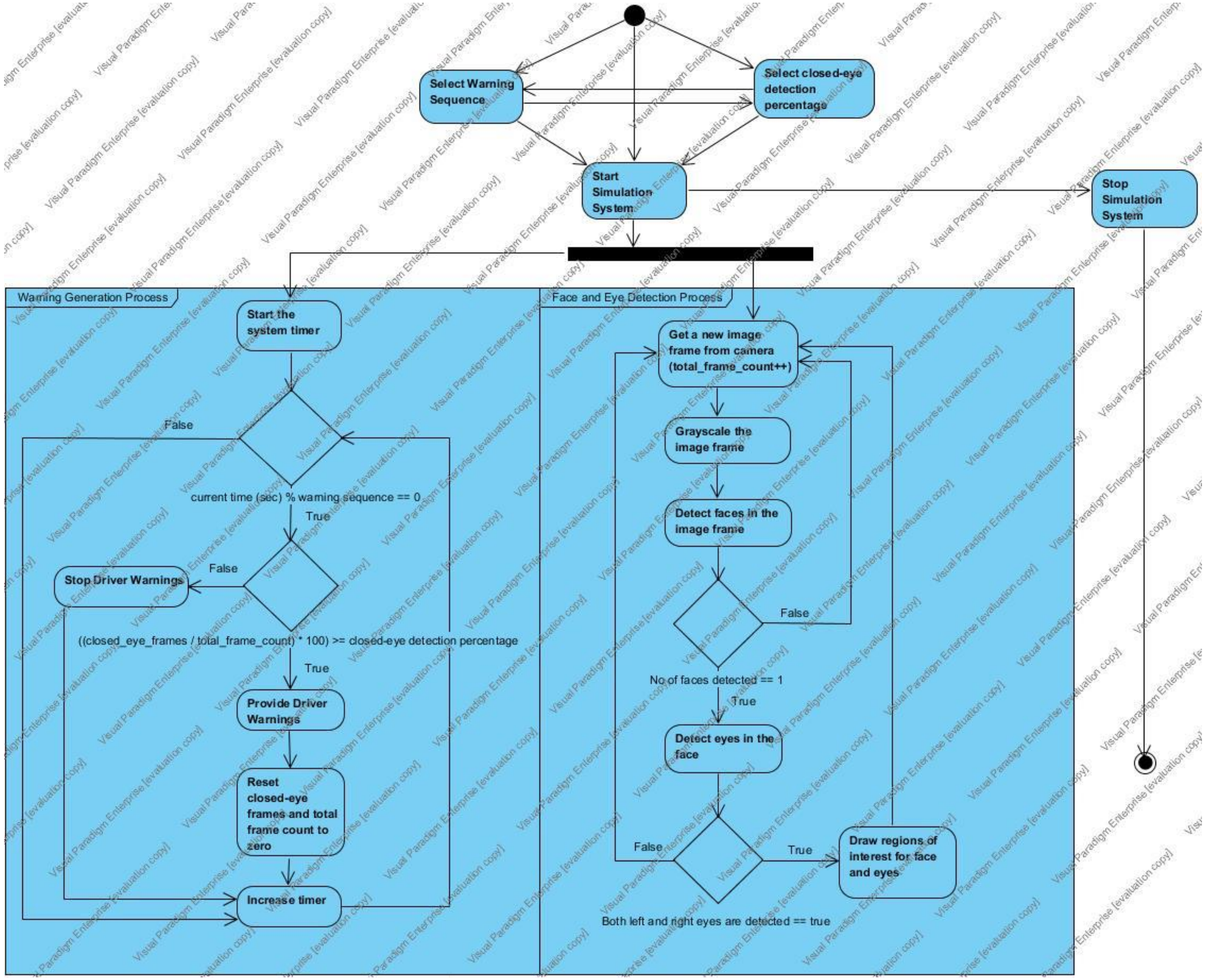

Select Warning Sequence
Select closed-eye detection percentage
Start Simulation System
Stop Simulation System
Warning Generation Process
Start the system timer
False
current time (sec) % warning sequence == 0
True
False
Stop Driver Warnings
((closed_eye_frames / total_frame_count) * 100) >= closed-eye detection percentage
True
Provide Driver Warnings
Reset closed-eye frames and total frame count to zero
Increase timer
Face and Eye Detection Process
Get a new image frame from camera (total_frame_count++)
Grayscale the image frame
Detect faces in the image frame
False
No of faces detected == 1
True
Detect eyes in the face
False
True
Draw regions of interest for face and eyes
Both left and right eyes are detected == true

Figure 13: Activity Diagram

Driver Safety and Drowsiness Detection Simulation system has two main processes which run concurrently. They are warning generation process and the other being face and eye detection process. The Driver can start the simulation system with the default values or can customize the closed-eye detection percentage and warning sequence parameters accordingly prior to starting the simulation. If the driver prefers he can also change the parameters after starting the simulation. Once the simulation is started both the above-mentioned main processes starts and runs asynchronously. Warning generation process depends on the face and eye detection process. Driver can also stop the simulation when required.

In the face and eye detection process, the integrated web camera will capture the live video stream. Video stream is a sequence of image frames and in this process all the image frames are distinguished from the video stream. Each of the image frames are converted to gray images (which is also known as gray-scaling) so that the noise will be removed from the color images and it will ease the computational process of those images. Grey-scaling will also eliminate all the unwanted color channels, so it will reduce the time to process images. Once the images have been converted to gray-scaled images the simulation uses the existing HaarCascade face and eye datasets in order to process the gray-scaled images to detect the face and opened-eyes. First it tries to detect the face in the image and if the simulation couldn't detect the face it grabs the next image frame while incrementing the "closed eye count" variable. If the face is detected the system makes sure that the detected face count is exactly one. This is since we are not interested in multiple face detection in this particular simulation. When the face is detected, the simulation will start detection of both left and rights eyes. If the eyes are not detected "closed eye count" variable is incremented, and it will move to the next image frame in the sequence. If both eyes are detected, then the system will draw a rectangular box around the face as well as the eyes to denote that the eyes have been detected by the system. This process continuously occurs until the simulation is forcefully terminated by the driver.

Warning generation process runs on a separate thread in the simulation. First it checks whether the current seconds spent in the process is equal to the warning sequence selected by the driver. If not, the simulation doesn't do anything. But if it's the selected warning sequence by the driver then the simulation calculates the actual closed-eye percentage for that time sequence as below.

Actual closed-eye percentage (%) = (number of frames with closed eyes/total frames) * 100

If the actual closed-eye percentage rate is greater than or equal to driver customized closed-eye percentage rate, then the alarm will be generated. Otherwise the alarm will be stopped. Last but not least, the system will also write all the driver statistics to a file as an added value so that these data can be used for machine learning purposes as future enhancements of this simulation.

4.6 Class Diagram

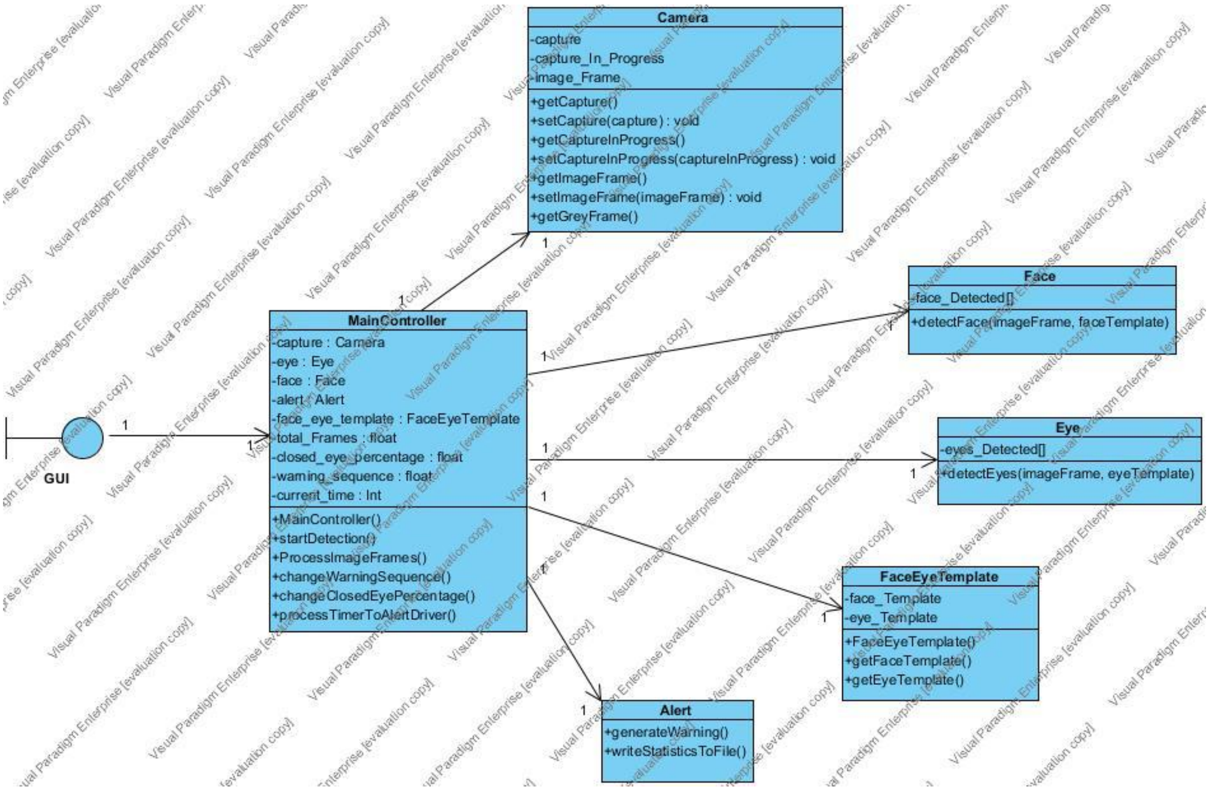


Figure 14: Class Diagram

In the simulation system, the team identified six main classes as shown in the above diagram and they are Alert, Camera, Face, Eye and FaceEyeTemplate and the controller. The controller class controls all the other classes and passes all the messages from the Graphical User Interface to classes and vice versa. Controller class more over runs concurrent processes which are warning process and face and eye detection process to produce warnings to the driver.

Camera class is responsible mostly for the web camera related functionalities such as accessing camera and live video stream, chopping live video stream into image frames, gray scaling etc.

FaceEyeTemplate class is responsible for loading the Haar Cascade face and eye datasets to the simulation system.

Face and Eye classes are accountable for face and eye detection processes respectively while the Alert class creates sound alarms and write driver statistics to the text file.

Each of the above-mentioned classes have a “one-to-one” relationship with the controller.

4.7 Sequence Diagram

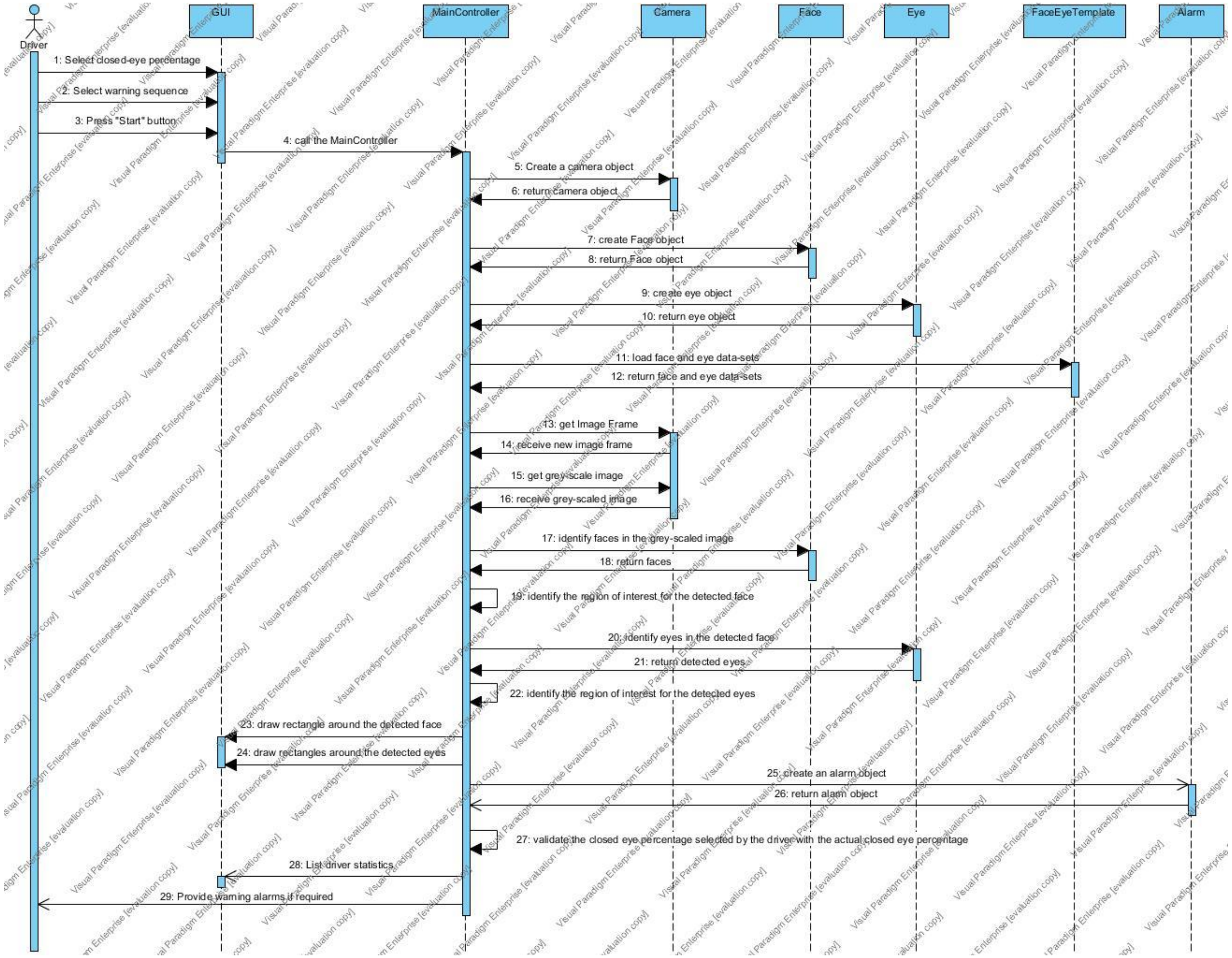

Driver
GUI
MainController
Camera
Face
Eye
FaceEyeTemplate
Alarm
1: Select closed-eye percentage
2: Select warning sequence
3: Press "Start" button
4: call the MainController
5: Create a camera object
6: return camera object
7: create Face object
8: return Face object
9: create eye object
10: return eye object
11: load face and eye data-sets
12: return face and eye data-sets
13: get Image Frame
14: receive new image frame
15: get grey-scale image
16: receive grey-scaled image
17: identify faces in the grey-scaled image
18: return faces
19: identify the region of interest for the detected face
20: identify eyes in the detected face
21: return detected eyes
22: identify the region of interest for the detected eyes
23: draw rectangle around the detected face
24: draw rectangles around the detected eyes
25: create an alarm object
26: return alarm object
27: validate the closed eye percentage selected by the driver with the actual closed eye percentage
28: List driver statistics
29: Provide warning alarms if required

Figure 15: Sequence Diagram

When the driver is navigated to the Graphical User Interface of the simulation system he/she can perform 3 main tasks. They are selection of closed eye percentage rate, selection of warning sequence and start the detection process by clicking on the “Start” button. After starting the detection process Main controller becomes responsible of the whole simulation.

MainController class initiates creation of objects such as Camera, Face, Eye, FaceEyeTemplate since these objects are accountable for different functionalities which are really important for further process of the simulation. First the controller captures the image from the live video stream and using Camera object it converts the image to a gray-scaled image. This gray-scaled image is then sent to the Face class and retrieved the face and the areas of the faces are is recognized. Now the controller communicates with the Eye class to identify the eyes in the image frame and the controller identifies the area of interest for eyes in it. When eyes are detected the controller draws a red rectangle around the face and both eyes so that it will give an idea to the user.

MainController class concurrently checks the actual closed-eye percentage as described in section 4.5 and if the percentage is less than or equal to driver selected closed-eye percentage it will create an object of Alert class and generate the warnings. The statistics will be displayed in the Graphical User Interface as well and also it will be written to a text file.

5. RISK MANAGEMENT

Risk management is a crucial process in any project. If the risks aren't properly identified and treated, it will affect the whole project ending up with a project failure. Project failures will not only affect the project team. The companies will lose their clients, partners and other stakeholders and also will give a bad reputation to the team and the responsible organization. These will affect the upcoming projects and the teams/organizations will be financially unstable to carry out further processes on projects. Hence, the risks which are associated with the proposed project will be identified initially. Once the risks have been identified we will be discussing how the mitigate the risk so that those risks will not affect the overall progress of the proposed project.

5.1 Identification of risks associated with the proposed project

1. **Lack of experts/skilled people in the research areas such as computer vision and machine learning**

In the proposed project the main research areas are video processing, image processing, face and eye detection and compare datasets to provide warnings. These concepts are relatively new to both the team members and this project will require a lot of research in order to identify on how to capture face and eyes from a live video stream and identify the eye open and closure states of the driver. Hence, this is identified as a potential risk in our project.

2. **Time management and estimations can be inaccurate**

The time allocated for the whole project is eight weeks. Within this 8-week period the team members have to make sure that the project deliverables are met within the deadlines. Since the proposed project has so much of research to be done and due to lack of expertise in the research areas, the time allocated for each task can consume more time than expected. This could affect the time of the following tasks and the resulting the project to fail.

3. **Continuous growth of the project scope and ambiguity of requirements**

So far in the proposed project we have identified our main research areas. They are video/image processing, face and eye detection and research on datasets. But when research continuously progresses, it could introduce new requirements and other concepts to the project which they team members wouldn't have even thought of. This could be one of the major risks in the project which could affect the project immensely.

4. **Potential data loss as a result of technology failures**

Technology risk is unavoidable in a project like this. No one can predict what will happen in the next minute of time. The development workstations might crash. If it happens it will be a shocking loss to the team member as it will affect completion of the project as well as affecting the final grades of the hot topic in software module as a student.

5. **Communication and collaboration between team members**

Both the team members are in the final quarter of the study program. Therefore, along with the hot topic module team has to focus on the final year project (last quarter) as well. There could be instances where the members will focus more on the final year project. If the work is not properly balanced, less communication, less team collaboration and no support to each other will become a burden for a single member of the team or both the members. This will create extra pressure in the team and it will affect the final year project as well.

6. **Scope of the proposed project is comparatively large within the eight-week time period**

The proposed project has a considerable amount of research areas. The team doesn't have the experts in these research areas and it can consume a lot of time for the team members to figure out the concepts and apply them to the proposed system. Hence, this could really affect the project in the current time allocations.

5.2 Risk Matrix

The numbering used in the section 4.1 will be used to uniquely identify the risks associated with the proposed project in the below risk matrix.

Table 2: Risk matrix

RISK IMPACT	***Severe***		**4**	**2**	**3**	**1, 6**
	Significant		**5**			
	Moderate					
	Minor					
	Minimal					
		Almost Never	***Possible but unlikely***	***Possible***	***Highly likely***	***Almost certain***
	LIKELIHOOD OF RISK OCCURANCE					

Table 3: Risk color codes

COLOUR	***RISK SEVERITY***
🟥	High risks
🟧	Moderate risks

Risk 1,2,3 are the high impact risks in the proposed project while 4 and 5 are moderate risks. By the above risk matrix, we have identified the severity of each risk and impact if occurred and below shows the priority of the identified risks.

1. Lack of experts/skilled people in the research areas such as computer vision and machine learning
2. Scope of the proposed project is comparatively large within the eight-week time period
3. Continuous growth of the project scope and ambiguity of requirements
4. Time management and estimations can be inaccurate
5. Potential data loss as a result of technology failures
6. Communication and collaboration between team members

In the next section "Risk Mitigation" we discuss the ways on how the team can mitigate the risks in order to make this proposed project a success.

5.3 Risk Mitigation

Table 4: Risk mitigation

IDENTIFIED RISKS	***RISK MITIGATION***
Lack of experts/skilled people in the research areas such as computer vision and machine learning	• Read articles, research papers, online resources and other materials related to computer vision and machine learning concepts and familiarize with them. • Identify the exact computer vision and machine learning concepts required for the proposed project (Video processing, Image processing, Datasets, Face and eye detection). • Do online tutorials and practice development tasks in above identified areas. • Request supervisor assistance.
Scope of the proposed project is comparatively large within the eight-week time period	• Divide the research areas of the proposed project among the team members. • Try to reduce the complexity of different concepts in the research areas by following online tutorials and programming videos. • Request supervisor assistance.
Continuous growth of the project scope and ambiguity of requirements	• Identify the project requirements at the initial phases of the project. • Identify the exact research areas for the proposed project. • Define the scope of the project. • Have frequent team meetings and consult supervisor for proper planning of the functionality of the system.
Time management and estimations can be inaccurate	• Identify all the tasks which needs to be executed in the proposed project. • Create a Gantt chart including various tasks, it's allocated time durations, milestones and the task deadlines. • Conduct frequent team meetings to monitor, review project progress and make amendments if required.
Potential data loss as a result of technology failures	• Frequent commits of project related work to a remote repository such as GitHub. • Take frequent back-ups of the project work to an external hard drive/USB flash drive.
Communication and collaboration between team members	• Carry out frequent team meetings. • Monitor the progress of each team member using GitHub remote repository. • Identify the issues related to the proposed project which each team member currently faces and support them to overcome such situations. • Consult supervisor for assistance.

6. IMPLEMENTATION

6.1 Selection of Technologies/Resources

Implementation was one of the main phases since a prototype was a main deliverable of this project. Hence the team had to work hard on the implementation phase to get a stable prototype for successful completion of the project. Team only had approximately two weeks to complete coding and due to lack of expertise in the computer vision concepts the team had to do more spikes in order to understand the concepts. Team understood the difficulty of implementing face and eye detection from scratch and started researching on the existing libraries so that the development time would be reduced and could come up with a better prototype. In the research for possible image processing third-party libraries team identified three main libraries which are popular in the development world. They are OpenCV, EmguCV and MATLAB. OpenCV is based on optimized C and C++. EmguCV is the wrapper class for OpenCV C/C++ libraries where C# is the programming language. MATLAB has its own scripting language for computer vision concepts. Therefore, the team needed to choose one from the above third-party libraries for the implementation of the proposed simulation system.

Even though performance wise choosing OpenCV was the best option one team member wasn't familiar with C++ programming language. Since the team has decided to follow the extreme programming (pair programming) agile method and lack of expertise in C++ team decided not to use OpenCV C/C++ library. MATLAB was also a possible option. But both the team members are new to MATLAB scripting language and couldn't spend time to learn a new language to implement the proposed simulation team decided to ignore MATLAB as well. Fortunately, the team was fluent with C# programming language and with lots of experience in that paradigm, team agreed upon choosing EmguCV as the third-party library to implement the proposed simulation.

In order to perform face and eye detection team needed to find possible data-sets. After research we found Haar-cascade classifiers for face and eyes which had been already trained for commercial use. The team did not train these data-sets since it could have been time consuming and it would have affected the overall development time. Hence, the team decided to go ahead with the existing data-sets in the market to develop the prototype.

The development environment was chosen as Microsoft Visual Studio 2017 while GitHub was used as code and document repository of the driver safety and drowsiness detection simulation. All the related documents and source code of the prototype can be accessed via the public GitHub URL here. https://github.com/ssomlk/Hot_Toipics_In_Software

To summarize below are the technologies and resources used in the project.

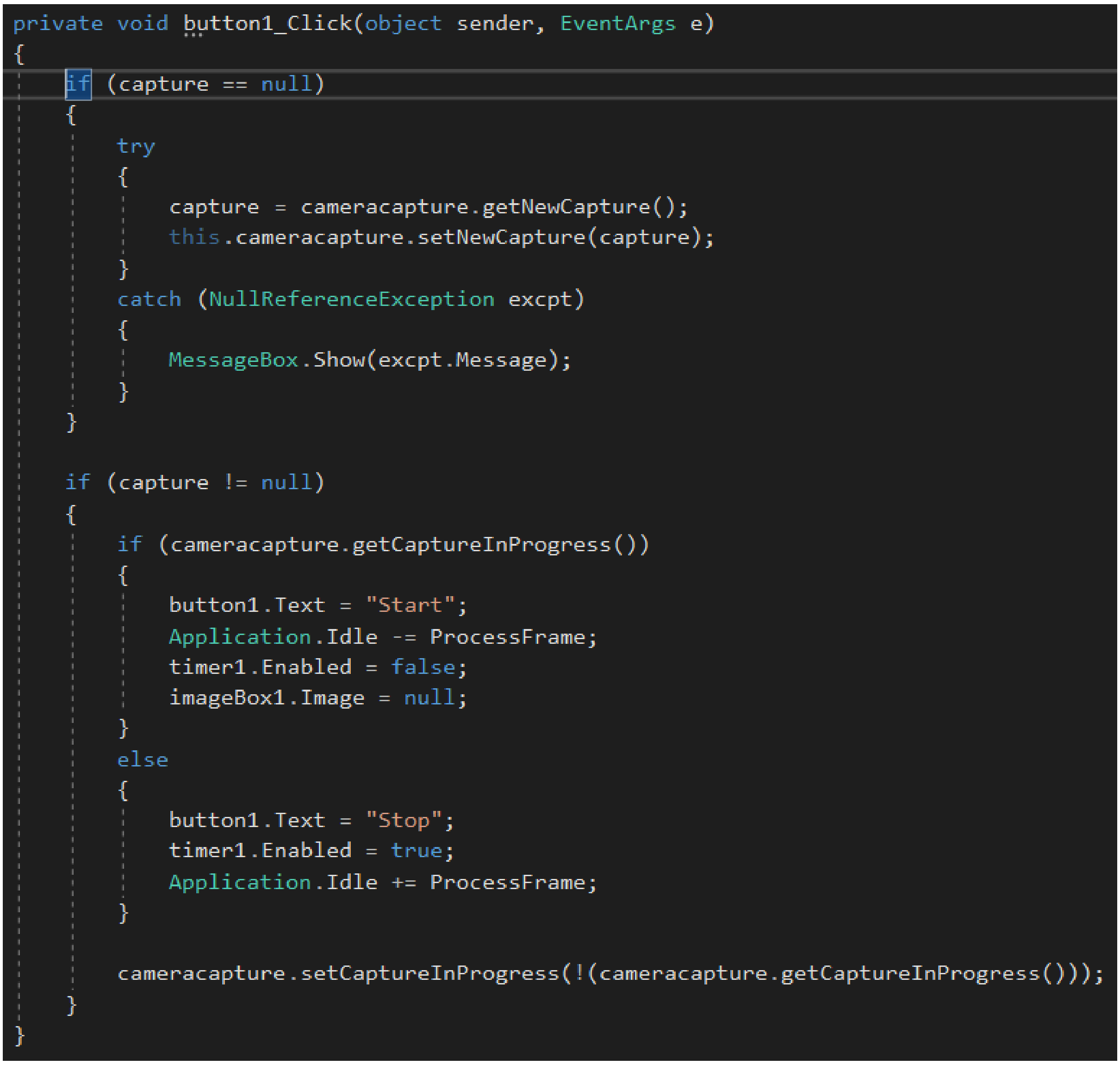
- EmgvCV – covers the computer vision concepts
- C# .Net – programming language
- Haar-Cascade classifiers – face and eye datasets
- Microsoft Visual Studio 2017 – Development environment
- GitHub – code and document repository

6.2 Coding

6.2.1 Graphical User Interface (GUI) of Simulation System

GUI of the driver safety and drowsiness detection simulation consists of 5 main components. They are image box (used to show the live video feed taken from the web camera), list box (lists down the driver statistics), two track bars (one for warning sequence selection and the other to select the eye percentage as driver's preference) and the button to start and stop the system. Apart from these visible components there is also a timer (an invisible visual studio component) running in the system to identify the time of the running system so that it will be responsible for generating the warning alarms and writing driver statistics to the file system. A screen shot of the GUI can be seen in the section 4.3 as well.

6.2.2 Starting and Stopping the Simulation System

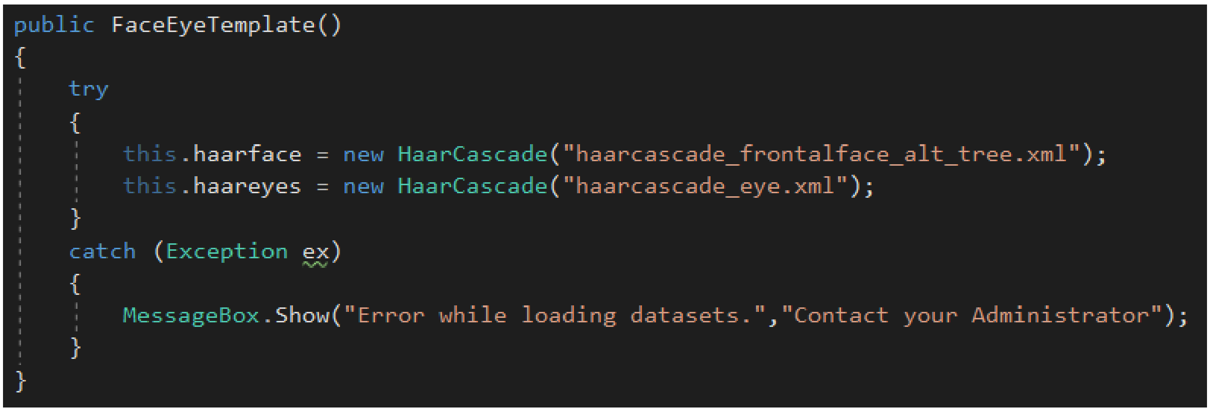


```
private void button1_Click(object sender, EventArgs e)
{
    if (capture == null)
    {
        try
        {
            capture = cameracapture.getNewCapture();
            this.cameracapture.setNewCapture(capture);
        }
        catch (NullReferenceException excpt)
        {
            MessageBox.Show(excpt.Message);
        }
    }

    if (capture != null)
    {
        if (cameracapture.getCaptureInProgress())
        {
            button1.Text = "Start";
            Application.Idle -= ProcessFrame;
            timer1.Enabled = false;
            imageBox1.Image = null;
        }
        else
        {
            button1.Text = "Stop";
            timer1.Enabled = true;
            Application.Idle += ProcessFrame;
        }

        cameracapture.setCaptureInProgress(!(cameracapture.getCaptureInProgress()));
    }
}
```

Figure 16: System startup and shutdown

In the simulation system when the button is clicked for the first time it starts up the processes of generating warnings as well as the face and eye detection which were mentioned in the activity diagram (section 4.5). System checks whether it can access the web camera in order to gain access to the camera video data before actually starting the face and eye detection process. Function “ProcessFrame” is responsible for processing the actual image frames captured from the camera so that the face and eyes can be detected for warning generation depending on the eye closed percentage.

6.2.3 Loading Face and Eye Data sets

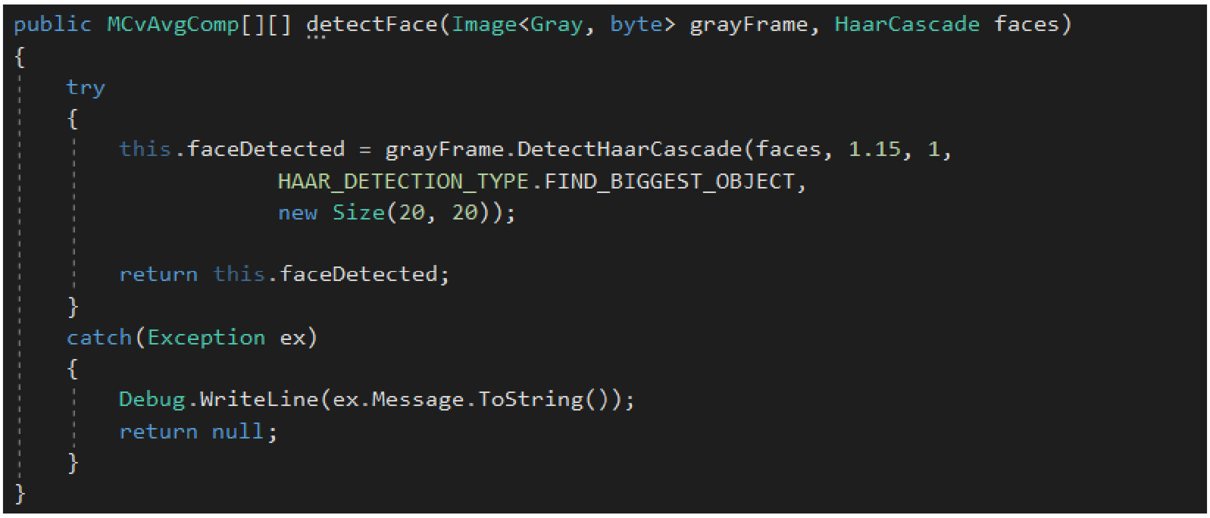


```
public FaceEyeTemplate()
{
    try
    {
        this.haarface = new HaarCascade("haarcascade_frontalface_alt_tree.xml");
        this.haareyes = new HaarCascade("haarcascade_eye.xml");
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error while loading datasets.","Contact your Administrator");
    }
}
```

Figure 17: Load face and eye data sets

When the system starts, initially the system loads all the face and eye datasets from the .xml files. “haarcascade_frontalface_alt_tree.xml” is responsible for loading face data while “haarcascade_eye.xml” is responsible for loading eye data to the system so that the system can compare and contrast the live data with existing datasets to detect face and eyes.

6.2.4 Face Detection

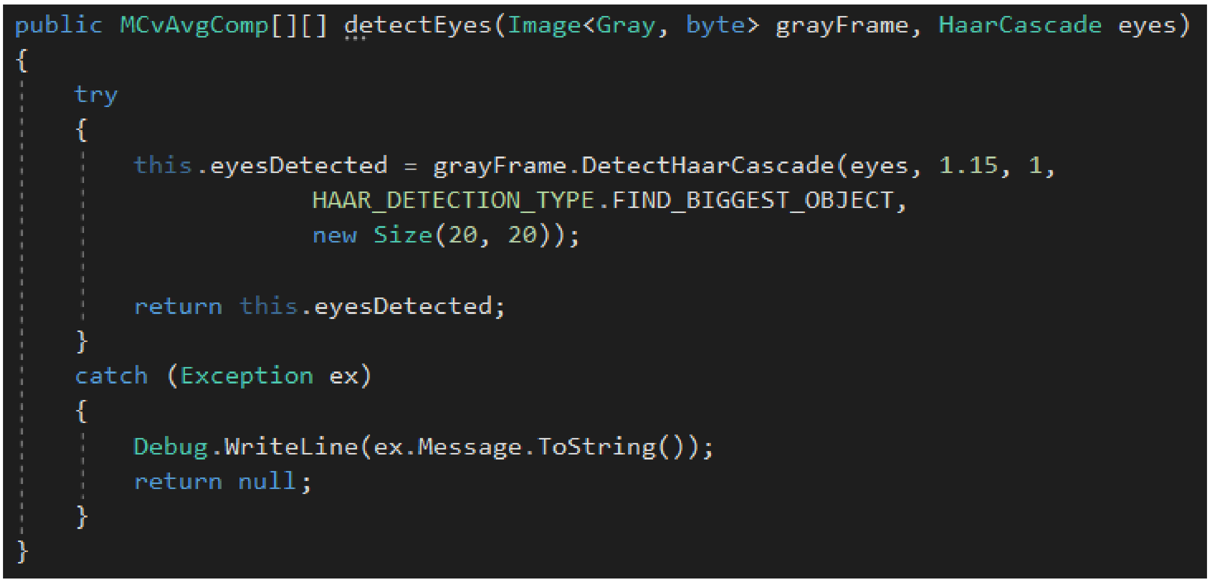


```
public MCvAvgComp[][] detectFace(Image<Gray, byte> grayFrame, HaarCascade faces)
{
    try
    {
        this.faceDetected = grayFrame.DetectHaarCascade(faces, 1.15, 1,
                    HAAR_DETECTION_TYPE.FIND_BIGGEST_OBJECT,
                    new Size(20, 20));

        return this.faceDetected;
    }
    catch(Exception ex)
    {
        Debug.WriteLine(ex.Message.ToString());
        return null;
    }
}
```

Figure 18: Face detection

Above shown code snippet is responsible for the actual face detection in the simulation. The image frame is passed along with the face dataset and the EmguCV “DetectHaarCascade” function detects and identifies the actual faces shown in the image frame. Once detected, all the detected faces are returned back to the “ProcessFrame” function to start eye detection in the image.

6.2.5 Eye Detection

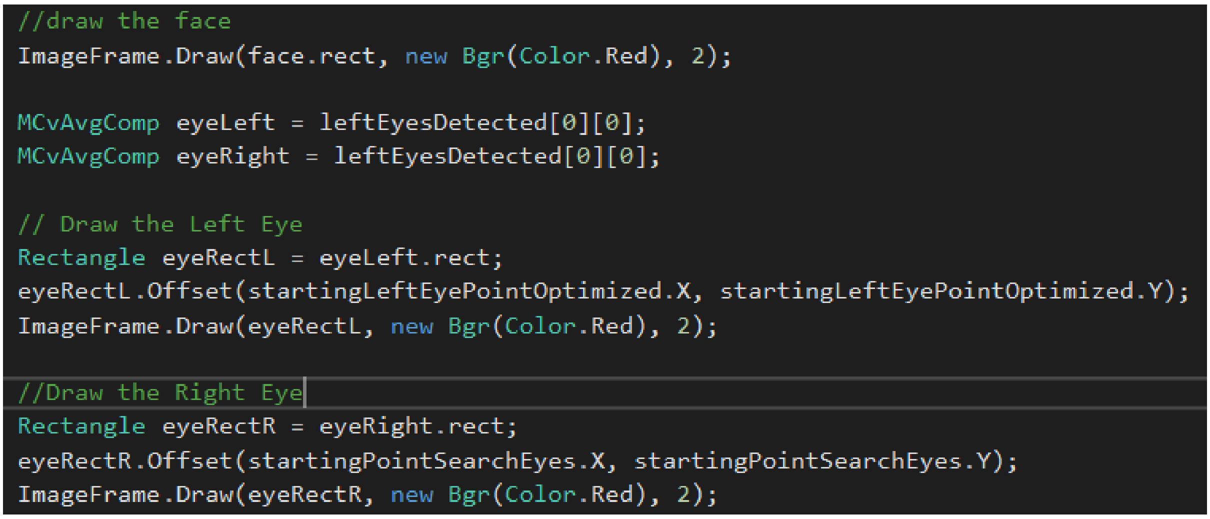


```
public MCvAvgComp[][] detectEyes(Image<Gray, byte> grayFrame, HaarCascade eyes)
{
    try
    {
        this.eyesDetected = grayFrame.DetectHaarCascade(eyes, 1.15, 1,
                    HAAR_DETECTION_TYPE.FIND_BIGGEST_OBJECT,
                    new Size(20, 20));

        return this.eyesDetected;
    }
    catch (Exception ex)
    {
        Debug.WriteLine(ex.Message.ToString());
        return null;
    }
}
```

Figure 19: Eye detection

Above shown code snippet is responsible for the actual eyes detection in the simulation. The image frame is passed along with the eye dataset and the EmguCV “DetectHaarCascade” function detects and identifies the actual eyes shown in the image frame. Once detected, all the detected eyes are returned back to the “ProcessFrame” function so that the simulation could draw red rectangles around the detected faces and eyes.

6.2.6 Draw Face and Eye Regions

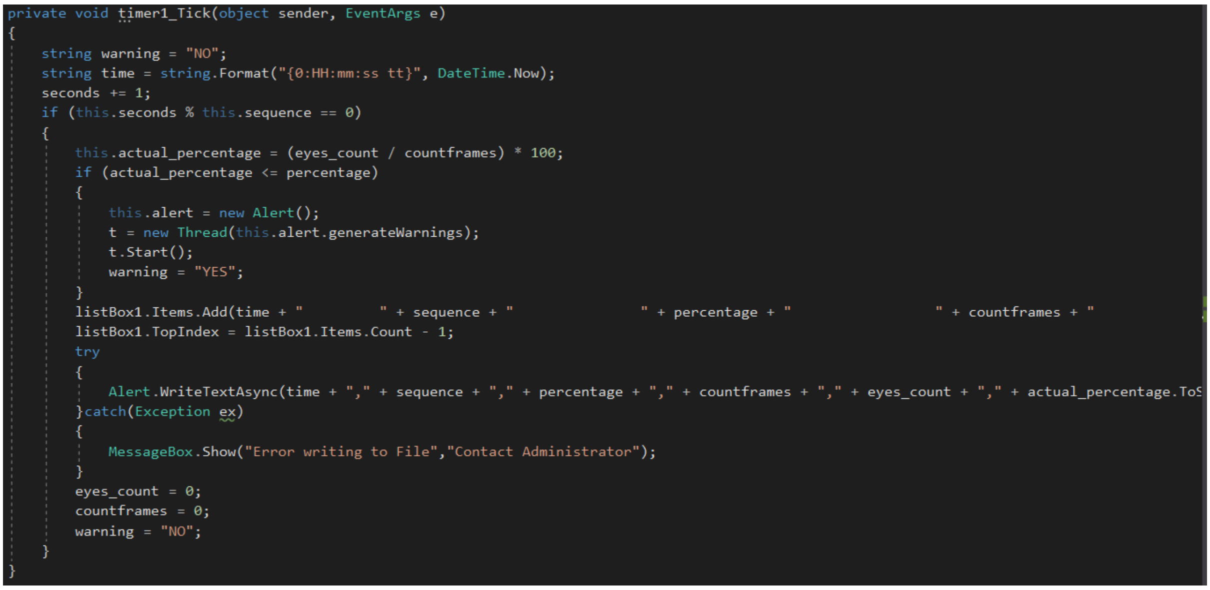


```
//draw the face
ImageFrame.Draw(face.rect, new Bgr(Color.Red), 2);

MCvAvgComp eyeLeft = leftEyesDetected[0][0];
MCvAvgComp eyeRight = leftEyesDetected[0][0];

// Draw the Left Eye
Rectangle eyeRectL = eyeLeft.rect;
eyeRectL.Offset(startingLeftEyePointOptimized.X, startingLeftEyePointOptimized.Y);
ImageFrame.Draw(eyeRectL, new Bgr(Color.Red), 2);

//Draw the Right Eye
Rectangle eyeRectR = eyeRight.rect;
eyeRectR.Offset(startingPointSearchEyes.X, startingPointSearchEyes.Y);
ImageFrame.Draw(eyeRectR, new Bgr(Color.Red), 2);
```

Figure 20: Draw face and eye regions

Separately identified face, left eye and right eye boundaries will be shown by a red rectangle drawn around them. This is an indication to the user that the face and eyes have been detected by the simulation system. If the eyes are not identified all of the above-mentioned boundaries will not be drawn.

6.2.7 Warning Generation Process

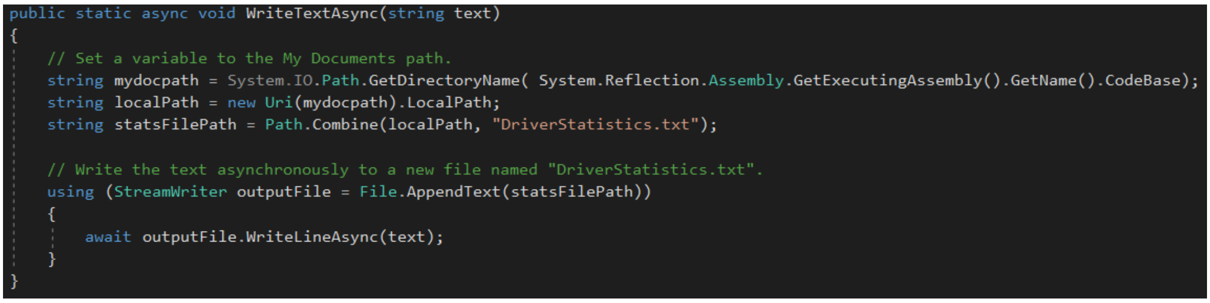


```
private void timer1_Tick(object sender, EventArgs e)
{
    string warning = "NO";
    string time = string.Format("{0:HH:mm:ss tt}", DateTime.Now);
    seconds += 1;
    if (this.seconds % this.sequence == 0)
    {
        this.actual_percentage = (eyes_count / countframes) * 100;
        if (actual_percentage <= percentage)
        {
            this.alert = new Alert();
            t = new Thread(this.alert.generateWarnings);
            t.Start();
            warning = "YES";
        }
        listBox1.Items.Add(time + "        " + sequence + "               " + percentage + "                 " + countframes + "
        listBox1.TopIndex = listBox1.Items.Count - 1;
        try
        {
            Alert.WriteTextAsync(time + "," + sequence + "," + percentage + "," + countframes + "," + eyes_count + "," + actual_percentage.ToS
        }catch(Exception ex)
        {
            MessageBox.Show("Error writing to File","Contact Administrator");
        }
        eyes_count = 0;
        countframes = 0;
        warning = "NO";
    }
}
```

Figure 21: Warning generation process

Above code snippet is accountable for warning alarm generation, updating the list box of driver statistics and saving those statistics into the text file. This function runs on a separate thread (since we are using the timer component). The actual percentage is calculated using the total images with detected eyes divided by the total image frames and multiplying it by 100. This gives you the actual eye percentage rate and if the actual rate is less than or equal to the driver selected eye percentage rate it will create another thread in order to generate the warning alarm.

6.2.8 Creating Driver Statistics File



```
public static async void WriteTextAsync(string text)
{
    // Set a variable to the My Documents path.
    string mydocpath = System.IO.Path.GetDirectoryName( System.Reflection.Assembly.GetExecutingAssembly().GetName().CodeBase);
    string localPath = new Uri(mydocpath).LocalPath;
    string statsFilePath = Path.Combine(localPath, "DriverStatistics.txt");

    // Write the text asynchronously to a new file named "DriverStatistics.txt".
    using (StreamWriter outputFile = File.AppendText(statsFilePath))
    {
        await outputFile.WriteLineAsync(text);
    }
}
```

Figure 22: Creating driver statistics file

Above code creates a text file named “DriverStatistics.txt” and append all the data received from the list box. The file is created in the Debug folder of the simulation system at the moment and this is an added value of the system. This is because the data created in this file can be used for deep learning so that the simulation can automatically identify the eye percentages and warning sequence which could be appropriate without human interaction.

7. TESTING AND ERROR HANDLING

Testing and Error handling is the most crucial part of this simulation. As this application mainly focuses on safety of the users (driver) in automobile industry. Therefore, testing needs to be done to make sure that the simulation is stable, and the users will not encounter any unexpected issues while using the application. Without proper testing and evaluation, these types of applications cannot be production ready. Driver safety and Drowsiness detection simulation is a proposed solution to make sure that the drivers are in appropriate conditions to drive a vehicle. If this application gives false information, it could affect the driver and also there can be lawsuits against the organization/owners of the product. Looking at the current status of the prototype the team can state that it is not production ready and needs a lot of research and testing to come up with a more stable version of the simulation.

Unit tests were carried out on below processes of the simulation.

- Start/Stop button functionality
- Image capture process using the camera
- Image gray-scaling
- Detect face
- Detect left and right eyes
- Generate warning alarms
- Show driver statistics in the list box
- Save driver statistics in the text file

7.1 Unit Tests

7.1.1 “Start” button functionality

Table 5: Start button functionality test case

ID	01
Test Title	“Start” button functionality
Description of Test	Tests the functionality of the “Start” button of the simulation system
Pre-Condition(s)	Start the simulation application in Microsoft Visual Studio 2017
Test Step(s)	1. Press the “Start” button of the simulation
Expected Outcome(s)	1. Label of the “Start” button should change to “Stop” 2. Image box should view the live video stream grabbed from the camera 3. Driver statistics should be populated in the list box 4. Driver statistics should be written to the “DriverStatistics.txt” file
Actual Outcome(s)	1. Button label is changed from “Start” to “Stop” 2. Image box is populated with the live camera feed 3. Driver statistics are shown in the list box 4. Driver statistics are written to the “DriverStatistics.txt” file
Post-Condition(s)	The simulation should be running until it’s stopped
Status (Fail/Pass)	Pass

Notes/Comments	The functionality has been tested approximately 250 times and it works as expected generating a success rate of 100%

7.1.2 “Stop” button functionality

Table 6: Stop button functionality test case

ID	02
Test Title	“Stop” button functionality
Description of Test	Tests the functionality of the “Stop” button of the simulation system
Pre-Condition(s)	Press the “Start” button and make sure the simulation is running
Test Step(s)	1. Press the “Stop” button of the simulation
Expected Outcome(s)	1. Label of the “Stop” button should change to “Start” 2. Image box should stop viewing the live video stream grabbed from the camera 3. Population of driver statistics in the list box should be paused 4. Simulation should stop writing driver statistics to the “DriverStatistics.txt” file
Actual Outcome(s)	1. Button label is changed from “Stop” to “Start” 2. Image box stops displaying the live video data 3. Population of driver statistics in the list box is paused 4. Simulation stops writing statistics to the “DriverStatistics.txt” file
Post-Condition(s)	The simulation should be stopped
Status (Fail/Pass)	Pass
Notes/Comments	The functionality has been tested approximately 250 times and it works as expected generating a success rate of 100%

7.1.3 Image capture process

Table 7: Image capturing process test case

ID	03
Test Title	Image capturing process using the camera
Description of Test	Tests the functionality of grabbing the image frames using the live video data from the camera
Pre-Condition(s)	Start the simulation application in Microsoft Visual Studio 2017
Test Step(s)	1. Press the “Start” button of the simulation
Expected Outcome(s)	1. Live video data stream should be visible in the image box of the simulation
Actual Outcome(s)	1. Video data is populated in the image box of the simulation
Post-Condition(s)	The simulation should be in the running state continuously visualizing live video data in the image box until the system is stopped
Status (Fail/Pass)	Pass
Notes/Comments	The functionality has been tested approximately 250 times and it works as expected generating a success rate of 100%

7.1.4 Image Gray-scaling

Table 8: Image capturing process test case

ID	04
Test Title	Image gray-scaling process
Description of Test	The image grabbed from the live video data is converted to a gray image in order to reduce noise
Pre-Condition(s)	The simulation should grab image frames from the live video data generated by the camera
Test Step(s)	1. Press the "Start" button of the simulation 2. Camera should push live video data to the image box.
Expected Outcome(s)	1. The colored image should be converted to a gray-scaled image
Actual Outcome(s)	1. Image is converted to a gray image
Post-Condition(s)	The simulation should be in the running state continuously visualizing live video data in the image box until the system is stopped
Status (Fail/Pass)	Pass
Notes/Comments	The functionality has been tested approximately 250 times and it works as expected generating a success rate of 100%

7.1.5 Detect Face

Table 9: Detect face test case

ID	05
Test Title	Face detection process
Description of Test	Once the image is fetched from the camera face is detected using Haar-cascade face datasets in the simulation
Pre-Condition(s)	The simulation should grab image frames from the live video data generated by the camera
Test Step(s)	1. Press the "Start" button of the simulation 2. Camera should push live video data to the image box. 3. Perform image gray-scaling automatically by the simulation 4. Detect face automatically by the simulation
Expected Outcome(s)	1. If the face is identified in the image the simulation should draw a red rectangle border around the face 2. If a face is not detected in the image the red rectangle border should be invisible.
Actual Outcome(s)	1. When the face is identified the simulation draws a red rectangle around the face 2. When the face is not detected a red rectangle is not drawn 3. Depending on the lighting conditions, camera quality and processing power of the computational resources the current simulation gives false positives. 4. Encountered system lags due to the same reasons mentioned above.

	5. Face detection was unsuccessful in bad lighting conditions and at night times
Post-Condition(s)	The simulation should be in the running state continuously visualizing live video data in the image box until the system is stopped
Status (Fail/Pass)	Partial (depending on the lighting conditions, camera quality and processing power of the computational resources)
Notes/Comments	The team cannot exactly measure the accuracy of the Face detection since this totally depends on the lighting conditions when the simulation is run, camera quality and processing power of the computational resources. Nevertheless, the team can state that in proper lighting conditions the simulation accuracy can be between 70 – 100 % and in improper lighting conditions this figure drops down to less than 5% or even 0%.

7.1.6 Detect Left and Right Eyes

Table 10: Detect left and right eyes test case

ID	06
Test Title	Left and Eye detection process
Description of Test	Once the face in the image is detected the simulation system performs eye detection.
Pre-Condition(s)	The simulation should grab image frames from the live video data generated by the camera
Test Step(s)	1. Press the “Start” button of the simulation 2. Camera should push live video data to the image box. 3. Perform image gray-scaling automatically by the simulation 4. Perform face detection automatically by the simulation 5. Perform left and right eye detection automatically by the simulation
Expected Outcome(s)	1. If both the eyes are detected the simulation draws two red rectangles around the left and right eyes. 2. If both eyes are not detected in the image, the red rectangle borders around the eyes should be invisible.
Actual Outcome(s)	1. When both the eyes are detected the simulation draws two red rectangles around the eyes 2. When the eyes are not detected two red rectangles are not drawn 3. Depending on the lighting conditions, camera quality and processing power of the computational resources the current simulation gives false positives. 4. Encountered system lags due to the same reasons mentioned above. 5. Eye detection was unsuccessful in bad lighting conditions and at night times 6. Eye detection was unsuccessful when the driver was wearing

	sun glasses
Post-Condition(s)	The simulation should be in the running state continuously visualizing live video data in the image box until the system is stopped
Status (Fail/Pass)	Partial (depending on the lighting conditions, camera quality and processing power of the computational resources)
Notes/Comments	The team cannot exactly measure the accuracy of the eye detection since this totally depends on the lighting conditions when the simulation is run, camera quality and processing power of the computational resources. Nevertheless, the team can state that in proper lighting conditions the simulation accuracy can be between 70 – 100 % and in improper lighting conditions this figure drops down to less than 5% or even 0%.

7.1.7 Generate Warning Alarms

Table 11: Generate warning alarm test case

ID	07
Test Title	Generate warning alarms
Description of Test	Depending on the actual eye closure percentage calculated by the simulation it will generate warning alarms or stop warning alarms
Pre-Condition(s)	The simulation should grab image frames from the live video data generated by the camera
Test Step(s)	1. Press the “Start” button of the simulation 2. Camera should push live video data to the image box. 3. Perform image gray-scaling automatically by the simulation 4. Perform face detection automatically by the simulation 5. Perform left and right eye detection automatically by the simulation
Expected Outcome(s)	1. If the actual opened-eye percentage is less than or equal to the driver selected eye percentage warning alarms will be generated. 2. The alarm status should be shown as “YES” in the driver statistics. 3. If the actual opened-eye percentage is greater than the driver selected eye percentage warning alarms will be not be generated. 4. In this case the alarm status should be shown as “NO” in the driver statistics.
Actual Outcome(s)	1. When the actual opened-eye percentage is less than or equal to the driver selected eye percentage warning alarms are generated. 2. Alarm status is shown as “YES” 3. When the actual opened-eye percentage is greater than the driver selected eye percentage warning alarms are generated. 4. Alarm status is logged as “NO”

Post-Condition(s)	The simulation should be in the running state continuously visualizing live video data in the image box until the system is stopped
Status (Fail/Pass)	Pass
Notes/Comments	The functionality has been tested approximately 250 times and it works as expected generating a success rate of 100%

7.1.8 Show driver statistics in the list box

Table 12: Show driver statistics in the list box test case

ID	08
Test Title	Show driver statistics in the list box of the simulation
Description of Test	Depending on the actual eye closure percentage calculated by the simulation it will generate the driver statistics and these stats are shown in the list box.
Pre-Condition(s)	Press the "Start" button and make sure the simulation is running
Test Step(s)	1. Press the "Start" button of the simulation 2. Press the "Stop" button of the simulation
Expected Outcome(s)	1. The user should be able to see the driver statistics in the list box after the start-up of the simulation 2. Stopping the simulation will stop logging the driver statistics in the list box
Actual Outcome(s)	1. When the simulation starts, driver statistics are shown in the list box 2. When the simulation is stopped, driver statistics are not logged in the list box
Post-Condition(s)	The simulation should be in the running state continuously visualizing live video data in the image box until the system is stopped
Status (Fail/Pass)	Pass
Notes/Comments	The functionality has been tested approximately 250 times and it works as expected generating a success rate of 100%

7.1.9 Save driver statistics in a text file

Table 13: Save driver statistics in a text file test case

ID	09
Test Title	Save driver statistics in a text file
Description of Test	Depending on the actual eye closure percentage calculated by the simulation it will generate the driver statistics and these stats are saved in a text file named "DriverStatistics.txt"
Pre-Condition(s)	Press the "Start" button and make sure the simulation is running
Test Step(s)	1. Press the "Start" button of the simulation 2. Press the "Stop" button of the simulation
Expected Outcome(s)	1. The user should be able to see the driver statistics written to the "DriverStatistics.txt" file after the start-up of the simulation

	2. Stopping the simulation should stop logging the driver statistics to the “DriverStatistics.txt” file
Actual Outcome(s)	3. When the simulation starts, driver statistics are saved to the text file 4. When the simulation is stopped, driver statistics are not logged to the text file
Post-Condition(s)	The simulation should be in the running state continuously visualizing live video data in the image box until the system is stopped
Status (Fail/Pass)	Pass
Notes/Comments	The functionality has been tested approximately 250 times and it works as expected generating a success rate of 100%

7.2 End User Tests

Driver safety and drowsiness detection simulation required end user testing so that the team could get an understanding what the end users thinks about the system. End users are new to the system and they tend to look at the simulation in a new perspective. This allows the development team to gather more information about the simulation, fresh ideas, and issues if exists so that it could be solved before releasing to the market. In this end user tests the development team main focused on the below three functionalities since they were most important in this proposed simulation.

1. Left and right eyes are in an opened state
2. Left and right eyes are in a closure state
3. Left eye or right eye is closed (only one eye)

7.2.1 Left and Right Eyes Are in An Opened State

Table 14: Left and right eyes are in an opened state test case

Test Subject	***Number of Tests Performed (T)***	***Number of Successful attempts (S)***	***Success Rate (%) = (S/T) *100***
#1	6	5	83.33
#2	10	9	90.00
#3	8	3	37.50
#4	14	11	78.57
#5	12	11	91.66
#6	11	8	72.72
#7	5	4	80.00
#8	9	6	66.66
#9	15	12	80.00
#10	20	13	65.00

7.2.2 Left and Right Eyes Are in a Closure State

Table 15: Left and right eyes are in a closure state test case

Test Subject	***Number of Tests Performed (T)***	***Number of Successful attempts (S)***	***Success Rate (%) = (S/T) *100***
#1	5	4	80.00
#2	4	4	100.00
#3	6	3	50.00
#4	2	2	100.00
#5	8	6	75.00
#6	7	6	85.71
#7	8	5	62.50
#8	18	15	83.33
#9	9	7	77.77
#10	10	9	90.00

7.2.3 Left Eye or Right Eyes is closed

Table 16: Left Eye or Right Eyes is closed test case

Test Subject	***Number of Tests Performed (T)***	***Number of Successful attempts (S)***	***Success Rate (%) = (S/T) *100***
#1	6	4	66.66
#2	8	7	87.50
#3	9	6	66.66
#4	3	1	33.33
#5	11	5	45.45
#6	10	6	60.00
#7	6	6	100.00
#8	9	8	88.88
#9	7	5	71.42
#10	12	10	83.33

7.2.4 End User Feedback

After running the above end user tests, team was able to get the below feedback from them. Note that the above tests were done in different lighting conditions which majorly affects the accuracy of the simulation. Nevertheless, processing power of the testing computational resource (laptop) and the quality of the web camera also affected the accuracy of proposed solution immensely.

- Live video data population in the simulation was stuck sometimes. Encountered system lag while running the simulation.
- Face and Eye detection process sometimes provided false positives resulting in incorrect warning alarms.
- Testing was done in bad lighting conditions and the simulation couldn't properly identify the face and eyes.
- Encountered system lags. Face and eye detection was ok for the most part but sometimes resulted in providing invalid alarms.
- System worked well.
- Face and eye detection worked well for the most part. Came across system lag.
- Face and eye detection doesn't work in bad lighting conditions. Development team must find a resolution for these criteria.
- Depending on the distance between the camera and the user it gives false alarms.
- Simulation gives false alarms when the driver wears a sun glass
- Realized that sometimes the simulation shows a NaN (Not a Number) value in the statistics. Is this an error/bug in the system?
- Simulation needs more testing.

8. LIMITATIONS

Even though the completion percentage of this project is 100%, still the prototype is not production ready. The simulation needs more and more testing and the current solution needs to be improved massively due to the fact that this is a safety application. The team has faced below limitations throughout the time span of this particular project which signifies that the solution is not production ready.

1. The simulation will not perform accurately and effectively during inacceptable illumination (light) conditions.
2. The simulation will not work completely at night times when there is no light at all.
3. Face and Eye detection algorithms do not labor during inacceptable illumination conditions including night times.
4. Eye detection is not possible when the user wears sun glasses while using the simulation. This is out of current scope of the proposed solution.
5. Driver must look forward at all times in order to identify the eye open and closure states. Eye states cannot be identified for the side views of the face.
6. Processing power of the computational resource (laptop) affects the performance and the accuracy of the proposed simulation.
7. Quality of the web camera used in the simulation affects the performance and the accuracy of the simulation.
8. Sometimes the camera doesn't capture frames and results in total frames count to be zero (0). This will also result in zero eye detection and will calculate the actual eye percentage to be (0/0) *100 = infinity. In computer world infinity is Not a Number (NaN). Therefore, current system alarm activation is disabled for such cases.

9. FUTURE WORK AND CONCLUSION

The team went through the test results, feedback received from the end users and limitations thoroughly. After analyzing these, the team identified possible future improvements which can be introduced to the current simulation to improve the accuracy and efficiency of the proposed simulation.

1. The main issue the team identified in the development phase was that the face and eye detection was not accurate in inacceptable illumination (light) conditions including at night times. As a solution, the normal web camera can be replaced with a night vision (Infra Red - IR) camera.
2. Current simulation is written on C# programming language. In order to get the competitive advantage on performance aspect, team could write the program in C/C++ programming language.
3. As a future enhancement current simulation program could be implemented on a FPGA chip so that it can be actually integrated to a vehicle.
4. Currently, the simulation detects the driver conditions only based on eye open and closure states. In the future, the team can in-cooperate other features such as head movements, head positions, positioning of ears etc. to detect driver conditions.
5. The team can integrate a new training model (machine learning for face and eye detection) for the existing simulation so that the user can train the simulation to use his/her own datasets to identify the driver conditions.
6. The simulation uses only two dimensions to identify the face and eyes at present. As an improvement the team could perform face and eye detection base on three dimensions view for better face and eye data extraction in images.
7. Currently saved driver statistics data (in DriverStatistics.txt file) can be used for machine learning so that the eye percentage and warning sequence selection processes can be automated.

In conclusion, the team believes that the project was above the expectations even though issues were encountered in the final prototype. Most of the issues are dependent on external factors such as lighting, processing power of resources, camera quality etc. Within the limited time frame the team was able to cover all the functionalities declared in the scope and successful in adding some extra functionalities requested by the supervisor. The team was able to produce a stable, working prototype which someone else could come in and work on, in order to improve more and more so that it can be actually production ready. The team learned concepts such as computer vision and external libraries in order to successfully complete the project and also the importance of testing a prototype before releasing it for production. We also believe that the lessons learnt from this research project will be an added advantage for us in our career development and personal satisfaction.

REFERENCES

Asp Technology Ltd. (2018). *Anti Sleep Pilot*. Retrieved from amazon: https://www.amazon.co.uk/Asp-Technology-Ltd-ASP001-Sleep/dp/B005C3EC18

Awais, M., Drieberg, M., & Badruddin, N. (2014). *Driver drowsiness detection using EEG power spectrum analysis.* Retrieved from ResearchGate: https://www.researchgate.net/publication/271481486_Driver_drowsiness_detection_using_EEG_power_spectrum_analysis

Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). *Manifesto for Agile Software Development.* Retrieved from agilemanifesto: http://agilemanifesto.org/

Bosch, R. (2012). *Preventing microsleep Bosch Driver Drowsiness Detection.* Retrieved from BoschChina: http://microsite.bosch.com.cn/life/en/invented-for-life/citizen/bosch-driver-drowsiness-detection.html

Cottingham, D. (2018). *Technology to keep you driving in your lane: lane departure warning & lane keep assist.* Retrieved from Driving tests: https://www.drivingtests.co.nz/resources/technology-to-keep-you-driving-in-your-lane-lane-departure-warning-lane-keep-assist/

Coxworth, B. (2011, January 4). *Anti Sleep Pilot detects drowsy drivers.* Retrieved from NEW ATLAS: https://newatlas.com/anti-sleep-pilot-monitors-driver-fatigue/17439/

cPRIME. (2018). *WHAT IS AGILE? WHAT IS SCRUM?* Retrieved from cprime: https://www.cprime.com/resources/what-is-agile-what-is-scrum/

Howard, B. (2017, February 27). *How does lane departure warning work?* Retrieved from ExtremeTech: https://www.extremetech.com/extreme/165320-what-is-lane-departure-warning-and-how-does-it-work

Jae-young, K. (2008, February 22). *I do not have a hand, I just play with my mind*. Retrieved from laxtha: http://www.laxtha.com/PCMinfView.asp?minf_idx=837&Brand_Name=QEEG-8&Model_Name=LXE3208&Info_Type=3

Ministry of Transport New Zealand. (2018, April 12). *Monthly road crash statistics update - March 2018.* Retrieved from Ministry of Transport: https://www.transport.govt.nz/resources/road-safety-resources/roadcrashstatistics/monthlyoverviewofcrashstatistics/monthly-road-crash-statistics-update-march-2018/

sinnaps. (2018). *AGILE PROJECT MANAGEMENT: THE SPRINT METHODOLOGY.* Retrieved from sinnaps: https://www.sinnaps.com/en/project-management-blog/agile-project-management-sprint-methodology

StopSleep Pty Ltd. (2017). *About: StopSleep® - Electronic Anti-Sleep Alarm.* Retrieved from stopsleep: http://www.stopsleep.com.au/about-3/

Stopsleep Pty Ltd. (2018). *Anti Sleep Alarm for Drivers. Warns up to 5 Minutes Before Drowsiness. Beep and Vibration Doze Alert. Car Truck Safety Driving Warning Device. Stay Awake Nap Detector Technology Alertness System*. Retrieved from amazon: https://www.amazon.com/Drivers-Drowsiness-Vibration-Technology-Alertness/dp/B00A33APOU

Tesla Inc. (2018). *Full Self-Driving Hardware on All Cars.* Retrieved from tesla: https://www.tesla.com/en_NZ/autopilot

Thorslund, B. (2003). Electrooculogram Analysis and Development of a System for Defining Stages of Drowsiness. Linköping, Sweden. Retrieved from https://www.diva-portal.org/smash/get/diva2:673960/FULLTEXT01.pdf

Verma, H. (2015, August 23). *Scrum Framework.* Retrieved from KnowledgeBlob: http://knowledgeblob.com/agile/scrum-framework/

APPENDIX A – WORK BREAKDOWN STRUCTURE

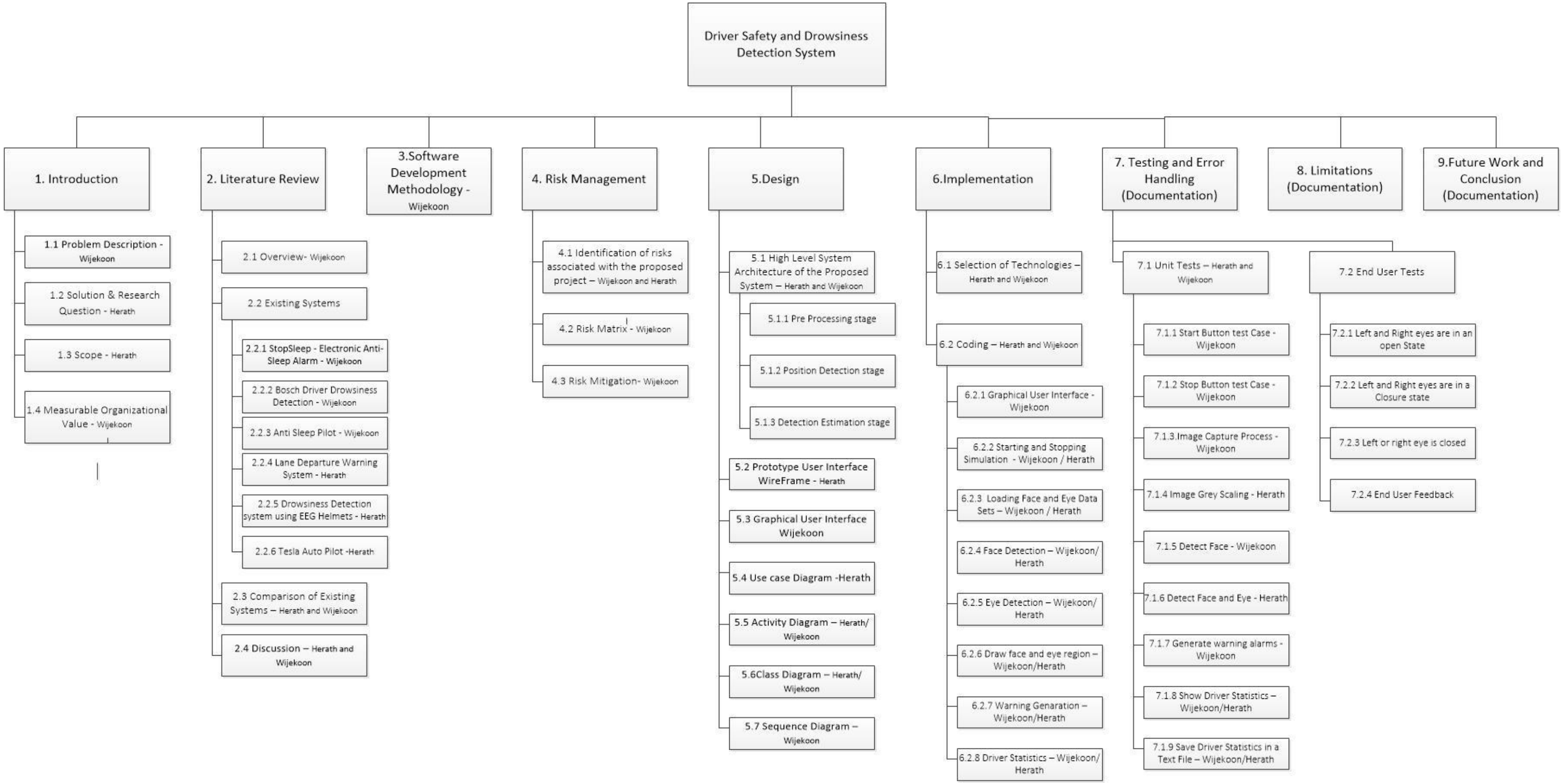
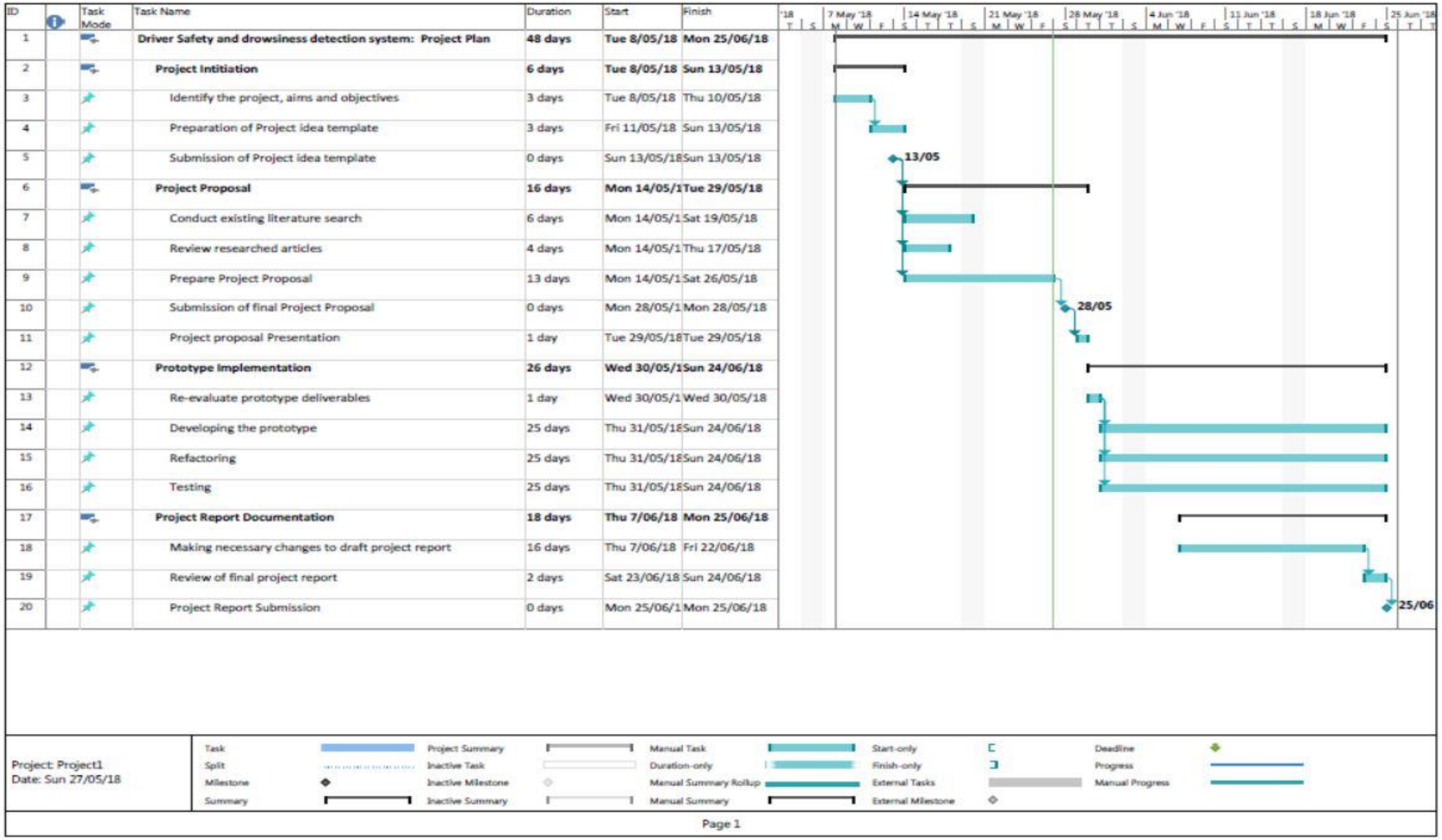

Driver Safety and Drowsiness Detection System
1. Introduction
1.1 Problem Description - Wijekoon
1.2 Solution & Research Question - Herath
1.3 Scope - Herath
1.4 Measurable Organizational Value - Wijekoon
2. Literature Review
2.1 Overview- Wijekoon
2.2 Existing Systems
2.2.1 StopSleep - Electronic Anti-Sleep Alarm - Wijekoon
2.2.2 Bosch Driver Drowsiness Detection - Wijekoon
2.2.3 Anti Sleep Pilot - Wijekoon
2.2.4 Lane Departure Warning System - Herath
2.2.5 Drowsiness Detection system using EEG Helmets - Herath
2.2.6 Tesla Auto Pilot -Herath
2.3 Comparison of Existing Systems – Herath and Wijekoon
2.4 Discussion – Herath and Wijekoon
3.Software Development Methodology - Wijekoon
4. Risk Management
4.1 Identification of risks associated with the proposed project – Wijekoon and Herath
4.2 Risk Matrix - Wijekoon
4.3 Risk Mitigation- Wijekoon
5.Design
5.1 High Level System Architecture of the Proposed System – Herath and Wijekoon
5.1.1 Pre Processing stage
5.1.2 Position Detection stage
5.1.3 Detection Estimation stage
5.2 Prototype User Interface WireFrame - Herath
5.3 Graphical User Interface Wijekoon
5.4 Use case Diagram -Herath
5.5 Activity Diagram – Herath/ Wijekoon
5.6Class Diagram – Herath/ Wijekoon
5.7 Sequence Diagram – Wijekoon
6.Implementation
6.1 Selection of Technologies – Herath and Wijekoon
6.2 Coding – Herath and Wijekoon
6.2.1 Graphical User Interface - Wijekoon
6.2.2 Starting and Stopping Simulation - Wijekoon / Herath
6.2.3 Loading Face and Eye Data Sets – Wijekoon / Herath
6.2.4 Face Detection – Wijekoon/ Herath
6.2.5 Eye Detection – Wijekoon/ Herath
6.2.6 Draw face and eye region – Wijekoon/Herath
6.2.7 Warning Genaration – Wijekoon/Herath
6.2.8 Driver Statistics – Wijekoon/ Herath
7. Testing and Error Handling (Documentation)
7.1 Unit Tests – Herath and Wijekoon
7.1.1 Start Button test Case - Wijekoon
7.1.2 Stop Button test Case - Wijekoon
7.1.3.Image Capture Process - Wijekoon
7.1.4 Image Grey Scaling - Herath
7.1.5 Detect Face - Wijekoon
7.1.6 Detect Face and Eye - Herath
7.1.7 Generate warning alarms - Wijekoon
7.1.8 Show Driver Statistics – Wijekoon/Herath
7.1.9 Save Driver Statistics in a Text File – Wijekoon/Herath
7.2 End User Tests
7.2.1 Left and Right eyes are in an open State
7.2.2 Left and Right eyes are in a Closure state
7.2.3 Left or right eye is closed
7.2.4 End User Feedback
8. Limitations (Documentation)
9.Future Work and Conclusion (Documentation)

Figure 23: Work Breakdown Structure

APPENDIX B – GANTT CHART



ID		Task Mode	Task Name	Duration	Start	Finish
1			**Driver Safety and drowsiness detection system: Project Plan**	**48 days**	**Tue 8/05/18**	**Mon 25/06/18**
2			**Project Intitiation**	**6 days**	**Tue 8/05/18**	**Sun 13/05/18**
3			Identify the project, aims and objectives	3 days	Tue 8/05/18	Thu 10/05/18
4			Preparation of Project idea template	3 days	Fri 11/05/18	Sun 13/05/18
5			Submission of Project idea template	0 days	Sun 13/05/18	Sun 13/05/18
6			**Project Proposal**	**16 days**	**Mon 14/05/1**	**Tue 29/05/18**
7			Conduct existing literature search	6 days	Mon 14/05/1	Sat 19/05/18
8			Review researched articles	4 days	Mon 14/05/1	Thu 17/05/18
9			Prepare Project Proposal	13 days	Mon 14/05/1	Sat 26/05/18
10			Submission of final Project Proposal	0 days	Mon 28/05/1	Mon 28/05/18
11			Project proposal Presentation	1 day	Tue 29/05/18	Tue 29/05/18
12			**Prototype Implementation**	**26 days**	**Wed 30/05/1**	**Sun 24/06/18**
13			Re-evaluate prototype deliverables	1 day	Wed 30/05/1	Wed 30/05/18
14			Developing the prototype	25 days	Thu 31/05/18	Sun 24/06/18
15			Refactoring	25 days	Thu 31/05/18	Sun 24/06/18
16			Testing	25 days	Thu 31/05/18	Sun 24/06/18
17			**Project Report Documentation**	**18 days**	**Thu 7/06/18**	**Mon 25/06/18**
18			Making necessary changes to draft project report	16 days	Thu 7/06/18	Fri 22/06/18
19			Review of final project report	2 days	Sat 23/06/18	Sun 24/06/18
20			Project Report Submission	0 days	Mon 25/06/1	Mon 25/06/18

Project: Project1
Date: Sun 27/05/18

Page 1

Figure 24: Gantt Chart

APPENDIX C – TEAM MEETINGS

Meeting of Minutes 1

Minutes of Meeting 1: *4.00 p.m. on 8th May 2018 at MIT Manukau Campus room-127*

Chairperson: *Wijekoon Somasiri (170001510)*

Minute Taker: *Herath Premarathne (170001825)*

Present: *Wijekoon Somasiri, Herath Premarathne*

Apologies: *None*

Absent (no apology received): *None*

Table 17: Minutes of Meeting 1

Agenda Item	**Description –** ***include a heading for each item recorded and a brief summary of the discussion.***	***Include the following four points:*** ***1. Action*** ***2. Person responsible*** ***3. Date action is to be completed***
1.	**Appointment of chairperson and recorder** Wijekoon was appointed to chair the meeting and Herath was appointed to take minutes of this meeting.	
2.	**Review of previous minutes and actions** None	
3	**Agenda items**	
	1. Team formation	***Action:*** Form the team with Wijekoon and Herath ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 8th May 2018
	2. Discuss possible project ideas	***Action:*** Research on cutting edge technologies such as computer vision, robotics, embedded systems etc. ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 11th May 2018
	3. Selection of the final project idea	***Action:*** Finalize to work on Driver safety and drowsiness detection system ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 8th May 2018

	4. Discuss the project Idea with the Supervisor - Fadi Fayes	***Action:*** Supervisor was satisfied with the project idea. Discuss the business value of the project Idea. Advised team members to research on existing solutions. Advised to find possible datasets for face and eye detection. ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 8th May 2018
	5. Start working on the draft project proposal template	***Action:*** Fill out draft project proposal template for supervisor verification ***Person Responsible:*** Herath ***Date to be completed:*** 13th May 2018
4	Date and time of next meeting 2.30 p.m. on 15th May 2018	
5	Close meeting: Record date and time the meeting closed 4.30 p.m. on 8th May 2018	

Meeting of Minutes 2

Minutes of Meeting 2: *2.30 p.m. on 15th May 2018 at MIT Manukau Campus room-127*

Chairperson: *Wijekoon Somasiri (170001510)*

Minute Taker: *Herath Premarathne (170001825)*

Present: *Wijekoon Somasiri, Herath Premarathne*

Apologies: *None*

Absent (no apology received): *None*

Table 18: Minutes of Meeting 2

Agenda Item	Description – *include a heading for each item recorded and a brief summary of the discussion.*	*Include the following four points:* *1. Action* *2. Person responsible* *3. Date action is to be completed*
1.	**Appointment of chairperson and recorder** Wijekoon was appointed to chair the meeting and Herath was appointed to take minutes of this meeting.	
2.	**Review of previous minutes and actions** 1. Team was formed 2. Discussed possible project ideas and finalized to continue on the Driver safety and drowsiness detection system 3. Got confirmation to proceed with the project idea by Fadi Fayes 4. Submission of template project proposal	
3	**Agenda items**	
	1. Research on existing driver drowsiness systems	***Action:*** Both team members must research on existing systems ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 19th May 2018
	2. Documentation of topic/description/problem description in the final project proposal document	***Action:*** Wijekoon should complete the allocated task in the project proposal documentation ***Person Responsible:*** Wijekoon ***Date to be completed:*** 21st May 2018
	3. Documentation of solution and research question in the final project proposal document	***Action:*** Herath should complete the allocated task in the project

		proposal documentation ***Person Responsible:*** Herath ***Date to be completed:*** 21st May 2018
	4. Documentation of measurable organizational values in the final project proposal document	***Action:*** Wijekoon should complete the allocated task in the project proposal documentation ***Person Responsible:*** Wijekoon ***Date to be completed:*** 21st May 2018
	5. Documentation of scope (aim and objectives) in the final project proposal document	***Action:*** Herath should complete the allocated task in the project proposal documentation ***Person Responsible:*** Herath ***Date to be completed:*** 21st May 2018
	6. Documentation of literature review in the final project proposal document	***Action:*** Herath and Wijekoon should complete the allocated task in the project proposal documentation ***Person Responsible:*** Herath ***Date to be completed:*** 21st May 2018
4	Date and time of next meeting 4.00 p.m. on 22nd May 2018	
5	Close meeting: Record date and time the meeting closed 3 p.m. on 15th May 2018	

Meeting of Minutes 3

Minutes of Meeting 3: *4.00 p.m. on 22nd May 2018 at MIT Manukau Campus room-127*

Chairperson: *Wijekoon Somasiri (170001510)*

Minute Taker: *Herath Premarathne (170001825)*

Present: *Wijekoon Somasiri, Herath Premarathne*

Apologies: *None*

Absent (no apology received): *None*

Table 19: Minutes of Meeting 3

Agenda Item	**Description – *include a heading for each item recorded and a brief summary of the discussion.***	***Include the following four points:*** ***1. Action*** ***2. Person responsible*** ***3. Date action is to be completed***
1.	**Appointment of chairperson and recorder** Wijekoon was appointed to chair the meeting and Herath was appointed to take minutes of this meeting.	
2.	**Review of previous minutes and actions** 1. Research on existing driver drowsiness systems 2. Documentation of topic/description/problem description in the final project proposal document 3. Documentation of solution and research question in the final project proposal document 4. Documentation of measurable organizational values in the final project proposal document 5. Documentation of scope (aim and objectives) in the final project proposal document 6. Documentation of literature review in the final project proposal document	
3	**Agenda items**	
	1. Documentation of software development methodology	***Action:*** Wijekoon and Herath should complete the allocated task in the project proposal documentation ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 27th May 2018
	2. Documentation of Risk Management	***Action:*** Wijekoon and Herath should complete the allocated task in the project proposal documentation ***Person Responsible:*** Wijekoon and Herath

		Date to be completed: 27th May 2018
	3. Documentation of the design of the proposed system	***Action:*** Wijekoon and Herath should complete the allocated task in the project proposal documentation ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 27th May 2018
	4. Create the Gantt chart	***Action:*** Wijekoon and Herath should complete the allocated task in the project proposal documentation ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 27th May 2018
	5. Create the Work Breakdown Structure (WBS)	***Action:*** Wijekoon and Herath should complete the allocated task in the project proposal documentation ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 27th May 2018
	6. Prepare the presentation slides	***Action:*** Wijekoon and Herath should complete the allocated task in the project proposal documentation ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 27th May 2018
4	Date and time of next meeting 5.00 p.m. on 29th May 2018	
5	Close meeting: Record date and time the meeting closed 4.45 p.m. on 22nd May 2018	

Meeting of Minutes 4

Minutes of Meeting 4: *5.00 p.m. on 29th May 2018 at MIT Manukau Campus room-127*

Chairperson: *Wijekoon Somasiri (170001510)*

Minute Taker: *Herath Premarathne (170001825)*

Present: *Wijekoon Somasiri, Herath Premarathne*

Apologies: *None*

Absent (no apology received): *None*

Table 20: Minutes of Meeting 4

Agenda Item	Description – *include a heading for each item recorded and a brief summary of the discussion.*	***Include the following four points:*** ***1. Action*** ***2. Person responsible*** ***3. Date action is to be completed***
1.	**Appointment of chairperson and recorder** Wijekoon was appointed to chair the meeting and Herath was appointed to take minutes of this meeting.	
2.	**Review of previous minutes and actions** 1. Documentation of software development methodology 2. Documentation of Risk Management 3. Documentation of the design of the proposed system 4. Create the Gantt chart 5. Create the Work Breakdown Structure (WBS) 6. Prepare the presentation slides	
3	**Agenda items**	
	1. Research on 3rd party libraries which can be used for face and eye detection	***Action:*** Wijekoon and Herath need to do the research on existing image processing 3rd party libraries ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 3rd June 2018
	2. Find out the research areas in image processing which needs to be used in face and eye detection	***Action:*** Wijekoon needs to identify the relevant image processing concepts ***Person Responsible:*** Wijekoon ***Date to be completed:*** 3rd June 2018

	3. Research on face and eye detection algorithms	***Action:*** Identify the possible algorithms to detect face and eye in an image ***Person Responsible:*** Herath ***Date to be completed:*** 3rd June 2018
4	Date and time of next meeting 5.00 p.m. on 08th June 2018	
5	Close meeting: Record date and time the meeting closed 5.30 p.m. on 29th May 2018	

Meeting of Minutes 5

Minutes of Meeting 5: *5.00 p.m. on 8th June 2018 at MIT Manukau Campus room-329*

Chairperson: *Wijekoon Somasiri (170001510)*

Minute Taker: *Herath Premarathne (170001825)*

Present: *Wijekoon Somasiri, Herath Premarathne*

Apologies: *None*

Absent (no apology received): *None*

Table 21: Minutes of Meeting 5

Agenda Item	Description – *include a heading for each item recorded and a brief summary of the discussion.*	*Include the following four points:* *1. Action* *2. Person responsible* *3. Date action is to be completed*
1.	**Appointment of chairperson and recorder** Wijekoon was appointed to chair the meeting and Herath was appointed to take minutes of this meeting.	
2.	**Review of previous minutes and actions** 1. Research on 3rd party libraries which can be used for face and eye detection 2. Find out the research areas in image processing which needs to be used in face and eye detection 3. Research on face and eye detection algorithms	
3	**Agenda items**	
	1. Draw Use Case Diagram	***Action:*** Herath needs to draw the use case diagram for the proposed solution ***Person Responsible:*** Herath ***Date to be completed:*** 10th June 2018
	2. Draw Activity Diagram	***Action:*** Herath needs to draw the Activity diagram for the proposed solution ***Person Responsible:*** Herath ***Date to be completed:*** 10th June 2018
	3. Draw Class Diagram	***Action:*** Wijekoon needs to draw the Class diagram for the

		proposed solution ***Person Responsible:*** Wijekoon ***Date to be completed:*** 10th June 2018
	4. Draw Sequence Diagram	***Action:*** Wijekoon needs to draw the Sequence diagram for the proposed solution ***Person Responsible:*** Wijekoon ***Date to be completed:*** 10th June 2018
	5. Create the Graphical User Interface (GUI) of the proposed solution	***Action:*** Complete the GUI using visual studio 2017 ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 12th June 2018
4	Date and time of next meeting 4.00 p.m. on 12th June 2018	
5	Close meeting: Record date and time the meeting closed 5.30 p.m. on 8th June 2018	

Meeting of Minutes 6

Minutes of Meeting 6: *4.00 p.m. on 12th June 2018 at MIT Manukau Campus room-127*

Chairperson: *Wijekoon Somasiri (170001510)*

Minute Taker: *Herath Premarathne (170001825)*

Present: *Wijekoon Somasiri, Herath Premarathne*

Apologies: *None*

Absent (no apology received): *None*

Table 22: Minutes of Meeting 6

Agenda Item	**Description – *include a heading for each item recorded and a brief summary of the discussion.***	***Include the following four points:*** ***1. Action*** ***2. Person responsible*** ***3. Date action is to be completed***
1.	**Appointment of chairperson and recorder** Wijekoon was appointed to chair the meeting and Herath was appointed to take minutes of this meeting.	
2.	**Review of previous minutes and actions** 1. Draw Use Case Diagram 2. Draw Activity Diagram 3. Draw Class Diagram 4. Draw Sequence Diagram 5. Create the Graphical User Interface (GUI) of the proposed solution	
3	**Agenda items**	
	1. Create camera, face, eye, face and eye template and Alarm classes and test the functionalities	***Action:*** Write code to create the entity classes ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 13th June 2018
	2. Write code to get access to the web camera and show video data in the image box in the simulation and test the functionalities	***Action:*** Write C# code to access the integrated laptop camera and view live stream in the image box ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 14th June 2018
	3. Write the code to detect the face and eye and test the functionalities	***Action:*** Write C# code to implement EmguCV face and eye detection

		Person Responsible: Wijekoon and Herath ***Date to be completed:*** 18th June 2018
4	Date and time of next meeting 3.30 p.m. on 19th June 2018	
5	Close meeting: Record date and time the meeting closed 4.30 p.m. on 12th June 2018	

Meeting of Minutes 7

Minutes of Meeting 7: *3.30 p.m. on 19th June 2018 at MIT Manukau Campus room-127*

Chairperson: *Wijekoon Somasiri (170001510)*

Minute Taker: *Herath Premarathne (170001825)*

Present: *Wijekoon Somasiri, Herath Premarathne*

Apologies: *None*

Absent (no apology received): *None*

Table 23: Minutes of Meeting 7

Agenda Item	Description – *include a heading for each item recorded and a brief summary of the discussion.*	*Include the following four points:* *1. Action* *2. Person responsible* *3. Date action is to be completed*
1.	**Appointment of chairperson and recorder** Wijekoon was appointed to chair the meeting and Herath was appointed to take minutes of this meeting.	
2.	**Review of previous minutes and actions** 1. Create camera, face, eye, face and eye template and Alarm classes and test the functionalities 2. Write code to get access to the web camera and show video data in the image box in the simulation and test the functionalities 3. Write the code to detect the face and eye and test the functionalities	
3	**Agenda items**	
	1. Draw red rectangles around the detected face and eyes and test the functionalities	***Action:*** Write C# code to implement red rectangular shapes around face and eye ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 20th June 2018
	2. Perform the calculation of the actual eye closure percentage to generate warnings and test the functionalities	***Action:*** Write Warning process code ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 20th June 2018
	3. Show the driver statistics in the list box and test the functionalities	***Action:*** Write C# code to show driver statistics in the listbox

		Person Responsible: Wijekoon and Herath ***Date to be completed:*** 20th June 2018
	4. Save the Driver statistics in the “DriverStatistics.txt” file and test the functionalities	***Action:*** Write C# code to save the generated driver statistics to the text file ***Person Responsible:*** Wijekoon and Herath ***Date to be completed:*** 20th June 2018
	5. Prepare final presentation slides	***Action:*** Prepare Microsoft Powerpoint slides for the final presentation ***Person Responsible:*** Herath ***Date to be completed:*** 21st June 2018
4	Date and time of next meeting 7.00 p.m. on 22nd June 2018	
5	Close meeting: Record date and time the meeting closed 4.00 p.m. on 19th June 2018	

Meeting of Minutes 8

Minutes of Meeting 8: *7.00 p.m. on 22nd June 2018 at MIT Student Village, Otara*

Chairperson: *Wijekoon Somasiri (170001510)*

Minute Taker: *Herath Premarathne (170001825)*

Present: *Wijekoon Somasiri, Herath Premarathne*

Apologies: *None*

Absent (no apology received): *None*

Table 24: Minutes of Meeting 8

Agenda Item	Description – *include a heading for each item recorded and a brief summary of the discussion.*	*Include the following four points:* *1. Action* *2. Person responsible* *3. Date action is to be completed*
1.	**Appointment of chairperson and recorder** Wijekoon was appointed to chair the meeting and Herath was appointed to take minutes of this meeting.	
2.	**Review of previous minutes and actions** 1. Draw red rectangles around the detected face and eyes and test the functionalities 2. Perform the calculation of the actual eye closure percentage to generate warnings and test the functionalities 3. Show the driver statistics in the list box and test the functionalities 4. Save the Driver statistics in the "DriverStatistics.txt" file and test the functionalities 5. Prepare final presentation slides	
3	**Agenda items**	
	1. Write Unit tests section in the final project documentation	***Action:*** Write unit test cases section ***Person Responsible:*** Wijekoon ***Date to be completed:*** 25th June 2018
	2. Write End User tests section in the final project documentation	***Action:*** Write User tests section ***Person Responsible:*** Herath ***Date to be completed:*** 25th June 2018
	3. Write Limitations section in the final project documentation	***Action:*** Write Limitations of the

		current prototype ***Person Responsible:*** Herath ***Date to be completed:*** 25th June 2018
	4. Write Future improvements and conclusion section in the final project documentation	***Action:*** Write future improvements and conclusion of the current prototype ***Person Responsible:*** Wijekoon ***Date to be completed:*** 25th June 2018
4	Date and time of next meeting None due to project completion	
5	Close meeting: Record date and time the meeting closed 7.30 p.m. on 22nd June 2018	