**TEAM MEMBERS**

1. Maradana Vijaya Krishna (A0178453W)
2. Putrevu Manoj Niyogi (A0213557E)
3. Sivasankaran Balakrishnan (A0065970X)

**Master of TECHNOLOGY INTELLIGENT SYSTEMS (PART-TIME) 2020  
  
SEMESTER 2 – Pattern Recognition and Machine Learning Systems (PRMLS)  
FINAL PROJECT REPORT**

**Driver Drowsiness Detection System**

Contents

[Executive Summary 2](#_Toc52980106)

[Objective 3](#_Toc52980107)

[Our Steps 4](#_Toc52980108)

[Technologies Used 4](#_Toc52980109)

[Driver Drowsiness Detection System Workflow 5](#_Toc52980110)

[The Model 6](#_Toc52980111)

[Our approach 8](#_Toc52980112)

[Project deliverables: 9](#_Toc52980113)

[Steps to run the Project in Windows Environment v10: 10](#_Toc52980114)

[Conclusions 12](#_Toc52980115)

[Limitations 12](#_Toc52980116)

[References: 13](#_Toc52980117)

[Appendix A: GRADUATE CERTIFICATE - Intelligent Reasoning Systems (IRS) Project Proposal 14](#_Toc52980118)

[Appendix B : Individual Project Report 18](#_Toc52980119)

# Executive Summary

Sleepiness can impair driving performance as much or more so than alcohol, studies show. (Dawson and Reid, 1997; Powell, 2001).

The American Automobile Association (AAA) estimates that one out of every six (16.5%) deadly traffic accidents, and one out of eight (12.5%) crashes requiring hospitalization of car drivers or passengers is due to drowsy driving. (AAA, 2010)

One analysis estimated the cost of automobile accidents attributed to sleepiness to be between $29.2 to $37.9 billion. (Leger, 1994)

(41%) admitted to having fallen asleep at the wheel at some point.; one in ten drivers (10%) reporting they did so within the past year. (AAA, 2010)

More than one-quarter of drivers (27%) admitting they had driven while they were “so sleepy that [they] had a hard time keeping [their] eyes open” within the past month (AAA, 2010)

Researchers estimate that more than 70 million Americans suffer from a sleep disorder. (Institute of Medicine, 2005) One of the most serious consequences of insufficient sleep is traffic accidents due to drowsy driving.

Experts suspect that even these disturbingly high figures underestimate the number of accidents or near-miss accidents due to drowsy driving because of drivers being unaware or not admitting they were drowsy at the time of the accident, or police not acquiring that information.

In spite of education to create awareness among driving community, policy initiatives by the Government to reduce the accidents numbers show no promising decline. In recent years, many of the luxury car companies have already moved in this direction but majority of the solutions are relatively employing complex sensors and devices which are costly and so is the need for simple, effective relatively low cost solutions that can be used by everyone.

Drowsy driving is a prevalent and serious public health issue that requires a simple low cost, fool proof continuous monitoring system that can be used by everyone.

# Objective

There are several types of solutions

* Steering pattern monitoring
* Vehicle position in lane monitoring
* Physiological measurement
* Driver face/eye monitoring

Steering pattern monitoring – It primarily uses steering input from electric power steering system. Learns driving behavior through steering input and compares it with the later rides. The system is based on the fact that the number of micro-corrections on the steering are lower than the one found in normal driving conditions[1, 2]. Any difference beyond threshold activates audio visual cues to draw attention. Monitoring this way only works when the driver actually steers the vehicle actively and does not activate automatic lane keeping system. The system can function reliably only at particular environments and is too dependent on the geometric characteristics of the road and, to a lesser extent, on the kinetic characteristics of the vehicle.

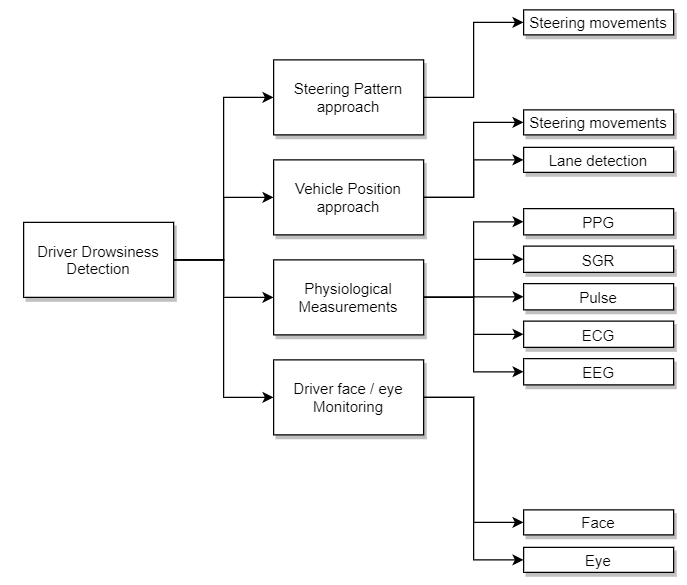
Vehicle position in lane monitoring – This method is based on the externally mounted camera(s) and associated software, which monitors the vehicle position relative to the lane. It depends on external factors such as road marking, weather, and light conditions.

Physiological measurement – Employs body sensors to measure parameters like brain activity, heart rate, skin conductance, muscle activity.

All the above types of models are relatively expensive.

Driver face/eye monitoring – This technique is simple and low-cost model relatively compared to other models and can be used in majority of the vehicles. The only limitation of this model is that the driver should not use dark colored spectacles that obscures the monitoring of eyes.

This technique monitors the state of eyes and if both the eyes are in a closed state for a defined threshold time, alarm is raised to draw attention of the driver.



*Figure 1 – Types of Solutions*

# Our Steps

The approach follows the CRISP-DM process. It consists of five stages, as listed below:

1. Understand the Business Requirements
2. Understanding & Analyzing the Data
3. Build the model
4. Verify / Optimize the model
5. Analyze the results and reveal the insights

The approach uses pre-trained model and follows technique termed as **Transfer Learning**. We have picked up the pretrained model from <http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2>.

The adopted approach is algorithmically simple, intuitive, and easy to implement. It is easily portable to different platforms. It is computationally in expensive as the training is done offline and the trained model is picked up for use.

# Technologies Used

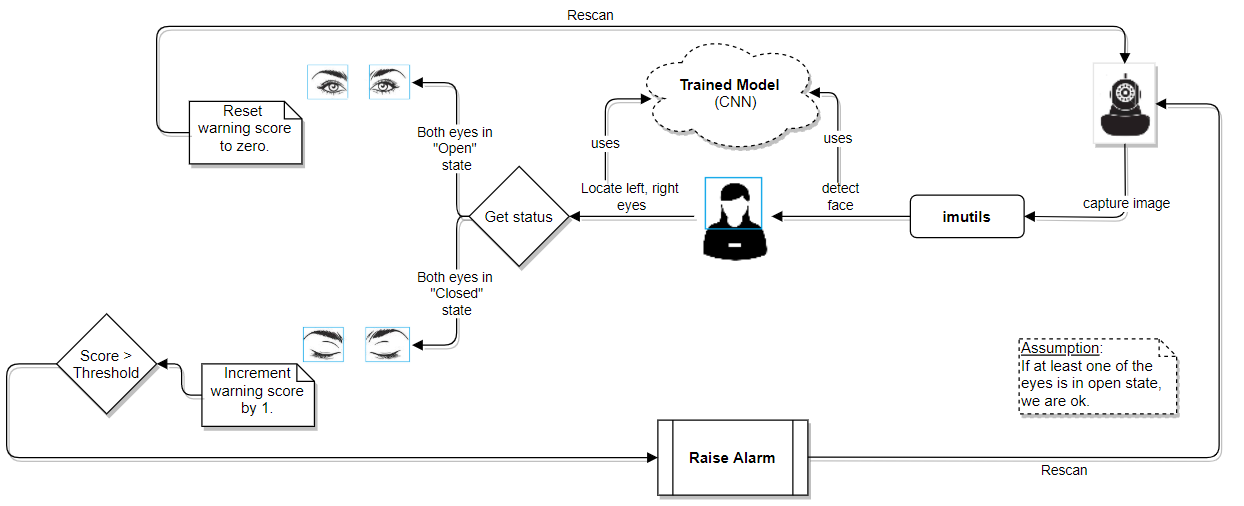
The model developed for the project work uses Python and following open source libraries:

* Numpy – Statistical and Scientific Computing
* Cv2 – Computer Vision library
* Dlib – Predictor and detector library
* Imutils – Utility library for object landmarks
* Matplotlib – Graphing library

**Environment used**

* PyCharm community edition for interactive computing environment for coding in python.

# Driver Drowsiness Detection System Workflow



*Figure 2 Workflow Diagram*

The workflow steps are detailed below:

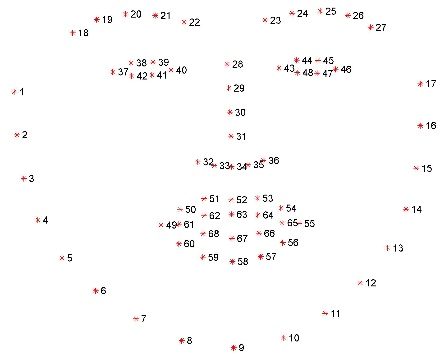
* The vehicle is fit with a camera device in an ideal location to be able to capture the face of driver.
* A python program, **detect\_drowsiness.py** captures the video images in color and converts to monochrome gray color images.
* From the gray image, the driver’s face, left eye, and right eye are detected with the help of **pre-trained model** available at <http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2>.
* The captured sub images of eyes are resized.
* Using the pre-trained model, the images are predicted for **open** or **closed** status.
* If both eyes are in a **closed** state, for a prescribed threshold score, the system treats it as a threat and then raises an alarm to alert the driver.
* If the driver is in a state of restlessness and keeps blinking at varied levels of fractions of time, the state of the eye is sensed and the score value is incremented / decremented based on the result of status to be closed / open.

# The Model

The source of the pretrained model we have used is from the **iBug** (<https://ibug.doc.ic.ac.uk/>) group of **Department of Computing, Imperial College London**. We used the technique, termed **Transfer Learning**, which is using the knowledge gained by a model that is well trained offline using real world data. The facial databases that are used include large collection of images of different subjects, poses, illumination, occlusions, etc. They have proposed a semi-automatic annotation methodology to annotate massive datasets to address issues with other techniques such as the following:

* Some techniques provide relatively low annotations when compared to the volume of images
* Low accuracy from few techniques
* Varying annotation models with each database

These issues make cross database experiments and comparisons difficult or infeasible. The model that iBug suggests a 68-point markup for annotating facial features as shown in the following figure.

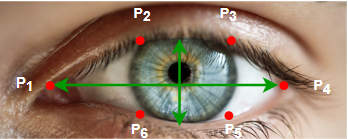
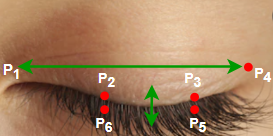


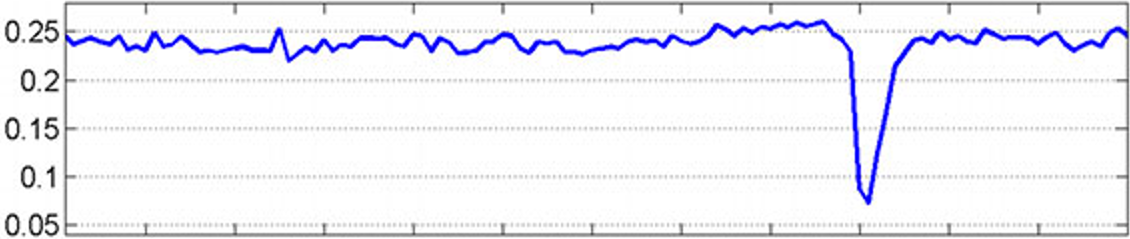
*Figure 3 68-point mark-up used for annotations*

Figure 3 shows the landmarks of features of face, starting from 1 through 68 as described below:

* Edge of face marking as 1 through 17
* Right eyebrow marking as 18 through 22
* Left eyebrow marking as 23 through 27
* Nose marking as 28 through 36
* Right eye marking as 37 through 42
* Left eye marking as 43 through 48
* Outer edges of mouth covering upper and lower lips with marks from 49 through 60
* Inner edges of mouth covering upper and lower lips with marks from 61 through 68

To be able to detect the state of eyes, we find the **Eye Aspect Ratio** of each eye. Each eye is represented by six marks each with (x, y) coordinates as shown in the figure 3.



The correlation between the width and height of the structure and can be expressed as a ratio, **Eye Aspect Ratio(EAR)** by **Soukupova and Cech** ([Real-time Eye Blink Detection using Facial Landmarks](http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf)) as

The aspect ratio will constant and greater than zero, when the eye is in open state and becomes zero when the eye is in closed state. This way we do not have to process the images of eye but can rely purely on the aspect ratio. The graph shows the aspect ratio between open and closed eyes. The dip indicates the aspect ratio when the eyelid is closed.

# Our approach

Step 1 – We have used a laptop with web cam facility to capture the images from video stream

Step 2 – Laptop with python and PyCharm editor installed for programming

Step 3 – Download and place extracted pretrained model viz., from [Shape Predictor](http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2) accessible to the program.

Step 4 – Program in python to detect faces from the video stream.

Step 5 – Resize and transform the image captured into gray scale as we are merely interested in capturing the state of eyes and it does not matter if the image is color or gray scale.

Step 6 – Apply facial landmark localization to extract eye regions from the face.

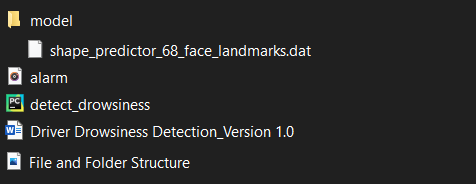
Step 7 – Compute the aspect ratio to determine if the eyes are closed or open.

Step 8 – Trigger alarm if the aspect ratio is below the threshold value for preset duration.

# Project deliverables:

1. The packet **DDDS.zip** will have the following files.
2. Driver Drowsiness Detection\_Version 1.0.docx (current document)
3. Model folder (folder holding pretrained model file viz., **shape\_predictor\_68\_face\_landmarks.dat** downloaded and extracted from [Shape Predictor](http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2))
4. **Detect\_drowsiness.py** (main program which uses detects face and the status of eyes)
5. Spawn a new thread to raise alarm to let the processing go uninterrupted for good performance.
6. Alarm.wav (used by main program to raise an alarm)
7. Img\_directory\_structure.png (image file showing the DDDS.zip structure of files)

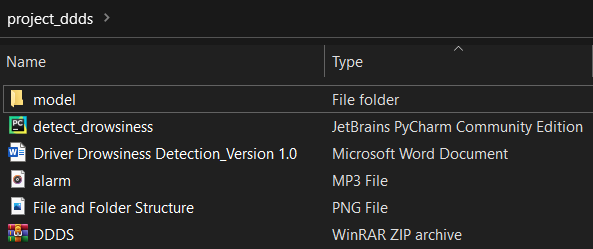
DDDS.zip files and folder structure:



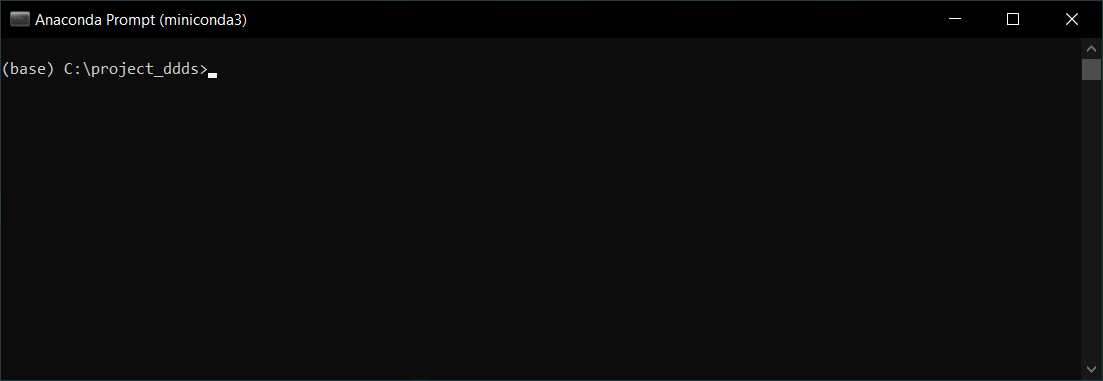
# Steps to run the Project in Windows Environment v10:

1. Programs are developed in
   1. python version 3.7.7,
   2. matplotlib 3.1.3
   3. Computer Vision (Open CV) 4.4.0
   4. imutils 0.5.3
   5. dlib 19.21.0
   6. scipy 1.4.1
   7. threading
2. Ensure you’ve python running environment along with the libraries is up to date.
3. Make sure you [install cmake and dlib libraries](https://medium.com/analytics-vidhya/how-to-install-dlib-library-for-python-in-windows-10-57348ba1117f).
4. Extract all files from the **DDDS.zip** to a folder, say **project\_ddds**. See [Pic 1](#_Pic_1).
5. Invoke Anaconda command environment. See [Pic 2](#_Pic_2).
6. Make sure you activate the environment with all libraries installed. See [Pic 3](#_Pic_3).
7. Run the program to activate the web cam connected to your laptop starts capturing video stream. See [Pic 4](#_Pic_4).
8. Program captures the face and highlight both the eyes in green color drawing closed polygon. See [Pic 5](#_Pic_5).
9. Program also overlays the EAR (Eye Aspect Ratio) continuously as text.
10. It prompts **drowsiness alert** as text when it detects closure of eyes and raises an audible alarm when the eyes are closed for sufficiently large enough time period to sense threat. See [Pic 6](#_Pic_6).
11. To stop the program from running, make sure the window showing the video is active and has focus then type the letter ‘q’ to quit.

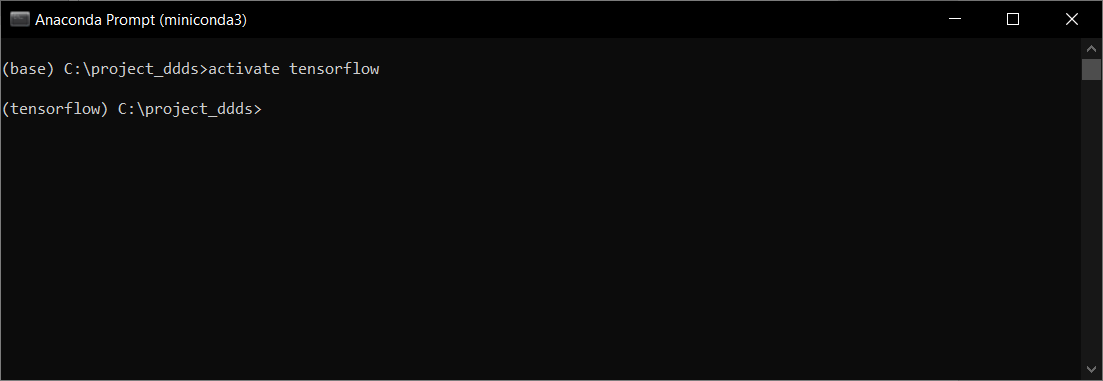
###### Pic 1

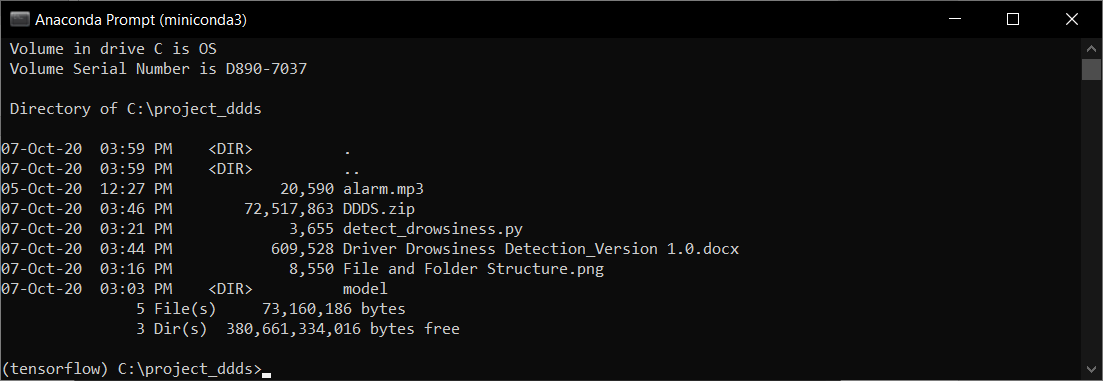


###### Pic 2

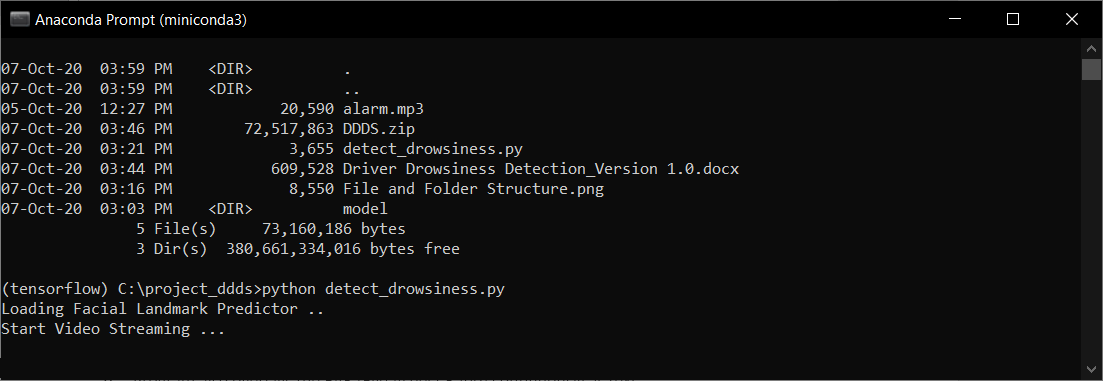


###### Pic 3





###### Pic 4



###### Pic 5

###### Pic 6

# Conclusions

Our objective is to implement a small easy to use, fail proof, low budget alert system to detect fatigue of driver during driving. We have studied several systems and implemented and successfully tested an alert system which uses a camera device fixed in front of the driver to detect fatigue by monitoring the state of eyes and warn through a beep.

# Limitations

* Lighting conditions
* Camera quality
* Positioning of camera
* Driver’s cooperation (not wearing dark colored spectacles)

# References:

###### S. H. Fairclough and R. Graham, “Impairment of driving performance caused by sleep deprivation or alcohol: A comparative study,” Human Factors: The Journal of the Human Factors and Ergonomics Society, vol. 41, no. 1, pp. 118–128, 1999.

###### R. Feng, G. Zhang, and B. Cheng, “An on-board system for detecting driver drowsiness based on multi-sensor data fusion using dempstershafer theory,” in Networking, Sensing and Control, 2009. ICNSC ’09. International Conference on Networking, 2009, pp. 897–902.

###### C. Sagonas, E. Antonakos, G, Tzimiropoulos, S. Zafeiriou, M. Pantic - Image and Vision Computing (IMAVIS), Special Issue on Facial Landmark Localisation "In-The-Wild". <https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/>

###### <https://drowsydriving.org/about/facts-and-stats/>

1. <https://www.nsc.org/road-safety/safety-topics/fatigued-driving>

# Appendix A: GRADUATE CERTIFICATE - Intelligent Reasoning Systems (IRS) Project Proposal

|  |
| --- |
| **Date of proposal:** 10 April 2020 |
| **Project Title:** |
| **Sponsor/Client:** *(Name, Address, Telephone No. and Contact Name)* |
| **Background/Aims/Objectives:** |
| **Requirements Overview:** |
| **Resource Requirements (please list Hardware, Software, and any other resources)**  Hardware proposed for consideration:  Software proposed for consideration:   * Pandas * Numpy * Matplotlib |
| **Number of Learner Interns required: (Please specify their tasks if possible)**  A team of three project members required to architect and implement this system. |
| **Methods and Standards:**   |  |  |  | | --- | --- | --- | | **Procedures** | **Objective** | **Key Activities** | |  | | **Requirement Gathering and Analysis** | The team should meet with ISS to scope the details of project and ensure the achievement of business objectives. | 1.        Gather & Analyze Requirements |  | | 2.        Define internal and External Design |  | | 3.        Prioritize & Consolidate Requirements |  | | 4.        Establish Functional Baseline |  | | **Technical Construction** | To develop the source code in accordance to the design. | 1.        Setup Development Environment |  | | To perform unit testing to ensure the quality before the components are integrated as a whole project | 2.        Understand the System Context, Design |  | | 3.        Perform Coding |  | | 4.        Conduct Unit Testing |  | | **Integration Testing and Acceptance testing** | To ensure interface compatibility and confirm that the integrated system hardware and system software meets requirements and is ready for acceptance testing. | 1.        Prepare System Test Specifications |  | | 2.        Prepare for Test Execution |  | | 3.        Conduct System Integration Testing |  | | 4.        Evaluate Testing |  | | 5.        Establish Product Baseline |  | |  |  | | **Acceptance Testing** | To obtain ISS user acceptance that the system meets the requirements. | 1.        Plan for Acceptance Testing |  | | 2.        Conduct Training for Acceptance Testing |  | | 3.        Prepare for Acceptance Test Execution |  | | 4.        ISS Evaluate Testing |  | | 5.        Obtain Customer Acceptance Sign-off |  | |  |  | | **Delivery** | Deploy the system into production (ISS standalone server) environment. | 1.        Software must be packed by following ISS’s standard |  | | 2.        Deployment guideline must be provided in ISS production (ISS standalone server) format |  | | 3.        Production (ISS standalone server) support and troubleshooting process must be defined. |  | |  |  | |

|  |
| --- |
| Team Name: **GROUP 1** |
| Project Title (repeated): **Driver Drowsiness Detection System** |
| System Name (if decided): |
|  |
| Team Member 1 Name: Anirban Kar Chaudhuri |
| Team Member 1 Matriculation Number: A0108517H |
| Team Member 1 Contact (Mobile/Email):  Mobile: 86118180  Email: anirban.karchaudhuri@gmail.com |
|  |
| Team Member 2 Name: MARADANA VIJAYAKRISHNA |
| Team Member 2 Matriculation Number: A0178453W |
| Team Member 2 Contact (Mobile/Email):  Mobile: 93896379  Email: mvskrishna@yahoo.com |
|  |
| Team Member 3 Name: Putrevu Manoj Niyogi |
| Team Member 3 Matriculation Number: A0213557E |
| Team Member 3 Contact (Mobile/Email):  Mobile: 94575890  Email: manojniyogi@yahoo.com |
|  |
| Team Member 4 Name: Sivasankaran Balakrishnan |
| Team Member 4 Matriculation Number: A0065970X |
| Team Member 4 Contact (Mobile/Email):  Mobile: 97379441  Email: bsivaa@gmail.com |

**Team Formation & Registration**

|  |  |  |
| --- | --- | --- |
| **For ISS Use Only** | | |
| **Program Name:** | **Project No:** | **Learner Batch:** |
| **Accepted/Rejected/KIV:** | | |
| **Learners Assigned:** | | |
| **Advisor Assigned:**  Contact: Mr. GU ZHAN / Lecturer & Consultant  Telephone No.: 65-6516 8021  Email: [zhan.gu@nus.edu.sg](mailto:zhan.gu@nus.edu.sg) | | |

# Appendix B : Individual Project Report

|  |
| --- |
| **Project Title:** Driver Drowsiness Detection System |
| **Team Member 4 Name:** |
| **Team Member 4 Matriculation Number:** |
|  |
|  |
| **Learn new technologies:**   1. I learned and became familiar in web development using html and css 2. Learnt deployment of micro web services using Python Flask library 3. Explored and discovered ways for writing efficient codes in Python as well as testing runtime |
| **My Involvement in the Project** |

|  |
| --- |
| **Project Title:** Driver Drowsiness Detection System |
| **Team Member 3 Name:** Putrevu Manoj Niyogi |
| **Team Member 3 Matriculation Number:** A0213557E |
|  |
|  |
| **Learn new technologies:**   1. I learned and became familiar in development using Python and flask frame work. 2. Acquired the working knowledge of Python libraries applicable for Machine Learning projects. 3. Learned and explored the data analysis / data discovery tools (Orange, Panda – Python Library). 4. Upon acquiring the above knowledge, looking forward to work in machine learning / data analysis related projects. |
| **My Involvement in the Project** |

|  |
| --- |
| **Project Title: Driver Drowsiness Detection System** |
| **Team Member Group 2 Name:** MARADANA VIJAYA KRISHNA |
| **Team Member Group 2 Matriculation Number:** A0178453W |
|  |
|  |
| **Technologies learned & applied:**   1. I had picked up and familiar with web development using Python and its flask framework. 2. Acquired the working knowledge of Python libraries applicable for Machine Learning projects. 3. Learned and explored the data analysis / data discovery tools (Orange, Panda – Python Library). 4. Also learned and impressed with working and implementation of ASR, speech recognition and natural language processing technologies. 5. Upon acquiring the above knowledge, looking forward to work in machine learning / data analysis related projects. |
| **My Involvement in the Project** |

|  |
| --- |
| **Project Title:** Driver Drowsiness Detection System |
| **Team Member 1 Name:** Sivasankaran Balakrishnan |
| **Team Member 1 Matriculation Number:** A0065970X |
|  |
|  |
| **Learn new technologies:**   1. I had picked up and familiar with web development using Python and its flask framework. 2. Acquired the working knowledge of Python libraries applicable for Machine Learning projects. 3. Learned and explored the data analysis / data discovery tools (Orange, Panda – Python Library). 4. Also learned and impressed with working and implementation of ASR, speech recognition and natural language processing technologies. 5. Upon acquiring the above knowledge, looking forward to work in machine learning / data analysis related projects. |
| **My Involvement in the Project** |