
TED UNIVERSITY

# CMPE 492

# Test Plan Report

# Project Name: Drive Safe-Off

**Team Members**

Gökçe BEKAR

Rabia Esra ŞENDUR

Rumeysa OMAY

Oğuzhan Uğur SARISAKALOĞLU

**Supervisor**

Venera ADANOVA

**Jury Members**

Aslı GENÇTAV

Orkunt SABUNCU

# Table of Contents

## Introduction

A part of The Drive Safe-Off is a feedback system in the risky settings. The entire product must be tested by various test quadrants that will check the assurance of usage. To sum up, this test plan enables our team to classify test, and it will be helpful for planning and identify the test resources early.

Unit test are critical to the success of an application. They really help to make sure that the functionality of the application works as it should because this testing asses to observe each functional part of software. Lastly, The test plan describes the period where quality will be testing the product and providing feedback and the total test period extended to seven weeks.

## Unit Testing

Unit testing is a software testing type that helps to test each unit or component of the software. For object-oriented programming, a unit can refer to the interface, such as class, or refer to a function or method. During the unit testing stages, we made our UML diagram as a reference. We performed unit tests for each individual component as shown in Figure 1 below. Thus, we made sure that each component in the UML diagram works correctly and effectively. Furthermore, since we realized our project gradually by taking our UML diagram as a reference, we ensured that the whole system works regularly and correctly with all the components of the system.

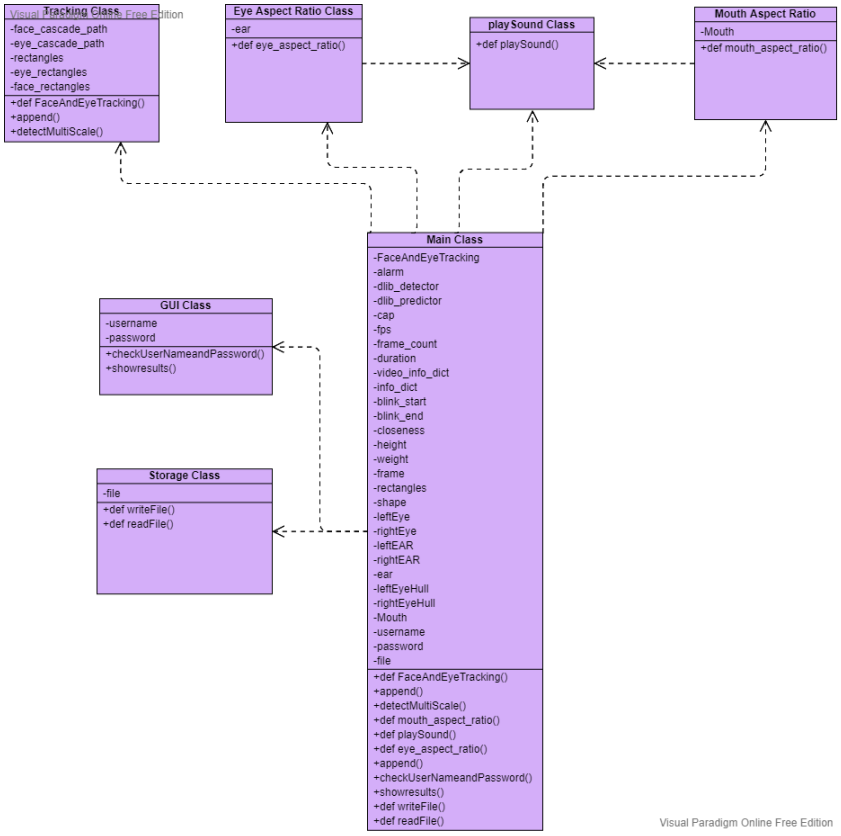


Figure 1: UML Diagram

**Testing Driver's Eye**

It is a test to test whether the driver falls asleep or keeps his eyes closed for a long time while driving. It will take video as its input, and this video will consist of a driver keeping its eye closed for a long time. The eyes will be detected for each frame of the video, and the EAR values of the detected eyes will be calculated. According to the result of the model, if the eye is closed for a long time (if the model predicts the value of being closed repeatedly), it will give a warning.

| | |
|---|---|
| ***Use case name*** | **DriversEye** |
| ***Input*** | Video |
| ***Feature to be tested*** | Alert if driver closes her/his eyes for a long time |
| ***Feature Pass/Fail Criteria*** | If it gives warning where her/his eyes are closed for a long time, it is pass, if it does not, it fails |
| ***Exit condition*** | Ends when the last frame in the video is taken |

**Testing Driver's Mouth**

It is a test to check whether the driver is yawning, or his mouth is open for a long time while driving. It will take video as its input, and this video will consist of a driver exhibiting flexing behavior. The video will detect the mouth for each frame, and the MAR value of the detected mouth will be calculated. If the output (MAR value) is high and constantly repeating, it will give a warning to the driver.

| | |
|---|---|
| ***Use case name*** | DriversMouth |
| ***Input*** | Video |
| ***Feature to be tested*** | Alert if driver opens her/his mouth for a long time |
| ***Feature Pass/Fail Criteria*** | If it gives warning where her/his mouth is opened for a long time, it is pass, if it does not, it fails |
| ***Exit condition*** | Ends when the last frame in the video is taken |

## Usability Testing

We aim to design the interface of our project as simple as possible to appeal to different types of users. With a simple and convenient interface as possible, users will be able to handle their transactions in a short time. When the user wants to run a program, this process will be carried out by pressing a key. Also, with the other interface page, the supervisor will log into the system by entering user information (e.g. username & password) easily. When it enters the system, it will be able to access the page with the data thanks to the titles in the navigation bar. As a result, we aim to ensure that they can handle the operations they want in an easy and effective manner and without any problems.

## Integration Testing

After completing our unit testing stages, we applied an integration test to observe whether the whole system works properly. Integration tests enabled us to observe the incompatibilities and performance losses in the system. In line with these observations, we had the chance to identify the deficiencies in the system. Also, we aim to identify the places where we need to make changes in the system and improve these parts.

**Testing the Driver's Behavior**

It is a test to see if the driver exhibits an anomaly behavior while driving. This anomaly behavior is that the eye must be closed for a long time and the mouth must be open for a long time, that is, the driver must be in a yawning state. It will take video as input. In the video content, there will be a driver exhibiting long stretch and long-term eye closing behavior. Long-term closure and yawning of the eye, which have been tested separately in unit tests, will be tested with the integration test. In the video, EAR and MAR values will be calculated after the eye and mouth are detected for each frame, EAR and MAR values will be calculated, it will warn the user if the eye is closed for a long time and the mouth is open for a long time. In this way, it will be seen that Tracking, Eye Aspect Ratio, Mouth Aspect Ratio classes how to integrate with each other.

| | |
|---|---|
| ***Use case name*** | Driver'sBehaviour |
| ***Input*** | Video |
| ***Feature to be tested*** | Alert if driver opens her/his mouth or closes her/his eyes for a long time |
| ***Feature Pass/Fail Criteria*** | If it gives warning where her/his mouth is opened or her/his eyes are closed for a long time, it is pass, if it does not, it fails |
| ***Exit condition*** | Ends when the last frame in the video is taken |

## System Testing

We made evaluations of the performance of our program with the observations we made during the system testing phase. While measuring the performance of our system, we considered the accuracy and failure of the program components. In this test phase, we used the UML diagram of our system as shown in Figure 1 as a reference. We calculated the accuracy and error rate of each component in our UML diagram. In this way, we found out whether the components in our system were working correctly. In addition, when it was a component has a high error rate, we carried out the necessary researches to increase performance and implemented them. For example, when we finished the tracking part in the system, we calculated the error rates to measure its performance. According to the results, we made the necessary adjustments in the code to improve the performance of the system (e.g. increasing fps rate, decreasing scale, etc.)

## Test Schedule

| Name | Desc |
|---|---|
| Prep | Two weeks of preparation for the test, which includes setting up the system for test |
| Test | Test plan execution and agile delivery of test results |
| Closure | Final summary of the findings of the alpha stage of testing |

As can be seen in Test Phases table, while creating the test plan, the intended draft is as follows: First of all, preparations must be made before the test can be performed**.** After the preparations are completed, the test phase is passed. The tests that have been created are made in this phrase. In the Clouse party, the final summary of the outputs will be made into a report.

| Week | Period | Topics/Activities |
|---|---|---|
| 2 | Prep | Program configuration and integration |
| 3 | Test | Unit Tests |
| 4 | Test | Integration Testing |
| 5 | Test | System Testing |
| 6 | Test | Usability Testing |
| 7 | Closure | Reporting And Analysis |

As can be seen in Schedule table, the development process and the testing process continued to be carried out together. Considering the schedule table, the first unit tests were made after the necessary preparations were made. Unit tests will be conducted to check the minimum applied functions in the components. When each component is completed, several modules working in an interconnected way will be tested with the integration test. After these two tests are completed, a usability test is applied to see if the system can be used by the user. A report is obtained from the results obtained.

| Feedback Type | Objectives |
|---|---|
| Bug Reports | Test quality, interoperability, and real-world performance |
| Suggestions | Measure acceptance, prioritize backlog, and generate new ideas |

At the end of the test period, we are going to finalize tests have done by two measurements. The result measurement we use consist of two type they are bug reports and suggestions. A bug report shows the unsuccessful test and enables us to investigate the bug with details.

## Glossary

**EAR(Eye Aspect Ratio):** Eye Aspect Ratio is an estimate of the eye opening state.

**MAR(Mouth Aspect Ratio):** Mouth Aspect Ratio is an estimate of the eye opening state.