|  |  |
| --- | --- |
| Academic Year : 2020-21 | |
| Subject: Software Engineering CLASS:SE (I) SEMESTER: IV | |
| ERP NO: 34  Teams No: 20 | **NAME**: Kaustubh Shrikant Kabra |
| Assignment No.: 6 | **Date of submission**:6th June,2021 |

**Title of assignment:** Perform manual testing of your mini project.

**Theory: Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Here are the benefits of using software testing:

* **Cost-Effective:**It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of software testing, it costs less to fix.
* **Security:**It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.
* **Product quality:**It is an essential requirement of any software product. Testing ensures a quality product is delivered to customers.
* **Customer Satisfaction:**The main aim of any product is to give satisfaction to their customers. UI/UX Testing ensures the best user experience.

# **Manual Testing**

Manual testing is a software testing process in which test cases are executed manually without using any automated tool. All test cases executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not. Test cases are planned and implemented to complete almost 100 percent of the software application. Test case reports are also generated manually.

Manual Testing is one of the most fundamental testing processes as it can find both visible and hidden defects of the software. The difference between expected output and output, given by the software, is defined as a defect. The developer fixed the defects and handed it to the tester for retesting.

Manual testing is mandatory for every newly developed software before automated testing. This testing requires great efforts and time, but it gives the surety of bug-free software. Manual Testing requires knowledge of manual testing techniques but not of any automated testing tool.

Manual testing is essential because one of the [software testing](https://www.javatpoint.com/software-testing-tutorial) fundamentals is "100% automation is not possible."

**Manual Testing #1:**

|  |  |
| --- | --- |
| **Test Case Field** | **Description** |
| **Test case ID:** | "TC\_ML\_1" indicating "Machine Learning Model Test Case#1." |
| **Test Priority:** | High |
| **Name of the Module**: | Driver drousiness\_model\_training module |
| **Test Designed by**: | Kaustubh S Kabra |
| **Date of test designed**: | 29th May,2021 |
| **Test Executed by**: | Kaustubh S Kabra |
| **Date of the Test Execution**: | 5th June,2021 |
| **Name or Test Title**: | Model Training |
| **Description/Summary of Test**: | Model training in machine language is the process of feeding an ML algorithm with data to help identify and learn good values for all attributes involved. |
| **Pre-condition**: | 1. Collect Data. 2. Prepare the data. 3. Choose the model |
| **Dependencies**: | Data Dependencies: The behavior of an ML system is dependent on the behavior and qualities of its input features. As the input data for those features changes, so too will your model.If the input data for that feature abruptly changes, your model's behavior might also abruptly change in undesirable ways. |
| **Test Steps**: | 1. Local development testing. 2. Testing in CI/CD. 3. Stage testing / Shadow testing. |
| **Test Data**: | 10K Images of Eyes. |
| **Expected Results**: | Detection of open or close eye. |
| **Post-Condition**: | Will be able to test the condition of drowsiness using eyes information. |
| **Actual Result**: | After test execution, it fulfills all test conditions. |
| **Status (Fail/Pass):** | **Pass** |

**Manual Testing #2:**

|  |  |
| --- | --- |
| **Test Case Field** | **Description** |
| **Test case ID:** | "TC\_RTD\_2" indicating "Real Time Detection Test Case#2." |
| **Test Priority:** | High |
| **Name of the Module**: | Drowsiness Detection |
| **Test Designed by**: | Kaustubh S Kabra |
| **Date of test designed**: | 29th May,2021 |
| **Test Executed by**: | Kaustubh S Kabra |
| **Date of the Test Execution**: | 5th June,2021 |
| **Name or Test Title**: | Real Time Detection Testing |
| **Description/Summary of Test**: | Real-time testing is the process of testing real-time computer systems. Checking cases generated must capture the functional and real time information needed to test systems. |
| **Dependencies**: | Camera Angle, Brightness on face, Image pixels. |
| **Test Steps**: | 1. Take image as input from a camera. 2. Detect the face in the image and create a Region of Interest (ROI). 3. Detect the eyes from ROI and feed it to the classifier. 4. Classifier will categorize whether eyes are open or closed. 5. Calculate score to check whether the person is drowsy. |
| **Test Data**: | Real time Image Capture data set. |
| **Expected Results**: | Display score of drowsiness and activation of alarm after score goes above limit score. |
| **Post-Condition**: | Display of Score and Status of Eye(Open or Closed) |
| **Actual Result**: | After test execution, it fulfil all test conditions. |
| **Status (Fail/Pass):** | **Pass** |

**Test Case #1:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TC\_ML\_1 | | | **Test Case Description** | | | | Model Training. | | | | | | | | | | | | |
| **Created By** | | Kaustubh S Kabra | | | **Reviewed By** | | | | Prof.SN Zaware | | | | | | **Version** | | | | | 1.1 | |
|  | | |  |  |  | | | | |  | |  | | | |  |  | |  |  |  |
| **QA Tester’s Log** | | Accuracy of detection of eye status in model in version 1.1 | | | | | | | | | | | | | |  |  | |  |  |  |
|  | | |  |  |  | | | | |  | |  | | | |  |  | |  |  |  |
| **Tester's Name** | | Kaustubh Kabra | | | **Date Tested** | | | 5th June,2021 | | | | | | **Test Case (Pass/Fail/Not Executed)** | | | | | | Pass | |
|  |  | | |  |  | |  |  | | |  | | |  | | | | |  |  |  |
| **S #** | **Prerequisites:** | | | | |  | | **S #** | | | **Test Data** | | | | | | | | | | |
| 1 | Collect Data. | | | | |  | | 1 | | | 10K Images of Eyes. | | | | | | | | | | |
| 2 | Prepare the data. | | | | |  | |  | | |  | | | | | | | | | | |
| 3 | Choose the model | | | | |  | |  | | |  | | | | | | | | | | |
|  |  | | |  |  | |  |  | | |  | | | | | | |  |  |  |  |
| **Test Scenario** | Verify on model training for eye status detection. | | | | | | | | | | | | | | | | |  |  |  |  |
|  |  | | |  |  | | | | |  | | |  | | |  | |  |  |  |  |
| **Step #** | **Step Details** | | | | **Expected Results** | | | | | | | | **Actual Results** | | | | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Local development testing. | | | | Performed as expected. | | | | | | | | As Expected | | | | | | Pass | | |
| 2 | Testing in CI/CD. | | | | Performed as expected. | | | | | | | | As Expected | | | | | | Pass | | |
| 3 | Stage testing /Shadow testing. | | | | Performed as expected. | | | | | | | | As Expected | | | | | | Pass | | |

**Test Case #2:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TC\_RTD\_2 | | | **Test Case Description** | | | | Real Time Detection Testing | | | | | | | | | | | | |
| **Created By** | | Kaustubh S Kabra | | | **Reviewed By** | | | | Prof.SN Zaware | | | | | | **Version** | | | | | 1.1 | |
|  | | |  |  |  | | | | |  | |  | | | |  |  | |  |  |  |
| **QA Tester’s Log** | | Real time drowsiness detection working in version 1.1 | | | | | | | | | | | | | |  |  | |  |  |  |
|  | | |  |  |  | | | | |  | |  | | | |  |  | |  |  |  |
| **Tester's Name** | | Kaustubh Kabra | | | **Date Tested** | | | 5th June,2021 | | | | | | **Test Case (Pass/Fail/Not Executed)** | | | | | | Pass | |
|  |  | | |  |  | |  |  | | |  | | |  | | | | |  |  |  |
| **S #** | **Prerequisites:** | | | | |  | | **S #** | | | **Test Data** | | | | | | | | | | |
| 1 | Working of Trained Model for detection. | | | | |  | | 1 | | | Image Capture | | | | | | | | | | |
|  |  | | | | |  | | 2 | | | Trained Model | | | | | | | | | | |
|  |  | | |  |  | |  |  | | |  | | | | | | |  |  |  |  |
| **Test Scenario** | Verify on Score and Detection of Drowsiness. | | | | | | | | | | | | | | | | |  |  |  |  |
|  |  | | |  |  | | | | |  | | |  | | |  | |  |  |  |  |
| **Step #** | **Step Details** | | | | **Expected Results** | | | | | | | | **Actual Results** | | | | | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Take image as input from a camera. | | | | 5 Image capture after every 1 second. | | | | | | | | As Expected | | | | | | Pass | | |
| 2 | Detect the face in the image and create a Region of Interest (ROI). | | | | Detection of image and creating the ROI as region of eye in image. | | | | | | | | As Expected | | | | | | Pass | | |
| 3 | Detect the eyes from ROI and feed it to the classifier | | | | Croping of eye region from image | | | | | | | | As Expected | | | | | | Pass | | |
| 4 | Classifier will categorize whether eyes are open or closed. | | | | Checking eye is open or closed. | | | | | | | | As Expected | | | | | | Pass | | |
| 5 | Calculate score to check whether the person is drowsy | | | | Display of score and activation od alarm after score crosses limit score. | | | | | | | | As Expected | | | | | | Pass | | |

**Conclusion:** Thus, we have performed manual testing of our project and have studied the test cases in detail. We have verified that our software matches the expected requirements and is running successfully in every test case.