TEST PLAN FOR

<<DRIVER ANALYSIS>>

ChangeLog

Version	Change Date	By	Description
001	06.11.23	Shivi Goel	Intial Draft

1 Introduction

The project aims to develop a system for analyzing the behavioral patterns of drivers and assessing their safety skills. The system will collect data on drivers' behavior, such as speed, vertical acceleration, lateral acceleration, roll, pitch, yaw and other driving-related actions. This data will be analyzed using machine learning algorithms to identify patterns and predict his safety skills. Overall, the goal is to improve road safety by encouraging safe driving habits and reducing the risk of accidents.

1.1 Scope

1.1.1 In Scope

Scope defines the features, functional or non-functional requirements of the software that **will be** tested

Functional Requirements:

Input Requirements: The input for this project contains various features on which the model is trained and tested. This includes speed, lateral acceleration, longitudinal acceleration, roll, pitch and yaw.

Driver Analysis Algorithm: The project shall implement several machine learning algorithm to combine the effect of various features and predict whether the person is safe or aggressive driver.

User Interface: The project should provide a user-friendly interface for users to check their details and understand the analysis of their driving skills.

Output Requirements: The project should be able to classify drivers as safe or aggressive.

Non-Functional Requirements:

Scalability: The model should be scalable to handle a large number of drivers and vehicles.

Maintainability: The model should be easy to maintain and update.

Portability: The model should be portable and should be able to be deployed on a variety of platforms.

Cost: The model should be cost-effective to develop and deploy.

1.1.2 Out of Scope

Out of Scope for our project will be fancy or artistic styles, special equipment, analyzing videos in real-time and advanced editing features for driving data.

1.2 Quality Objective

Here make a mention of the overall objevtie that you plan to achieve withou your testing

Some objectives of your testing project could be

- Ensure the Application Under Test conforms to functional and non-functional requirements
- Ensure the AUT meets the quality specifications defined by the client
- Bugs/issues are identified and fixed before go live
- Optimize the ML model for efficiency and speed to enable rapid evaluation of driving skills while preserving high-quality accuracy.
- Emphasize a user-friendly and intuitive interface to enhance the user experience.
- Ensure cross-device and cross-browser compatibility to guarantee the system's functionality on a variety of platforms and operating systems. This approach ensures that a broader user base can access and utilize the system effectively for evaluating driving skills.

1.3 Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like

- **QA Analyst** – Shivi Goel
- **Test Manager** – Prof. Shreela Pareek
- **Configuration Manager** – Prof. Neha Shukla
- **Developers** – Shivi Goel, Shikha Dixit, Sneha Jaiswal
- **Installation Team**- Prof. Shreela Pareek, Prof. Neha Shukla, Shivi Goel, Shikha Dixit, Sneha Jaiswal

2 Test Methodology

2.1 Overview

To ensure that our project runs well, we are employing an iterative testing strategy. This means we test it in small phases, beginning with determining whether each component works independently. Then we observe how the various parts interact with one another. We continue to test as we make improvements and add new features. This manner, we can ensure that our project always works properly, even after modifications.

2.2 Test Levels

Test Levels define the Types of Testing to be executed on the Application Under Test (AUT). The Testing Levels primarily depends on the scope of the project, time and budget constraints.
We aim to test our project at the following levels :
1) Unit Testing: This is the lowest level of testing and focuses on individual components or functions within the software. Developers often perform unit tests to verify that specific parts of the code work correctly.
2) Integration Testing: This level of testing checks how different components or modules of the software work together. It ensures that integrated parts of the software function as intended.
3) System Testing: At this level, the entire system is tested as a whole. It verifies that the software meets its specified requirements and functions properly in its intended environment.

2.3 Test Completeness

Here you define the criterias that will deem your testing complete.
For instance, a few criteria to check Test Completeness would be

- 100% test coverage
- All Manual & Automated Test cases executed
- All open bugs are fixed or will be fixed in next release
- Completed comprehensive tests for evaluating driving behaviors, confirming the successful analysis of diverse input data, all of which align with our established criteria for assessing driving skills.

3 Test Deliverables

TEST CASES:

S.No.	Input							Expected Output	Actual Output	Remarks
	Speed	Vertical_Acc	Lateral_Acc	Longitudinal_Acc	Roll	Pitch	Yaw			
1	46.5	-0.052	-0.009	-0.044	-1.474	0.019	-0.73	Safe	Safe	Pass
2	48.8	-0.074	0.056	-0.067	-1.602	0.226	1.934	Safe	Safe	Pass
3	56.5	-0.016	-0.224	-0.135	-1.654	0.022	1.786	Aggressive	Aggressive	Pass
4	61.8	-0.053	-0.059	0.003	-1.57	-0.071	-2.085	Safe	Safe	Pass
5	63.2	-0.064	0.033	-0.008	-1.497	0.017	-0.617	Safe	Safe	Pass
6	112.8	0.02	-0.02	-0.014	-1.513	0.059	1.923	Safe	Safe	Pass
7	138.5	0.036	0.005	-0.018	-1.567	-0.008	0.457	Aggressive	Aggressive	Pass
8	109.1	-0.005	-0.006	-0.016	-1.487	0.032	0.133	Safe	Safe	Pass
9	108.9	-0.019	0.001	0.007	-1.524	0.022	2.038	Safe	Safe	Pass
10	144.1	0.04	-0.021	0.031	-1.543	0.07	3.037	Aggressive	Aggressive	Pass
11	140.6	0.056	0.023	-0.004	-1.593	-0.007	2.828	Aggressive	Aggressive	Pass

BOUNDARY VALUE ANALYSIS:

Speed		
	Boundary Value	

Less than 10	Min = 10
	Min+1 = 11
	Nominal = 64.35
	Max-1= 127.7
	Max = 128.7
		Greater than 128.7

Safe	Unsafe / Safe	Unsafe

Vertical Acceleration		
	Boundary Value	

Less than -0.188	Min = -0.188
	Min+1 = -0.187
	Nominal = -0.0515
	Max-1= 0.084
	Max = 0.085
		Greater than 0.085

Unsafe	Unsafe / Safe	Unsafe

Lateral Acceleration		
	Boundary Value	

Less than -0.391	Min = -0.391
	Min+1 = -0.390
	Nominal = -0.131
	Max-1= 0.128
	Max = 0.129
		Greater than 0.129

UnSafe	Unsafe / Safe	Unsafe

EQUIVALENCE CLASS PARTITION:

Partition	Input Factor	Range	Output	Remarks
Partition 1	Speed	>=128.7	Aggressive	All factors are above range
	Vertical Acc.	>=0.085		
	Lateral Acc.	-0.391>= && >=0.130		
Partition 2	Speed	<=128.7	Safe/ Aggressive	All factors are out of range
	Vertical Acc.	<=0.085		
	Lateral Acc.	-0.391<= && <=0.130		
Partition 3	Speed	>=128.7	Aggressive	One factor out of range
	Vertical Acc.	>=0.085		
	Lateral Acc.	-0.391<= && <=0.130		

S.No.	Input							Expected Output	Actual Output	Remarks
	Speed	Vertical_Acc	Lateral_Acc	Longitudinal_Acc	Roll	Pitch	Yaw			
1	128.5	-0.074	-0.148	-0.058	-1.626	-0.014	-2.051	Safe	Safe	Pass
2	128.7	-0.074	-0.158	-0.058	-1.626	0.32	1.642	Aggressive	Aggressive	Pass
3	145.6	-0.161	-0.505	-0.321	-1.717	-0.444	-0.498	Aggressive	Aggressive	Pass
4	132	-0.085	-0.335	-0.068	-1.634	-0.02	2.092	Aggressive	Aggressive	Pass
5	131.9	-0.084	-0.334	-0.068	-1.633	-0.292	-0.587	Aggressive	Aggressive	Pass
6	135.5	-0.095	-0.395	-0.089	-1.648	-0.18	0.059	Aggressive	Aggressive	Pass
7	45.2	0.074	0.124	0.087	-1.412	0.006	-1.814	Safe	Safe	Pass

4 Resource & Environment Needs

4.1 Testing Tools

1. Selenium: Selenium is an open source umbrella project for a range of tools and libraries aimed at supporting browser automation.
2. Mantis Bug Tracking Tool: Mantis Bug Tracker is a free and open source, web-based bug tracking system. The most common use of Mantis BT is to track software defects.

4.2 Test Environment

It mentions the minimum **hardware** requirements that will be used to test the Application.

Following **software's** are required in addition to client-specific software.

- Windows 8 and above
- Office 2013 and above
- MS Exchange, etc.

5 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
API	Application Program Interface
AUT	Application Under Test