TEST PLAN FOR

<GESTURE VIRTUAL MOUSE >

ChangeLog

Version	**Change Date**	**By**	**Description**
version number	Date of Change	Name of person who made changes	Description of the changes made
1.0	11/10/2023	Ayush and Deepanshu	Fix bugs

1 Introduction

The testing strategy and methodologies employed for the "Gesture Control Virtual Mouse" project are critical components of ensuring the functionality, reliability, and overall quality of the software. This section provides an overview of the test processes, workflows, and methodologies that have been applied to achieve the project's testing objectives.

1.1 Scope

The scope of the testing effort is defined to outline the boundaries of the testing process. It clarifies what aspects of the software will be thoroughly tested and what will not be covered.

1.1.1 In Scope

In Scope encompasses the features, functional, and non-functional requirements of the "Gesture Control Virtual Mouse" software that will be rigorously tested. This includes but is not limited to:

Gesture recognition accuracy: Testing the software's ability to accurately recognize a range of predefined gestures.
Cursor control: Verifying that the virtual mouse cursor responds appropriately to recognized gestures.
Functional gestures: Testing functional gestures like left-click, right-click, zoom, scroll, and others.
Usability: Assessing the user interface for ease of use and user-friendliness.
Compatibility: Ensuring compatibility with various hand recognition devices and operating systems.
Performance: Evaluating the software's performance in terms of response time and resource utilization.

1.1.2 Out of Scope

Out of Scope, conversely, delineates the features, functional, or non-functional requirements of the "Gesture Control Virtual Mouse" software that will not be subjected to testing. These may include:

Hardware testing: Testing of the actual hardware devices used for hand recognition, as this falls outside the software's purview.
Third-party software: Assessing the performance or compatibility of other software applications not directly related to the virtual mouse functionality.
Non-standard gestures: Unconventional gestures or highly specialized gestures that are not part of the predefined set.
Accessibility testing: In-depth analysis of accessibility features for users with disabilities, which might be a separate testing process.
The scope defines the boundaries within which the testing team will operate, ensuring that efforts are concentrated on critical aspects of the "Gesture Control Virtual Mouse" software. This delineation assists in planning and executing an efficient and effective testing process to achieve the project's quality and performance goals.

1.2 Quality Objective

The quality objective for the "Gesture Control Virtual Mouse" testing project is to ensure the software meets a set of defined criteria, including both functional and non-functional requirements. The primary objectives of this testing effort are as follows:

Conformance to Functional and Non-functional Requirements: The foremost objective is to validate that the Application Under Test (AUT), in this case, the "Gesture Control Virtual Mouse" software, aligns with both its functional and non-functional requirements. This entails confirming that the software accurately recognizes and responds to predefined gestures, meets usability and performance expectations, and adheres to compatibility and security standards.

Client Quality Specifications: The testing project aims to verify that the AUT complies with the quality specifications outlined by the client or stakeholders. This involves assessing whether the software aligns with the client's expectations and delivers the intended user experience.

Bug Identification and Resolution: Another key objective is to detect and report any bugs, issues, or anomalies present in the software. The goal is to identify these issues early in the testing process, allowing for timely resolution before the software's release or "go live." This includes both functional defects, such as incorrect gesture recognition, and non-functional issues like performance bottlenecks or usability challenges.

1.3 Roles and Responsibilities

- Ayush Kumar and Deepanshu Singh done all QA Analyst, Develop and Test Manager
- Configuration Manager: Mr Pradeep Tyagi

2 Configuration Manager Test Methodology

2.1 Overview

The Waterfall model is a traditional project management and software development approach that is characterized by a linear and sequential sequence of phases. It is structured into distinct phases, each of which must be completed before the next one begins. The typical phases of the Waterfall model include Requirements, Design, Implementation, Testing, Deployment, and Maintenance.

2.2 Test Levels

1. Unit Testing: Performed at the code level, this level of testing focuses on individual components and their correctness.
2. Integration Testing: Validates the interaction between different components to ensure they work cohesively as a whole.
3. System Testing: This level verifies that the complete system meets its requirements and performs its intended functions as a unified entity.
4. Acceptance Testing: Conducted by end-users or stakeholders, acceptance testing ensures that the software meets business and user needs

2.3 Suspension Criteria and Resumption Requirements

Data Quality Issues: If significant data quality issues are identified during the analysis and model development stages, testing may be suspended to address these issues. This could include issues such as incomplete or inaccurate data, which could adversely affect the reliability of the machine learning models.

Model Development Challenges: If the machine learning model development encounters unforeseen difficulties or obstacles, suspending testing to reevaluate the modeling process might be necessary. This could include issues like overfitting, underfitting, or inadequate model performance.

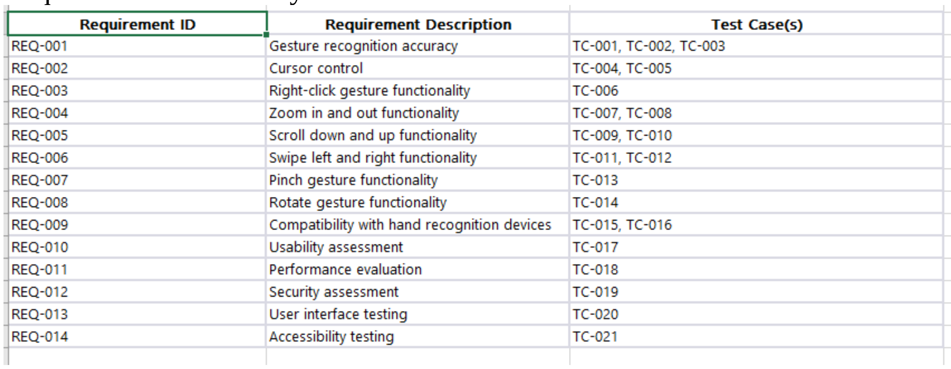
Resource Constraints: If resource constraints, such as hardware or software limitations, impede the testing process, suspension criteria may be invoked to address these limitations and ensure testing can resume effectively.

2.4 Test Completeness

- 80% test coverage
- All Manual test cases have been executed.
- All open bugs are fixed Test Deliverables

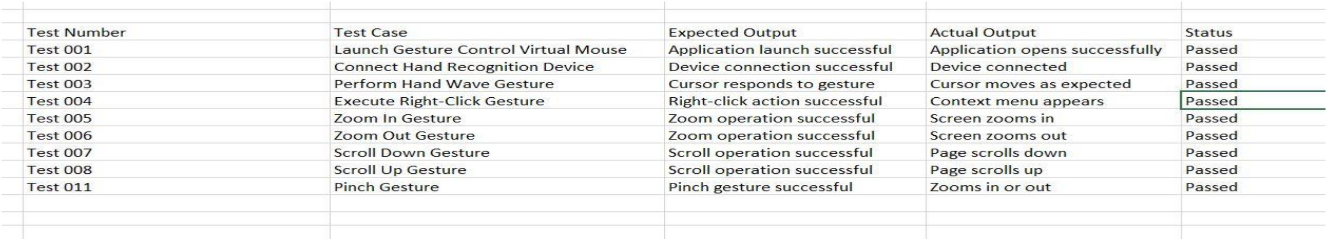
3 Test Deliverables

- Requirement Traceability Matrix



Requirement ID	Requirement Description	Test Case(s)
REQ-001	Gesture recognition accuracy	TC-001, TC-002, TC-003
REQ-002	Cursor control	TC-004, TC-005
REQ-003	Right-click gesture functionality	TC-006
REQ-004	Zoom in and out functionality	TC-007, TC-008
REQ-005	Scroll down and up functionality	TC-009, TC-010
REQ-006	Swipe left and right functionality	TC-011, TC-012
REQ-007	Pinch gesture functionality	TC-013
REQ-008	Rotate gesture functionality	TC-014
REQ-009	Compatibility with hand recognition devices	TC-015, TC-016
REQ-010	Usability assessment	TC-017
REQ-011	Performance evaluation	TC-018
REQ-012	Security assessment	TC-019
REQ-013	User interface testing	TC-020
REQ-014	Accessibility testing	TC-021

- Test Cases



Test Number	Test Case	Expected Output	Actual Output	Status
Test 001	Launch Gesture Control Virtual Mouse	Application launch successful	Application opens successfully	Passed
Test 002	Connect Hand Recognition Device	Device connection successful	Device connected	Passed
Test 003	Perform Hand Wave Gesture	Cursor responds to gesture	Cursor moves as expected	Passed
Test 004	Execute Right-Click Gesture	Right-click action successful	Context menu appears	Passed
Test 005	Zoom In Gesture	Zoom operation successful	Screen zooms in	Passed
Test 006	Zoom Out Gesture	Zoom operation successful	Screen zooms out	Passed
Test 007	Scroll Down Gesture	Scroll operation successful	Page scrolls down	Passed
Test 008	Scroll Up Gesture	Scroll operation successful	Page scrolls up	Passed
Test 011	Pinch Gesture	Pinch gesture successful	Zooms in or out	Passed

4 Resource & Environment Needs

4.1 Testing Tools

Required to test the project is manual testing

4.2 Test Environment

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

- Windows 10 or macOS 10.15 or later
- Minimum 4 GB RAM
- Minimum Intel Core i3 or equivalent processor

- Webcam and microphone for input

5 Terms/Acronyms

TERM/ACRONYM	DEFINITION
API	Application Program Interface
AUT	Application Under Test
GVM	Gesture Virtual Mouse
HMI	Human Machine Interface