## Test Plan: Smart Doorbell Camera

Table of Contents

[1. Introduction 1](#_Toc190301075)

[2. Scope 1](#_Toc190301076)

[3. Assumptions 2](#_Toc190301077)

[4. Test Strategy 2](#_Toc190301078)

[5. Test Environment 2](#_Toc190301079)

[6. Defect Management Cycle 3](#_Toc190301080)

[7. Entry and Exit Criteria 3](#_Toc190301081)

[8. Roles and Responsibilities 3](#_Toc190301082)

ChangeLog

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| 0.1 | 10/02/2025 | Suguna K | Drafted the Plan |
|  |  |  |  |

# 1. Introduction

This document outlines the test plan for the Smart Doorbell Camera, focusing on automation and feature validation. The product is a battery-powered, wireless doorbell camera equipped with a single button, a camera, a motion sensor, and advanced facial recognition technology. It connects to a local hub, which provides internet access. Users can access camera settings, view videos, and monitor battery life through a mobile-responsive website. The unique selling point of the product is its advanced facial recognition technology, enabling users to identify known individuals upon arrival. Videos are automatically tagged with the names of recognized individuals, all accessible via the web interface.

# 2. Scope

This test plan covers the functional and non-functional testing of the Smart Doorbell Camera, including:

* **Core Functionality:** Button press, video recording, motion detection, wireless connectivity, and battery life.
* **Facial Recognition:** Accuracy of identification, tagging of videos, and management of known individuals via the web interface.
* **Web Interface:** Functionality, usability, performance, and security.
* **Installation and Setup:** Ease of setup and connection to the hub and web interface.
* **Performance Testing:** Battery life, video quality under various conditions, and network performance.
* **Security Testing:** Vulnerability assessment of the device and web interface.

# 3. Assumptions

* The doorbell connects to a local hub, which then connects to the internet.
* The web interface is compatible with major web browsers (e.g., Chrome, Firefox, Safari).
* Detailed requirements specifications are available for all features.
* A stable test environment is provided, including network infrastructure and test devices.
* The CLI is available for testing during development but will be disabled in production.
* Facial recognition models are pre-trained and provided by the development team.
* Test data (e.g., known individuals' facial data) is available for validation.
* Battery life is expected to be similar to other battery-powered doorbell cameras, depending on usage and environmental factors.

# 4. Test Strategy

The testing strategy will focus on automation and feature validation, utilizing a combination of techniques:

* **Functional Testing:** Verifying all features work as expected, using both manual and automated tests.
* **Facial Recognition Testing:** Rigorous testing of the facial recognition accuracy, including edge cases and variations in lighting and angles.
* **Integration Testing:** Verify communication between doorbell, local hub, and webapp.
* **Regression Testing:** Ensuring existing functionality is not broken by new code changes, primarily through automated test suites.
* **Performance Testing:** Evaluating battery life, video quality, and network performance under various conditions.
* **Usability Testing:** Assessing the ease of use of the web interface.
* **Security Testing:** Identifying potential vulnerabilities in the device and web interface.

# 5. Test Environment

* **Hardware:**
  + Doorbell camera (development unit with CLI enabled).
  + Local hub.
  + Mobile devices (iOS and Android) for webapp testing.
* **Software:**
  + Webapp (latest version).
  + Facial recognition engine.
  + Test automation frameworks (Pytest, Cypress).
* **Network:**
  + Stable Wi-Fi connection for wireless communication.
  + Simulated internet connection for cloud-based testing.

# 6. Defect Management Cycle

The following defect management cycle will be used to track and resolve defects:

1. **Identification:** Testers identify and document defects encountered during testing. This includes detailed descriptions, steps to reproduce, severity level, and any supporting evidence (e.g., screenshots, logs).
2. **Logging:** Defects are logged in a defect tracking system (e.g., Jira, Bugzilla, Xray). Each defect is assigned a unique ID and relevant attributes.
3. **Triage:** The test lead and development team triage the defects, prioritizing them based on severity and impact.
4. **Assignment:** Defects are assigned to developers for resolution.
5. **Resolution:** Developers fix the defects and provide a resolution status.
6. **Verification:** Testers retest the fixed defects to ensure they are resolved.
7. **Closure:** If the verification is successful, the defect is closed. If not, the defect is reopened and reassigned.

# 7. Entry and Exit Criteria

* **Entry Criteria:** Stable build of the doorbell firmware and web interface, test environment set up, test cases designed.
* **Exit Criteria:** All planned tests executed, all critical and high-priority defects resolved and retested, test summary report generated. Additionally, all defects within the defined scope should have a final status (Closed or Deferred).

# 8. Roles and Responsibilities

* **Test Lead:** Responsible for planning, executing, and reporting on testing activities, including managing the defect tracking process and ensuring timely resolution of defects.
* **Test Engineers:** Responsible for executing test cases, reporting defects, and developing automated tests.
* **Developers:** Responsible for fixing defects and supporting testing activities.
* **Test Manager:** Oversees the testing process and provides resources.