

Part I. Description of Overall Test Plan

For our testing strategy, we wanted to prioritize modularity by making sure all components (Hardware, Software, and Database) worked together. Hardware tests will verify the Raspberry Pi and camera's ability to capture clear images in various conditions. Software tests will assess OpenCV's accuracy in bird identification, UI responsiveness, and database functionality. Database tests will check data consistency and performance. We will use normal, abnormal, and boundary testing to evaluate expected and edge-case behavior, with blackbox and whitebox approaches for feature validation and algorithm accuracy. Unit tests will ensure individual module reliability, while integration tests will verify seamless interaction between components. Performance metrics like speed, power consumption, and identification accuracy will be our main consideration in terms of successful vs unsuccessful tests.

Part II. Test Case Descriptions

TC1.1 – Camera Image Capture Test

TC1.2 Ensure the camera module captures clear images in normal conditions.

TC1.3 The Raspberry Pi will capture an image under default daylight conditions. The image will be analyzed for resolution and clarity.

TC1.4 Inputs: Raspberry Pi camera capture command.

TC1.5 Expected Outputs: A clear, focused image with distinguishable details.

TC1.6 Normal

TC1.7 Blackbox

TC1.8 Functional

TC1.9 Unit Test

TC2.1 – Camera Low-Light Performance Test

TC2.2 Verify that the camera captures usable images in low-light conditions.

TC2.3 The Raspberry Pi camera will capture images at dusk or with limited lighting.

The image will be checked for noise and visibility.

TC2.4 Inputs: Low-light environment and camera capture command.

TC2.5 Expected Outputs: A visible image with minimal noise and enough clarity to distinguish features.

TC2.6 Boundary

TC2.7 Blackbox

TC2.8 Performance

TC2.9 Unit Test

TC3.1 – Motion Detection Accuracy Test

TC3.2 Ensure the system correctly detects bird movement.

TC3.3 Simulated bird movement in front of the camera will trigger OpenCV motion detection. Detection success rate will be recorded.

TC3.4 Inputs: Video feed with moving objects (bird-like models and other objects).

TC3.5 Expected Outputs: Movement is detected when a bird enters the frame but not for irrelevant objects like wind-blown leaves.

TC3.6 Normal

TC3.7 Whitebox

TC3.8 Functional

TC3.9 Unit Test

TC4.1 – False Positive Motion Detection Test

TC4.2 Verify that the system does not falsely trigger detection for non-bird motion.

TC4.3 Objects such as leaves, shadows, or insects will be introduced to test the motion detection filter.

TC4.4 Inputs: Video feed with non-bird motion (e.g., moving leaves, shadows).

TC4.5 Expected Outputs: System correctly ignores non-bird movements.

TC4.6 Abnormal

TC4.7 Whitebox

TC4.8 Functional

TC4.9 Unit Test

TC5.1 – Bird Classification Test

TC5.2 Ensure the image classification model correctly identifies bird species.

TC5.3 Images of different bird species will be fed into the OpenCV model to check classification accuracy.

TC5.4 Inputs: Labeled bird images.

TC5.5 Expected Outputs: Correct bird species identified with at least 85% accuracy.

TC5.6 Normal

TC5.7 Blackbox

TC5.8 Functional

TC5.9 Unit Test

TC6.1 – False Identification Test

TC6.2 Test if the system incorrectly identifies non-bird objects as birds.

TC6.3 Images of non-bird objects (squirrels, leaves, hands) will be processed to check if they are misidentified.

TC6.4 Inputs: Images of non-bird objects.

TC6.5 Expected Outputs: The system does not misclassify non-bird objects as birds.

TC6.6 Abnormal

TC6.7 Blackbox

TC6.8 Functional

TC6.9 Unit Test

TC7.1 – UI Responsiveness Test

TC7.2 Ensure the QML-based UI remains responsive during system operation.

TC7.3 The UI will be tested for lag, responsiveness, and crashes while bird identification is running.

TC7.4 Inputs: Continuous user interactions while image processing occurs.

TC7.5 Expected Outputs: UI remains responsive with no significant delays or crashes.

TC7.6 Normal

TC7.7 Blackbox

TC7.8 Performance

TC7.9 Integration Test

TC8.1 – Camera Connectivity Test

TC8.2 Ensure the Raspberry Pi maintains a stable connection with the camera.

TC8.3 The camera will be continuously assessed for an extended period, with random disconnects simulated.

TC8.4 Inputs: Camera connection status checks over time.

TC8.5 Expected Outputs: The camera remains stable; any disconnections are automatically recovered.

TC8.6 Abnormal

TC8.7 Whitebox

TC8.8 Functional

TC8.9 Integration Test

TC9.1 – Power Consumption Test

TC9.2 Measure power consumption during different operational states.

TC9.3 Power usage will be recorded while idle, capturing images, and identifying birds.

TC9.4 Inputs: Device running in different modes, power measurement tools.

TC9.5 Expected Outputs: Power usage remains within expected limits and does not cause system failures.

TC9.6 Normal

TC9.7 Whitebox

TC9.8 Performance

TC9.9 Integration Test

TC10.1 – Outdoor Durability Test

TC10.2 Ensure the system remains functional in outdoor conditions.

TC10.3 The feeder will be tested under different weather conditions (humidity, rain, temperature variations).

TC10.4 Inputs: Exposure to various environmental conditions over time.

TC10.5 Expected Outputs: System remains operational with no critical failures.

TC10.6 Boundary

TC10.7 Blackbox

TC10.8 Performance

TC10.9 Integration Test

TC11.1 – Database Entry Consistency Test

TC11.2 Ensure bird data entries in the database are correctly stored and retrieved without corruption.

TC11.3 Bird images and metadata (species, food type, location) will be inserted, retrieved, and compared for integrity.

TC11.4 Inputs: Sample bird data entries into the database.

TC11.5 Expected Outputs: Retrieved data matches the originally inserted values.

TC11.6 Normal

TC11.7 Whitebox

TC11.8 Functional

TC11.9 Integration Test

TC12.1 – Similar Bird Matching Accuracy Test

TC12.2 Ensure the app suggests similar-looking bird species accurately.

TC12.3 An image of a bird will be uploaded, and the system will generate a list of visually similar birds from the database.

TC12.4 Inputs: Bird images for comparison.

TC12.5 Expected Outputs: The system correctly suggests at least three visually similar birds with an accuracy above 80%.

TC12.6 Normal

TC12.7 Blackbox

TC12.8 Functional

TC12.9 Unit Test

TC13.1 – Database Performance Under Load Test

TC13.2 Assess database response time when handling multiple simultaneous queries.

TC13.3 Multiple concurrent database requests (inserts, queries, deletions) will be performed to check system efficiency.

TC13.4 Inputs: High-volume database transactions.

TC13.5 Expected Outputs: Query response time remains below a predefined threshold without failures.

TC13.6 Boundary

TC13.7 Whitebox

TC13.8 Performance

TC13.9 Integration Test

TC14.1 – Offline Mode Functionality Test

TC14.2 Verify if the app continues working in limited or no internet connectivity scenarios.

TC14.3 The system will attempt to capture and classify a bird image while the network is disabled, then sync data once reconnected.

TC14.4 Inputs: Network loss during app operation.

TC14.5 Expected Outputs: The app functions normally and syncs stored data once connectivity is restored.

TC14.6 Abnormal

TC14.7 Blackbox

TC14.8 Functional

TC14.9 Integration Test

TC15.1 – Multi-Bird Detection Test

TC15.2 Ensure the system can accurately identify multiple birds in a single frame.

TC15.3 An image with multiple birds of different species will be processed, and the system's ability to distinguish and classify them will be recorded.

TC15.4 Inputs: Image with multiple birds.

TC15.5 Expected Outputs: Each bird in the image is identified separately, with at least 85% accuracy.

TC15.6 Normal

TC15.7 Blackbox

TC15.8 Functional

TC15.9 Unit Test

TC16.1 – Edge Case Image Quality Test

TC16.2 Evaluate system performance on blurred, cropped, or obstructed bird images.

TC16.3 The app will be tested with images that have poor focus, partial bird visibility, or background obstructions.

TC16.4 Inputs: Blurred, cropped, or obstructed bird images.

TC16.5 Expected Outputs: System correctly identifies birds when sufficient features are visible and rejects unusable images.

TC16.6 Boundary

TC16.7 Blackbox

TC16.8 Functional

TC16.9 Unit Test

Part III. Test Case Matrix: summarizes the test case coverage (items 1, 6-9 in a tabular format)

	Test Case ID	Normal/Abnormal	Blackbox/Whitebox	Functional/Performan	Unit/Integration
1	TC1	Normal	Black	Function	Unit
2	TC2	Boundary	Black	Performance	Unit
3	TC3	Normal	White	Function	Unit
4	TC4	Abnormal	White	Function	Unit
5	TC5	Normal	Black	Function	Unit
6	TC6	Abnormal	Black	Function	Unit
7	TC7	Normal	Black	Performance	Integration
8	TC8	Abnormal	White	Function	Integration
9	TC9	Normal	White	Performance	Integration
10	TC10	Boundary	Black	Performance	Integration
11	TC11	Normal	White	Function	Integration
12	TC12	Normal	Black	Function	Unit
13	TC13	Boundary	White	Performance	Integration
14	TC14	Abnormal	Black	Function	Integration
15	TC15	Normal	Black	Function	Unit
16	TC16	Boundary	Black	Function	Unit