



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Faculty of
Computing

SECP3106-01 APPLICATION DEVELOPMENT (WBL)

Semester 2 2024/2025

<<Kutip Smart Waste Collection System>>

Software Test Document

Group: GardeniaProMax

Group Members:

No	Name	Matric No
1	Wong Jun Ji	A22EC0117
2	Chen Pyng Haw	A22EC0042
3	Koh Li Hui	A22EC0059
4	Lee Yik Hong	A21BE0376
5	Tiew Chuan Rong	A22EC0112

Section: 01

Lecturer: Dr. Haza Nuzly Bin Abdull Hamed

Date: 12/6/2025

Software Test Document

for

Kutip Smart Waste Collection System

Version 1.0 approved

Prepared by GardeniaProMax

Universiti Teknologi Malaysia

12/6/2025

TABLE OF CONTENTS

1 Introduction	3
1.1 Document overview	3
1.2 Abbreviations and Glossary	3
1.2.1 Abbreviations	3
1.2.2 Glossary	3
1.3 References	3
1.3.1 Project References	3
1.3.2 Standard and regulatory References	4
1.4 Conventions	4
2 Tests preparations	5
2.1 Choose sub-section name	5
2.1.1 Hardware preparation	5
2.1.2 Software preparation	5
2.1.3 Other test preparation	5
2.1.4 Safety, security and privacy precautions	5
3 Tests descriptions	6
3.1 Choose sub-section name	6

● Introduction

●.1 Document overview

This Software-Test Description (STD) defines all tests to be executed for Kutip Smart Waste Collection System version 1.0, as identified in the master Software-Test Plan (STP-KUTIP-01). Tests are organised into four logical groups—Integration, Verification, Performance, and Security/Safety—and cover both web-admin and driver mobile flows, the IoT edge stack, AI image recognition, and Mapbox-based scheduling.

●.2 Abbreviations and Glossary

●.2.1 Abbreviations

Abbr.	Meaning
AI	Artificial Intelligence
API	Application Programming Interface
CRUD	Create-Read-Update-Delete
KPI	Key Performance Indicator
RLS	Row-Level Security (Supabase)
RBAC	Role-Based Access Control
SUT	System Under Test
YOLOv8	You-Only-Look-Once v8 object-detection model

●.2.2 Glossary

Bin – Physical smart container instrumented with ultrasonic fill sensor.

Driver App – Mobile interface used by waste-collection truck drivers.

Pickup Event – Logical record representing an attempt to service a bin.

●.3 References

●.3.1 Project References

#	Document Identifier	Document Title
[R1]	[SRS]	Software Requirements Specification for Kutip v1.0 (20 May 2025)
[R2]	[SDD]	Software Design Document for Kutip v1.0 (20 May 2025)
[R3]	[STP]	Software Test Plan for Kutip v1.0 (to be issued)
[R4]	[MAPBOX]	Mapbox Optimization API v1 Docs
[R5]	[YOLO]	Ultralytics YOLOv8 Tech Report

●.3.2 Standard and regulatory References

#	Document Identifier	Title / Source
[STD-01]	ISO/IEC 29119-1 ... -5:2023	<i>Software and Systems Engineering — Software Testing (Parts 1–5)</i>
[STD-02]	ISO/IEC 25010:2011	<i>Systems and Software Quality Requirements and Evaluation (SQuaRE) — System and software quality models</i>
[STD-03]	ISO/IEC 27001:2022	<i>Information Security, Cybersecurity and Privacy Protection — Information security management systems — Requirements</i>
[STD-04]	OWASP ASVS v4.0	<i>Application Security Verification Standard</i>
[STD-05]	ETSI EN 303 645:2020	<i>Cyber Security for Consumer Internet of Things</i>
[STD-06]	RFC 9110 (HTTP Semantics)	IETF Standard
[REG-MY-01]	Personal Data Protection Act 2010 (PDPA Malaysia)	Malaysian Government Gazette
[REG-MY-02]	Occupational Safety & Health Act 1994 (OSHA Malaysia)	Dept. of Occupational Safety and Health
[LIC-01]	Mapbox Terms of Service (May 2024)	Mapbox

[LIC-02]	Ultralytics YOLOv8 AGPL-3.0 Licence	Ultralytics
----------	-------------------------------------	-------------

All testing activities shall ensure compliance with the standards and regulations listed in Table 1.3-2. Any deviation will be documented and approved by the QA lead and the client (Troom Tech Sdn Bhd).

- **Tests preparations**

This section contains tasks and recommendations before executing tests.

Describe the tests preparation tasks, for a phase, a category of tests, or even a single test (in this case, use the test identifier).

Give a name to sub-sections, which matches the scope of described tasks.

- .1 Choose sub-section name**

The sub-section name may be:

- Testing phase xxx
- or category of test xxx
- or test xxx if a single test deserves special preparation
- or some other logic to group tests

The tests impacted by the preparation shall be listed, add a list or add a reference to a list of tests made in the software test plan or a list of test in §3 test description.

- .1.1 Hardware preparation**

The following hardware components and configurations are required to execute system integration and functionality tests for the Kutip Smart Waste Collection System. These preparations ensure all physical devices used during testing are operational, properly connected, and simulating real-world conditions:

Hardware Component	Description / Setup Requirement
Laptop/PC for Admin Dashboard	A laptop or desktop with modern web browser (Google Chrome recommended), minimum 8GB RAM, for running the Next.js admin dashboard and accessing Supabase backend.
Laptop/PC for IoT Simulator and Image Detection Server	Dedicated device running the <code>camera_server.py</code> with access to laptop camera simulating bin plate detection. Requires working built-in webcam or external USB camera.
Laptop Camera (Truck Camera Simulation)	The built-in laptop webcam is configured to simulate the bin plate detection system mounted on waste collection trucks. Ensure clear video capture quality and stable positioning during testing.
Network Router/Switch (if applicable)	Stable internet connectivity for connecting devices, accessing Supabase, Google Drive, and Mapbox APIs. Local network must support real-time testing without interruptions.
Peripheral Devices	Mouse, keyboard, and external monitor (optional) for ease

	of test execution and monitoring.
Test Environment Isolation	Ensure the testing devices are connected to a dedicated, controlled network to avoid interference with production environments or external unauthorized systems.

Hardware Setup Steps:

1. Verify laptop/PC meets minimum specifications and is operational.
2. Ensure camera drivers are installed, and the webcam provides clear video feed.
3. Position laptop camera appropriately to simulate bin plate visibility as per real-world truck conditions.
4. Confirm stable internet access to Supabase, Mapbox services, and Google Drive APIs.
5. Ensure all devices are on the same network for seamless integration between simulator, dashboard, and backend.
6. Conduct hardware functionality check (e.g., test video stream at `http://localhost:8000/stream`) before proceeding with software setup.

Notes:

- For camera testing, use printouts or physical mock-ups of known bin plates to verify OCR and detection accuracy.
- Avoid running resource-heavy background processes on testing laptops to maintain system responsiveness.
- Maintain safe workspace conditions with cables secured to avoid disconnection during testing.

●.1.1 Software preparation

Objective:

Install, configure, and validate all software components needed to conduct the system integration and functional tests for the Kutip Smart Waste Collection System.

Backend System Preparation (Supabase)

Steps:

1. Supabase Project Initialization

- Create or select the dedicated Supabase project instance for testing.
- Verify project region, storage quotas, and API rate limits.

2. Database Schema Setup

- Deploy schema for tables:
 - bins (metadata, status, geolocation)
 - trucks (fleet details, driver assignments)
 - schedules (planned collections)
 - iot_events (sensor data, photos)
- Apply any necessary migrations.

3. Seed Test Data

- Populate the database with representative entries:
 - ~20 test bins across various simulated locations
 - ~3 trucks with dummy driver records
 - ~5 scheduled routes
- Use SQL scripts or Supabase CLI to automate seeding.

4. Storage Configuration

- Create buckets for storing bin photos and IoT sensor images.
- Define public/private access policies and CORS rules.

5. API Keys and Environment Variables

- Generate and secure:
 - SUPABASE_ANON_KEY
 - SUPABASE_SERVICE_ROLE_KEY
- Store in .env files used by the dashboard and simulator scripts.

6. Realtime Subscriptions

- Enable Realtime updates on `iot_events` and `schedules` tables.
- Validate that changes propagate to the frontend in real time.

Frontend (Next.js Dashboard) Preparation

Steps:

1. Code Retrieval and Installation

- Clone the Next.js repository.
- Install Node.js (≥ 18).
- Run `npm install` to install dependencies.

2. Environment Configuration

- Create `.env.local` with:

`NEXT_PUBLIC_SUPABASE_URL=`

`NEXT_PUBLIC_SUPABASE_ANON_KEY=`
- Confirm correct API base URLs.

3. Build and Start

- Build production bundle:

npm run build

npm start
- Verify app loads test data and connects to Supabase.

4. **CORS and API Proxying**

- Configure next.config.js if necessary for proxying API calls.
- Validate all routes are accessible.

5. **Browser Testing**

- Confirm the dashboard displays:
 - Bin status
 - Truck assignments
 - Schedule overviews
 - Realtime updates

IoT Sensor Simulator Preparation

Steps:

1. **Simulator Setup**

- Use a Node.js script or Postman Runner to simulate:
 - Bin status updates
 - Location tracking
 - Photo uploads

2. **Configuration**

- Define frequency of data emission:

- Bin status: every 30 seconds
- Location: every 10 seconds
- Photos: every 2 minutes

3. Dry Run

- Validate data ingestion into Supabase tables.
- Confirm test images upload to storage.

API Validation Tools Preparation

Steps:

1. Tool Configuration

- Prepare Postman/Insomnia collections covering:
 - CRUD APIs for bins and trucks
 - Scheduling endpoints
 - IoT event ingestion

2. Test Execution

- Run collections to verify:
 - 200 OK responses
 - JSON schema validation
 - Proper authorization handling

Network and Security Configuration

Steps:

1. Connectivity Validation

- Confirm network paths between:
 - IoT simulator
 - Supabase
 - Dashboard
- Verify ports and DNS resolution.

2. SSL/TLS Checks

- Ensure HTTPS is enforced end-to-end.

3. Firewall Rules

- Restrict access to authorized tester IP ranges.

Logging and Monitoring Setup

Steps:

1. Enable Supabase Logs

- Activate logs for:
 - Database queries
 - Storage uploads
 - Realtime events

2. Frontend Debugging

- Use Next.js development mode (`npm run dev`) when troubleshooting.

3. Notifications

- Configure optional alerts or logs aggregation (e.g., Slack notifications).

●.1.2 Other test preparation

Describe here any other specific task to do before tests.

●.1.3 Safety, security and privacy precautions

During the testing phase of the Kutip Smart Waste Collection System, it is paramount to adhere to strict safety, security, and privacy protocols to protect personnel, data, and system integrity.

1. Safety Precautions:

- **Test Environment Integrity:** Ensure the physical and virtual test environments are stable and configured correctly to prevent system crashes, unexpected behaviors, or data corruption during testing.
- **Equipment Handling:** Be mindful of the testing equipment (e.g., laptops, monitors, network cables). Ensure proper handling to prevent damage and maintain a tidy, organized workspace to avoid accidental disconnections or hazards.
- **Tester Well-being:** Ensure testers maintain ergonomic postures and take regular breaks during prolonged testing sessions to prevent eye strain or physical discomfort.

2. Security Precautions:

- **Controlled Access:** Access to test environments and sensitive system components must be strictly controlled. Only authorized personnel should have access to the test server, databases, and application code.
- **Secure Credentials:** Use strong, unique passwords for all test accounts. Avoid hardcoding credentials in test scripts. After testing, ensure all temporary test credentials are revoked or reset.
- **Vulnerability Scanning:** Where appropriate, conduct basic security scans on the test environment to identify common vulnerabilities before widespread testing begins.

- **Data Integrity:** Ensure test data is robust but does not compromise system integrity. Avoid introducing malicious data that could impact the system's stability or security post-testing.
- **Network Security:** Ensure the testing network is secure and isolated from production environments to prevent any accidental data leakage or system compromise.

3. Privacy Precautions:

- **Anonymized/Mock Data:** Crucially, testing must primarily use anonymized or mock data, not real customer or personal identifiable information (PII). This is especially important for modules dealing with customer details, bin locations (even if virtual), and driver personal information.
- **Data Minimization:** Only use the minimum amount of data required for testing purposes. Avoid replicating entire production databases for testing if not absolutely necessary.
- **Access Control for Test Data:** Limit access to sensitive test data sets (even if anonymized) to only those testers who require it for their specific tasks.
- **Compliance Awareness:** Be aware of relevant data protection regulations and ensure testing practices do not inadvertently violate these regulations, especially when dealing with location data or user profiles.
- **Secure Data Disposal:** Establish procedures for the secure disposal or sanitization of test data once testing is complete, particularly if any real (even if anonymized) or sensitive data was used.
- **Location Data Sensitivity:** Recognize that real-time location tracking of bins and trucks, while crucial for the system, also carries privacy implications. Ensure testing of this feature uses mock locations or is conducted in a way that does not expose actual private locations without explicit consent.

- **Tests descriptions**

This section outlines the detailed descriptions for each test case, organized by the system's main functional areas as per the User Acceptance Test Plan. Each test includes its ID, a description, verified requirements, initial conditions, inputs, data collection methods, expected outputs, assumptions, and the step-by-step procedure.

- .1 **General Functionality Testing**

Test ID	Same ID as in test plan	
Test description	Same as in test plan	
Verified Requirement	SRS-REQ-001	Verification method: I,A,D,T
Initial conditions	The state of software before test	You may reference a procedure or it may be the result of previous test
Tests inputs	Input data from any test tool, input files name and location	You may reference a procedure to use the test tool
Data collection actions	Recording and post processing of output data	You may reference a procedure to record data with a test tool
Tests outputs	Output data files names and location, logs ...	Give unique name out output data files.
Assumptions and constraints	If any, may be limited access to a tool, license ...	
Expected results and criteria	List here the results of test	And the criteria to evaluate the result
Test procedure		
Step number	Operator actions	Expected result and evaluation criteria
1	Start foo	Foo is started

Test ID	T-SRS-REQ-001	
Test desc.	Verify that the user manual contains the intended use	
Verif. Req.	SRS-REQ-001,	Inspection
Init. Cond.	XXX Software is started and idle	
Tests inputs	N/A	
Data collection	N/A	
Tests outputs	N/A	
Assum & constr	N/A	
Expected results and criteria	The user manual contains the intended use	Same IU as the one found in risk analysis report ref. xxx
Test procedure		
Step number	Operator actions	Expected result and eval crit
1	Open user manual, doc ref xxx, section: Introduction	The IU is located in the introduction, the text is the same as the one found in risk analysis report ref. xxx
2	Open Help/User manual	The online user manual is opened
3	Go to Section 1	The section 1 is displayed, it contains the IU as the one found in risk analysis report ref. xxx

●.2 Admin Page Functionality Testing

3.2.1 Dashboard Functionality

Test ID	T-SRS-REQ-001	
Test desc.	Verify that the xxx software allows to choose on protocol in the list of recorded protocols	
Verif. Req.	SRS-REQ-001,	Demonstration
Init. Cond.	XXX Software is started and idle	
Tests inputs	N/A	
Data collection	N/A	
Tests outputs	N/A	
Assum & constr	N/A	
Expected results and criteria	Chosen protocol is selected	See last step for criteria
Test procedure		
Step number	Operator actions	Expected result and eval crit
1	Open the list of protocols	The list of protocol is displayed
2	Select a protocol	The protocol summary is displayed
3	Validate the choice	The protocol is selected. The name of the protocol is displayed at the top of the screen.

3.2.2 Bins Management Functionality

Test ID	A-004	
Test desc.	Bins - View List	
Verif. Req.	System displays a comprehensive list of all registered bins.	
Init. Cond.	Logged in as Admin.	
Tests inputs	N/A	
Data collection	Record Pass/Fail status and any comments/improvements.	
Tests outputs	Table/list of bins.	
Assum & constr	Pre-existing bins are registered in the system.	
Expected results and criteria	A list of all registered bins is displayed.	
Test procedure		
Step number	Operator actions	Expected result and eval crit
1	Click on "Bins" under Operational Tables in the navigation menu.	The Bins page loads, displaying a list of all registered bins.

Test ID	A-005	
Test desc.	Bins - Add Customer	
Verif. Req.	System allows administrators to add new customers.	
Init. Cond.	Logged in as Admin, on the Bins page.	
Tests inputs	New customer details (e.g., name, address).	
Data collection	Record Pass/Fail status and any comments/improvements.	

Tests outputs	Confirmation message for successful customer addition.	
Assum & constr	Valid customer input is provided.	
Expected results and criteria	New customer is successfully added to the system.	
Test procedure		
Step number	Operator actions	Expected result and eval crit
1	On Bins page, locate "Add Customer" option.	"Add Customer" button/link is visible.
2	Click and follow prompts to add a new customer.	Customer details form appears, successful submission confirms addition.

Test ID	A-006	
Test desc.	Bins - Add Bin by Customer	
Verif. Req.	System allows administrators to add new bins and associate them with existing customers.	
Init. Cond.	Logged in as Admin, on the Bins page.	
Tests inputs	Bin location on map, selection of existing customer, required bin details (e.g., bin plate, label).	
Data collection	Record Pass/Fail status and any comments/improvements.	
Tests outputs	Confirmation message for successful bin addition.	
Assum & constr	An existing customer is available for selection. Map services are functional.	
Expected results and criteria	New bin is successfully added and associated with the selected customer.	
Test procedure		
Step number	Operator actions	Expected result and eval crit
1	On Bins page, locate "Add Bin" option.	"Add Bin" button is visible.
2	Select bin location on the map.	Map allows selection of a point for bin placement.
3	Choose an existing customer.	Customer dropdown/selector is populated with existing customers.
4	Input required bin details.	Form fields for bin details are present and allow input.
5	Click "Create Bin".	New bin is successfully added and associated with the selected customer.

Test ID	A-007	
Test desc.	Bins - View/Edit Bin Details	
Verif. Req.	System allows viewing and editing of existing bin details.	
Init. Cond.	Logged in as Admin, on the Bins list page.	
Tests inputs	Selection of a specific bin, modified editable details.	
Data collection	Record Pass/Fail status and any comments/improvements.	

Tests outputs	Bin details displayed, updated information saved successfully.	
Assum & constr	At least one bin exists in the list.	
Expected results and criteria	All details for the selected bin are viewable, and editable details can be successfully updated.	
Test procedure		
Step number	Operator actions	Expected result and eval crit
1	On Bins list, select a specific bin.	Bin is selected.
2	Click "Details" button to view its details.	Bin details page opens, displaying all information.
3	Attempt to click "Edit" button to edit some details and save changes.	Editable fields allow modification, and changes are saved upon submission.

3.2.3 Trucks Management Functionality

Test ID	A-008	
Test desc.	Trucks View List	
Verif. Req.	System displays a comprehensive list of all registered trucks.	
Init. Cond.	Logged in as Admin.	
Tests inputs	N/A	
Data collection	Record Pass/Fail status and any comments/improvements.	
Tests outputs	Table/list of trucks.	
Assum & constr	Pre-existing trucks are registered in the system.	
Expected results and criteria	A list of all registered trucks details is displayed.	
Test procedure		
Step number	Operator actions	Expected result and eval crit
1	Click on "Trucks" under Operational Tables in the navigation menu.	The Trucks page loads, displaying a list of all registered trucks.

Test ID	A-009	
Test desc.	Trucks - Create New Truck	
Verif. Req.	System allows administrators to add new trucks.	
Init. Cond.	Logged in as Admin, on the Trucks page.	
Tests inputs	Required truck details (e.g., plate no., status) and selection of existing driver.	
Data collection	Record Pass/Fail status and any comments/improvements.	
Tests outputs	Confirmation message for successful truck addition.	
Assum & constr	Valid truck input is provided.	
Expected results and criteria	New truck is successfully added to the system.	
Test procedure		

Step number	Operator actions	Expected result and eval crit
1	On Trucks page, locate "Create New Truck" option.	"Create New Truck" button is visible.
2	Input required truck details.	Form fields for truck details are present and allow input.
3	Click "Create Truck".	New truck is successfully added to the system.

Test ID	A-010	
Test desc.	Trucks View/Edit Truck Details	
Verif. Req.	System allows viewing and editing of existing truck details.	
Init. Cond.	Logged in as Admin, on the Trucks list page.	
Tests inputs	Selection of a specific truck, modified editable details.	
Data collection	Record Pass/Fail status and any comments/improvements.	
Tests outputs	Truck details displayed, updated information saved successfully.	
Assum & constr	At least one truck exists in the list.	
Expected results and criteria	All details for the selected truck are viewable, and editable details can be successfully updated.	
Test procedure		
Step number	Operator actions	Expected result and eval crit
1	On Trucks list, select a specific truck.	Truck is highlighted or selected.
2	Click "Details" button to view its details.	Truck details page opens, displaying all information.
3	Attempt to edit some details and save changes.	Editable fields allow modification, and changes are saved upon submission.

3.2.4 Driver Page Functionality

Test ID	D-003	
Test desc.	Camera Page - Scan Bin	
Verif. Req.	System allows the driver to scan bins using the camera module.	
Init. Cond.	Logged in as Driver. Camera server running.	
Tests inputs	Physical bin plate (In "BinPlate_Image_Test" folder", with upside down position).	
Data collection	Record Pass/Fail status, detected plate number, confidence score.	
Tests outputs	Detected plate number and confidence displayed in Driver App.	
Assum & constr	Camera server and YOLOv8 model operational, bin plate visible and clear.	
Expected results and criteria	Bin plate is successfully detected and displayed with confidence percentage.	
Test procedure		
Step number	Operator actions	Expected result and eval crit

1	Click "Open Camera" on Driver interface.	Camera stream activates, live video visible.
2	Present bin plate to camera.	System detects bin plate, displays number and confidence.

Test ID	D-004	
Test desc.	Camera Page - Auto Update Status	
Verif. Req.	System automatically updates bin status after successful scan.	
Init. Cond.	Logged in as Driver, schedule with assigned bins active.	
Tests inputs	Bin plate from existing system record.	
Data collection	System logs, confirmation message on Driver interface.	
Tests outputs	Bin status updated (e.g., "Collected"), confirmation displayed.	
Assum & constr	Bin plate matches system record, backend and Supabase operational.	
Expected results and criteria	Bin status updated in system, driver receives confirmation.	
Test procedure		
Step number	Operator actions	Expected result and eval crit
1	Scan bin plate using Camera page.	Bin plate detected, system validates match.
2	Observe status update.	Bin status updated, confirmation shown on Driver interface.