ZOPAC Test Plan & Report

Zach Kaiser, Colin Miller, Preston Beachum, Alex Grommesh, Othman Al Taie

11/25/2025

1 Testing Objectives

Client:

- The overall test objective for the client is to verify that the data is displayed properly and correctly. Here are some things we are going to do to validate all of our test cases:
    - Ensure login functionality is working and communicating with the database properly. Storing user credentials gets sent properly through the API to the database, previous users are able to be fetched from the database and granted access to the system, and functionality for logging out.
    - Ensure all components on the heatmap are displayed properly.
        - Handling display before and after user uploads an image
        - Verifying heatmap “blobs” and making sure the density pattern is displayed accurately
        - Handling clicks on different components of the heatmap page to make sure the modals display properly and close properly

Endpoint

The objective of the endpoint system tests is to verify that the Wi-Fi scanning pipeline is fully functional and error-free. These tests will evaluate if the endpoints can detect nearby devices, capture Wi-Fi probe data, log the information locally, and transmit the data to the server. Additionally, the system must notify the server when the endpoint comes online. There are three goals for the test:

1. **Validate Endpoint boot behavior**

    a. Ensure that the endpoint starts all the required services automatically on boot and immediately sends an online status to the server.

2. **Verify the Wi-Fi scan data pipeline**

a. Confirm that each endpoint detects nearby probe requests and generates scan entries containing the MAC address, RSSI, Timestamp

b. These values must be parsed, logged into a local JSONL file, and transmitted to the server

3. **Confirm local logging of transmitted scans**

a. Ensure that all transmitted records are also written into a log file on the endpoint to help with debugging.

Server:

The server API testing objective is to validate that all endpoints handle requests correctly, communicate with the database properly, and return expected responses. Specific goals include:

1. Data Transmission Validation

• Verify the server accepts WiFi scan data from endpoints

• Confirm device location queries return accurate positioning data

• Validate floorplan upload and retrieval functionality

2. Authentication & Authorization

• Verify API key authentication for endpoint requests

• Confirm user authentication APIs handle signup and login correctly

3. Database Integration

• Validate all API endpoints connect to PostgreSQL successfully

• Confirm data persistence and retrieval operations work correctly

2 Test Procedures

Testing Tools:

- Client:
    - For the client, we chose to use Playwright as we had experience using it in class and know how effective it is. To test the heatmap component, ensure you are in the Heatmap directory (client/src/components/heatmap) and run "**npm run test"** in the terminal. This will run the dedicated Playwright test and generate a new window with the test results.
    - To install playwright, make sure you are in the client directory and run "**npm install"**. This will install all of the dependencies needed for the client, Playwright being one of them. Next, run **"npx playwright install"**. This allows playwright to access browsers and test amongst chromium, firefox, and webkit.

Endpoint Test Procedures

Before testing, ensure each endpoint has required dependencies installed:

- sudo apt update
- sudo apt install -y tcpdump
- sudo apt install -y python3 python3-pip

Additionally:

- The endpoint must have the full WiFi_Project repository pulled
- setup_monitor.sh, capture_wifi.sh, and parser_scan.py must be executable
- Systemd services must be installed and enabled

Test Case: Endpoint Sends Status Notification on Boot

Preparation

1. Ensure the endpoint systemd service responsible for status reporting is installed:
    - sudo systemctl status wifi-status.service
2. Ensure the server is running.

Steps

1. Reboot the endpoint:
    - sudo reboot
2. After reconnecting, verify the service status:
    - systemctl status wifi-status.service
3. On the server, check for the latest status

Expected Result

- Server database receives a new online status entry
- Timestamp matches the reboot time
- HTTP 200 logged on server

Test Case: Endpoint Begins Wi-Fi Scanning on Boot

Preparation:

1. Ensure wifi-capture.service is enabled:
    a. sudo systemctl is-enabled wifi-capture.service
2. Ensure the server is running

Steps:

1. Reboot the Pi:
    a. sudo reboot
2. After reconnecting, verify the service status:
    a. systemctl status wifi-capture.service
3. Confirm log file has been created:
    a. ls ~/WiFi_Project/logs
4. Confirm the server is receiving the scanned data- seen in the 201 messages.

Expected Results:

- Log files are created without manual intervention
- Parsed jsonl contains valid entries
- · Server is receiving the parsed scan data

Test Case: Scan Records Contain MAC, RSSI, Timestamp

Steps

1. Inspect recent parsed JSONL file:
    a. tail -n 3 ~/WiFi_Project/logs/parsed_wlan1_*.jsonl
2. Validate the presence of the required fields

Expected Results

- All scans contain the required fields

SERVER

Server/API:

For server API testing, we use curl-based bash scripts to validate all API endpoints. The test suite is located in tests/api/ directory.

To run the API tests:

1. Ensure the server is running (either locally at http://localhost:3000 or deployed at https://seng401-project-zopac.fly.dev)

2. Set environment variables:

```
export API_BASE_URL="https://seng401-project-zopac.fly.dev"

export ENDPOINT_API_KEY="your_api_key_here"
```

3. Navigate to the tests directory:

```
cd tests/api
```

4. Run the test suite:

```
./run-api-tests.sh
```

The script will output color-coded results for each test case and provide a summary at the end.

3 Traceability Matrix

SERVER

Our traceability matrix is maintained in a Google Sheet accessible here:

The matrix includes:
- Test Case IDs mapped to User Story numbers
- Test descriptions and expected results
- Pass/fail status for each test execution
- Date of last test run
- Notes on any failures or bugs discovered

Key test cases include:
- TC-API-001: Endpoint scan data submission (maps to WiFi scanning user stories)
- TC-API-002: Device location query (maps to localization user stories)
- TC-API-003: Endpoint status monitoring (maps to endpoint health user stories)
- TC-API-004: Floorplan upload/retrieval (maps to UI floorplan user stories)
- TC-API-005: User authentication (maps to login/signup user stories)
- TC-CLIENT-001 through TC-CLIENT-007: Heatmap UI component tests (maps to visualization user stories)

TRACABILITY MATRIX SPREADSHEET

https://docs.google.com/spreadsheets/d/1HzlqYJWfODvPNhpvpbX0OfOhOV2Mi8hgEOcHb4Axicw/edit?usp=sharing

4 Accuracy Analysis

SERVER

Test Setup:
- Floorplan: [Describe the test location - e.g., "Cameron Hall Computer Lab, 50ft x 30ft room"]

- Device positions: [Specify exact positions - e.g., "Device 1: Northwest corner (5ft, 25ft), Device 2: Center (25ft, 15ft)"]
- Endpoint positions: [Specify where you placed each Pi - e.g., "EP1: (0, 0), EP2: (50, 0), EP3: (0, 30), EP4: (50, 30)"]
- Minimum samples: 100 probe requests per device over a 10-minute period
- Ground truth: Physical measurements using tape measure from corner reference point

Test Procedure:
1. Upload floorplan image to system via UI
2. Configure endpoint positions in UI to match physical placement
3. Place test devices at known positions
4. Run WiFi scanning for 10 minutes
5. Compare heatmap output positions to known physical positions
6. Calculate position error (distance between predicted and actual position)

Endpoint
To evaluate the accuracy, we will measure how closely the generated heatmap matches the physical location of the mobile devices.
Required Elements:
- Floorplan
    - Upload the floorplan of the room being used
- Endpoint locations
    - X/Y positions of the endpoints
- Device Placement
    - Place 3-6 phones in fixed positions, and measure the positions to scale with the floorplan

Client

In order to fully test the functionality of the client, we are planning on using multiple heatmaps in order to ensure that the endpoints can be configured properly. This will ultimately handle all test cases, adjusting the location of endpoints, and handling less than 4 endpoints.

Procedure:
1. Upload a different map
2. Drag endpoints to different locations throughout the heatmap
3. Validate accuracy of the heatmap information

5 Analysis of Results

SERVER

As of November 24, 2025:

API/Server Tests (5 tests):
- TC-API-001: PASS ✓
- TC-API-002: FAIL ✗ (column name mismatch: ep.floor vs floor_number)
- TC-API-003: PASS ✓
- TC-API-004: PASS ✓
- TC-API-005: PASS ✓

Pass Rate: 4/5 (80%)

Bugs & Issues:
- Database schema column mismatch: queries use `ep.floor` but database has `floor_number` (IDENTIFIED - fix in progress)
- Need to deploy fix and retest TC-API-002

Features Not Tested:
- System performance under high device density (>50 devices)
- Long-term stability (24+ hour continuous operation)
- Network interruption recovery
- Concurrent user access to UI

Endpoint

```
PS C:\seng 401\seng401-project-zopac> pytest

==========================================================================
test session starts
==========================================================================

platform win32 -- Python 3.12.7, pytest-8.3.3, pluggy-1.5.0

rootdir: C:\seng 401\seng401-project-zopac

plugins: cov-5.0.0

collected 52 items



endpoint\tests\test_aggregator.py .....
[  9%]

endpoint\tests\test_config.py ....................
[ 48%]

endpoint\tests\test_parser_scan.py ........
[ 63%]
```

```
endpoint\tests\test_send_status.py ........
[ 78%]

endpoint\tests\test_shipper.py .......
[ 92%]

endpoint\tests\test_stream.py ....
[100%]

============================================================================
52 passed in 1.28s
============================================================================

PS C:\seng 401\seng401-project-zopac>
```

As of November 24th, all endpoint python files have been tested and passed with full marks.

CLIENT

All playwright tests pass on the heatmap and the 56/60 pass on the login page. We found some small issues while testing that were easy to fix. The React Bootstrap Modal components were having an issue communicating with playwright, but after some extensive testing, we were able to resolve the issue.

We still have to work out some small issues with the Webkit browser, which is the only one not passing all tests.

6 Conclusions

SERVER

Release Readiness:
Based on our testing results, the system is [ready/not ready] for Demo Day release. All core API endpoints are functional and passing automated tests. The deployment to Fly.io is stable with no database connection errors.

Required Actions Before Demo Day (December 4):
1. Complete accuracy analysis with physical device testing - Assigned to: Alex
2. Run and document all Playwright client tests - Assigned to: Zach, Colin
3. Verify endpoint boot behavior on all Raspberry Pis - Assigned to: Alex
4. Create testing GitHub issues for each user story - Assigned to: Preston

5. Complete traceability matrix with all test results - Assigned to: All
6. Fix any remaining bugs discovered during testing - Assigned to: Preston/Othman
7. Prepare demo script and rehearse presentation - Assigned to: All

Action Assignments:
- Preston: 4,5,6,7
- Zach: 2, 5, 7
- Colin: 2, 5, 7
- Alex: 1, 3, 5, 7
- Othman: 5,6,7

Endpoint
Required Actions Before Demo Day

- Fix remaining bugs identified in testing
- Validate endpoint accuracy with real-world data
- Ensure all endpoints start correctly via systemd
- Perform a full end-to-end test with all team members present
- Refresh database and redeploy containers

Client

- We are very close to being ready, but not yet. We have everything containerized and deployed, but are having small testing issues with some niche heatmap tests. We are also getting ready to fully test real-time data.
- We are going to validate all playwright tests and ensure all of the data communication is working and displayed on the heatmap properly.
- Zach will be responsible for ensuring the playwright testing and Colin will be responsible for making sure the heatmap is displaying properly with the data.

Server

- Real close to wrapping it up. I need to see everything come together first. But the database seems to be accepting the Pi's info well.
- Need to do a little more testing on using all 4 pi's at the same time
- Preston will be responsible making sure all api's work and database tables are setup correctly
- The server is 80% ready - core functionality works but needs the floor column fix deployed before Demo Day.