Project Acceptance Tests

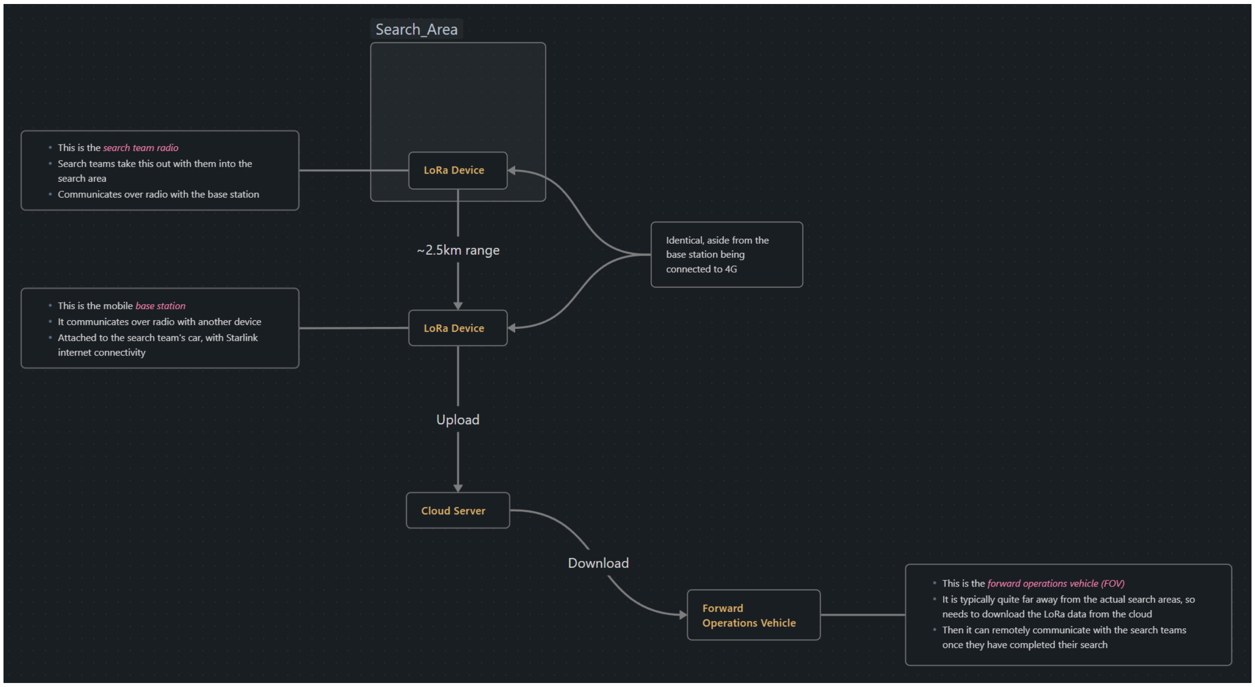
Objectives

This document outlines the testing strategy for the ‘LoRa Devices’ project which aims to provide a lightweight system of tracking search teams involved in incidents with large and remote search spaces. The testing strategy consists of:

- **Unit tests** - to ensure correct functionality of discrete features of the software such as GPS data capture/ transmission via LoRa and login/ authentication processes. To minimise sources of external error stubs are used where appropriate.

- **Integration tests** - to ensure data can be reliably accessed from cloud server and data can be reliably sent to cloud server from mobile base-station.

- **System tests** - to evaluate the entire system in a semi real-world scenario, ensuring reliable service from the start of a search operation through till the end.



- This is the *search team radio*
- Search teams take this out with them into the search area
- Communicates over radio with the base station

Identical, aside from the base station being connected to 4G

- This is the mobile *base station*
- It communicates over radio with another device
- Attached to the search team's car, with Starlink internet connectivity

- This is the *forward operations vehicle (FOV)*
- It is typically quite far away from the actual search areas, so needs to download the LoRa data from the cloud
- Then it can remotely communicate with the search teams once they have completed their search

Unit testing is to be carried out across all subsystems and are to be primarily completed by members who developed that piece of the software. The results will be verified by other group members to eliminate any observer biases. Unit tests will be performed immediately after a feature is completed and before it is integrated into the broader system to minimise the risk of software bugs. Stubs will be used to control inputs to a subsystem when undergoing unit testing.

The entire team will be present during integration tests and system tests so that we can accurately simulate the testing environment. These practical tests will be conducted primarily at UWA campus with possible testing at the CSU facility, under the guidance of our client. For these tests, some team members will take on the role of the client operating from the Forward Operations Vehicle (FOV), while others will simulate the search units operating from the mobile base station. This setup will allow us to test the communication flow, ensuring that GPS data are correctly transmitted from the mobile base station, streamed to the cloud, and accessed by the FOV unit.

System tests will follow the same paradigm as integration tests, however additional team members will enact the roles of search units on a simulated search. This arrangement tests the flow of data from the `Search_Area` all the way down to the FOV across an entire search. System testing will likely be a lengthy testing process and will take place towards the end of development, after many of the subsystems it relies upon are already in place.

Tests were designed to cater to features that were deemed most important to our client, based on the “$100” project requirements test.

$100 Test:

- $40 - LoRa/Meshtastic GPS tracking/communication
- $40 - Base station-secure cloud server connection/data management
- $10 - Export the paths as a KMZ/GPX or similar file format
- $10 - Visualisation of map data via web client

Authentication Test (Unit test)

Ensures that only authenticated users can access information hosted on cloud services. Denies access to unknown users.

Test Specification

Test will validate successful login of users with correct credentials. Test will also measure server response to one/ many unsuccessful login attempts from unknown or incorrect credentials

Test Methods:

Credentials are measured against a local stub database which is loaded with test credentials. Login attempts with both valid and invalid credentials are made. Flooding of login attempts with invalid credentials.

Test Description

- Location of test: UWA CSSE Building.
- Equipment: Computer with relevant authentication script.
- Means of Control: Test cases of manual data entry and through driver (Selenium) to test flooding attempts.
- Test Data
    - Input credentials as strings
    - Progress through testing by interacting with frontend software.
    - Successful login yields the next page, whereas failed login returns the original login page, without authenticating the user.
    - System must log all data from login attempts and notify when login status for a session has changed.
- Procedures: Complete test cases consisting of single login attempts with both valid and invalid credentials, and concurrent failed login attempts.

Test Analysis Report

Function:

- Verifies that authentication procedure gives reliable results and function as expected
- Successful if the system allows access to registered users and prevents access to unknown, unauthorised users.

Performance:

- Evaluates server's ability to handle multiple login attempts at the same time.
- Successful if server responses are reliable and timely to all requests.

Data Measures:

- All test cases will have an expected result to be compared with the server's response.
- Successful if and only if all responses align with expected results.

Accuracy of Geolocation Data Test (Integration test)

Geolocation data, when viewed as a client through a web-app, should be accurate and timely. Coordinates should update in real-time and in the case of momentary loss of signal, and display the last known position of that node.

Test Specification

Verifies the reliability of routes traced by a team member with a predetermined path. Measures system response to momentary loss of connection, to simulate loss of line of sight.
with neighbours. When connection drops, the node's last known location should still display on the client's end. The client should be notified that the node is offline. When node regains connection, it should gracefully begin tracing the route with an updated position.

Test Description

- Location of test: UWA James Oval
- Equipment: Preconfigured Meshtastic devices, cloud server setup, and computer with internet connection.
- Means of Control: Data gathered from real gps measurements from known location and uploaded to server by the software.
- Data
  - Input Geolocation data from Meshtastic device.
  - Initial input command to commence search with pre configured hardware.
  - Output observed by client machine to see if route is being tracked accurately across test environment (across the width of James Oval)
  - System messages (timeout message, ack testing) to notify loss of connection from tracking device.
- Procedures: A team member with a pre configured Meshtastic device *J* traces a calculated path across James Oval. Another member with an identically configured device *B* requests and receives Geolocation data of *J* and transmits this data to the cloud server. A third member views this data from the web server to ensure that the actual path taken aligns with what is presented through the web interface.

Test Analysis Report

Function:

- Verify the seamless and accurate communication of GPS data between mobile base-station and the FOV.
- Successful if the displayed route mirrors the route that was actually taken by a team member with a test tracker.

Performance:

- Evaluates latency in render-time from when there is a marked change in geoposition, to when that change is reflected on the client side (FOV).
- Successful if latency is generally below 1 minute.

Data Measures:

- System latency in displaying updated GPS information.
- Accuracy of GPS data against known coordinates of actual path travelled.
- System response to crashed node.

End-to-End Reliable Service Test (System test)

Ensures system's ability to provide reliable service through the duration of a search operation. System should respond gracefully in the event nodes drop connection.

Test Specification:

Verifies the system's ability to maintain reliable service during a search operation by simulating connection drops and testing recovery processes.
System should display GPS information to authorised clients in a timely manner.
System should automatically recover in these cases and continue updating routes for remaining live nodes.

Test Description:

- Location of test: UWA James Oval and UWA CSSE Computer Lab [2.05]
- Equipment: Preconfigured Meshtastic devices, cloud server setup with pre programmed test credentials, and computer with internet connection.
- Means of Control: All logins entered manually to simulate initialisation of real search. All data entry/transmission should be handled by the system.
- Test Data
    - Initial input will be a set of test user credentials and a cloud server setup with a test database. As search gets carried out. GPS data is automatically generated and transmitted by Meshtastic devices over LoRa.
    - Search begins with hardware that is configured to communicate over a shared channel.
    - Output of all routes traced by the relevant search team will be rendered on the client's screen.

    - System Messages will also be displayed in the event of disconnection/ reconnection.
- Procedures: Team members with Meshtastic trackers will travel a pre-configured route. Another member will monitor the routes on the client machine. Connection drops will be simulated to mimic signal loss, and the observer will record how the system responds to disconnections and reconnections, focusing on maintaining accurate position data.

Test Analysis Report

Function:

- Ability to provide continuous and reliable service throughout a search operation, including handling and recovering from connection drops.
- Successful if the system responds gracefully to all disconnections throughout search and displays accurate system information to the client machine.

Performance:

- Accuracy of route tracing across multiple disconnections of different nodes when conducting a search.
- Successful if the system is able to serve reliable information across the entire search.

Data Measures:

- Client is consistently shown the last known position of disconnected nodes.
- Clients receive timely notifications about offline nodes, and data resumes updating without manual intervention from other online nodes.

Frontend Robustness Test (System test)

Ensures the frontend softwares robustness, namely its capacity to handle large amounts of data, unexpected inputs/edge cases, and long periods of sustained usage. This test specifically refers to non-authentication cases, namely GPS data inputs and connections from base stations.

Test Specification:

Verifies the frontend software's capability to handle a heavy load of input data over a sustained period of time.
The frontend software should handle the data accordingly and display it typically.
If necessary, under particularly heavy load, it should gracefully shutdown and provide a helpful error message.
Additionally, the frontend software should be able to handle unexpected inputs/edge cases of data entry from the cloud server, including malicious inputs such as SQL injection.

The frontend software should alert the user that an unexpected/malicious input was received and continue functioning without error.

Test Description:

- Location of test: n/a
- Equipment: Frontend software setup in development environment, preferably in testing mode able to take inputs from a fake cloud server.
- Means of control: Selenium testing automation.
- Test Data:
    - Selenium automated to connect ~20 base stations and begin transmitting GPS data from those base stations at a higher-than-expected rate.
    - Selenium automated to input various malicious/unexpected data.
        - GPS data that is significantly far away from the search area, such as in a different country or off the map entirely.
        - Malicious data that could impact the running of the software.
        - (Possibly) Connection from an unknown/suspicious base station.
    - The output of both tests will be rendered in a txt file (or on the command line) for the developer to evaluate.
    - System messages will also be displayed in the event of the software gracefully exiting.
- Procedures: Developers will activate the Selenium tests in a controlled development environment and monitor the results.

Test Analysis Report

Function:

- Software has the capacity to handle large amounts of data input from multiple base stations over a sustained period of time.
- Software is able to understand unexpected/malicious input and provide helpful error messages to the user.
- Software is able to gracefully shutdown as necessary, such as when under too much load.

Performance:

- Simply, the ability to not crash when placed under heavy load, to continue operating as normal.
- Successful if not crashed, data is handled correctly OR in the event of a crash, a helpful error message is supplied.

Data Measures:

- Successful if GPS data from all connected devices is displayed correctly.
- Users receive helpful error messages in the event of a crash/disconnection.