Test Plan

1. **Introduction**

**The project being tested involves a mapping and routing system designed to simulate the addition of routes to a base map. This system includes functionalities for populating the map with buildings, defining routes, and adding these routes to the map for visualization.**

**The primary objective of this test plan is to verify the correctness and robustness of these functionalities, ensuring that the map and routes are accurately created, routes are properly added to the map, and the system behaves as expected under various scenarios.**

**The testing aims to identify any defects or issues and ensure the software meets the specified requirements.**

1. **Scope**
   1. What Will Be Tested
      1. The creation and population of the base map with buildings: This involves testing the populateMap function to ensure it correctly initializes a Map structure with predefined building locations. The test will verify that the map's dimensions and building placements match the expected values. Test Activity: Ensure the map structure is initialized without any errors or unexpected behaviors.
      2. The definition and retrieval of specific routes (blue, green, and yellow): This includes testing the functions getBlueRoute, getGreenRoute, and getYellowRoute to ensure they return routes with the correct sequence of points and the correct route symbols. Test Activity: Check that the sequence of points in each route matches the expected path.
      3. The shortest path calculation between two points on the map: This involves testing the shortestPath function to ensure it accurately calculates the shortest path between two points on the map, avoiding buildings and other obstacles. Test Activity: Check that the function returns an empty route when no path exists or when the start and destination points are the same, and ensure that the calculated path does not pass through buildings or other obstacles.
      4. Other functions created along the milestone: This includes testing any additional functions developed throughout the project milestone, ensuring they work as intended and integrate smoothly with the existing system. Test activities: Identify and list all additional functions developed during the milestone, write test cases to cover the functionality and execute the test cases and verify that the functions perform correctly under various conditions.
   2. What Will Not Be Tested
      1. Advanced routing algorithms beyond the provided implementations: This means that any advanced routing techniques or optimizations not included in the current implementation will not be tested. Test activity: Enhancements or modifications to the existing routing algorithms that are not part of the original codebase.
2. **Environment Requirements**
   1. Hardware Requirements:
      1. Test computer with at least 4GB of RAM.
      2. Processor: Apple (ARM-based) or Intel
   2. Software Requirements:
      1. Operating System: Windows, macOS or any Unix-based OS
      2. Compiler: GCC or Clang
      3. Testing Tools: Built-in C standard libraries and any additional tools required for automated testing.
   3. Test Harness:
      1. Microsoft CppUnitTest framework or other pre-existing set of testing tools such as CUnit or Unity for C.
      2. Custom test scripts to automate the execution of the test cases.
3. **Test Schedule**
   1. Milestone 2:
      1. Setup the test environment.
      2. Write initial test cases for basic functionalities (e.g., map population, route creation).
   2. Milestone 3:
      1. A set of Blackbox tests as test documents (in an Excel file).
      2. Test data for the functions created at MS3.
   3. Milestone 4:
      1. Finish implementing/coding the functions.
      2. Whitebox tests.
   4. Milestone 5:
      1. Integration tests document.
   5. Milestone 6:
      1. Final testing report listing tested conducted.