Proposed Mapping Services Test Cases

*Document drafted in test cases by function (see Psuedocode document in Box). Required paths accounted for.*

**recieveAndNotify:**

Happy Path

* orderFromUser created in backend server has all valid values
  + Approximate information based on destination address (On error: catch exception redirect)
  + Fetch appropriate vehicle for use (On error: intercept. Retry request)
  + Create dispatch record based on vehicle information (On error: catch exception. Redirect)
  + Desired action/output = send createRoute(dispatchRecord) back to supply side server (If return = error (null, exception, ETA = 0): catch exception. Redirect)

Alternative Path

* orderFromUser created in backend server contains falsified/invalid values (Ex. Destination Address = Antarctica)
  + Approximate information based on destination address (On error: catch exception. Redirect)
  + Fetch appropriate vehicle for use
    - On unfound (Service Type): throw exception “Service Unavailable at This Time”
    - On unfound (Invalid location): throw exception “Service Unavailable in Stated location

Exception Path

* orderFromUser contains null/error values that are invalid
  + Approximate position based on destination address
    - On error null/invalid address: throw exception “Destination address not determined”
    - On error null/invalid service: throw exception “Service type not provided”
    - TBD based on order parameters (On error/invalid: throw exception “Invalid”)

**createRoute:**

Happy Path

* dispatchRecord created by recieveAndNotify contains all valid values
  + Pull vehicle information and destination address from dispatchRecord (On error: catch exception. Redirect)
  + Set start and endpoint for route
  + Use start and end point to send HTTP request to Directions API (On error: catch exception, retry. On retry exhaust, redirect)

Alternative Path

* dispatchRecord contains falsified/invalid values
  + Pull vehicle information and destination address from dispatchRecord
    - On error (vehicle inconsistency): throw exception “Unable to determine Vehicle”
  + Set start and endpoint for route
  + Use start and end point to send HTTP request to Directions API (On error: catch exception, retry. On retry exhaust, redirect)

Exception Path

* dispatchRecord contains null/error values that are invalid
  + Pull vehicle information and destination address from dispatchRecord
    - On error null/invalid vehicle information: throw exception “Invalid Dispatch Record”
    - On error null/invalid destination address: throw exception “Invalid Dispatch Record”

**displayMap:**

Singular Pathway

* Instantiate map (HHTP request to Maps and Directions API) (On error: retry. On retry exhaust, throw exception “Map is unable to display at this time”)

**deploy**

Singular Pathway

* Route
  + displayMap(route) (On error: retry. On retry exhaust, throw exception “Route cannot be determined”)
  + Send HTTP request to directions API to dynamically travel route (On error: retry. On retry exhaust, throw exception “Route cannot be displayed at this time”)