Car App

**Test Plan**

*Purpose: [Insert description of doc here]*

**Team members: Michael Edwards**

**Date/Version: 07/06/2022**

# Overview

## Purpose

<The purpose of this document is to define:

* The context of the testing,
* The test methodology,
* The test environment
* The types of tests
* The standards that will apply
* The test deliverables
* The test scope, including functional and non-functional tests.>

## Context

<Describe the context of the application and its testing. What are the benefits of testing? Who are the stakeholders and what are their roles in the testing?>

<INSERT CLASS DIRAGRAM HERE>

## Methodology

<This section should describe the methodologies being used in this test plan. What methodologies are we using and why? What types of testing are they? (Automated?) what are the benefits of using these methodologies? \*This is the most important section that you elaborate on\*>

## Test environment

<What tools and frameworks will be used (Unit Testing framework?) >

# Testing Plan

## Scope of Testing

### Functional tests

<Insert a table here for your functional (manual) tests. Consider data inputs, steps to take and expected and actual outcomes. Fill this out.>

* Test Case: Add a car to the inventory
  + Testing Steps
    - Fill out all textboxes under **Add/Inspect/Remove Cars**
      * Fill ‘Rego’ textbox with an 8 letter string that is unique from all currently existing car regos in the inventory. (Use ‘35671JRT’)
      * Select a make from the make dropdown menu. (Select ‘White’)
      * Select a model from the model dropdown menu. (Select ‘Truck’)
      * Fill ‘Year’ textbox with a number greater than 1900. (Use ‘2005‘)
      * Fill ‘Budget’ textbox with a non-negative number. (Use ‘7000’)
      * Fill ‘Car Position’ with a number corresponding to an empty space according to ‘Empty spots’ listed above, or leave blank. (Use ‘20’)
    - Click ‘Add Car’ button to attempt to add car to inventory.
  + Expected Outcome
    - A Messagebox should appear confirming the success of adding car to inventory.
    - The car inventory Listbox should update with the newly added car.
    - The ‘Empty spots’ should update to show that spot 20 is gone.
* Test Case: Search car inventory
  + Testing Steps
    - Fill out one set of textboxes under **Search Cars** with input then attempt to search. After each search select the ‘Reset’ button to clear all search textboxes.
      * Fill ‘Rego’ textbox with a string corresponding to an existing rego in the car inventory, then search. (Use ‘90045QER’)
      * Select a make from the dropdown menu then attempt a search.
      * Select a model from the dropdown menu then attempt a search.
      * Fill ‘Year’ textboxes with a min and/or max value, then attempt a search.
      * Fill ‘Budget’ textboxes with a min and/or max value, then attempt a search.
      * Fill ‘Position’ textbox with a value corresponding to a car position, then attempt a search.
  + Expected Outcome
    - Each search attempt should fill out the Listbox with entries that match the given criteria. If there are no matches, the Listbox will be empty. Searching with an empty textbox/textboxes/dropdowns set to none will repopulate Listbox.
* Test Case: Remove car from inventory
  + Testing Steps

## Debugging Screenshot

<Take a screenshot of you debugging your application using a breakpoint (The red circle in visual studio that stops the application so you can check values at run time).>

