**Car Dealer Showroom**

**Test Plan**

*Purpose: describe the context of the testing, methodology, environment,*

*types of tests, the standards, deliverables and test scope.*

**Team members: Halina Samulenka**

**Date/Version: 10/06/2021**

# Overview

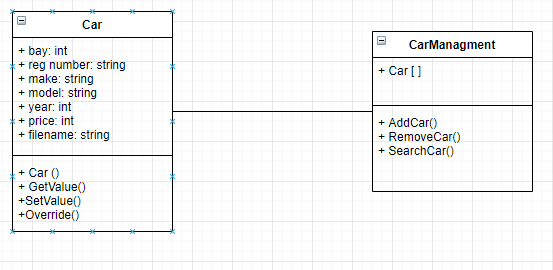
## Purpose

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?



“Car Dealer Showroom” is a car inventory showroom program that can house up to 20 cars. Each car is assigned to a numbered parking bay. The program must provide the ability for the user to add, remove or search for cars. The program must use an array to hold the car inventory details, include validation and demonstrate examples of array search.

One of the main stakeholder of this program is the manager of a car dealership, who is the first user of this program. The manager need to test the program to sure that he is satisfied. Also developer of the program must be sure that all client’s requirements are met.

The benefits of testing:

* It provides better program design and higher code quality
* It reduces costs
* It simplifies the debugging process
* It provides documentation of the system
* It ensures that the customer or end- user is satisfied.
* It ensures the all the requirements should be met.

## Methodology

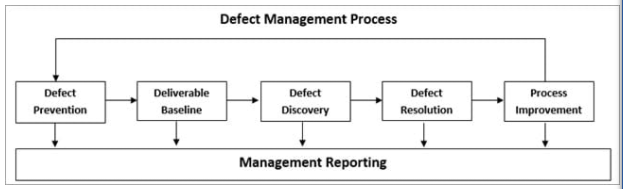
This section explains the testing methodology that will be performed by the project team for the Car Dealer Showroom project. It defines the overall testing approach and test activities. Its purpose is to document:

* How testing will be performed;
* What resources are needed and their responsibilities, as well as the schedule for testing
* How will defects and test cycles be managed?
* Any risks or dependencies
* Formal SDLC methodologies that will be followed.

Well known testing methodologies may include V model, Waterfall model, and iterative or agile methodologies. Explain the rationale for choosing a particular model.

Testing will include functional (unit testing) and non-functional testing (black-box technique)). Selecting proper testing methodologies also incorporates choosing a proper model in SDLC. The Project testing will be performed into Agile methodology. We will use Q1(The Automated quadrant) for Unit Test and Q2 (Automated and Manual quadrant) for functional test. One of the reason why we chose this methodology these are resources are allocated entirely to one project. There is a single project manager as a contact person and he is responsible for the project. All project tasks must be finished within 18 weeks. All dependencies in the project are a type of risk (for example, to add new car we need to check empty car bay or not).

Defect life cycle starts from the stage when the defect is found or raised in the process of testing and ends when a defect is closed either by ensuring that it’s not reproducible or rejected by a developer.



We choose Agile model because:

* It is small project
* As testing begins at the start of the project, errors can be fixed in the middle of the project.
* It needs a less documentation
* Every iteration has its own testing phase
* Testers and developers work closely
* User acceptance is performed at the end of every sprint.

## Test environment

What platforms will the app run on? Does it need to connect to a backend system? Describe the environment (hardware, software, network) that needs to be set up for testing. What tools and frameworks will be used? Is there a plan for resource provisioning?

The app will run on Visual Studio and needn`t to connect to a backend system. WPF testing requires PC with Windows OS and Visual Studio installation. We will use .NET framework Unit Testing tools.

## Types of Tests

Describe the types of tests that will be done.

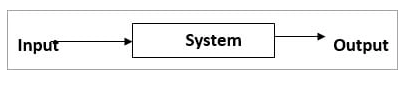
In addition to the functional and non-functional tests, what stages/levels of testing will there be and who will be responsible for performing them?

Will UI tests be conducted in the same way as other functional tests? ie. Will you use interactive testing, automated unit testing or a combination?

Will there be black box testing, white box testing, or a combination?

The functional and non-functional testing tests will be done.

The non-functional test comes under a [black box testing](https://www.softwaretestinghelp.com/black-box-testing/) technique. This technique looks like



We will use checklist to ensure that no important aspect is left without testing. Checklist will include:

* Check how to add the Car
* Check how to remove the Car
* Check how work the searching functions

We will do also User Interface Testing. This test will evaluate the GUI. GUI should provide data about car in the car parking bay. GUI should provide tooltips to make it easy to add, remove and search the car.

Scalability testing will be done to verify if the application is capable enough to handle 20 positions.

We will use also automated unit test to ensure that each part in the software meet the customer`s requirements (add car, remove car, etc.)

## Test standards

What standards will be used for measuring the quality of the software? What processes will be used for ensuring quality assurance standards are met?

Will test cycles be iterated until there are no more defects, or will defect fixes be prioritised?

ISO 9000 and CMMI model Standards will be used for measuring the quality of the software.

Each process area in CMMI represents an activity needs to be followed to achieve a Maturity Level. We will use following process in CMMI: project planning, risk management, requirement management, technical solution, validation, verification, organizational performance management, etc.

According to the ISO 9000 we will use three general styles of process: management processes, business processes, and support processes.

Test cycles will be iterated until there are no more defects.

## Test deliverables

How will test results be captured or recorded? What are the test deliverables?

Who will signing off at the end of the test cycle and what will be the handover process?

The reports of the automated unit test will be automatically generated in the program. Non-functional testing results will be recorded in this report.

Test Deliverables are the test artifacts which are given to the stakeholders of a software project during the [SDLC (Software Development Life Cycle)](https://www.softwaretestingmaterial.com/sdlc-software-development-life-cycle/). These are test strategy, test plan and estimation, test scenario, test cases and test data, RTM, test summary report, test closure report, incident report.

The project manager will sign off at the test cycle. The handover process will include handover documentation, handover criteria and the review process prior to handover.

# Testing Plan

## Scope of Testing

### Functional tests

List the functional tests here or in a spreadsheet.

The list of functional tests must cross-reference the items in the requirements specifications.

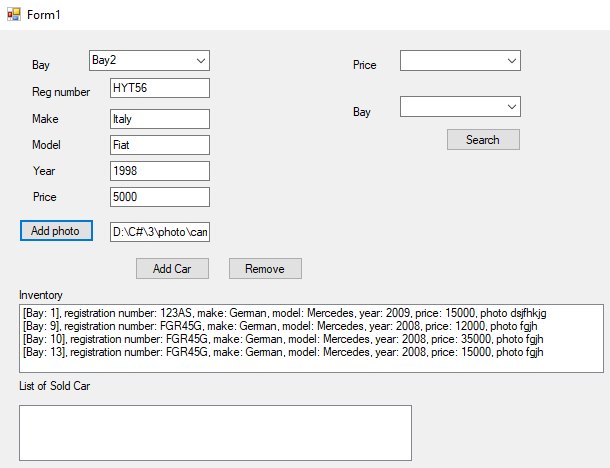
Describe what input data will be used in the test cases.

Include the following elements:

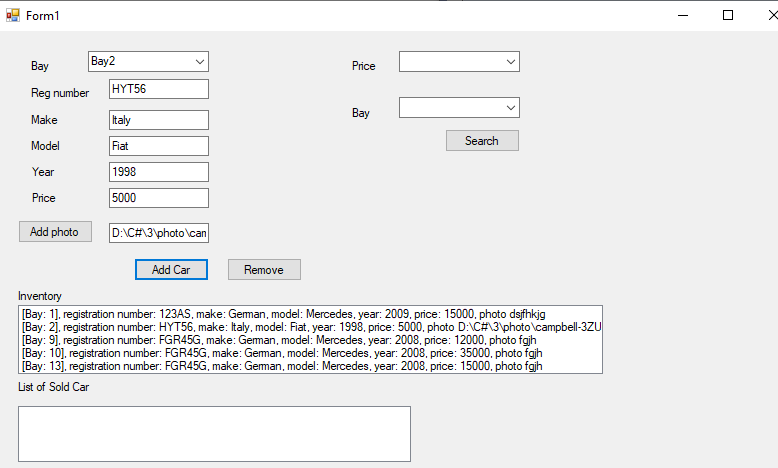
* Requirement number
* Test number
* Description or steps in the test
* Input data
* Expected results
* Pass or fail result for each iteration/date
* Priority/severity

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Inputs | Steps | Expected result | Actual result | Pass or Fail result | Comments |
| **1a** | **Add car valid (add car to the empty bay)** | **Car details** | **Fill details, press add button** | **Add new car to Car inventory** | **Add new car to Car inventory** | **Pass** |  |
| **1b** | **Add car invalid ( add car to the occupied parking space)** | **Car details** | **Fill details, press add button** | **Doesn`t add new car to Car inventory** | **Doesn`t add new car to Car inventory** | **Pass** |  |
| **2a** | **Remove car** | **Selected item in List Car Inventory** | **Select Car in List inventory, press Remove button** | **Remove selected car from List inventory car to the List of Sold car** | **Remove selected car from List inventory car to the List of Sold car** | **Pass** |  |
| **3a** | **Sequential search to find cars in a price range** | **Selected price range** | **Select price range, press Search button** | **Find all Cars in Car Inventory with the corresponding price** | **Find all Cars in Car Inventory with the corresponding price** | **Pass** |  |
| **3b** | **Directly access to inventory array using car bay** | **Selected car bay** | **Select car bay, press Search button** | **Find car in Car inventory with the corresponding bay** | **The program found all cars from Sequential search with selected price and added car with selected car bay** | **Fail** | **The program must be fixed** |
| **3c** | **Directly access to inventory array using car bay (after the fixing problem)** | **Selected car bay** | **Select car bay, press Search button** | **Find car in Car inventory with the corresponding bay** | **Find car in Car inventory with the corresponding bay** | **Pass** |  |

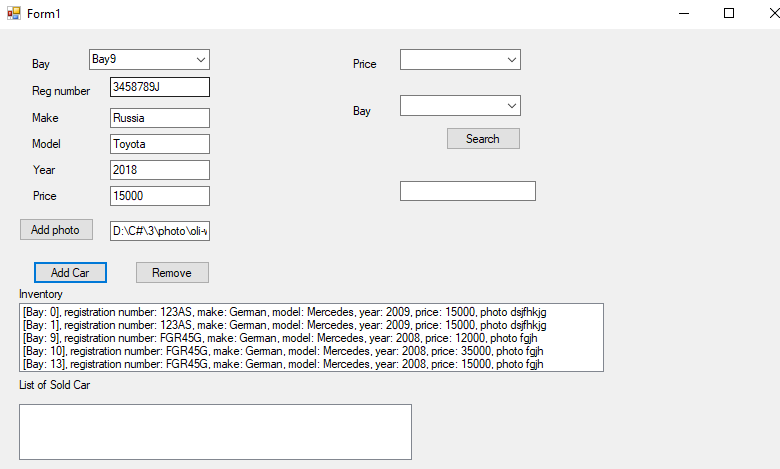
**Test 1a.**



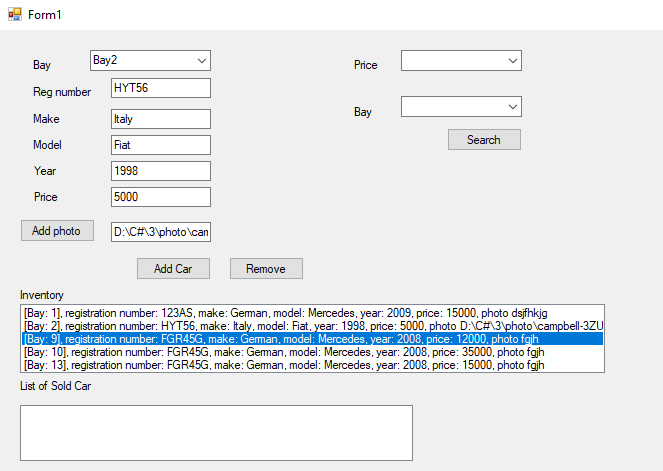
Result



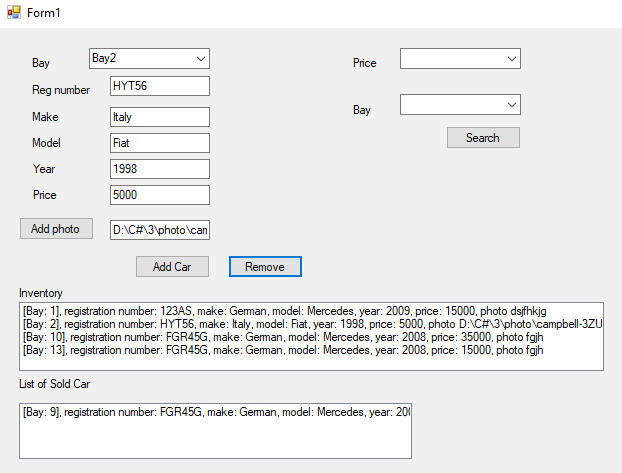
**Test 1b**



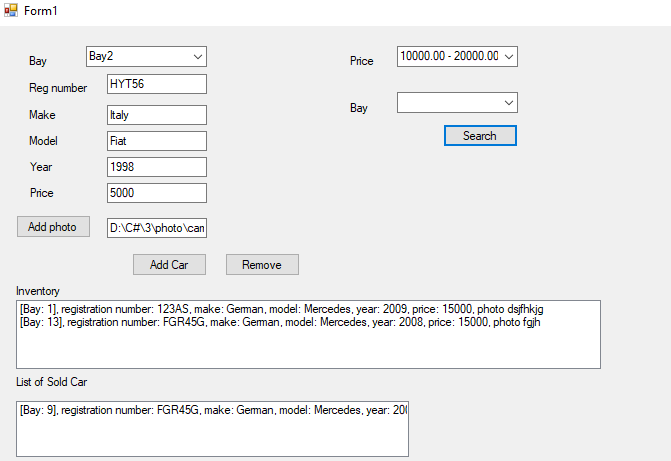
**Test 2a.**



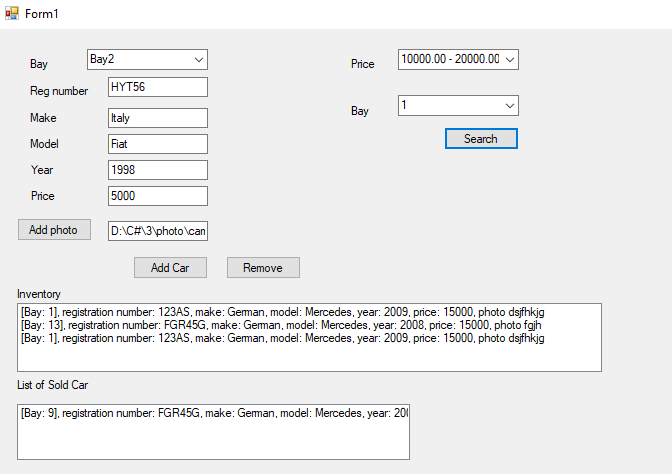
**Result**



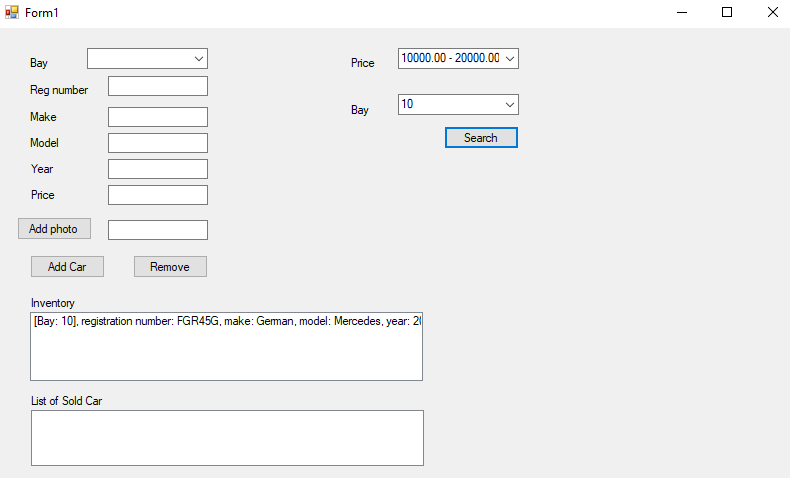
**Test 3a.**



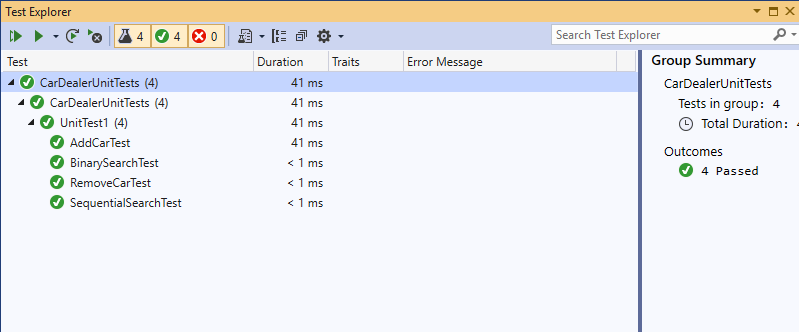
**Test 3b.**



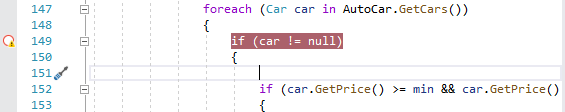
**Test 3c.**



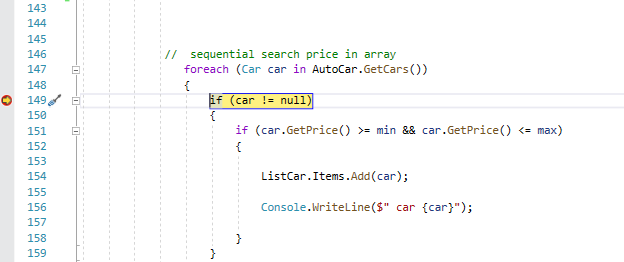
**Unit Test results**



**Breakpoints** mark a point in the code where the execution will take a break.



When we debug, execution pauses at the breakpoint, before the code on that line is executed. The breakpoint symbol shows a yellow arrow



### Performance tests

Determine what performance test metrics will be used to evaluate performance. Consider memory usage, CPU and response time.

Will it be feasible to conduct load testing?

To evaluate performance, we will use client-side metrics (transaction response time, throughput, number of passed transactions, etc.) and system performance metrics (CPU usage, memory usage, disk usage, etc.).

Load testing measures system performance as the workload increases. That workload could mean concurrent users or transactions. The program is not intended to be used by concurrent users or transactions.

### Other non-functional tests

Discuss other non-functional tests such as security, usability, scalability, etc.

Can any metrics be captured related to the usage of the application?

Usability is generally measured using a number of observable and quantifiable metrics. The main usability metrics, that can be captured, include effectiveness, efficiency and satisfaction.

Application security metrics are: Application security process metrics, including on what level the business is meeting security standards and policies; Application security risk metrics that link vulnerability, threat and incident measurements to the businesses risk objectives; and lastly, metrics surrounding security in the [SDLC](https://www.checkmarx.com/glossary/software-development-life-cycle-sdlc/) – metrics showing value in the tools and resources in identifying root causes of [vulnerabilities](https://www.checkmarx.com/entity/vulnerability-computing/) and mitigating their risk.

Scalability test can be captured related to the usage 20 car bays.

### Out of scope

Describe what is out of scope from a testing perspective for the project team. For example, the user may be responsible for testing usability or functionality that requires business domain knowledge. Have the requirements been assigned a MoSCoW prioritisation?

User may have setup problem to test the app. The requirements have been assigned a MoSCow prioritisation into four components: must-haves, should-haves, could-haves, and won't-haves.