

# Master test plan for the course project

## SWE 326: Software Testing and Quality Assurance



Haitham Alsaeed	201648500
Mohammad Eskandarani	201615200
Ahmed Almoairfi	201634600
Moaz Khan	201513950

## History

Version	Date	Author	Changes
1.0	3/14/2020	Haitham Alsaeed	Filling Section 1 (1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4) and Section 2

## Table of Contents

1. Introduction	4
1.1. The Project	4
1.2. Overview	4
1.2.1. Test Organization	4
1.2.2. System Under test (SUT)	4
1.2.3. Testing types	4
1.2.4. Project Phases	4
1.2.5. Features to be tested	4
1.2.6. Features not to be tested	5
1.1. Test Environment and Tools	5
1.2. Document Structure	5
2. Static Code Analysis	5
2.1. Manual Code Inspection	5
2.2. Automatic Static Code Analysis	6
2.2.1. (Name of Selected Static Code Analysis Tool)	6
2.2.2. Produced reports	6
3. System Test	6
3.1. System Requirements	6
3.2. System Test Cases	6
3.3. Report on the Execution of System Test Cases	7
4. Unit Testing	7
5. Conclusion	7

## 1. Introduction

### 1.1. The Project

The project is about testing a car cruise control system in three phases; the first phase is “Static Code Analysis”. The aim of the first phase is to test the system without executing it. Second phase is “System Testing”; the aim of it is to test the whole system as a black box. The third phase is “Unit Testing”; the aim of it is to test the units of the system as white boxes with the help of some testing tools.

### 1.2. Overview

#### 1.2.1. Test Organization

Teams are composed of four members each. Each team member will act as a team leader for one phase (sub-phase). Provide the assigned team leaders (where applicable) :

**Phase 1 (Static Code Analysis):** Haitham Alsaeed

**Phase 2 (System Testing):** Mohammad Eskandarani

**Phase 3 (Unit Testing):** Moaz Khan

**Phase 4 (Presentation):** Ahmed Almoairfi

#### 1.2.2. System Under test (SUT)

The main idea of the system is to allow the driver of a car to maintain speed without pressing the accelerator pedal.

The system consists of five classes and one interface, with a total of 36 methods, 522 lines of code, 26 if statements, and 3 loops.

#### 1.2.3. Testing types

We aim at conducting the following three types of testing:

- Static code analysis
- System testing
- Unit testing

#### 1.2.4. Project Phases

The project consists of four phases: The first phase is the static code analysis to the cruise control project which will help in finding certain kinds of problems that can easily be recognized using a predefined checklist. The second phase is System Testing of the cruise control project. The plan should be developed then provides a report of the system testing execution. The third phase is Unit Testing of the cruise control project. In this phase each

unit will be tested separately using a tool to automate the test like JUnit to repeat the test easily. First by designing the test cases then executing the test cases and reporting the result. The fourth phase is presenting the result of the testing. In this phase we will show the result of the previous three phases.

### **1.2.5. Features to be tested**

Describe briefly what aspects of the software will be tested.

### **1.2.6. Features not to be tested**

What types of testing are not addressed in this test plan?

## **1.1. Test Environment and Tools**

This section describes the test environment and tools used during this project.

- Eclipse tool
- Tool checkstyle for static code inspection.
- JUnit environment for Unit Testing.
- Codecover tool for execution coverage

## **1.2. Document Structure**

This document follows (not strictly) the template for test plans in IEEE standard for software test documentation [IEEE829].

## **2. Static Code Analysis**

### **2.1. Manual Code Inspection**

Class Name: SpeedControl

Time Report:

Time spent on Familiarization with the Code: 10 minute

Time spent on Filling in Findings Report: 10 minute

Time spent on Actual Inspection:20 minute

Findings:

#	Description	Line #	Class Name	Associated Checklist Item #
1	Placement of curly braces wasn't done consistently to others	23	SpeedControl	2.1

2	The names are not used consistently for the same concept	10, 11	SpeedControl	2.4
3	The contractor input can be null which will cause an error in the result of calling disp.disabled()	13	SpeedControl	3.3
4	The method doesn't have a comment	18, 22, 26, 35	SpeedControl	7.2
5	The variable has no description	9, 11	SpeedControl	7.3
6	int 10 in set speed should be used as constant that has clear name	23	SpeedControl	7.5
7	int 500 in set wait should be used as constant that has clear name	45	SpeedControl	7.5
8	the class doesn't have to string method		SpeedControl	10.1

Class Name: CruiseDisplay

Time Report:

Time spent on Familiarization with the Code: 24 minutes

Time spent on Filling in Findings Report: 75 minutes

Time spent on Actual Inspection: 30 minutes

Findings:

#	Description	Line #	Class Name	Associated Checklist Item #
1	There is no {.	53, 55, 57, 59	CruiseDisplay	2.1

2	No comments to describe these lines of code.	20, 28, 33, 66	CruiseDisplay	2.3
3	Bad naming for constant	10	CruiseDisplay	2.5
4	No input check	41, 45, 65	CruiseDisplay	3.1
5	There are no descriptions for the methods.	24, 41, 45, 65, 73, 78, 83	CruiseDisplay	7.2
6	The purpose of these variables is not clear	20 to 22	CruiseDisplay	7.3
7	Bad naming for recorded speed.	8	CruiseDisplay	7.4
8	Bad naming for the methods.	73, 78	CruiseDisplay	7.4
9	Not a general method.	45	CruiseDisplay	8.1
10	No <code>toString</code> method.	5	CruiseDisplay	10.1

Class Name: CruiseControl

Inspector's name: Moaz Khan

Time Report:

Time spent on Familiarization with the Code: 11 minutes

Time spent on Filling in Findings Report: 68 minutes

Time spent on Actual Inspection: 68 minutes

Findings:

#	Description	Line #	Class Name	Associated Checklist Item #

1	Bracket '{' should be moved up one line to stay consistent.	25	CruiseControl	2.1
2	Whole block should be moved up two lines and back two indentations to stay consistent.	115-117	CruiseControl	2.1
3	Cont. on 2, the line should be moved back one indentation to stay consistent.	116	CruiseControl	2.2
4	Brackets '}' should be moved back one space to align vertically with its block.	49, 57, 65, 73, 80, 87, 94	CruiseControl	2.1
5	Bracket '}' should be moved forward one space to align vertically with its block.	106	CruiseControl	2.1
6	Each Button's initialization and addActionListener(){} can be in its own method to simplify the init() method.	43-49, 51-57, 59-65, 67-73, 75-80, 82-87, 89-94	CruiseControl	6.1
7	Methods should have a comment block to describe its use.	24, 32, 108	CruiseControl	7.2
8	Comments on instance variables to describe their purpose, restrictions, and relationships	13-22	CruiseControl	7.3
9	Height and width of the frame should be constant variables.	27	CruiseControl	7.5
10	setVisible() should use a variable 'visibility'.	28	CruiseControl	7.5

11	Variables should be private.	13-22	CruiseControl	9.2
12	A comment that describes the purpose of the class should be provided.	1	CruiseControl	7
13	No <code>toString()</code> method.		CruiseControl	10.1
14	Variable names could be better.	13-15	CruiseControl	7.4

Class Name: Controller

Time Report:

Time spent on Familiarization with the Code: 6 minutes

Time spent on Filling in Findings Report: 50 minutes

Time spent on Actual Inspection: 30 minutes

Findings:

#	Description	Line #	Class Name	Associated Checklist Item #
1	Statements are in one line.	12, 39	Controller	2.1
2	Incorrect place for braces.	17	Controller	2.1
3	Incorrect place for braces and statements are in one line.	22, 34, 46, 55	Controller	2.1
4	There are no braces for if statement.	27	Controller	2.1
5	The statement in the same line as "if" statement.	27	Controller	2.2

6	Bad naming convention for instance variable.	9, 10	Controller	2.5
7	No comment to describe the variable.	10	Controller	7.3
8	No comments to describe the methods.	15, 20, 25, 32, 37, 44, 53	Controller	7.2
9	The name of the variable does not clarify the purpose.	9	Controller	7.4
10	No <code>toString</code> method.	3	Controller	10.1

Class Name: CarSpeed

Inspector's name: Moaz Khan

Time Report:

Time spent on Familiarization with the Code: <1 minute

Time spent on Filling in Findings Report: 8 minutes

Time spent on Actual Inspection: 8 minutes

Findings:

#	Description	Line #	Class Name	Associated Checklist Item #
1	Parameter usage is not clear from its name.	7	CarSpeed	7.4
2	Comments on the purpose of the interface should be provided.	1	CarSpeed	7
3	Comments to describe the use of each method should be provided.	5, 7	CarSpeed	7.2

Class Name: CarSimulator

Time Report:

Time spent on Familiarization with the Code: 25 minute

Time spent on Filling in Findings Report: 4 hours

Time spent on Actual Inspection: 4 hours

Findings:

#	Description	Line #	Class Name	Associated Checklist Item #
1	There is no {	55,57,82,14 2,145,147,1 54,157,170, 171,175,191 ,192	CarSimulator	2.1
2	There is no }	57,59,84,14 4,147,149,1 56,157,170, 171,175,191 ,192	CarSimulator	2.1
3	no comment to explain this lines of code	4,14,28,32, 34,45,49,55, 68,77,82,93, 105,109,117 ,125,135,14 1,152,161,1 88,195	CarSimulator	2.3
4	Indentation is used in wrong way	169 to 176	CarSimulator	2.2
5	Bad naming for “flen”	94	CarSimulator	2.4
6	Bad constant naming	15,16	CarSimulator	2.5
7	Not using camel case	29,30,31,33, 94,95,164,1 65	CarSimulator	2.5
8	Meaningless variable name	34,46,50,69, 93,94,95,96, 97,164,165	CarSimulator	2.5
9	Inefficient complex code	189,194	CarSimulator	4.1

10	No comment to describe this method	23,33,46,50, 69,78,93,10 6,126,135,1 41,153,162, 196	CarSimulator	7.2
11	No comment to describe this instance variables	29,30,31	CarSimulator	7.3
12	Bad naming for instance variables	29,30,31	CarSimulator	7.4
13	Bad naming for method's parameter variables	46,50,69,78, 93,106,189	CarSimulator	7.4
14	Bad naming for local variables	34,94,95,96, 97,164,165	CarSimulator	7.4
15	This variables visibility is not stated (public, protected, private)	29,30,31	CarSimulator	9.1
16	No <code>toString</code> method	199	CarSimulator	10.1

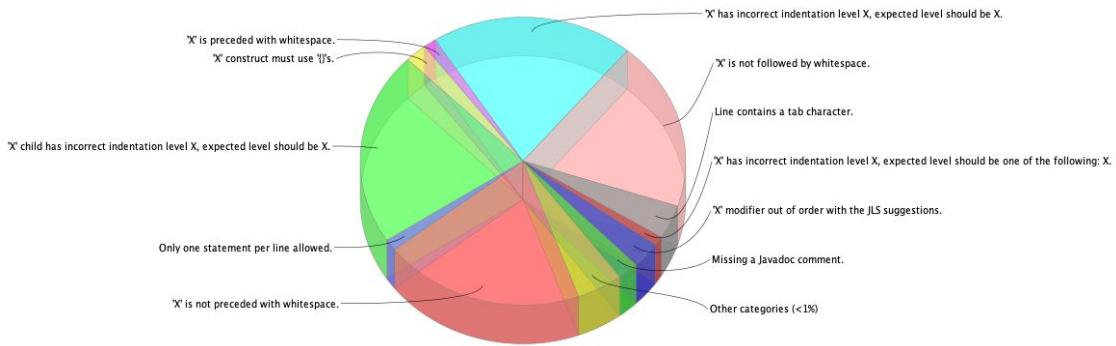
## 2.2. Automatic Static Code Analysis

### 2.2.1. checkstyle tool

Eclipse tool to help in automating the static analysis and give a report of the result.

### 2.2.2. Produced reports

# SWE-326 Master Test Plan



Checkstyle violation type	Occurrences
Name 'X' must match pattern 'X'.	6
'X' is not preceded with whitespace.	170
'X' at column X should have line break after.	8
Distance between variable 'X' declaration and its first usage is X, but allowed X. Consider making that variable final if you still need to store its value in advance (before m...	1
Using the '*' form of import should be avoided - X.	5
Empty catch block.	2
Only one statement per line allowed.	11
'X' child has incorrect indentation level X, expected level should be X.	152
'X' at column X should be alone on a line.	1
'X' construct must use '{}'s.	19
'X' is preceded with whitespace.	3
'X' at column X should be on the same line as the next part of a multi-block statement (one that directly contains multiple blocks: if/else-if/else, do/while or try/catch/finally).	1
'X' has incorrect indentation level X, expected level should be X.	138
'X' is not followed by whitespace.	161
'X' at column X should be on the previous line.	8
Line contains a tab character.	33
Array brackets at illegal position.	1
'X' has incorrect indentation level X, expected level should be one of the following: X.	9
'X' modifier out of order with the JLS suggestions.	25
Missing a Javadoc comment.	13
Comment has incorrect indentation level X, expected is X, indentation should be the same level as line X.	2

Resource	In Folder	Line	Message
CarSimulator.java	/SWE326P1/src/appCruise	1	Package name 'appCruise' must match pattern "[a-zA-Z][a-zA-Z0-9]*\$".
CarSpeed.java	/SWE326P1/src/appCruise	1	Package name 'appCruise' must match pattern "[a-zA-Z][a-zA-Z0-9]*\$".
Controller.java	/SWE326P1/src/appCruise	1	Package name 'appCruise' must match pattern "[a-zA-Z][a-zA-Z0-9]*\$".
CruiseDisplay.java	/SWE326P1/src/appCruise	1	Package name 'appCruise' must match pattern "[a-zA-Z][a-zA-Z0-9]*\$".
SpeedControl.java	/SWE326P1/src/appCruise	1	Package name 'appCruise' must match pattern "[a-zA-Z][a-zA-Z0-9]*\$".
CruiseControl.java	/SWE326P1/src/appCruise	5	Package name 'appCruise' must match pattern "[a-zA-Z][a-zA-Z0-9]*\$".

## SWE-326 Master Test Plan

Resource	In Folder	Line	Message
⑤ CruiseDisplay.java	/SWE326P1/src/appCruise	36	WhitespaceAround: '-' is not preceded with whitespace.
⑤ CruiseDisplay.java	/SWE326P1/src/appCruise	36	WhitespaceAround: '-' is not preceded with whitespace.
⑤ SpeedControl.java	/SWE326P1/src/appCruise	36	WhitespaceAround: '=' is not preceded with whitespace.
⑤ SpeedControl.java	/SWE326P1/src/appCruise	36	WhitespaceAround: '=' is not preceded with whitespace.
⑤ SpeedControl.java	/SWE326P1/src/appCruise	36	WhitespaceAround: '=' is not preceded with whitespace.
⑤ Controller.java	/SWE326P1/src/appCruise	37	WhitespaceAround: '(' is not preceded with whitespace.
⑤ Controller.java	/SWE326P1/src/appCruise	38	WhitespaceAround: '!' is not preceded with whitespace.
⑤ Controller.java	/SWE326P1/src/appCruise	38	WhitespaceAround: '(' is not preceded with whitespace.
⑤ CruiseDisplay.java	/SWE326P1/src/appCruise	38	WhitespaceAround: '-' is not preceded with whitespace.
⑤ CruiseDisplay.java	/SWE326P1/src/appCruise	38	WhitespaceAround: '-' is not preceded with whitespace.
⑤ Controller.java	/SWE326P1/src/appCruise	40	WhitespaceAround: ';' is not preceded with whitespace.
⑤ SpeedControl.java	/SWE326P1/src/appCruise	41	WhitespaceAround: '=' is not preceded with whitespace.
⑤ SpeedControl.java	/SWE326P1/src/appCruise	42	WhitespaceAround: '-' is not preceded with whitespace.
⑤ SpeedControl.java	/SWE326P1/src/appCruise	42	WhitespaceAround: ';' is not preceded with whitespace.
⑤ SpeedControl.java	/SWE326P1/src/appCruise	43	WhitespaceAround: '=' is not preceded with whitespace.
⑤ Controller.java	/SWE326P1/src/appCruise	44	WhitespaceAround: '(' is not preceded with whitespace.
⑤ SpeedControl.java	/SWE326P1/src/appCruise	44	WhitespaceAround: '*' is not preceded with whitespace.
⑤ Controller.java	/SWE326P1/src/appCruise	45	WhitespaceAround: '=' is not preceded with whitespace.
⑤ Controller.java	/SWE326P1/src/appCruise	46	WhitespaceAround: '=' is not preceded with whitespace.
⑤ Controller.java	/SWE326P1/src/appCruise	46	WhitespaceAround: ')' is not preceded with whitespace.
⑤ SpeedControl.java	/SWE326P1/src/appCruise	47	WhitespaceAround: ';' is not preceded with whitespace.
⑤ Controller.java	/SWE326P1/src/appCruise	48	WhitespaceAround: '=' is not preceded with whitespace.
⑤ SpeedControl.java	/SWE326P1/src/appCruise	48	WhitespaceAround: '=' is not preceded with whitespace.
⑤ Controller.java	/SWE326P1/src/appCruise	53	WhitespaceAround: '(' is not preceded with whitespace.
⑤ CarSimulator.java	/SWE326P1/src/appCruise	54	WhitespaceAround: '*' is not preceded with whitespace.
⑤ Controller.java	/SWE326P1/src/appCruise	54	WhitespaceAround: '=' is not preceded with whitespace.
⑤ CruiseDisplay.java	/SWE326P1/src/appCruise	54	WhitespaceAround: '*' is not preceded with whitespace.
⑤ Controller.java	/SWE326P1/src/appCruise	55	WhitespaceAround: '=' is not preceded with whitespace.
⑤ Controller.java	/SWE326P1/src/appCruise	55	WhitespaceAround: ')' is not preceded with whitespace.
⑤ CruiseDisplay.java	/SWE326P1/src/appCruise	56	WhitespaceAround: '*' is not preceded with whitespace.
⑤ CarSimulator.java	/SWE326P1/src/appCruise	59	WhitespaceAround: '*' is not preceded with whitespace.
⑤ CarSimulator.java	/SWE326P1/src/appCruise	61	WhitespaceAround: '*' is not preceded with whitespace.
⑤ CarSimulator.java	/SWE326P1/src/appCruise	61	WhitespaceAround: '**' is not preceded with whitespace.

SWE-326 Master Test Plan

CarSimulator.java	/SWE326P1/src/appCruise	190	'WhitespaceAround: '=' is not preceded with whitespace.
CarSimulator.java	/SWE326P1/src/appCruise	191	'WhitespaceAround: '< is not preceded with whitespace.
CarSimulator.java	/SWE326P1/src/appCruise	191	'WhitespaceAround: ';' is not preceded with whitespace.
CarSimulator.java	/SWE326P1/src/appCruise	192	'WhitespaceAround: '> is not preceded with whitespace.
CarSimulator.java	/SWE326P1/src/appCruise	192	'WhitespaceAround: ',' is not preceded with whitespace.
CarSimulator.java	/SWE326P1/src/appCruise	192	'WhitespaceAround: '-' is not preceded with whitespace.

Resource	In Folder	Line	Message
Controller.java	/SWE326P1/src/appCruise	13	'\n' at column 5 should have line break after.
Controller.java	/SWE326P1/src/appCruise	17	'\n' at column 7 should have line break after.
Controller.java	/SWE326P1/src/appCruise	22	'\n' at column 7 should have line break after.
SpeedControl.java	/SWE326P1/src/appCruise	23	'\n' at column 26 should have line break after.
Controller.java	/SWE326P1/src/appCruise	34	'\n' at column 7 should have line break after.
SpeedControl.java	/SWE326P1/src/appCruise	36	'\n' at column 26 should have line break after.
Controller.java	/SWE326P1/src/appCruise	46	'\n' at column 7 should have line break after.
Controller.java	/SWE326P1/src/appCruise	55	'\n' at column 6 should have line break after.

Resource	In Folder	Line	Message
CruiseControl.java	/SWE326P1/src/appCruise	35	Distance between variable 'isfixed' declaration and its first usage is 6, but allowed 3. Consider making that variable final if you still need to store its value in advance (before...

## SWE-326 Master Test Plan

Resource	In Folder	Line	Message
CarSimulator.java	/SWE326P1/src/appCruise	3	Using the '*' form of import should be avoided - java.awt.*.
CruiseDisplay.java	/SWE326P1/src/appCruise	3	Using the '*' form of import should be avoided - java.awt.*.
CruiseControl.java	/SWE326P1/src/appCruise	7	Using the '*' form of import should be avoided - java.awt.*.
CruiseControl.java	/SWE326P1/src/appCruise	8	Using the '*' form of import should be avoided - java.awt.event.*.
CruiseControl.java	/SWE326P1/src/appCruise	9	Using the '*' form of import should be avoided - javax.swing.*.

Resource	In Folder	Line Message
SpeedControl.java	/SWE326P1/src/appCruise	47 Empty catch block.
CarSimulator.java	/SWE326P1/src/appCruise	... Empty catch block.

Resource	In Folder	Line Message
Controller.java	/SWE326P1/src/appCruise	13 Only one statement per line allowed.
SpeedControl.java	/SWE326P1/src/appCruise	14 Only one statement per line allowed.
SpeedControl.java	/SWE326P1/src/appCruise	15 Only one statement per line allowed.
SpeedControl.java	/SWE326P1/src/appCruise	19 Only one statement per line allowed.
Controller.java	/SWE326P1/src/appCruise	22 Only one statement per line allowed.
SpeedControl.java	/SWE326P1/src/appCruise	23 Only one statement per line allowed.
Controller.java	/SWE326P1/src/appCruise	34 Only one statement per line allowed.
SpeedControl.java	/SWE326P1/src/appCruise	36 Only one statement per line allowed.
Controller.java	/SWE326P1/src/appCruise	39 Only one statement per line allowed.
Controller.java	/SWE326P1/src/appCruise	46 Only one statement per line allowed.
Controller.java	/SWE326P1/src/appCruise	55 Only one statement per line allowed.

SWE-326 Master Test Plan

## SWE-326 Master Test Plan

---

§	CruiseDisplay.java	/SWE326P1/src/appCruise	66	'method def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	67	'method def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	68	'method def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	69	'method def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	70	'method def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	74	'method def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	75	'method def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	79	'method def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	80	'method def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	84	'method def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	85	'method def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	15	'ctor def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	16	'ctor def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	25	'method def' child has incorrect indentation level 8, expected level should be 4.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	28	'if' child has incorrect indentation level 16, expected level should be 6.
§	CruiseDisplay.java	/SWE326P1/src/appCruise	29	'if' child has incorrect indentation level 16, expected level should be 6.
§	SpeedControl.java	/SWE326P1/src/appCruise	42	'while' child has incorrect indentation level 16, expected level should be 8.
§	SpeedControl.java	/SWE326P1/src/appCruise	43	'while' child has incorrect indentation level 16, expected level should be 8.
§	SpeedControl.java	/SWE326P1/src/appCruise	44	'while' child has incorrect indentation level 16, expected level should be 8.
§	SpeedControl.java	/SWE326P1/src/appCruise	45	'while' child has incorrect indentation level 12, expected level should be 8.
§	SpeedControl.java	/SWE326P1/src/appCruise	28	'if' child has incorrect indentation level 10, expected level should be 6.

Resource	In Folder	Line	Message
§ Controller.java	/SWE326P1/src/appCruise	13	'}' at column 46 should be alone on a line.

Resource	In Folder	Line	Message
§ CarSimulator.java	/SWE326P1/src/appCruise	...	'if' construct must use '()'s.
§ CarSimulator.java	/SWE326P1/src/appCruise	...	'if' construct must use '()'s.
§ CarSimulator.java	/SWE326P1/src/appCruise	...	'if' construct must use '()'s.
§ CarSimulator.java	/SWE326P1/src/appCruise	55	'if' construct must use '()'s.
§ CarSimulator.java	/SWE326P1/src/appCruise	57	'else' construct must use '()'s.
§ CarSimulator.java	/SWE326P1/src/appCruise	82	'for' construct must use '()'s.
§ CarSimulator.java	/SWE326P1/src/appCruise	...	'if' construct must use '()'s.
§ CarSimulator.java	/SWE326P1/src/appCruise	...	'if' construct must use '()'s.
§ CarSimulator.java	/SWE326P1/src/appCruise	...	'if' construct must use '()'s.
§ CarSimulator.java	/SWE326P1/src/appCruise	...	'if' construct must use '()'s.
§ CarSimulator.java	/SWE326P1/src/appCruise	...	'else' construct must use '()'s.
§ CarSimulator.java	/SWE326P1/src/appCruise	...	'if' construct must use '()'s.
§ CarSimulator.java	/SWE326P1/src/appCruise	...	'if' construct must use '()'s.
§ Controller.java	/SWE326P1/src/appCruise	27	'if' construct must use '()'s.
§ CruiseDisplay.java	/SWE326P1/src/appCruise	53	'if' construct must use '()'s.
§ CruiseDisplay.java	/SWE326P1/src/appCruise	55	'else' construct must use '()'s.
§ CruiseDisplay.java	/SWE326P1/src/appCruise	57	'if' construct must use '()'s.
§ CruiseDisplay.java	/SWE326P1/src/appCruise	59	'else' construct must use '()'s.

Resource	In Folder	Line	Message
§ Controller.java	/SWE326P1/src/appCruise	16	'}' is preceded with whitespace.
§ Controller.java	/SWE326P1/src/appCruise	21	'}' is preceded with whitespace.
§ Controller.java	/SWE326P1/src/appCruise	45	'}' is preceded with whitespace.

Resource	In Folder	Line	Message
§ Controller.java	/SWE326P1/src/appCruise	46	'}' at column 50 should be on the same line as the next part of a multi-block statement (one that directly contains multiple blocks: if/else-if/else, do/while or try/catch/finally).

SWE-326 Master Test Plan

## SWE-326 Master Test Plan

8	CruiseDisplay.java	/SWE326P1/src/appCruise	17	'ctor def rourly' has incorrect indentation level 4, expected level should be 2.
9	CruiseDisplay.java	/SWE326P1/src/appCruise	20	'member def type' has incorrect indentation level 4, expected level should be 2.
10	CruiseDisplay.java	/SWE326P1/src/appCruise	21	'member def type' has incorrect indentation level 4, expected level should be 2.
11	CruiseDisplay.java	/SWE326P1/src/appCruise	22	'member def type' has incorrect indentation level 4, expected level should be 2.
12	CruiseDisplay.java	/SWE326P1/src/appCruise	24	'method def modifier' has incorrect indentation level 4, expected level should be 2.
13	CruiseDisplay.java	/SWE326P1/src/appCruise	26	'if' has incorrect indentation level 12, expected level should be 4.
14	SpeedControl.java	/SWE326P1/src/appCruise	41	'while' has incorrect indentation level 10, expected level should be 6.
15	SpeedControl.java	/SWE326P1/src/appCruise	4	'class def ident' has incorrect indentation level 1, expected level should be 0.

## SWE-326 Master Test Plan

SWE-326 Master Test Plan

Resource	In Folder	Line Message
Controller.java	/SWE326P1/src/appCruise	17 `(` at column 7 should be on the previous line.
Controller.java	/SWE326P1/src/appCruise	22 `(` at column 7 should be on the previous line.
Controller.java	/SWE326P1/src/appCruise	34 `(` at column 7 should be on the previous line.
Controller.java	/SWE326P1/src/appCruise	46 `(` at column 7 should be on the previous line.
Controller.java	/SWE326P1/src/appCruise	55 `(` at column 6 should be on the previous line.
Controller.java	/SWE326P1/src/appCruise	13 `(` at column 5 should be on the previous line.
CruiseControl.java	/SWE326P1/src/appCruise	25 `(` at column 5 should be on the previous line.
CruiseControl.java	/SWE326P1/src/appCruise	... `(` at column 9 should be on the previous line.

Resource	In Folder	Line Message
CarSimulator.java	/SWE326P1/src/appCruise	35 Line contains a tab character.
CarSimulator.java	/SWE326P1/src/appCruise	36 Line contains a tab character.
CarSimulator.java	/SWE326P1/src/appCruise	37 Line contains a tab character.
CarSimulator.java	/SWE326P1/src/appCruise	38 Line contains a tab character.
CarSimulator.java	/SWE326P1/src/appCruise	39 Line contains a tab character.
CarSimulator.java	/SWE326P1/src/appCruise	40 Line contains a tab character.
CarSimulator.java	/SWE326P1/src/appCruise	41 Line contains a tab character.
CarSimulator.java	/SWE326P1/src/appCruise	... Line contains a tab character.
CruiseControl.java	/SWE326P1/src/appCruise	26 Line contains a tab character.
CruiseControl.java	/SWE326P1/src/appCruise	27 Line contains a tab character.
CruiseControl.java	/SWE326P1/src/appCruise	28 Line contains a tab character.
CruiseControl.java	/SWE326P1/src/appCruise	35 Line contains a tab character.
CruiseControl.java	/SWE326P1/src/appCruise	42 Line contains a tab character.
CruiseControl.java	/SWE326P1/src/appCruise	50 Line contains a tab character.
CruiseControl.java	/SWE326P1/src/appCruise	58 Line contains a tab character.
CruiseControl.java	/SWE326P1/src/appCruise	66 Line contains a tab character.
CruiseControl.java	/SWE326P1/src/appCruise	74 Line contains a tab character.
CruiseControl.java	/SWE326P1/src/appCruise	81 Line contains a tab character.
CruiseControl.java	/SWE326P1/src/appCruise	88 Line contains a tab character.
CruiseControl.java	/SWE326P1/src/appCruise	95 Line contains a tab character.
CruiseDisplay.java	/SWE326P1/src/appCruise	30 Line contains a tab character.
CruiseDisplay.java	/SWE326P1/src/appCruise	31 Line contains a tab character.
CruiseDisplay.java	/SWE326P1/src/appCruise	32 Line contains a tab character.
CruiseDisplay.java	/SWE326P1/src/appCruise	26 Line contains a tab character.
CruiseDisplay.java	/SWE326P1/src/appCruise	27 Line contains a tab character.
CruiseDisplay.java	/SWE326P1/src/appCruise	28 Line contains a tab character.
CruiseDisplay.java	/SWE326P1/src/appCruise	29 Line contains a tab character.
SpeedControl.java	/SWE326P1/src/appCruise	28 Line contains a tab character.
SpeedControl.java	/SWE326P1/src/appCruise	41 Line contains a tab character.
SpeedControl.java	/SWE326P1/src/appCruise	42 Line contains a tab character.
SpeedControl.java	/SWE326P1/src/appCruise	43 Line contains a tab character.
SpeedControl.java	/SWE326P1/src/appCruise	44 Line contains a tab character.
SpeedControl.java	/SWE326P1/src/appCruise	45 Line contains a tab character.

Resource	In Folder	Line	Message
CarSimulator.java	/SWE326P1/src/appCruise	...	Array brackets at illegal position.

## SWE-326 Master Test Plan

---

Resource	In Folder	Line Message
§ CruiseControl.java	/SWE326P1/src/appCruise	49 'object def rcurl' has incorrect indentation level 9, expected level should be one of the following: 8, 10, 12.
§ CruiseControl.java	/SWE326P1/src/appCruise	57 'object def rcurl' has incorrect indentation level 9, expected level should be one of the following: 8, 10, 12.
§ CruiseControl.java	/SWE326P1/src/appCruise	64 'method def rcurl' has incorrect indentation level 13, expected level should be one of the following: 10, 12, 14.
§ CruiseControl.java	/SWE326P1/src/appCruise	65 'object def rcurl' has incorrect indentation level 9, expected level should be one of the following: 8, 10, 12.
§ CruiseControl.java	/SWE326P1/src/appCruise	72 'method def rcurl' has incorrect indentation level 13, expected level should be one of the following: 10, 12, 14.
§ CruiseControl.java	/SWE326P1/src/appCruise	73 'object def rcurl' has incorrect indentation level 9, expected level should be one of the following: 8, 10, 12.
§ CruiseControl.java	/SWE326P1/src/appCruise	80 'object def rcurl' has incorrect indentation level 9, expected level should be one of the following: 8, 10, 12.
§ CruiseControl.java	/SWE326P1/src/appCruise	87 'object def rcurl' has incorrect indentation level 9, expected level should be one of the following: 8, 10, 12.
§ CruiseControl.java	/SWE326P1/src/appCruise	94 'object def rcurl' has incorrect indentation level 9, expected level should be one of the following: 8, 10, 12.

Resource	In Folder	Line Message
§ CarSimulator.java	/SWE326P1/src/appCruise	8 'private' modifier out of order with the JLS suggestions.
§ CarSimulator.java	/SWE326P1/src/appCruise	9 'private' modifier out of order with the JLS suggestions.
§ CarSimulator.java	/SWE326P1/src/appCruise	10 'private' modifier out of order with the JLS suggestions.
§ CarSimulator.java	/SWE326P1/src/appCruise	11 'private' modifier out of order with the JLS suggestions.
§ CarSimulator.java	/SWE326P1/src/appCruise	12 'private' modifier out of order with the JLS suggestions.
§ CarSimulator.java	/SWE326P1/src/appCruise	15 'static' modifier out of order with the JLS suggestions.
§ CarSimulator.java	/SWE326P1/src/appCruise	16 'static' modifier out of order with the JLS suggestions.
§ CarSimulator.java	/SWE326P1/src/appCruise	17 'static' modifier out of order with the JLS suggestions.
§ CarSimulator.java	/SWE326P1/src/appCruise	18 'static' modifier out of order with the JLS suggestions.
§ CarSimulator.java	/SWE326P1/src/appCruise	19 'static' modifier out of order with the JLS suggestions.
§ CarSimulator.java	/SWE326P1/src/appCruise	20 'static' modifier out of order with the JLS suggestions.
§ CarSimulator.java	/SWE326P1/src/appCruise	21 'static' modifier out of order with the JLS suggestions.
§ Controller.java	/SWE326P1/src/appCruise	4 'static' modifier out of order with the JLS suggestions.
§ Controller.java	/SWE326P1/src/appCruise	5 'static' modifier out of order with the JLS suggestions.
§ Controller.java	/SWE326P1/src/appCruise	6 'static' modifier out of order with the JLS suggestions.
§ Controller.java	/SWE326P1/src/appCruise	7 'static' modifier out of order with the JLS suggestions.
§ CruiseDisplay.java	/SWE326P1/src/appCruise	10 'static' modifier out of order with the JLS suggestions.
§ SpeedControl.java	/SWE326P1/src/appCruise	39 'public' modifier out of order with the JLS suggestions.
§ SpeedControl.java	/SWE326P1/src/appCruise	5 'static' modifier out of order with the JLS suggestions.
§ SpeedControl.java	/SWE326P1/src/appCruise	6 'static' modifier out of order with the JLS suggestions.
§ SpeedControl.java	/SWE326P1/src/appCruise	7 'private' modifier out of order with the JLS suggestions.
§ SpeedControl.java	/SWE326P1/src/appCruise	8 'private' modifier out of order with the JLS suggestions.
§ SpeedControl.java	/SWE326P1/src/appCruise	9 'private' modifier out of order with the JLS suggestions.
§ SpeedControl.java	/SWE326P1/src/appCruise	10 'private' modifier out of order with the JLS suggestions.
§ SpeedControl.java	/SWE326P1/src/appCruise	11 'private' modifier out of order with the JLS suggestions.

Resource	In Folder	Line Message
§ CarSimulator.java	/SWE326P1/src/appCruise	... Missing a Javadoc comment.
§ CarSimulator.java	/SWE326P1/src/appCruise	33 Missing a Javadoc comment.
§ CarSimulator.java	/SWE326P1/src/appCruise	50 Missing a Javadoc comment.
§ CarSimulator.java	/SWE326P1/src/appCruise	... Missing a Javadoc comment.
§ CarSimulator.java	/SWE326P1/src/appCruise	... Missing a Javadoc comment.
§ CarSimulator.java	/SWE326P1/src/appCruise	... Missing a Javadoc comment.
§ CarSimulator.java	/SWE326P1/src/appCruise	... Missing a Javadoc comment.
§ CarSimulator.java	/SWE326P1/src/appCruise	... Missing a Javadoc comment.
§ CruiseControl.java	/SWE326P1/src/appCruise	24 Missing a Javadoc comment.
§ CruiseControl.java	/SWE326P1/src/appCruise	32 Missing a Javadoc comment.
§ CruiseDisplay.java	/SWE326P1/src/appCruise	45 Missing a Javadoc comment.
§ CruiseDisplay.java	/SWE326P1/src/appCruise	65 Missing a Javadoc comment.
§ CruiseDisplay.java	/SWE326P1/src/appCruise	24 Missing a Javadoc comment.

Resource	In Folder	Line Message
§ CarSimulator.java	/SWE326P1/src/appCruise	79 Comment has incorrect indentation level 7, expected is 8, indentation should be the same level as line 80.
§ CruiseControl.java	/SWE326P1/src/appCruise	34 Comment has incorrect indentation level 8, expected is 5, indentation should be the same level as line 35.

### 3. System Test

#### 3.1. System Requirements

Requirements	Description
R1	To activate Cruise Control, the engine should be turned on.
R2	The activation of the CC will hold the vehicle speed at the selected value.
R3	When the CC is enabled, accelerating the vehicle, will disable the CC.
R4	When the CC is enabled, braking will disable the CC.
R5	The user can turn the engine off anytime.
R6	When CC is enabled, the CC GUI should store and display the correct speed.
R7	Pressing the resume button should set the CC to last fixed speed.
R8	When we turn the engine on, the CC should be off.
R9	When the engine is turned off, the CC should be turned off.
R10	The odometer should show the correct distance when the CC is enabled/disabled.
R11	When the engine is off, clicking on the CC on/off should have no impact.
R12	System should display the Cruise control status (Green for on, Red for off).
R13	When the CC is turned on, clicking on the button “on” should have no impact.
R14	When the CC is turned off, clicking on the button “off” should have no impact.
R15	The CC cannot be set for a speed less than 40km/h.
R16	The user should be able to run turn the CC on and off with no limit.
R17	When the engine is turned off then on, the CC should be turned off and showing the initial speed as zero.

## 3.2. System Test Cases

Design your test cases according to the following test case template

Test Case Title	Activation of the Cruise Control while the engine is off.
Test Case ID	TC1-1
Requirement IDs	R1
Author	Mohammad Eskandarani
Description	This test case ensures that the Cruise Control cannot be turned on while the engine is turned off.
Initial State	The Cruise Control is turned off.
Pre-conditions (if applicable)	The engine should be turned off.
Test steps (inputs, actions, etc.)	Input: press the on button. Action: see the cruise status.
Expected Output	The Cruise Control is off.
Priority (High/Medium/Low)	Medium
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	The holding of the speed while the speed is 20 km/h.
Test Case ID	TC2-1
Requirement IDs	R2
Author	Mohammad Eskandarani
Description	Testing the Cruise Control while the car is moving at a speed of 20 km/h.
Initial State	The Cruise Control is turned off.

Pre-conditions (if applicable)	The engine is turned on. The speed is 20 km/h.
Test steps (inputs, actions, etc.)	Accelerate the car until it reaches a speed of 20 km/h. Press the on button. See the speed of the car.
Expected Output	The speed remains 20 km/h.
Priority (High/Medium/Low)	High
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	The holding of the speed while the speed is 70 km/h.
Test Case ID	TC2-2
Requirement IDs	R2
Author	Mohammad Eskandarani
Description	Testing the Cruise Control while the car is moving at a speed of 70 km/h.
Initial State	The Cruise Control is turned off.
Pre-conditions (if applicable)	The engine is turned on. The speed is 70 km/h.
Test steps (inputs, actions, etc.)	Accelerate the car until it reaches a speed of 70 km/h. Press the on button. See the speed of the car.
Expected Output	The speed remains 70 km/h.
Priority (High/Medium/Low)	High
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	The acceleration of the car and the Cruise Control is turned on.
Test Case ID	TC3-1
Requirement IDs	R3
Author	Mohammad Eskandarani
Description	Testing whether the Cruise Control is turned off when pressing the accelerate button or not.
Initial State	The Cruise Control is turned on.
Pre-conditions (if applicable)	Not applicable.
Test steps (inputs, actions, etc.)	Accelerate the car to a specific speed, turn the Cruise Control on, then press the acceleration button again
Expected Output	The Cruise Control is turned off.
Priority (High/Medium/Low)	High
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	The breaking of the car and the Cruise Control is turned on.
Test Case ID	TC4-1
Requirement IDs	R4
Author	Mohammad Eskandarani
Description	Testing whether the Cruise Control is turned off when pressing the brake button or not.
Initial State	The Cruise Control is turned on.
Pre-conditions (if applicable)	Not applicable.
Test steps (inputs, actions, etc.)	Accelerate the car to a specific speed, turn the Cruise Control on, then press the brake button.
Expected Output	The Cruise Control is turned off.
Priority (High/Medium/Low)	High

Test Type (Functional, Performance, stress, etc.)	Functional
---	------------

Test Case Title	Turning off the engine when the car is not moving.
Test Case ID	TC5-1
Requirement IDs	R5
Author	Mohammad Eskandarani
Description	Testing the turning off the engine when the car is not moving.
Initial State	The engine is turned on.
Pre-conditions (if applicable)	The car is not moving.
Test steps (inputs, actions, etc.)	Turn the engine on, then turn the engine off.
Expected Output	The engine is turned off.
Priority (High/Medium/Low)	High
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	Turning off the engine when the car is moving, and the Cruise control is turned off.
Test Case ID	TC5-2
Requirement IDs	R5
Author	Mohammad Eskandarani
Description	Testing the turning off the engine when the car is moving, and the Cruise Control is turned off.
Initial State	The engine is turned on.
Pre-conditions (if applicable)	The car is moving. The Cruise Control is turned off.

Test steps (inputs, actions, etc.)	Turn the engine on, accelerate the car, then turn the engine off.
Expected Output	The engine is turned off.
Priority (High/Medium/Low)	High
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	Turning off the engine when the car is moving, and the Cruise control is turned on.
Test Case ID	TC5-3
Requirement IDs	R5
Author	Mohammad Eskandarani
Description	Testing the turning off the engine when the car is moving, and the Cruise Control is turned on.
Initial State	The engine is turned on.
Pre-conditions (if applicable)	The car is moving. The Cruise control is turned on.
Test steps (inputs, actions, etc.)	Turn the engine on, accelerate the car, turn the Cruise Control on, then turn the engine off.
Expected Output	The engine is turned off.
Priority (High/Medium/Low)	High
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	GUI of the Cruise Control at a speed of 50 km/h.
Test Case ID	TC6-1
Requirement IDs	R6
Author	Haitham Alsaeed

Description	Testing the value displayed in the GUI of the Cruise Control.
Initial State	The value in the GUI is 0.
Pre-conditions (if applicable)	
Test steps (inputs, actions, etc.)	Accelerate the car until it reaches 50 km/h, then enable the Cruise Control.
Expected Output	The GUI displays 50 km/h
Priority (High/Medium/Low)	Medium
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	Resume button of the Cruise Control.
Test Case ID	TC7-1
Requirement IDs	R7
Author	Haitham Alsaeed
Description	Testing whether the resume button restores the last fixed speed stored by the Cruise Control or not.
Initial State	The car is moving.
Pre-conditions (if applicable)	The engine is on.
Test steps (inputs, actions, etc.)	Enable the Cruise Control when the car is moving, then disable it, then press the resume button.
Expected Output	The Cruise Control is set to last fixed speed.
Priority (High/Medium/Low)	Low
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	The state of the Cruise Control when turning the engine on.
-----------------	---

Test Case ID	TC8-1
Requirement IDs	R8
Author	Haitham Alsaeed
Description	Testing whether the Cruise Control is turning off state when turn the engine on or not.
Initial State	The Cruise Control is turned off.
Pre-conditions (if applicable)	The engine is turned off.
Test steps (inputs, actions, etc.)	Turn the engine on, then see the state of the Cruise Control.
Expected Output	The Cruise Control is off
Priority (High/Medium/Low)	High
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	The state of the Cruise Control after turning the engine off.
Test Case ID	TC9-1
Requirement IDs	R9
Author	Haitham Alsaeed
Description	Testing the state of the Cruise Control after turning the engine off.
Initial State	The Cruise Control is on
Pre-conditions (if applicable)	The engine is on
Test steps (inputs, actions, etc.)	Enable the Cruise Control, then turn the engine off.
Expected Output	The Cruise Control is turned off.
Priority (High/Medium/Low)	High
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	Odometer when CC is disabled and car is moving
Test Case ID	TC10-1
Requirement IDs	R10
Author	Moaz Khan
Description	We are testing the accuracy of the odometer.
Initial State	Car is not moving
Pre-conditions (if applicable)	Engine is on
Test steps (inputs, actions, etc.)	Make the car move by accelerating. Monitor the odometer and calculate if it's accurate based on the speed and time.
Expected Output	Odometer increases by one for every kilometer.
Priority (High/Medium/Low)	Low
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	Odometer when CC is disabled and car stops
Test Case ID	TC10-2
Requirement IDs	R10
Author	Moaz Khan
Description	We are testing if the odometer stops when the car stops.
Initial State	Car is moving
Pre-conditions (if applicable)	Engine is on
Test steps (inputs, actions, etc.)	Make the car move by accelerating. Then make the car stop by braking. Monitor the odometer for a few seconds.
Expected Output	Odometer should stay the same.
Priority (High/Medium/Low)	Low
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	Odometer when CC is enabled
Test Case ID	TC10-3
Requirement IDs	R10
Author	Moaz Khan
Description	We are testing the accuracy of the odometer when the CC is enabled.
Initial State	Car is moving.
Pre-conditions (if applicable)	Engine is on and the CC is off.
Test steps (inputs, actions, etc.)	Make the car move by accelerating. Then turn on the CC. Monitor the odometer and calculate if it's accurate based on the speed and time.
Expected Output	Odometer increases by one for every kilometer.
Priority (High/Medium/Low)	Low
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	CC on button while the engine is off.
Test Case ID	TC11-1
Requirement IDs	R11
Author	Moaz Khan
Description	We are seeing if the CC on button does anything if the engine is off.
Initial State	Engine is off.
Pre-conditions (if applicable)	
Test steps (inputs, actions, etc.)	Click the CC on button.
Expected Output	Nothing happens.
Priority (High/Medium/Low)	Medium

Test Type (Functional, Performance, stress, etc.)	Functional
---	------------

Test Case Title	CC off button while the engine is off.
Test Case ID	TC11-2
Requirement IDs	R11
Author	Moaz Khan
Description	We are seeing if the CC off button does anything if the engine is off.
Initial State	Engine is off.
Pre-conditions (if applicable)	
Test steps (inputs, actions, etc.)	Click the CC off button.
Expected Output	Nothing happens.
Priority (High/Medium/Low)	Medium
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	CC enabled status
Test Case ID	TC12-1
Requirement IDs	R12
Author	Moaz Khan
Description	The CC should show that it is enabled when it is and a green light.
Initial State	CC is disabled
Pre-conditions (if applicable)	Engine is on
Test steps (inputs, actions, etc.)	Accelerate to above 40km/h. Click the CC on button.
Expected Output	CC status shows "Enabled" and the light turns green.
Priority (High/Medium/Low)	High

Test Type (Functional, Performance, stress, etc.)	Functional
---	------------

Test Case Title	CC disabled status
Test Case ID	TC12-2
Requirement IDs	R12
Author	Moaz Khan
Description	The CC should show that it is disabled when it is and a red light.
Initial State	CC is enabled
Pre-conditions (if applicable)	Engine is on
Test steps (inputs, actions, etc.)	Click the CC off button.
Expected Output	CC status shows “Disabled” and the light turns red.
Priority (High/Medium/Low)	High
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	Impact of clicking on while CC is on
Test Case ID	TC13
Requirement IDs	R13
Author	Moaz Khan
Description	There should be no impact of clicking the CC’s on button if it’s already on.
Initial State	CC is enabled
Pre-conditions (if applicable)	Engine is on
Test steps (inputs, actions, etc.)	Click the CC on button.
Expected Output	Nothing happens.
Priority (High/Medium/Low)	High

Test Type (Functional, Performance, stress, etc.)	Functional
---	------------

Test Case Title	No impact on clicking turn off if the CC is turned off.
Test Case ID	TC14
Requirement IDs	R14
Author	Ahmed Almoairfi
Description	Testing Off button when the CC is turn off. Then Off button should have no impact.
Initial State	The engine is turn on.
Pre-conditions (if applicable)	The Cruise Control is turned off.
Test steps (inputs, actions, etc.)	Pressing off button.
Expected Output	There should be no impact
Priority (High/Medium/Low)	High
Test Type (Functional, Performance, stress, etc.)	Functional

Test Case Title	The Can't maintain the speed if it less than 40Km/h
Test Case ID	TC15-1
Requirement IDs	R15
Author	Ahmed Almoairfi
Description	Testing setting the speed when the speed is 39Km/h.
Initial State	The engine is turn on.

Pre-conditions (if applicable)	The car is moving. the brake pedal is not depressed.
Test steps (inputs, actions, etc.)	Accelerate the speed of the car to be 39Km/h. Turn of the Cruise Control.
Expected Output	The speed should not be set.
Priority (High/Medium/Low)	Medium
Test Type (Functional, Performance, stress, etc.)	Performance

Test Case Title	The Can't maintain the speed if it less than 40Km/h
Test Case ID	TC15-2
Requirement IDs	R15
Author	Ahmed Almoairfi
Description	Testing setting the speed when the speed is 29Km/h.
Initial State	The engine is turn on.
Pre-conditions (if applicable)	The car is moving. the brake pedal is not depressed.
Test steps (inputs, actions, etc.)	Accelerate the speed of the car to be 29Km/h. Turn of the Cruise Control.
Expected Output	The speed should not be set.
Priority (High/Medium/Low)	Medium
Test Type (Functional, Performance, stress, etc.)	Performance

Test Case Title	Testing the quality of CC turn buttons
-----------------	--

Test Case ID	TC16
Requirement IDs	R16
Author	Ahmed Almoairfi
Description	Turning the CC on and off for 100 times continually
Initial State	The engine is turn on.
Pre-conditions (if applicable)	The Cruise Control is turned off
Test steps (inputs, actions, etc.)	Turn CC on and off for 100 time continually.
Expected Output	The CC should work fine at the end.
Priority (High/Medium/Low)	High
Test Type (Functional, Performance, stress, etc.)	stress

Test Case Title	CC should be off and the speed is zero when the engine is turn off then on
Test Case ID	TC17
Requirement IDs	R17
Author	Ahmed Almoairfi
Description	Testing when the engine is turned off then the speed should be zero and CC off when the engine is turned on again.
Initial State	The engine is turned on.
Pre-conditions (if applicable)	The car in moving. The Cruise Control is turned on.
Test steps (inputs, actions, etc.)	Turn off the engine. Turn on the engine

Expected Output	The CC should be off. The car speed should be zero
Priority (High/Medium/Low)	High
Test Type (Functional, Performance, stress, etc.)	Functional

### 3.3. Report on the Execution of System Test Cases

Fill in the Test Execution Report Template:

TC ID	Verdict (Pass/Fail/ Pass with exception)	Comments (if fail or pass with exception (Actual output vs.))	Bug ID	Execution Date	Author	Engineer
TC1-1	Pass	-	-	4/3/2020	Mohammad Eskandarani	
TC2-1	Pass	-	-	4/3/2020	Mohammad Eskandarani	
TC2-2	Pass	-	-	4/3/2020	Mohammad Eskandarani	
TC3-1	Pass	-	-	4/3/2020	Mohammad Eskandarani	
TC4-1	Fail	The Cruise Control was not turned off.	B1	4/3/2020	Mohammad Eskandarani	
TC5-1	Pass	-	-	4/3/2020	Mohammad Eskandarani	
TC5-2	Pass	-	-	4/3/2020	Mohammad Eskandarani	
TC5-3	Pass	-	-	4/3/2020	Mohammad Eskandarani	
TC6-1	Fail	The value displayed by the GUI is wrong	B4	4/4/2020	Haitham Alsaeed	
TC7-1	Pass	-	-	4/4/2020	Haitham	

					Alsaeed	
TC8-1	Pass	-	-	4/4/2020	Haitham Alsaeed	
TC9-1	Pass	-	-	4/4/2020	Haitham Alsaeed	
TC10-1	Fail	Odometer is too fast.	B2	4/4/2020	Moaz Khan	
TC10-2	Pass	-	-	4/4/2020	Moaz Khan	
TC10-3	Fail	Odometer is too fast.	B2	4/4/2020	Moaz Khan	
TC11-1	Pass	-	-	4/4/2020	Moaz Khan	
TC11-2	Pass	-	-	4/4/2020	Moaz Khan	
TC12-1	Pass	-	-	4/4/2020	Moaz Khan	
TC12-2	Pass	-	-	4/4/2020	Moaz Khan	
TC13	Pass with Exception	Clicking “on” sometimes resets the CC.	B3	4/4/2020	Moaz Khan	
TC14	pass	-	-	4/4/2020	Ahmed Almoairfi	
TC15-1	Fail	The speed is maintain	B5	4/4/2020	Ahmed Almoairfi	
TC15-2	Fail	The speed is maintain	B5	4/4/2020	Ahmed Almoairfi	
TC16	pass	-	-	4/4/2020	Ahmed Almoairfi	
TC17	pass	-	-	4/4/2020	Ahmed Almoairfi	

**Bug List**

Bug ID	Test Case ID	Detailed Description	Severity (High/Medium/Low)	Date found	Submitter
B1	TC4-1	The Cruise control is not turned off when pressing the break.	High	4/3/2020	Mohammad Eskandarani
B2	TC10-1/TC10-3	The odometer is going too fast based on the speed scale specified in the requirements.	Low	4/4/2020	Moaz Khan
B3	TC13	In the moment where the speed decreases before the CC can adjust it, clicking “on” resets the CC to the lower speed.	Low	4/4/2020	Moaz Khan
B4	TC6-1	The value displayed by the GUI is more than the actual speed by 20.	Medium	4/4/2020	Haitham Alsaeed
B5	TC15-1/TC15-2	The speed should not be maintain when the speed is less than 40Km/h.	Low	4/4/2020	Ahmed

## 4. Unit Testing

Fill in this section in Phase 3.

This section should contain:

- The code for the unit test cases

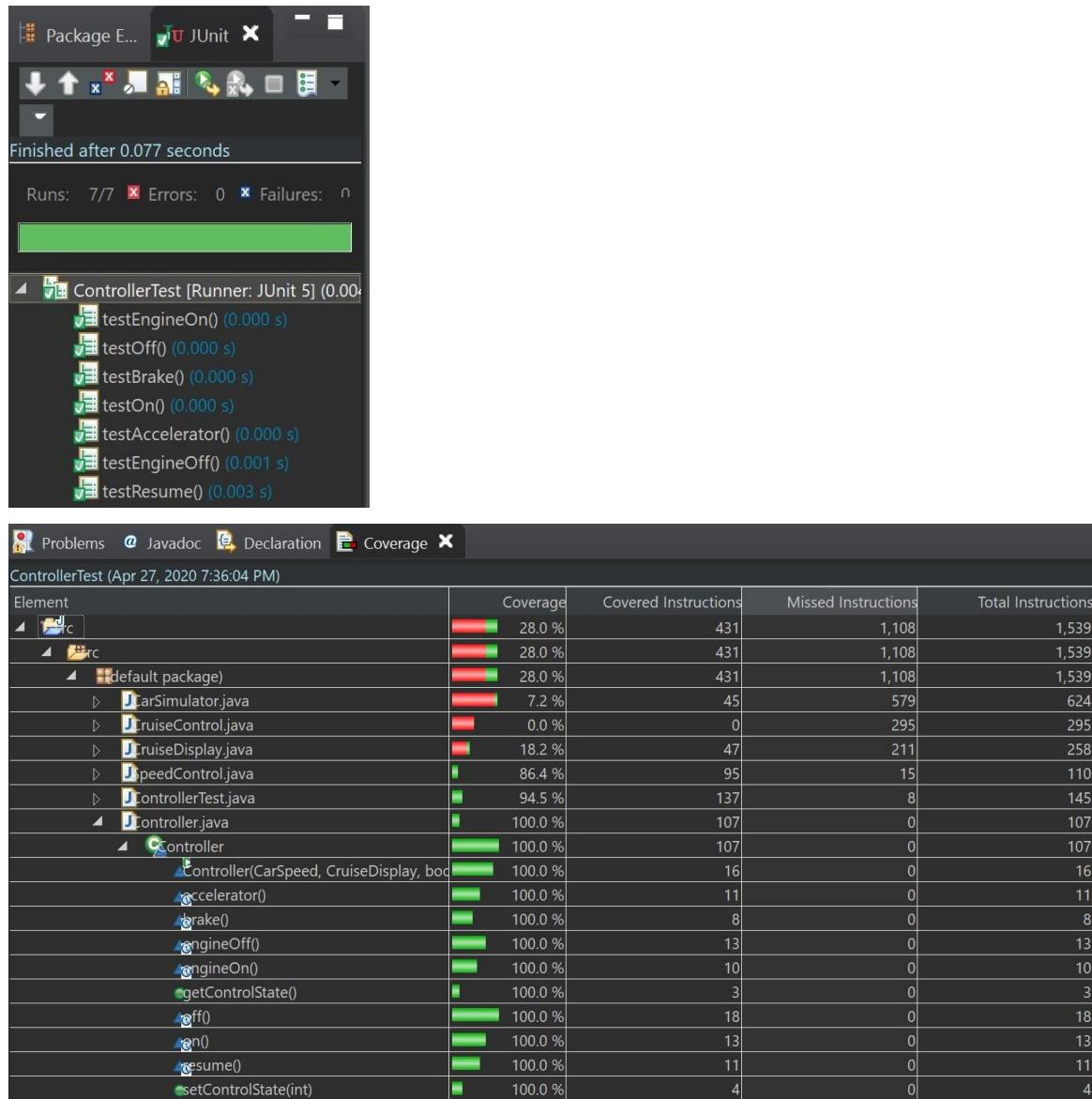
A summary of execution of the unit test cases (Screen shots showing the execution results and achieved coverage (using codecov tool: <http://codecov.org/>)).

Controller.java test case:

```

1  /*
2   * To correctly test the class Controller.java we added a setter and getter
3   * for the variables 'controlState'.
4   */
5  import static org.junit.jupiter.api.Assertions.*;
6
7  import org.junit.jupiter.api.BeforeEach;
8  import org.junit.jupiter.api.Test;
9
10 class ControllerTest extends junit.framework.TestCase {
11
12     Controller c;
13
14     @BeforeEach
15     public void setup() throws Exception {
16         c = new Controller(new Partimulator(), new CrisistDisplay(), true);
17     }
18
19     @Test
20     void testBrake() {
21         c.setControlState(Controller.CRUISING);
22         c.brake();
23         assertTrue(c.getControlState() == Controller.STANDBY);
24     }
25
26     @Test
27     void testAccelerator() {
28         c.setControlState(Controller.IDLE);
29         c.accelerator();
30         assertEquals(c.getControlState() == Controller.STANDBY);
31     }
32
33     @Test
34     void testEngineOff() {
35         c.setControlState(Controller.ACTIVE);
36         c.engineOff();
37         assertEquals(c.getControlState() == Controller.INACTIVE);
38     }
39
40     @Test
41     void testEngineOn() {
42         c.setControlState(Controller.INACTIVE);
43         c.engineOn();
44         assertEquals(c.getControlState() == Controller.ACTIVE);
45     }
46
47     @Test
48     void testOn() {
49         c.setControlState(Controller.ACTIVE);
50         c.on();
51         assertEquals(c.getControlState() == Controller.CRUISING);
52     }
53
54     @Test
55     void testOff() {
56         c.setControlState(Controller.CRUISING);
57         c.off();
58         assertEquals(c.getControlState() == Controller.STANDBY);
59         c.off();
60         assertEquals(c.getControlState() == Controller.INACTIVE);
61     }
62
63     @Test
64     void testResume() {
65         c.setControlState(Controller.STANDBY);
66         c.resume();
67         assertEquals(c.getControlState() == Controller.CRUISING);
68     }
69
70 }
71

```



## CarSimulator.java test case:

```

import org.junit.Ignore;
import org.junit.jupiter.api.Test;

class CarSimulatorTest {

    @Test
    void testAccelerate() {
        CarSimulator carS=new CarSimulator();
        double prevThrottle=carS.getThrottle();
        carS.accelerate();

        assertEquals(0, carS.getBrakepedal());
        assertEquals(prevThrottle+1,carS.getThrottle(),0.1);

        for(int i=0; i<carS.getMaxThrottle()+3; i++) {
            carS.accelerate();
        }
        assertEquals(carS.getMaxThrottle(),carS.getThrottle(),0.1);
    }

    @Test
    void testBrake() {
        CarSimulator carS=new CarSimulator();
        carS.setThrottle(10);
        carS.brake();
        assertEquals(0,carS.getThrottle(),0.1);
    }

    @Test
    void testRun() throws InterruptedException {
        CarSimulator carS=new CarSimulator();
        carS.engineOn();
        Thread.sleep(1000/CarSimulator.ticksPerSecond);
        carS.run();
        assertEquals(0, (carS.getThrottle() /CarSimulator.airResistance - 2*carS.getBrakepedal()) /CarSimulator.ticksPerSecond,0.1);
        assertEquals(0, (carS.getSpeed()/36.0)/CarSimulator.ticksPerSecond,0.1);
    }

    @Test
    void testSetThrottle() {
        CarSimulator carS=new CarSimulator();
        carS.engineOn();
        carS.setThrottle(-10);
        assertEquals(0, carS.getThrottle(),0.1);

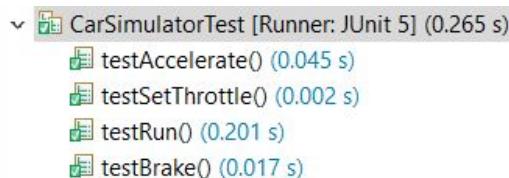
        carS.setThrottle(0);
        assertEquals(0, carS.getThrottle(),0.1);

        carS.setThrottle(10);
        assertEquals(10, carS.getThrottle(),0.1);

        carS.setThrottle(100);
        assertEquals(100, carS.getThrottle(),0.1);

        carS.setThrottle(5);
        assertEquals(5, carS.getThrottle(),0.1);
    }
}

```



CarSimulator.java	18.4 %	116	513	629
CarSimulator	18.4 %	116	513	629
drawOdo(Graphics, int)	0.0 %	0	87	87
run()	9.8 %	9	83	92
drawSpeedometer(Graphics)	0.0 %	0	80	80
update(Graphics)	0.0 %	0	73	73
drawMark(Graphics, int)	0.0 %	0	67	67
backdrop()	0.0 %	0	59	59
drawControl(Graphics, int)	0.0 %	0	34	34
brake()	54.5 %	12	10	22
engineOff()	0.0 %	0	9	9
accelerate()	84.0 %	21	4	25
paint(Graphics)	0.0 %	0	4	4
getSpeed()	0.0 %	0	3	3
CarSimulator()	100.0 %	25	0	25
engineOn()	100.0 %	18	0	18
getBrakepedal()	100.0 %	3	0	3
getMaxThrottle()	100.0 %	2	0	2
getThrottle()	100.0 %	3	0	3
setThrottle(double)	100.0 %	23	0	23

SpeedControl.java test case:

```
1 import static org.junit.jupiter.api.Assertions.*;
2
3 import org.junit.jupiter.api.BeforeEach;
4 import org.junit.jupiter.api.Test;
5
6 class SpeedControlTest extends junit.framework.TestCase {
7
8     SpeedControl sc;
9     CarSimulator cs;
10
11    @BeforeEach
12    public void setUp() throws Exception {
13        cs = new CarSimulator();
14        sc = new SpeedControl(cs, new CruiseDisplay());
15    }
16
17    @Test
18    void testRecordSpeed() {
19        sc.recordSpeed();
20        assertTrue(sc.getSetSpeed() == cs.getSpeed());
21    }
22
23    @Test
24    void testClearSpeed() {
25        sc.setState(SpeedControl.DISABLED);
26        sc.clearSpeed();
27        assertTrue(sc.getSetSpeed() == 10);
28    }
29
30    @Test
31    void testEnableControl() {
32        sc.setState(SpeedControl.DISABLED);
33        sc.enableControl();
34        assertTrue(sc.getState() == SpeedControl.ENABLED);
35    }
36
37    @Test
38    void testDisableControl() {
39        sc.setState(SpeedControl.ENABLED);
40        sc.disableControl();
41        assertTrue(sc.getState() == SpeedControl.DISABLED);
42    }
43
44    @Test
45    void testRun() {
46        sc.setState(SpeedControl.ENABLED);
47        sc.run();
48        assertTrue(sc.getError() == (sc.getSetSpeed()-cs.getSpeed())/6.0);
49        assertTrue(sc.getSteady() == sc.getSetSpeed()/12.0);
50    }
51
52 }
53 }
```

## 5. Conclusion

The results of the static testing show that the source code had some issues with consistency, code organization and a few syntax errors. These issues can be resolved by using the refactoring features of any Java IDE. The system testing stage revealed 5 bugs that were inconsistent with the requirements of the project. However, only one was “High” severity. Unit tests were done using JUnit. Setters and getters for private variables were added to correctly test all the methods. GUI elements were not tested as that is part of integration testing. Unit tests did not show any errors.