

# **UNIVERSITY OF CENTRAL LANCASHIRE**

**School of Engineering**

**EL2311 SOFTWARE DEVELOPMENT 2  
ASSIGNMENT, 2018-19**

## **Data Storage and Retrieval in SQLite Database using C#**

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

# ABSTRACT

The notion of transferring data into relational SQLite databases via XML files is gaining popularity. The report deals with the documentation of the development process of the ‘Autonomous Vehicle Configuration Details (AVCD) App’ which is a software solution to allow an organisation to store configuration details of vehicles in XML format and transfer the data into the given relational database so that the users with different access levels can retrieve information from the database for analysis.

# TABLE OF CONTENT

1. Introduction.....	3-4
1.1 Purpose.....	3
1.2 Customer Statement of Requirement .....	3
1.3 Definitions, acronyms and abbreviations.....	4
2. System Requirements.....	5 -7
2.1 Enumerated Functional Requirements.....	5-6
2.2 Enumerated Non- Functional Requirements.....	7
3. Development Description.....	8-16
3.1 Parsing XML file using XmlDocument.....	8-9
3.2 Loading parsed XML data into relational SQLite database.....	10-12
3.3 User Interface Design and Implementation.....	13-16
4. Design of Tests .....	17-21
5. Discussion .....	21
6. Conclusion and future work .....	21
7. References .....	22
8. Appendix.....	23-26

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

# 1. INTRODUCTION

This section gives a scope description and overview of everything included in this report. Besides, the purpose for this document is described and a list of abbreviations and definitions is provided.

## 1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the 'Autonomous Vehicle Configuration Details (AVCD) App'. It will illustrate the requirements, test plans and implementations in the development of the system.

## 1.2 Customer Statement of Requirements

### (a.) Problem Statement

The organisation requires a software application to allow permanent storage of different configuration details of the autonomous vehicles which are in XML format to the SQLite Database and then allow users to retrieve the stored information to perform analysis and critical comparison of the vehicle specifications using data queries. The AVCD App is a Windows Form App (.NET Framework) created in C# to allow two types of users (expert user and novice user) to interact with the application and to get search results for queries run on the 'courseworkdb' database. Each vehicle can be specified in terms of engine (Petrol/Electric), with unique ID, range (in metres), weight (in kilograms), a carry capacity (in kilograms) and top speed (in metres/second). Besides a petrol engine has additional specifications for noise level (in decibels) and number of cylinders. Whereas, electric engine has specification details stating whether the motor is brushless (indicated as 0) or brushed (indicated as 1) and engine voltage (in volts).

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

### 1.3 Definitions, Acronyms and Abbreviations

TERM	DEFINITION
Stakeholder	Any person who has interaction with the system and is not a developer.
Expert User	Someone who has access to run custom queries on the database. The expert user cannot edit, update, delete or change the database by typing a custom SQLite query in the textbox.
Novice User	Someone who has access to run only basic queries from the available menu on the interface.
AVCD App	Autonomous Vehicle Configuration Details Application
Database	A physical storage Entity with tables/columns containing data.
Foreign Key	A Foreign Key is a column in a database table whose values match a Primary Key in a different table to link those two tables together.
XML file	An electronic communication message format that contains data.
GUI	Graphical User Interface.

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Date: 7-4-2019

## 2. System Requirements

### 2.1 Enumerated Functional Requirements

The functional requirements are tabulated as follows:

REQ1(a)	The system should allow to load and parse XML files containing the configuration details of vehicles into given three tables in the 'courseworkdb' SQLite database using user interface.
REQ1(b)	The system application must not crash under any circumstance and be able to handle invalid XML files loaded into the system.
REQ2	The application should allow permanent storage of the configuration details of the vehicles each with unique vehicle ID into given database. That is, overwriting configuration details for a vehicle with same vehicle ID in the database file should not be permitted in the system.
REQ3(a)	The system application should allow the expert user to run custom SQL queries on the database and have the search results displayed in a 'txtResult' textbox.
REQ3(b)	The expert user must not be allowed to insert, edit, update, delete or overwrite configuration details of vehicles in the database.
REQ3(c)	The application should return sensible errors in the 'txtResult' textbox when invalid SQL query is typed in the 'CustomQueryBox' textbox.
REQ4(a)	The system application should allow the novice user to run basic search queries on the database without typing in any SQL queries using menu driven interface on the application and have the search results displayed in a 'NoviceUserBox' textbox.
REQ4(b)	The Novice user should be able to access the average top speed of all vehicles, only electric vehicles and only petrol vehicles from the database using three radio buttons.
REQ4(c)	The Novice user should be able to access the top five vehicles in terms of range, weight, carry capacity and top speed of all vehicles, only electric vehicles and only petrol vehicles from the database using 12 radio buttons on the application.

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Date: 7-4-2019

REQ4(d)	The Novice user should be able to view the time duration in seconds a chosen vehicle running at top speed with constant range could for in the 'NoviceUserBox' textbox by selecting a particular vehicle ID from the combo box after checking the 'time duration vehicle' radio button. That is retrieving the range and top speed for the chosen vehicle in the application and calculating the time duration for the corresponding vehicle.
REQ5	Allow the GUI to remain responsive for the users to run queries on database whilst loading and storing the XML files in the database file by implementing background threads.

REQ1(b) has a very high priority as the system application should not crash under any circumstance in any stage to prevent loss of crucial data. REQ5 has a lower priority as this functionality would be included depending on the time and resources available. REQ3(b) is paramount considering security of the application.

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

## 2.2 Enumerated Non-Functional Requirements

REQ1	Privacy should be enforced between privileged and non-privileged users. Expert user is privileged user who can use novice user's simple menu-driven queries interface section. Whereas, Novice user is a non-privileged user and are not allowed to know the query constraints stated for the expert user.
REQ2	The system must be user-friendly and usable by non-experts who have no knowledge of writing SQL command queries and database theory by using menu-driven interface.
REQ3	A novice user should run only one query at a time, a constraint set by allowing select only one radio button to retrieve information from database.
REQ4(a)	The system should provide an appropriate level of performance for expert users. The elapsed time between sending custom written SQL query to the database after the press of Query button and displaying the retrieved information on the txtResult text box should be minimal.
REQ4(b)	The system should provide an appropriate level of performance and minimal time to display retrieved information for the chosen menu on the NoviceUserBox text box for novice users.

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

### **3. Development Description**

#### **3.1 Parsing XML file using XmlDocument:**

Xml file can be parsed using different techniques based on different .NET Framework classes and their associated methods. For example, XmlTextReader, XmlDocument, XPathDocument, XmlSerializer, DataSet, etc., For this application the Xml files are parsed using XmlDocument technique and the XML file contents are loaded into object classes (here, vehicle.cs, petrol.cs and electric.cs) where petrol.cs and electric.cs inherit from base class vehicle.cs using object-oriented programming technique in C# as shown in the figure 1 below.

Each vehicle node is described inside the parent node in the XML file with eight child nodes to represent its corresponding configuration details like engine, vehicleID, range, weight, carryCapacity, topspeed and two other properties depending on engine type ('Petrol' or 'Electric').

Student name: Mrunal Prakash Gavali

Student Id : G20753327

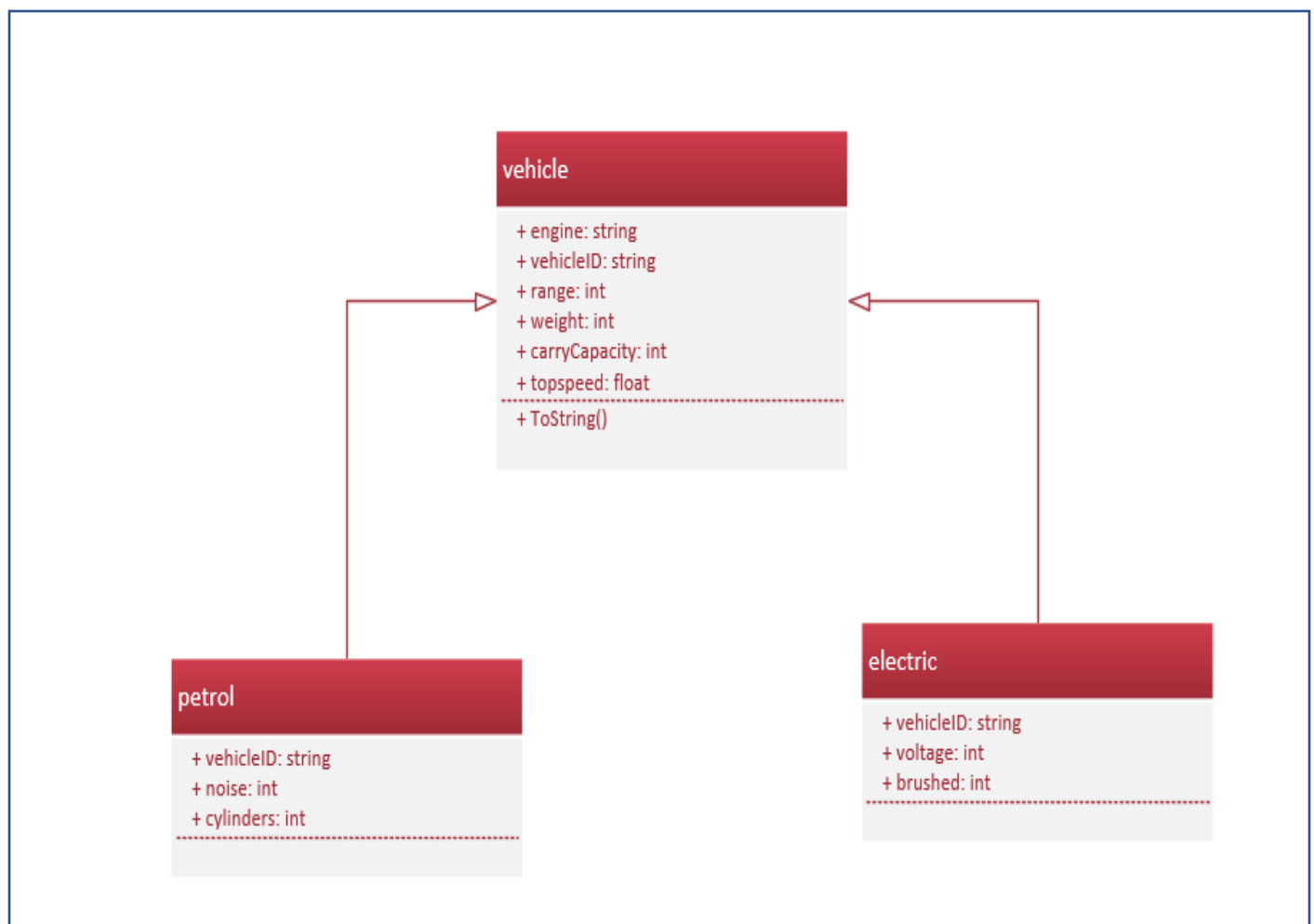
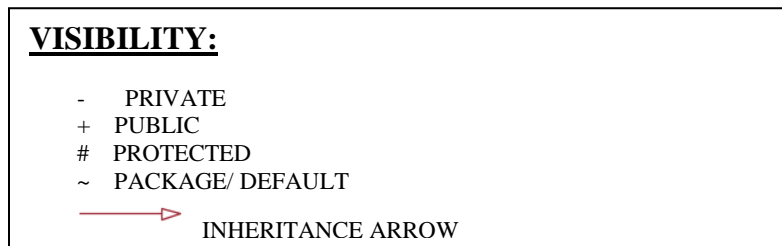
Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019



**Figure 1 : Class Diagram: vehicle (base class), petrol (derived class) and electric (derived class).**



Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

### 3.2 Loading parsed XML data into relational SQLite database:

This process involves mapping each element in the XML file versus the corresponding columns and tables in the database. This process is done using **INSERT SQL COMMAND** as shown in the code snippet below.

```
// insert into Vehicle table in courseworkdb.db database
string temp9 = "INSERT INTO vehicle(engine, vehicleID, range, weight,
carryCapacity, topspeed) VALUES('" + Engine + " ', '" + VehicleID + "', " +
Range + ", " + Weight + ", " + CarryCapacity + " , " + Topspeed + "); ";

string r9 = dbAccess.sqlChange(temp9);
```

**Figure 2 : INSERT SQL command to insert XML values into vehicle table in database.**

Note: SQLite Database connection is done using the given dbAccess.cs file

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

## XML file

```
<?xml version="1.0" encoding="UTF-8"?>
- <VehicleConfigurationDetails>
  - <vehicle>
    <engine>Petrol</engine>
    <vehicleID>Vehicle01</vehicleID>
    <range>1500</range>
    <weight>60</weight>
    <carryCapacity>12</carryCapacity>
    <topspeed>25</topspeed>
    <noise>60</noise>
    <cylinders>2</cylinders>
  </vehicle>
  - <vehicle>
    <engine>Electric</engine>
    <vehicleID>Vehicle02</vehicleID>
    <range>2500</range>
    <weight>30</weight>
    <carryCapacity>10</carryCapacity>
    <topspeed>23</topspeed>
    <voltage>40</voltage>
    <brushed>0</brushed>
  </vehicle>
</VehicleConfigurationDetails>
```



Relational Database file viewed using DB browser for SQLite

Table: Vehicle

	vehicleID	engine	range	weight	carryCapacity	topspeed
	Filter	Filter	Filter	Filter	Filter	Filter
1	Vehicle01	Petrol	1500	60	12	25.0
2	Vehicle02	Electric	2500	30	10	23.0

Table: Petrol

	vehicleID	noise	cylinders
	Filter	Filter	Filter
1	Vehicle01	60	2

Table: Electric

	vehicleID	voltage	brushed
	Filter	Filter	Filter
1	Vehicle02	40	0

**Figure 3: Mapping XML data versus database columns/tables.**

Student name: Mrunal Prakash Gavali

Student Id : G20753327

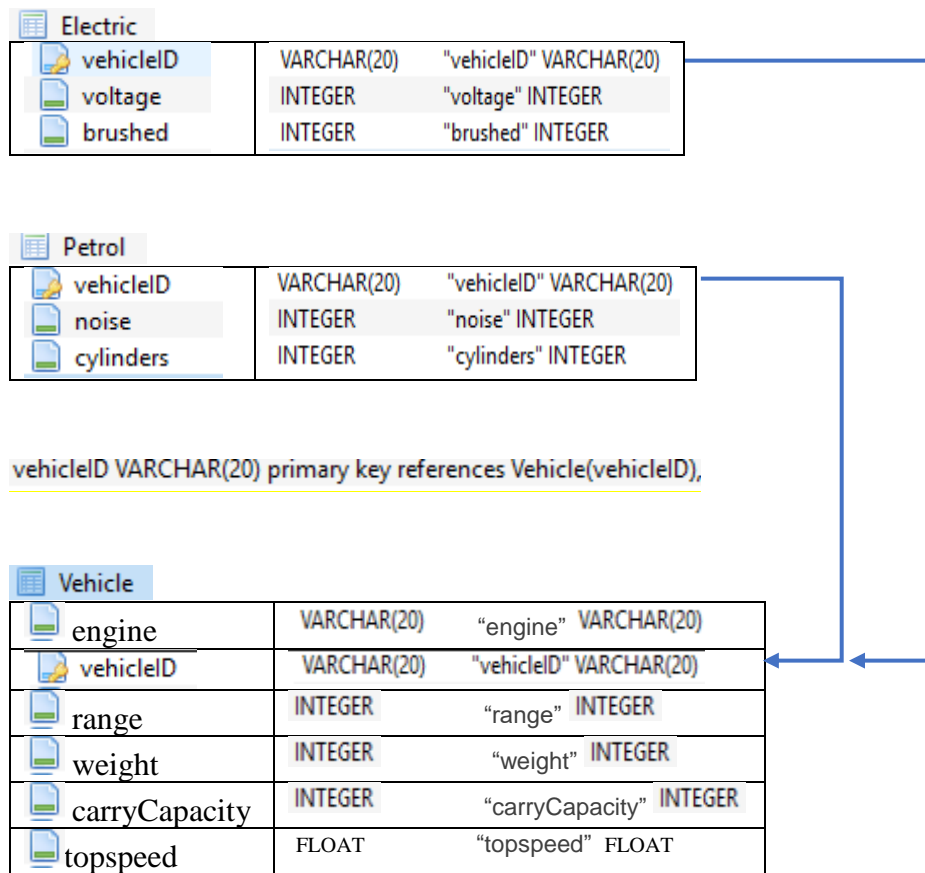
Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Date: 7-4-2019

The main data structure in the application is the Database. There are three tables in the given 'courseworkdb' SQLite database: Vehicle, Electric and Petrol. Vehicle ID is the foreign key in the linking the Petrol and Electric tables to the vehicle table. The following data types are used to store the variables in the table.

**Figure 4: Database Tables and schema diagram for Vehicle, Petrol and Electric tables**



Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Date: 7-4-2019

### **3.3 User Interface Design and Implementation**

The OpenFileDialog control in C# has a filter to allow user to select only files with .xml extension. So the user always uploads the file with the correct .xml extension. The program then needs to check if the uploaded XML file is invalid or badly formed. The program handles this exception using try-catch-finally block and prevents the application from crashing by letting the user know that invalid XML file has been uploaded and prompts user to load a valid XML file as shown in the figure below. There is a vehicles.xml file in the bin -> debug of the program used for test case plan 3.

Student name: Mrunal Prakash Gavali

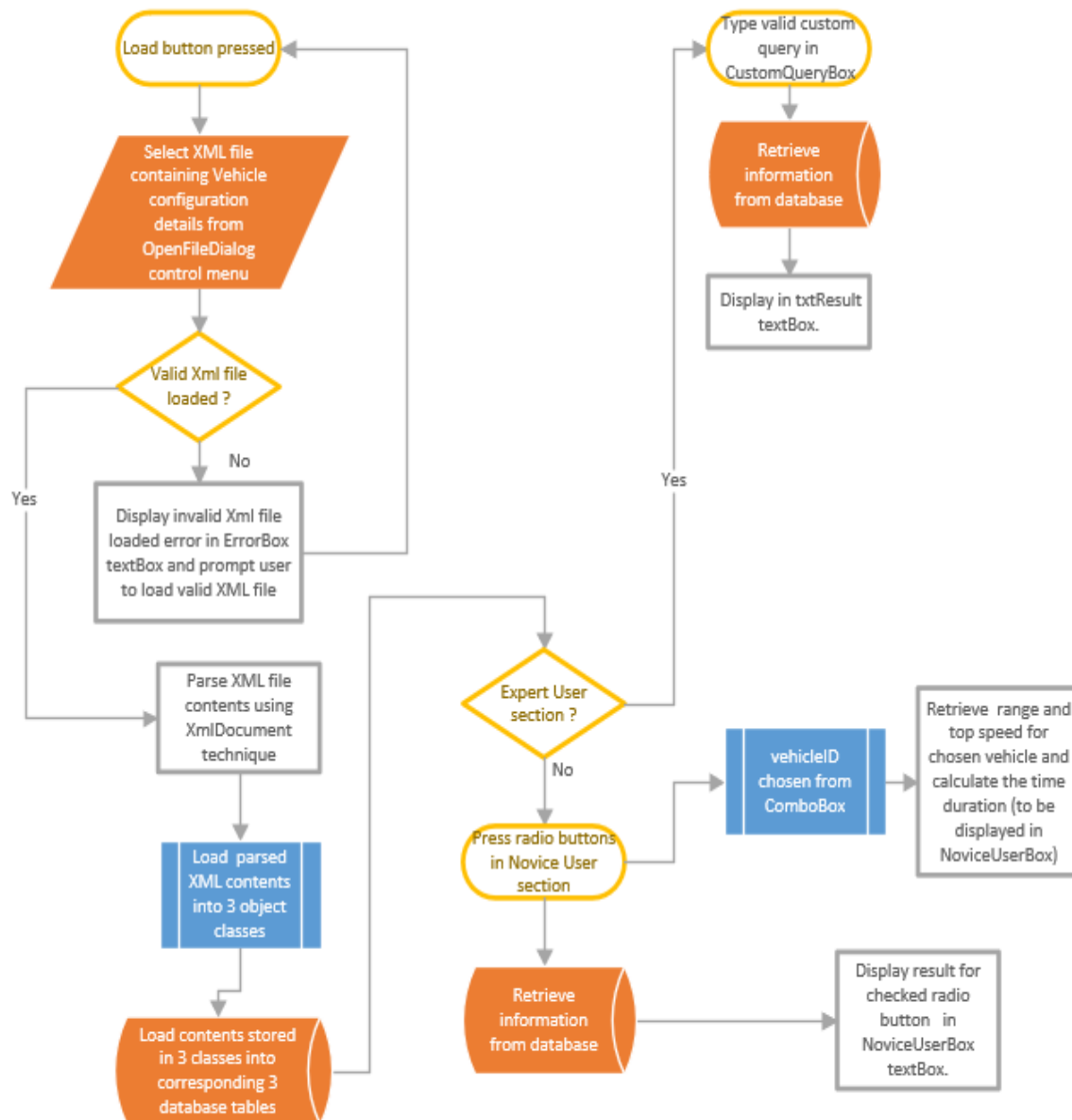
Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

**Figure 5: Data/Process Flow chart for the designed application**



Student name: Mrunal Prakash Gavali

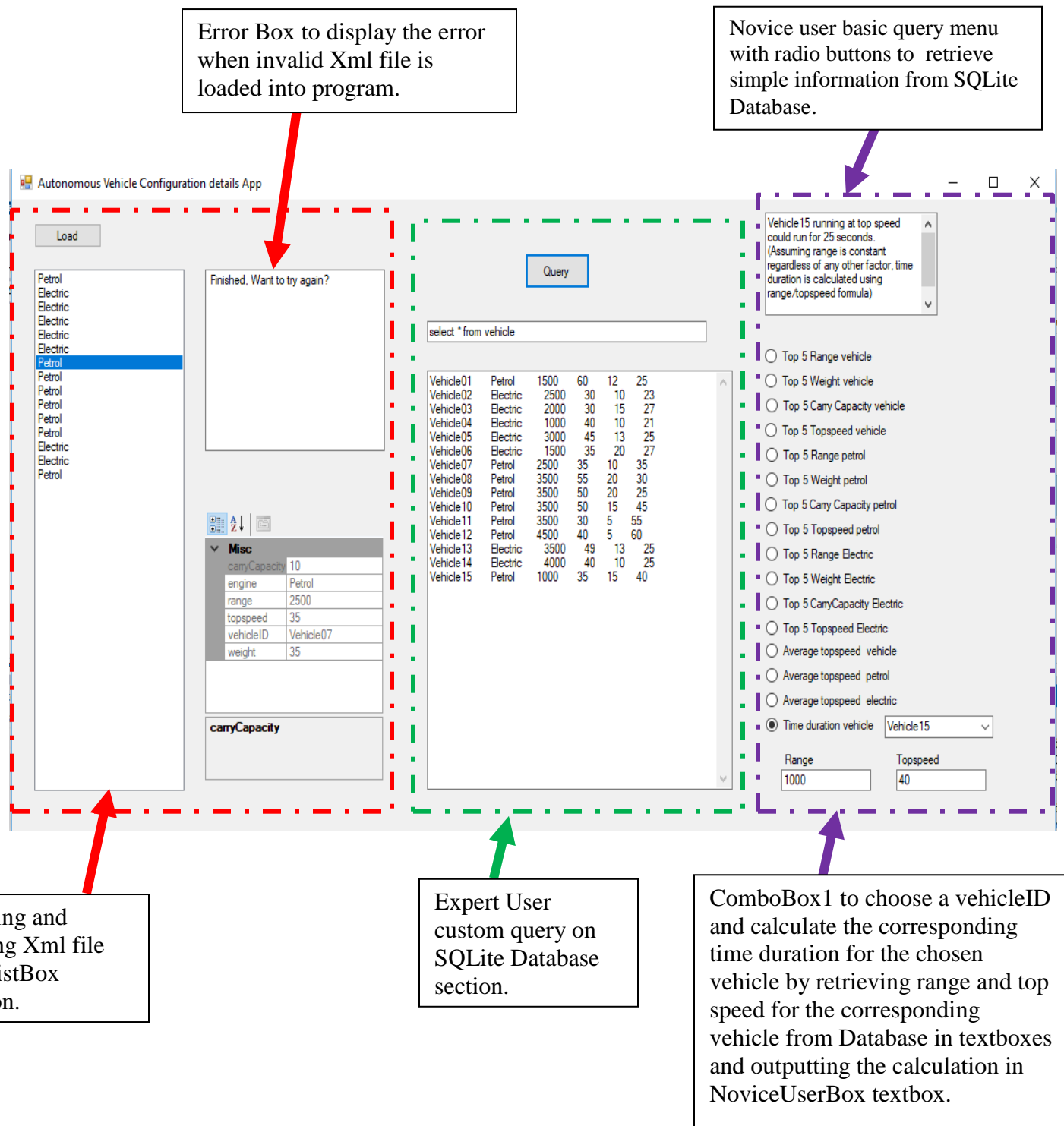
Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

**Figure 6 : Autonomous Vehicle Configuration Details (AVCD) Application user interface**



Student name: Mrunal Prakash Gavali

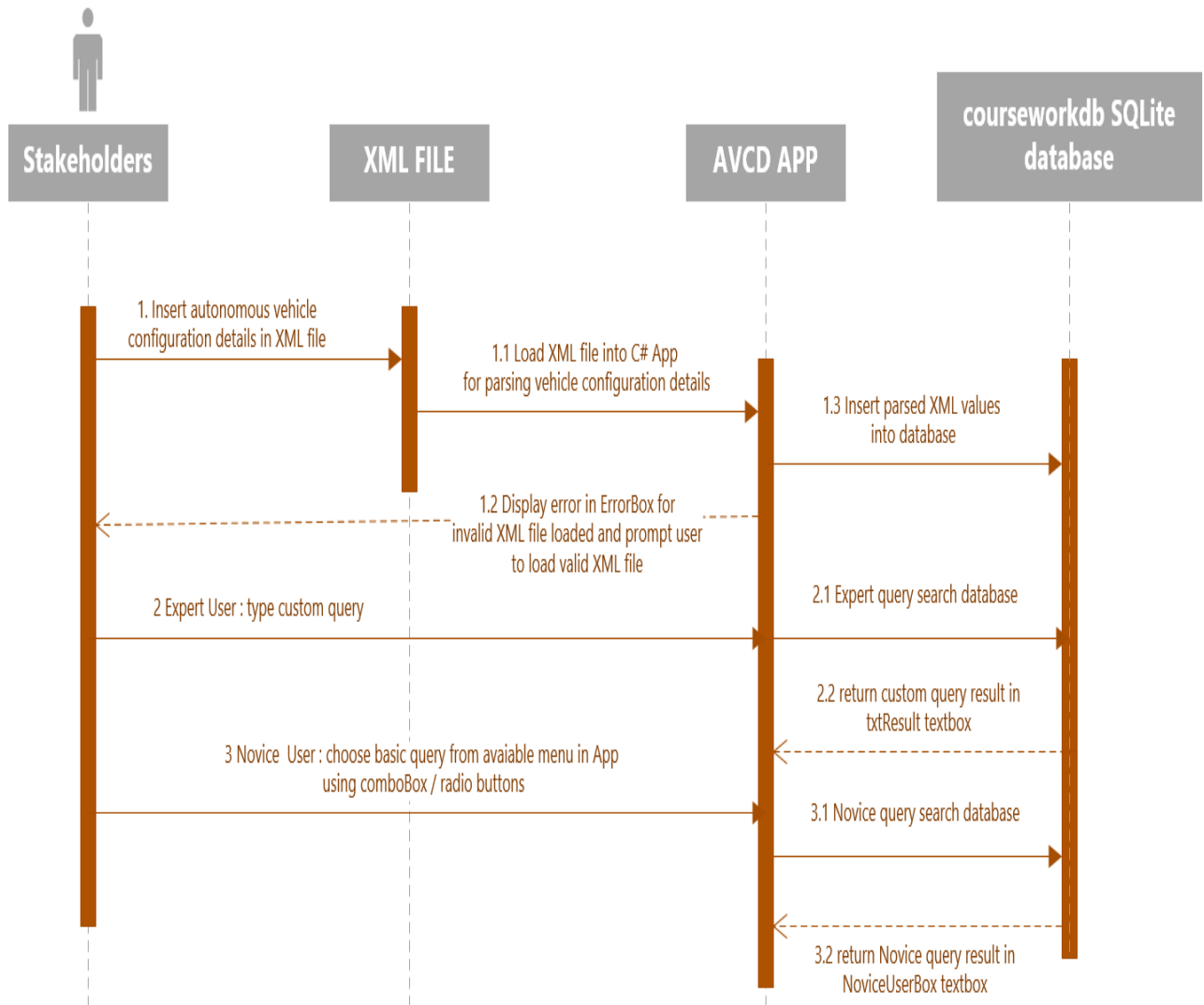
Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

**Figure 7 : Sequence Diagram for designed application**



Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019



## 4. Design of Tests

Following tables list different test case plans considered to determine the correctness of implemented program.

### 4.1 Test case plan 1: Upload XML file

Unit to test: Load button pressed and corresponding result in listBox.

Assumptions: The program has displayed the GUI screen and is waiting for user to press load button to upload the XML file.

**Table 1 : Test Sheet for loading and parsing Xml file into C# program**

Input	Expected Results	Result	Comments
User uploads a valid XML file with desired format.	The file should be Successfully loaded and the listBox on GUI is populated.	Pass if result meets the expected result.	This test case shows that user loads a valid XML file.
User uploads an XML file with random data format.	The listBox is not populated and ErrorBox text box displays error and prompts user to load valid XML file.	Pass if result meets the expected result.	This test case shows that XML file with improper format will not be allowed to upload.
XML file with same vehicleID used twice.	The listBox is not populated and ErrorBox text box displays error and prompts user to load valid XML file.	Pass if result meets the expected result.	This test case checks that vehicle ID is unique so that the corresponding configuration details are not overwritten.
XML file with missing elements.	The listBox is not populated and ErrorBox text box displays error and prompts user to load valid XML file.	Pass if result meets the expected result.	This test case checks that user loads a suitable XML file for mapping data into relational database.

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

#### 4.2 Test case plan 2: Write Custom queries to database as per functional requirements REQ3(a), REQ3(b) and REQ3(c).

Unit to test: Expert User Section

Assumptions: The program has now populated the listBox after loading valid XML file and is waiting for expert user to write custom query in CustomQueryBox text box and press Query button to get search results from database.

**Table 2: Test Sheet for running custom queries by expert user on database**

String Input for SQL query commands written	Expected Results	Result	Comments
select Distinct(vehicleID) from petrol	Data retrieved from database table 'Petrol' and displayed in txtResult textbox.	Pass if result meets the expected result.	This test case shows that expert user typed a valid query in CustomQueryBox text box.
SELECT * from Electric	Data retrieved from database table 'Electric' and displayed on txtResult textbox.	Pass if result meets the expected result.	This test case shows that the queries written to database ignores the case(uppercase/lowercase) of the SQL command.
select * from ahb	txtResult text box displays logic error message.	Pass if result meets the expected result.	This test case shows that expert user typed an invalid query in CustomQueryBox text box.
select * vehicle	txtResult text box displays syntax error message.	Pass if result meets the expected result.	This test case shows that expert user typed an invalid query in CustomQueryBox text box.
INSERT INTO vehicle VALUES('Petrol', 'vehicle100', 1500, 45, 10, 25.9,60, 4)	"Don't change the database" displayed in txtResult text box.	Pass if result meets the expected result.	This test case shows that the expert user is not allowed to insert new values in the database.

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

DELETE from vehicle WHERE range = 1500	“Don't change the database” displayed in txtResult text box.	Pass if result meets the expected result.	This test case shows that the expert user is not allowed to delete values in the database.
UPDATE vehicle SET operator = 'HONDA' WHERE range = 1500	“Don't change the database” displayed in txtResult text box.	Pass if result meets the expected result.	This test case shows that the expert user is not allowed to update values in the database.

#### 4.3 Test case plan 3: Testing implementation of functional requirements REQ4(a), REQ4(b), REQ4(c), and REQ4(d) of the system.

Unit to test: Novice User Section

Assumptions: The program has now populated the listBox after loading valid XML file and is waiting for novice user to select basic query from the available menu to display search results from database in the NoviceUserBox text box..

**Table 3: Test Sheet to run basic queries from menu for novice user for vehicle.xml (refer to appendix to view contents in vehicles.xml file) file loaded in the application.**

Input selected	Expected Result in NoviceUserBox	Actual Result in NoviceUserBox	Pass/Fail
Top 5 Range vehicle radio button checked	Vehicle12 4500 Vehicle14 4000 Vehicle08 3500 Vehicle09 3500 Vehicle10 3500	Vehicle12 4500 Vehicle14 4000 Vehicle08 3500 Vehicle09 3500 Vehicle10 3500	Pass
Top 5 Weight vehicle radio button checked	Vehicle01 60 Vehicle08 55 Vehicle09 50 Vehicle10 50 Vehicle13 49	Vehicle01 60 Vehicle08 55 Vehicle09 50 Vehicle10 50 Vehicle13 49	Pass
Top 5 Carry Capacity vehicle radio button checked	Vehicle06 20 Vehicle08 20 Vehicle09 20 Vehicle03 15 Vehicle10 15	Vehicle06 20 Vehicle08 20 Vehicle09 20 Vehicle03 15 Vehicle10 15	Pass

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Date: 7-4-2019

Top 5 Topspeed vehicle radio button checked	Vehicle12 60 Vehicle11 55 Vehicle10 45 Vehicle15 40 Vehicle07 35	Vehicle12 60 Vehicle11 55 Vehicle10 45 Vehicle15 40 Vehicle07 35	Pass
Top 5 Range petrol radio button checked	Vehicle12 4500 Vehicle08 3500 Vehicle09 3500 Vehicle10 3500 Vehicle11 3500	Vehicle12 4500 Vehicle08 3500 Vehicle09 3500 Vehicle10 3500 Vehicle11 3500	Pass
Top 5 Weight petrol radio button checked	Vehicle01 60 Vehicle08 55 Vehicle09 50 Vehicle10 50 Vehicle12 40	Vehicle01 60 Vehicle08 55 Vehicle09 50 Vehicle10 50 Vehicle12 40	Pass
Top 5 Carry Capacity petrol radio button checked	Vehicle08 20 Vehicle09 20 Vehicle10 15 Vehicle15 15 Vehicle01 12	Vehicle08 20 Vehicle09 20 Vehicle10 15 Vehicle15 15 Vehicle01 12	Pass
Top 5 Topspeed petrol radio button checked	Vehicle12 60 Vehicle11 55 Vehicle10 45 Vehicle15 40 Vehicle07 35	Vehicle12 60 Vehicle11 55 Vehicle10 45 Vehicle15 40 Vehicle07 35	Pass
Top 5 Range Electric radio button pressed	Vehicle14 4000 Vehicle13 3500 Vehicle05 3000 Vehicle02 2500 Vehicle03 2000	Vehicle14 4000 Vehicle13 3500 Vehicle05 3000 Vehicle02 2500 Vehicle03 2000	Pass
Top 5 Weight Electric radio button pressed	Vehicle13 49 Vehicle05 45 Vehicle04 40 Vehicle14 40 Vehicle06 35	Vehicle13 49 Vehicle05 45 Vehicle04 40 Vehicle14 40 Vehicle06 35	Pass
Top 5 CarryCapacity Electric radio button pressed	Vehicle06 20 Vehicle03 15 Vehicle05 13 Vehicle13 13 Vehicle02 10	Vehicle06 20 Vehicle03 15 Vehicle05 13 Vehicle13 13 Vehicle02 10	Pass
Top 5 Topspeed Electric radio button pressed	Vehicle03 27 Vehicle06 27 Vehicle05 25 Vehicle13 25 Vehicle14 25	Vehicle03 27 Vehicle06 27 Vehicle05 25 Vehicle13 25 Vehicle14 25	Pass
Average topspeed vehicle radio button pressed	32.53333333333333	32.53333333333333	Pass
Average topspeed petrol radio button pressed	39.375	39.375	Pass

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

Average topspeed Electric radio button pressed	24.7142857142857	24.7142857142857	Pass
Time duration vehicle radio button pressed and 'Vehicle15' vehicleID chosen from comboBox	Vehicle15 running at top speed could run for 25 seconds. (Assuming range is constant regardless of any other factor, time duration is calculated using range/topspeed formula)	Vehicle15 running at top speed could run for 25 seconds. (Assuming range is constant regardless of any other factor, time duration is calculated using range/topspeed formula)	Pass

## 5. DISCUSSION

The user is supposed to start running the application with the assumption that the database is blank and that no XML file has previously been loaded into it. The comboBox in the novice user section displays only those vehicles which are in the XML file currently loaded in the program when the application is running. The program allows the user to load XML file for each vehicle separately as well as in groups by allowing multiple press of the load button. The vehicleID is unique and even if the different configurations are written for the same vehicleID during different press of load button, the configuration details written first time for that vehicle are not overwritten thus meeting the requirements of the system application.

## 6. Conclusion and Future Work

Numerous useful features have been incorporated into this application and the application has met its primary objectives listed in the requirements section. However, the application can include additional features in the long run. For instance, this application can be extended into an android application that can provide enhanced mobility to the user and allow accessing the configuration details of the autonomous vehicles anywhere, anytime. The application can further have additional functionality to allow user to login with unique username and password. The application would then communicate with the database to authenticate the user. Authentication would allow the user to continue the session to perform queries on database. The new users would have to register as either expert user or novice user with the application and this information of registered user, which would determine the level of access, would be stored in the database. This would enhance organisational security by having record of every user who accessed the database.

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

## REFERENCES

1. Thomas D., Hunt A., "Learning to Love Unit Testing", in The Software Testing & Quality Engineering Magazine, January/February 2002, volume 4, issue 1. Pp.32-38.
2. Fowler M., ed., Refactoring: Improving the Design of Existing Code, Addison Wesley Longman, 1999.
3. Beck K., extreme Programming explained: Embrace Change, Addison Wesley Longman, 2000.
4. Fewster M, Graham D., Software Test Automation: Effective use of test execution tools, Addison Wesley, 1999.
5. Kit E., Software Testing in the Real World: improving the process, Addison Wesley, 1995.
6. Dustin E., Rashke E., Paul J., Automated Software Testing: Introduction, Management and Performance, Addison Wesley, 1999.
7. Binder R., Testing Object-Oriented Systems: Models, Patterns and Tools, Addison Wesley, 2000.
8. Humphrey W., A Discipline for Software Engineering, Addison Wesley, 1995.
9. McConnell S., Code Complete, Microsoft Press, 1993.
10. Maguire S., Writing Solid Code, Microsoft Press, 1993.
11. Whittaker J.A, How to break software: A Practical Guide to Testing, AddisonWesley, 2002.
12. Sommerville I., Software Engineering 7th Ed. Addison-Wesley, 2004.

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

# APPENDIX

## vehicles.xml file

```
<?xml version="1.0" encoding="utf-8" ?>
<VehicleConfigurationDetails>

  <vehicle>
    <engine>Petrol</engine>
    <vehicleID>Vehicle01</vehicleID>
    <range>1500</range>
    <weight>60</weight>
    <carryCapacity>12</carryCapacity>
    <topspeed>25</topspeed>
    <noise>60</noise>
    <cylinders>2</cylinders>
  </vehicle>

  <vehicle>
    <engine>Electric</engine>
    <vehicleID>Vehicle02</vehicleID>
    <range>2500</range>
    <weight>30</weight>
    <carryCapacity>10</carryCapacity>
    <topspeed>23</topspeed>
    <voltage>40</voltage>
    <brushed>0</brushed>
  </vehicle>

  <vehicle>
    <engine>Electric</engine>
    <vehicleID>Vehicle03</vehicleID>
    <range>2000</range>
    <weight>30</weight>
    <carryCapacity>15</carryCapacity>
    <topspeed>27</topspeed>
    <voltage>30</voltage>
    <brushed>1</brushed>
  </vehicle>

  <vehicle>
    <engine>Electric</engine>
    <vehicleID>Vehicle04</vehicleID>
    <range>1000</range>
    <weight>40</weight>
    <carryCapacity>10</carryCapacity>
```

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

```

    <topspeed>21</topspeed>
    <voltage>45</voltage>
    <brushed>1</brushed>
</vehicle>

<vehicle>
    <engine>Electric</engine>
    <vehicleID>Vehicle05</vehicleID>
    <range>3000</range>
    <weight>45</weight>
    <carryCapacity>13</carryCapacity>
    <topspeed>25</topspeed>
    <voltage>35</voltage>
    <brushed>0</brushed>
</vehicle>

<vehicle>
    <engine>Electric</engine>
    <vehicleID>Vehicle06</vehicleID>
    <range>1500</range>
    <weight>35</weight>
    <carryCapacity>20</carryCapacity>
    <topspeed>27</topspeed>
    <voltage>45</voltage>
    <brushed>1</brushed>
</vehicle>

<vehicle>
    <engine>Petrol</engine>
    <vehicleID>Vehicle07</vehicleID>
    <range>2500</range>
    <weight>35</weight>
    <carryCapacity>10</carryCapacity>
    <topspeed>35</topspeed>
    <noise>60</noise>
    <cylinders>2</cylinders>
</vehicle>

<vehicle>
    <engine>Petrol</engine>
    <vehicleID>Vehicle08</vehicleID>
    <range>3500</range>
    <weight>55</weight>
    <carryCapacity>20</carryCapacity>
    <topspeed>30</topspeed>
    <noise>55</noise>
    <cylinders>4</cylinders>
</vehicle>

<vehicle>

```

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019



```
<engine>Petrol</engine>
<vehicleID>Vehicle09</vehicleID>
<range>3500</range>
<weight>50</weight>
<carryCapacity>20</carryCapacity>
<topspeed>25</topspeed>
<noise>70</noise>
<cylinders>4</cylinders>
</vehicle>
```

```
<vehicle>
  <engine>Petrol</engine>
  <vehicleID>Vehicle10</vehicleID>
  <range>3500</range>
  <weight>50</weight>
  <carryCapacity>15</carryCapacity>
  <topspeed>45</topspeed>
  <noise>75</noise>
  <cylinders>2</cylinders>
</vehicle>
```

```
<vehicle>
  <engine>Petrol</engine>
  <vehicleID>Vehicle11</vehicleID>
  <range>3500</range>
  <weight>30</weight>
  <carryCapacity>5</carryCapacity>
  <topspeed>55</topspeed>
  <noise>40</noise>
  <cylinders>2</cylinders>
</vehicle>
```

```
<vehicle>
  <engine>Petrol</engine>
  <vehicleID>Vehicle12</vehicleID>
  <range>4500</range>
  <weight>40</weight>
  <carryCapacity>5</carryCapacity>
  <topspeed>60</topspeed>
  <noise>30</noise>
  <cylinders>4</cylinders>
</vehicle>
```

```
<vehicle>
  <engine>Electric</engine>
  <vehicleID>Vehicle13</vehicleID>
  <range>3500</range>
  <weight>49</weight>
  <carryCapacity>13</carryCapacity>
```

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019

```
<topspeed>25</topspeed>
<voltage>30</voltage>
<brushed>0</brushed>
</vehicle>

<vehicle>
  <engine>Electric</engine>
  <vehicleID>Vehicle14</vehicleID>
  <range>4000</range>
  <weight>40</weight>
  <carryCapacity>10</carryCapacity>
  <topspeed>25</topspeed>
  <voltage>20</voltage>
  <brushed>1</brushed>
</vehicle>

<vehicle>
  <engine>Petrol</engine>
  <vehicleID>Vehicle15</vehicleID>
  <range>1000</range>
  <weight>35</weight>
  <carryCapacity>15</carryCapacity>
  <topspeed>40</topspeed>
  <noise>65</noise>
  <cyinders>2</cyinders>
</vehicle>

</VehicleConfigurationDetails>
```

Student name: Mrunal Prakash Gavali

Student Id : G20753327

Project title: Data Storage and Retrieval in SQLite Database using C#

Tutor name : Carl Berry

Data: 7-4-2019