Cognizant

COGNIZANT

Cognizant Academy

Vehicle Repair Service Centre

C#, ADO .Net Knock Out Challenge

Version 1.0

	Prepared By / Last Updated By	Reviewed By	Approved By
Name	Gnanasekar R		
Role			
Signature			
Date			

Cognizant

Table of Contents

Cognizant

1.0 Introduction

1.1 Purpose of this document

JAWA Bike Service Centre wants to automate their business process .

They want to store the bike details coming for service .

Simliarly update the bike colour based on the Id.

The focus is to automate the below mentioned requirements.

1. Add the bike details to the database
2. Update the bike colour based on bike id in the database
3. Calculate the repair cost based on bike CC and bike's kilometer

1.2 Definitions & Acronyms

Definition / Acronym	Description
Req	Requirement

1.3 Project Overview

This project captures the various concepts, techniques and skills learnt and helps to put them into practice using C# with ADO.NET. Admittedly, this would be at a scaled-down level since the purpose is to let the associate experience the various concepts learned in C# as an individual. The individual associate is expected to create a console-based application within the specified time.

1.4 Scope

The scope of the system is explained through its following modules

1. Add the bike details to the database.

2. Update the bike colour based on bike id in the database.

3. Calculate the repair cost based on bike CC and bike's kilometer

1.5 Target Audience

Learner Level

1.6 Software Requirement

1.6.1 Software Requirements

#	Item	Specification/Version
1.	C#	6
2.	ADO.NET	4.5
3	ORACLE SERVER	18C

Note: All the required hardware and software will be provided in the Tekstac platform.

2.0 Functional Requirements

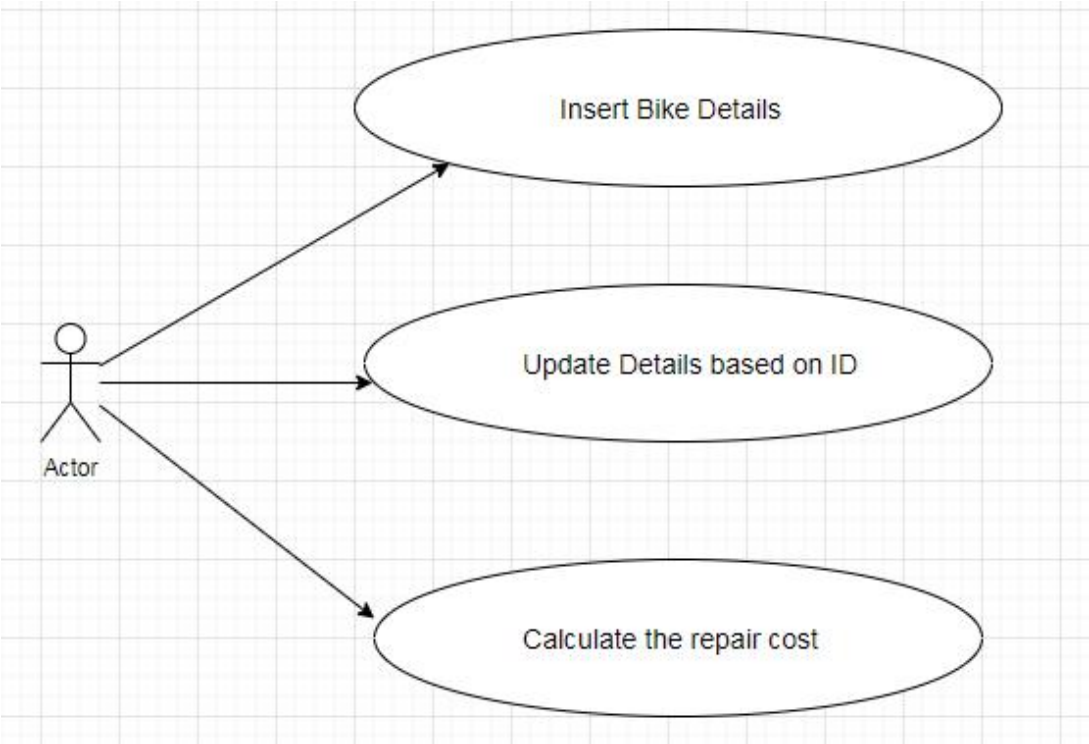
2.1 Functional Requirements

Req. #	1
Req. Name	Insert the bike details into the database
Req. Description	Get the **bike details** from the User. Insert the Bike details into the database.
Actors/ Users	Manager
Comments	The manager is responsible for inserting the data to the database.

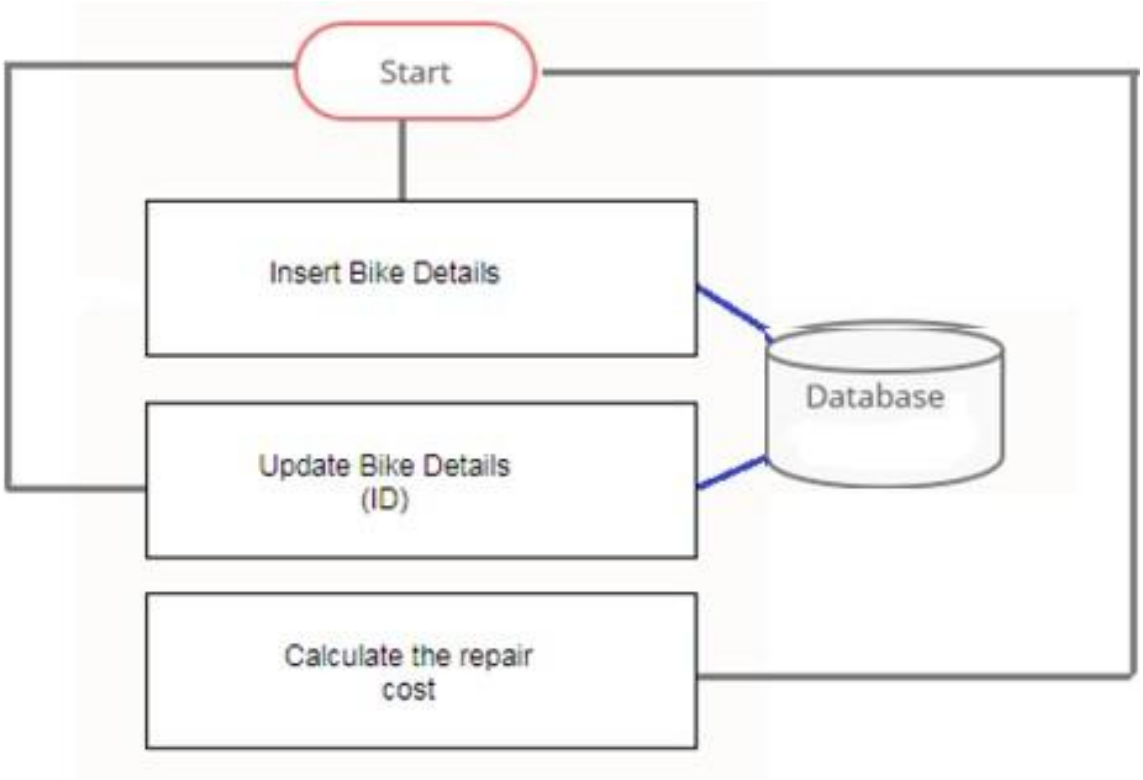
Req. #	2
Req. Name	Update the bike details based on the bike id
Req. Description	Get the **BikeId, BikeColour** from the user. Update the bike's colour in the database based on bike id
Actors/ Users	Manager
Comments	The manager is responsible for updating the details in the database

Req. #	3
Req. Name	Calculate the repair cost based on bike CC and bike's Kilometer.
Req. Description	Get the **CC and kilometer** from user. Calculate the repair cost based on the conditions
Actors/ Users	User
Comments	The User is responsible for calculating repair cost

2.2 Use case Diagram



2.3 System Architecture Diagram



2.4 Sample Input/Output

Sample Input 1 :

1. Add the bike details

2. Update the bike colour

3. Calculate the repair cost

4. Exit

Enter your choice:

1

Enter bike id :

4

Enter bike model :

SUZUKI GIXER

Enter bike colour :

Blue

Enter cubic capacity :

250

Enter bike price :

260000

Sample Output 1 :

Bike details added successfully

Sample for your reference :

BikeId	BikeModel	BikeColour	CubicCapacity	BikePrice
1	TVS XL	Blue	100	60000
2	KTM Duke	Orange	250	250000
3	BULLET	Black	500	275000
4	SUZUKI GIXER	Blue	250	260000

Sample Input 2 :

1. Add the bike details

2. Update the bike colour

3. Calculate the repair cost

4. Exit

Enter your choice:

1

Enter bike id :

4

Enter bike model :

Ninja

Enter bike colour :

Blue

Enter cubic capacity :

1000

Enter bike price :

500000

Sample Output 2 :

Bike details cannot be added

Sample Input 3 :

1. Add the bike details

2. Update the bike colour

3. Calculate the repair cost

4. Exit

Enter your choice:

2

Enter bike id :

2

Enter bike colour to update :

White

Sample Output 3 :

Bike colour updated successfully

Sample for your reference :

BikeId	BikeModel	BikeColour	CubicCapacity	BikePrice
1	TVS XL	Blue	100	60000
2	KTM Duke	**White**	250	250000
3	BULLET	Black	500	275000
4	SUZUKI GIXER	Blue	250	260000

Sample Input 4 :

1. Add the bike details

2. Update t e bike colour

3. Calculate the repair cost

4. Exit

Enter your choice:

3

Enter cubic capacity :

100

Enter bike's current kilomet :

5550

Sample Output 4 :

Calculated repair cost = 1495

Sample Inpput 5 :

1. Add the bike details

2. Update t e bike colour

3. Calculate the repair cost

4. Exit

Enter your choice:

4

Sample Output 5 :

Thank you

3.0 Design Specification

3.1 Data Design

Table Structure:

Table name: Bike	
Column Name	**Data type**
BikeId	NUMBER
BikeModel	VARCHAR2(100)
BikeColour	VARCHAR2(100)
CubicCapacity	NUMBER
BikePrice	NUMBER

Design Constraints:

- Use **ORACLE SERVER** database to store the data. The database name is "Bike". **This is already created for you in Tekstac.**
- The table names and the column names should be the same as specified in the table structure.
- Connection string is given in **DBConnection.cs,** Should use the **'connStr'** static variable for DB connections, which is also provided as part of code skeleton. THIS IS GIVEN ONLY FOR YOUR REFERENCE. You need NOT change this.

Sample Data is Already inserted in the Bike table.

BikeId	BikeModel	BikeColour	CubicCapacity	BikePrice
1	TVS XL	Blue	100	60000
2	KTM Duke	Orange	250	250000
3	BULLET	Black	500	275000

Note: The code skeleton will be available in the Tekstac platform

3.2 Component Details for identified Use Cases

3.2.1 Insert the bike detail into the database

In the 'Main' get the following values,

Bike Id,

Bike Model,

Bike Colour,

Cubic Capacity,

Bike Price

Insert the Bikedetails into the table

3.2.2 Update the bike colour by bike id in the database

In the 'Main' get the following values,

Bike Id, Bike colour

Update the Bike Colour in the database based on the Bike Id

3.2.3 Calculate the repair cost based on bike cc and bike's Kilometer

In the 'Main' get the following values,

Cubic capacity, kilometer

Repair cost to be calculated based on the below condition:

NOTE : Service cost = 1000(fixed rate)

If bike kilometer is **between 0 – 5000km** then repair cost = 50 for all cc's

If bike kilometer is **between 5001 – 10000km** then,

For 100cc

Cost = (serviceCost +500)– 5% of cc

For 250cc

Cost = (serviceCost + 700) – 8% of cc

For 500cc

Cost = (serviceCost + 900)– 10% of cc

If bike kilometer is **above 10000km** then,

For 100cc

Cost = (serviceCost +600)– 5% of cc

For 250cc

Cost = (serviceCost + 800) – 8% of cc

For 500cc

Cost = (serviceCost + 1000)– 10% of cc

3.3 Component Specification

Class Name :Bike(model class)

Responsibility:

This model object holds the state of the booking detail at all point-in-time.

Type(Class)	Properties
Bike	**int** BikeId **String** BikeModel **String** BikeColour **int** CubicCapacity **int** BikePrice

Note : Keep all the properties as **'public'**.

3.3.1 Insert Bike Details

Type(Class)	Method	Responsibilities
BikeManagement	public void AddBikeDetails(Bike bike)	This method is used to add the bike details into the database and once the

		details gets added into the database then print **"Bike details added successfully"** in main method, if cubic capacity is other than 100, 250, 500 then don't add the details into the datebase and print **"Bike details cannot be added".**

3.3.2 Update Bike colour by Id

Type(Class)	Method	Responsibilities
BikeManagement	public bool UpdateBikeColourById (int id, String bikeColour)	This method is used to update the bike's colour in the database based on the given bike id and return **true** and print **"Bike colour updated successfully"** once the bike details gets updated else return **false** and print **"Bike colour cannot be updated"**

3.3.3 Calculate repair cost by cc and price

Type(Class)	Method	Responsibilities
BikeManagement	public int CalculateBikeRepairCost(int cc, int bikeKilometer)	This method is used to calculate the bike repair cost and return it. Use the given conditions to calculate the cost. if

		cubic capacity is other than **100, 250, 500** then return **0.**

Class Name : DBHandler(DAO class)

3.3.4 GetConnection Method

Responsibility:

This method should connect to the database by reading the database details from the **DBConnection.cs** file and it should return the connection object.

Type(Class)	Method	Resources
DBHandler	public OracleConnection GetConnection()	DBConnection.cs file contains the databaseconnection details.

3.4 General Design Constraints

1. The fields/properties/method/class name should be correctly specified as given in the document.
2. Keep all the classes as '**public**'
3. Do not change the DBConnection.cs file.
4. Do not change the namespace name.

4.0 Submission

4.1 Code submission instructions

1. Do not change the code skeleton given as your code will be auto evaluated.
2. You can validate your solution against sample test cases

during the assessment duration.

3. Your last submitted solution will be considered for detailed evaluation.

4. Make sure to submit the solution before the time limit. After the assessment duration you will not be allowed to submit the solution.

5.0 Change Log

<table>
<tr><th></th><th colspan="4">Changes Made</th></tr>
<tr><td>V1.0.0</td><td colspan="4">Initial baseline created on <dd-Mon-yy> by <Name of Author></td></tr>
<tr><td rowspan="3">Vx.y.z</td><td colspan="4"><Please refer the configuration control tool / change item status form if the details of changes are maintained separately. If not, the template given below needs to be followed></td></tr>
<tr><td>Section No.</td><td>Changed By</td><td>Effective Date</td><td>Changes Effected</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

6.0 Evaluation Areas

S.No	Description
1.	Declaration of properties in the class Bike
2.	Declaration of methods in the class DBHandler
3.	Declaration of methods in the class BikeManagement
4.	Implementation to create a valid database connection object
5.	Implementation to Insert Bike details into the database
6.	Implementation to Update Bike colour in the database
7.	Calculate the repair cost based on cc and Bike's current

	kilometer