# **6.1 Introduction**

This chapter is dedicated to discuss the development process of the solution. Starting with the development methodology, areas such as language and tool selection. Later on architectures and coding standards also have being discussed. Several coding samples also provided from the real project.

# **6.2 Software Development-Technology Considerations**

In order to develop a successful system which meets the end user expectations, it is very important to use appropriate tools in the development process. Use of any inappropriate tools will only leads to develop a system with unnecessary errors and faults and it can affect the user expectations in efficiency and reliability of the solution. It is very vital to use appropriate computer language and any other necessary tools in order to develop a successful system. So these technologies and tools will help to develop the system within a minimum development time. The main objective of TBMVCS is to do a fast, easy, accurate claiming process via tablet pc. And this is done by connecting to the company database via web service with the use of mobile networks. So it is very important to consider some factors such as platform supportiveness and the efficiency of the system. And in order to meet above mention factors we must use the most appropriate tools available in the market to develop the system.

Technological considerations - followed during the development of the system

• Efficiency and Performance

• Re-usability and flexibility

• Object oriented development support

# **6.2.1 Development Language Selection**

In modern programming field there are many programming languages and many more are introducing day by day. Core function of the system is vehicle claiming through the tablet pc, researchers have been done on many software development languages in order to recognize the appropriate language that helps efficient data retrieval and updating from the company database via web service. Since the entire solution has a 3 main components namely the ground office system, web service and tablet application it was vital to select a language that will support all these three feature so that it will reduce the learning time of different languages. Several aspects such as build powerful web based applications, powerful, flexible, Simplified Data Access, platform dependency, build fast mobile applications were considered.

For the development of ground office system main technology considerations were ASP.net platform, java platform and PHP platform. Java and asp.net provide greater object oriented programming feasibility while PHP work as a fast backend server scripting language. Visual Studio IDE provide ASP.NET development environment with rich inbuilt functionalities for faster programming. As for the web service it was considered about coding a Visual studio web service, PHP web service and also WCF web service. WCF was considered to be a far better option than other web service types Microsoft has provided.

Another key part of the solution was the development of tablet applications. Main platform considerations were android, windows and IOS. In order to develop an IOS app it requires to learn Objective C language and it also need a Mac computer which are far more expensive than the others. Android is an open source platform with free available tools. There are lot of resources available for android development as well. Windows tablet pc was feasible to build on the same environment which was used to build the ground office system and the time taken for the learning curve was fairly less than android. Visual studio provides an in built emulator for testing purposed as well.

After considering several options ASP.Net was selected as the main programming platform to develop the ground office system since ASP.Net provides an easier and stable environment to create great web applications and it supports object oriented development greatly. Although it has to be purchased for commercial use, it can be purchased for free for university students from Dream spark. Bootstrap, a free CSS library was use to design the interfaces. C# programming language was used to implement the business logics in the backend of the system.

As for the initial step tablet application was developed for windows surface tab. It is using XAML for interface development and C# for backend coding. To connect the Tab application and the web system, WCF web service technology was selected since it can be developed using the same tools which are used to develop the system. Visual Studio 2013 was selected as the main tool for developing the solution while using open source text editors such as brackets were used wherever necessary.

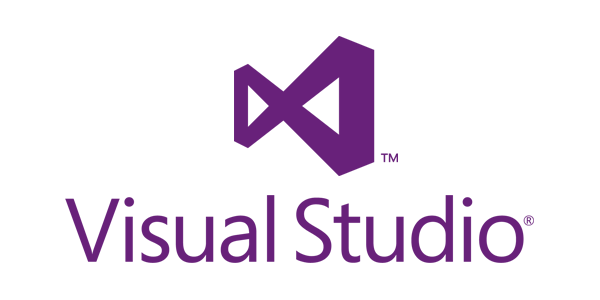
# **6.2.2 Database Selection**

Database which is most important aspect of the overall system which handles all the relevant details related to insurance as well as claims. In order to fulfill the need of database operation this system has been used SQL Server Management Studio which enables to access and manage the database engine. Another reason for using this was it was free to purchase using Dreamspark account. Management Studio brings graphical tools for database management together with a rich development environment. Database is maintained at a central cloud server such that it can be accessed from wherever necessary.

# **6.2.3 Development Tool Selection**

After selecting the development language and database type necessary tools were selected for the development. This includes tools for build and debug environment, integrated development environments, database servers, plugins and frameworks and so on. Some of the key tools which were used are described below.

**Visual Studio 2013 Premium Version**



Visual Studio was used as the main development environment and IDE for the web application, tab application, web service and the smartphone application. In order to test and emulate the tab application and smartphone application separate emulate tool set provided by Microsoft was installed. This provide very rich tools for debugging which made the developer task less complex. This was downloaded for free from Microsoft Dreamspark.

**DevExpress Developer Framework**



DevExpress framework was integrated to the visual studio since it provided very powerful tools for representing and manipulating data grids. DevExpress is a software product company that providing the UI controls to the Borland Delphi, and Active X controls to the Visual Studio to make easy development and used for high quality user experience.

**Gimp**



This was used for photo editing and icon creating works. It is the Free and Open Source, cross-platform image editor available for GNU/Linux, OS X, Windows and more. GIMP provides the tools needed for high quality image manipulation. From retouching to restoring to creative composites, the only limit is your imagination.

**Bootstrap**



Bootstrap was used for front end development and styling purposes. With Bootstrap, you get extensive and beautiful documentation for common HTML elements, dozens of custom HTML and CSS components, and awesome jQuery plugins. It is an open source styling library.

**Microsoft Visio 2013**



Microsoft Visio 2013 is used to data modeling, develop the diagrammatical designs and data analyzing designs to the project. Mostly used in the designing stage of the project as well as system analyzing stage.

**Font Awesome Icon Font**



Font Awesome gives you scalable vector icons that can instantly be customized — size, color, drop shadow, and anything that can be done with the power of CSS. This is an iconic font designed for Bootstrap

**Metro UI CSS & JavaScript Frame Work**



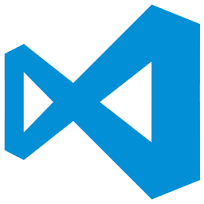
The front-end framework for developing projects on the web in Windows Metro Style. Metro UI CSS developed with the advice of Microsoft to build the user interface and include: general styles, grid, layouts, typography, components, and built-in icons. Metro UI CSS is open source and has MIT licensing model.

**SQL Server 2012**



In the Enterprise industrial level with the .Net develop framework based development and most recommended database engine is Microsoft SQL Server. As the current requirement to develop the application, .Net framework has been used. According to that, as bettor database environment, SQL Server has been selected.

**Visual Studio Code**



Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications. Visual Studio Code is free and have stylish UI design which offer various text editing options.

A custom development has been selected as the development strategy in order to develop the system. This method has been selected after considering some vital factor such as complexity of the system, development time, etc. Main reason of using this development strategy is building the system interfaces at initially which will used to develop system prototypes and develop the databases with accordance with interfaces in order to develop the web service in a fast manner. Then the tab application was developed while developing the some modules of web application as well.

Components that has been recognized as vital and to be built at first are,

* Database Development
* Interfaces of all components
* Policy information module and vehicle spare part module of web application
* Web service
* Tab application
* Claim approval module
* Rest of the web application components
* Mobile application

# **6. 3 Database Development**

SQL server 2012 was used for database related activities. All the tables were created in the design view and a naming convention was used to identify the tables easily. Following figure shows the tables in the SQL server views.

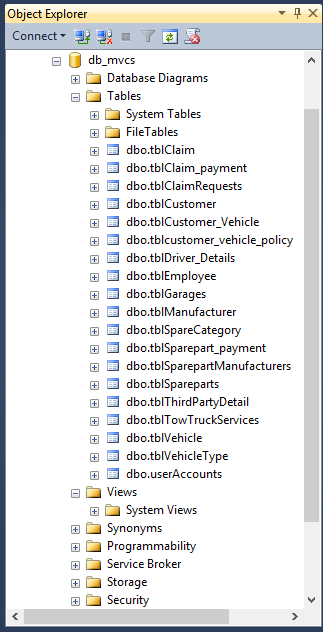


Figure : SQL Server Tables

In order to query to the database tables stored procedures has being used in the entire web application. A stored procedure is simply a way to put blocks of code inside the database. Stored procedures are compiled once and stored in executable form, so procedure calls are quick and efficient. Executable code is automatically cached and shared among users. This lowers memory requirements and invocation overhead. Stored procedures increase scalability by isolating application processing on the server. Following figure shows some of the stored procedures declared in the application.

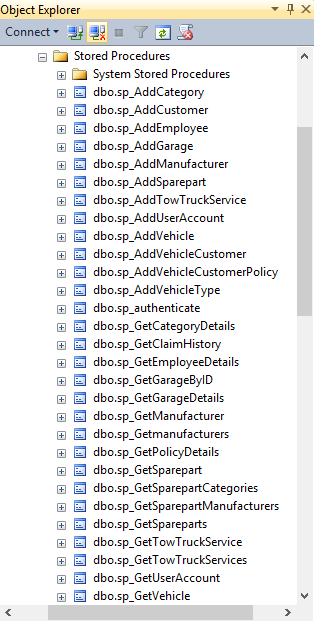


Figure : SQL Server Stored Procedures

Below shown is an example stored procedure taken from this solution.

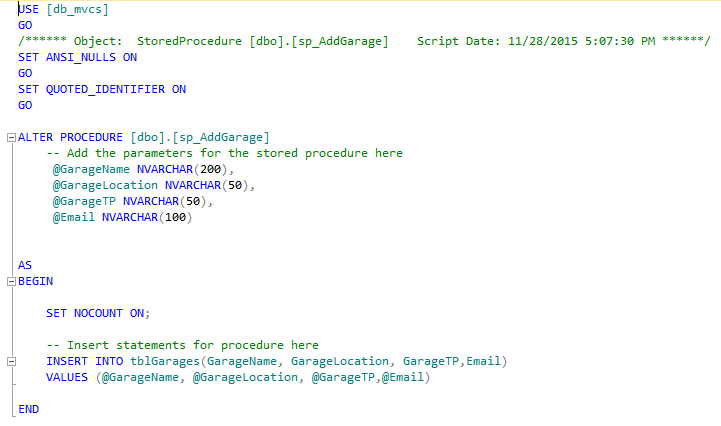


Figure : Sample Stored Procedure

# **6. 4 Web Application Development Architecture**

A layered architecture approach has being adopted when developing the web application. WCF web service also has being added as a layer in the project. Therefore main layers in the web application solution are data layer, business layer, model layer, view layer and the WCF service layer.

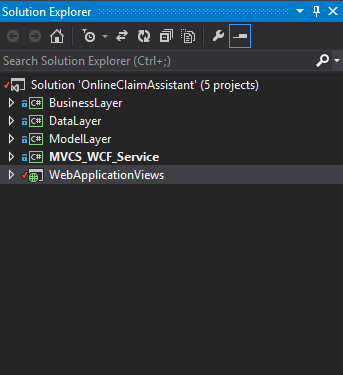


Figure : Project Structure

# **6.4.1 View Layer**

This layer is the visible layer for end users. All the front end designs are coded in this layer. Views are created with aspx extension web forms connecting to a single master page. In the master page all the libraries such as bootstrap, font awesome are imported using their CDN libraries.



Figure : CDN Librbay Links

Basic html and asp.net tags have being used to develop the front end. In the code behind communication with the model layer and business layer are maintained. All the button click events are coded in code behind using c#.

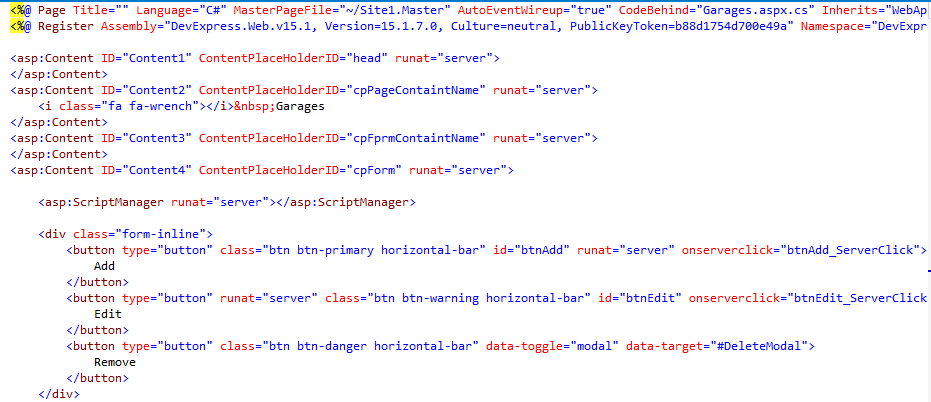


Figure : Sample from Front End

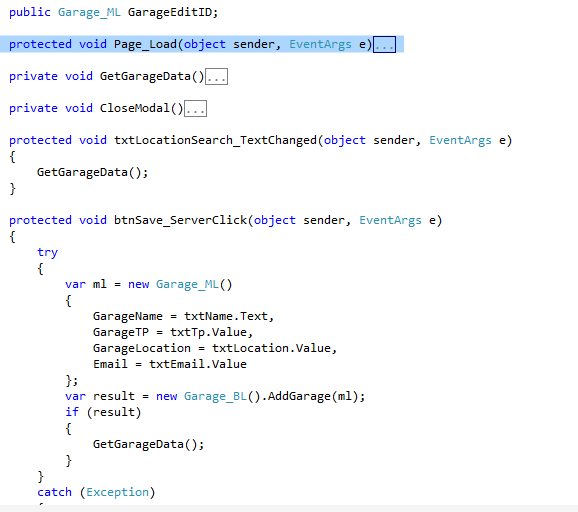


Figure : Sample from Code behind

# **6.4.2 Model Layer**

Model layer was used to define the model of the available class objects. Defaults constructor as being overridden in order to set values in the instance of creating objects. After creating an object of a particular class it’s being passed as parameters for other layers when processing. A sample of a model class is illustrated in the below code.

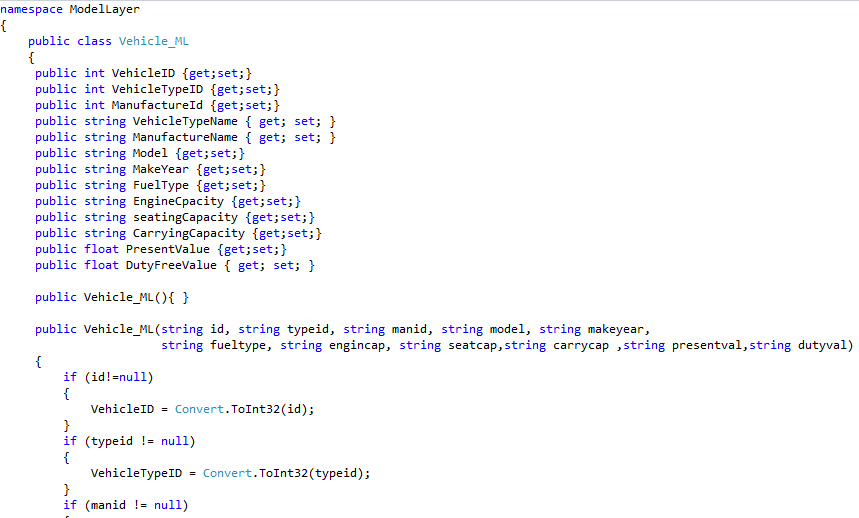


Figure : Sample Class from Model Layer

# **6.4.3 Business Layer**

This layer handles all the business logics can communications with the data layer. All the parameters required for a transaction with the database are provided at the business layer.



Figure : Sample Clsss from Business Layer

# **6.4.4 Data Layer**

Data layer is the responsible layer for communication with the database. All the constants need for are stored in a separate class called Constants. This layer contains the data base connection class as well. Data connection string is stated in the web.config file so that it can be easily changed in the implementation phase.

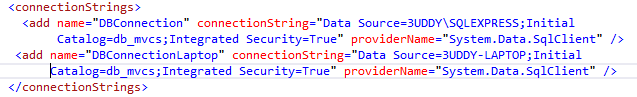


Figure : Connection Strings

It will be inherited by the database access controller class. Following figure shows a sample code from a data layer class. This layer classes are developed such that it can be reusable as a DLL reference for a different project.



Figure : Sa,ple Code from Data Layer Class

# **6.4.5 WCF Web service**

This is the layer where the web service is created for the communication with tab application and mobile application. Web service will also communicate with the database through data layers. In the service there is an interface class also to abstract the exposed methods. In the service class all the implementations of the methods are done. In web services simply method calls are done.

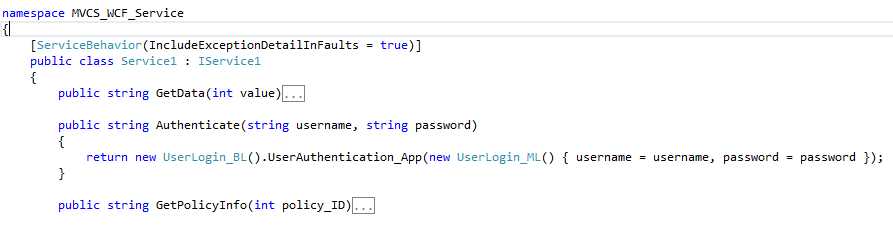
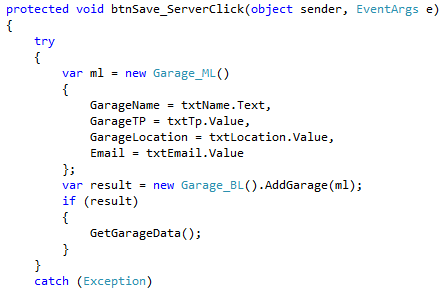


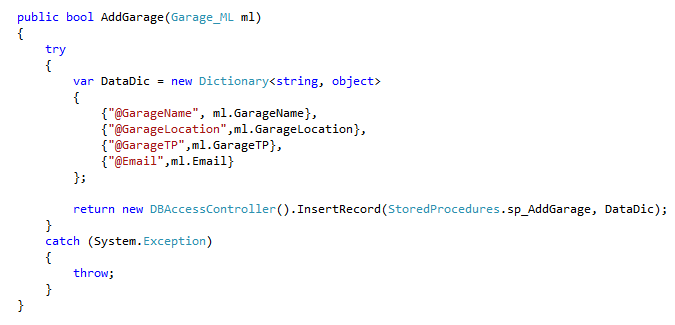
Figure : Sample Code from Web Service

# **6.4.6 Communication between Layers**

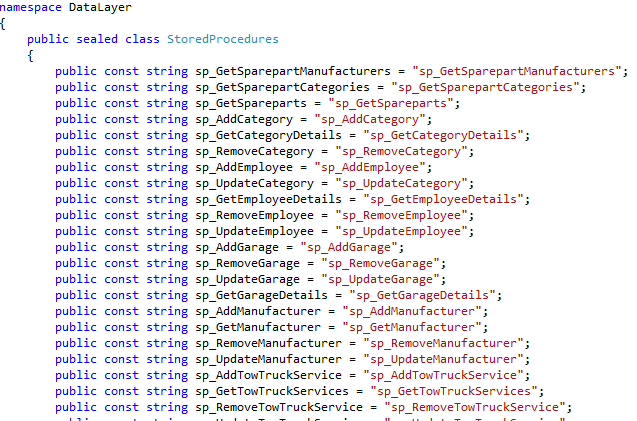
This section will discuss how the communication will happen between the layers in the time of execution. Let’s consider a situation where we are in add a garage user interface and we are adding a new garage in to database. When the save button is clicked from the front end aspx file save button click event will be fired.



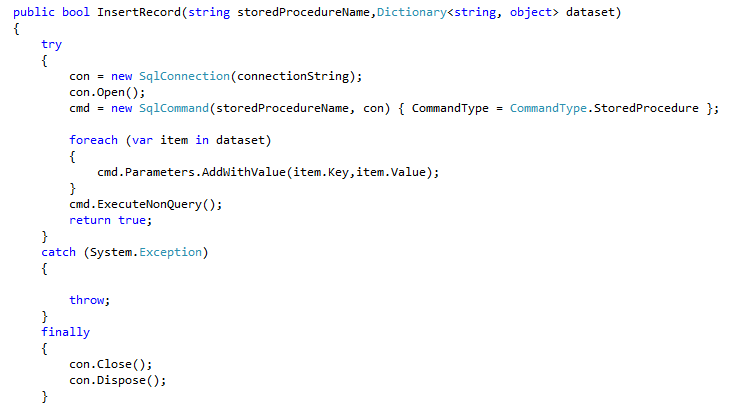
In that method a new object will be create from Garage model class in the model layer. Then in the object initialize constructor values from the text boxes will be assigned to the models attributes. Then a new object from the Garage business layer class will be created and a method name AddGarage () will be invoked passing the model object we created as the parameter.



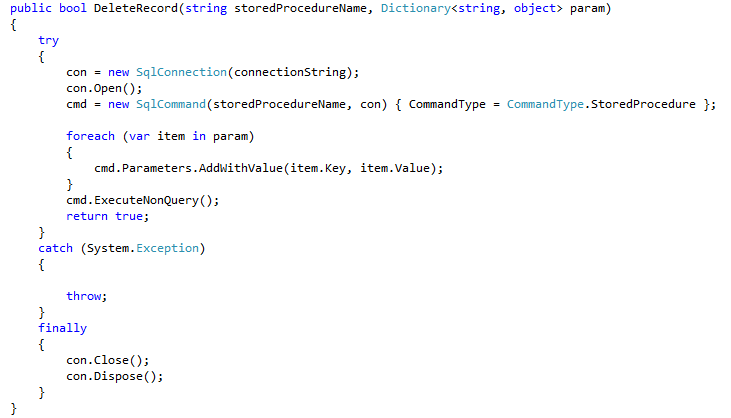
In the business layer Garage classes AddGarage() methods a new Dictionary type object will be created with passed values through the Garage model object. Dictionary is a visual studio data structure which uses key value pairs. Then a new object will be created from data layers DBAccessController class and a method name Insert Record will be called passing stored procedure and data dictionary as parameters. In here the stored procedures are stored in a class called constants.



It is considered as a good coding practice to keep constants in a one separate sealed class. The data layers insert method is shown below.



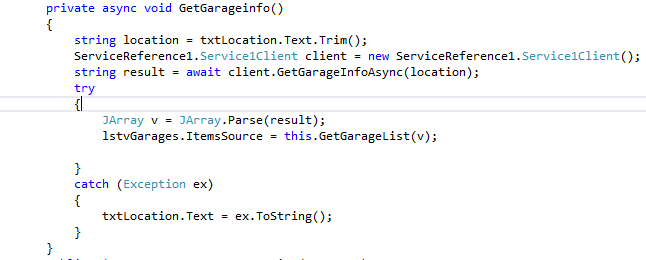
In this method it is looped though the data dictionary to get the values and pass them as parameters for the stored procedures. By coding this way it is possible to use this same insert method to carry out any other insert operation of data. Similar approach has being use for other operations for example update and remove also.



When considering the WCF web service it will also communicate with the database through above discussed communication methodology.

# **6.5 Tablet PC and Smartphone Application Development Architecture**

Similar architecture has being used here also namely layered architecture. Since there are very less business logics have being written in tab application only two layers namely view and model layers. Interfaces are developed with XAML and backend coding was done using c#. In the code behind call events for web services id coded.



# **6.6 Summary**

This chapter was dedicated to provide a discussion about development and development techniques as well as the development process of the application by following best practices of the software development. Tools techniques, coding impartments, architectures that used in the development of the applications were also discussed in this chapter.