**Chapter 3: DISCRIPTION OF TOOLS**

**3.1 Visual Studio**

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps. [2] Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. [3] It can produce both native code and managed code. Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).[3]

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++ and C++/CLI (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), F# (as of Visual Studio 2010[3]) and Type Script (as of Visual Studio 2013 Update 2).[4]Support for other languages such as Python, Ruby, Node.js, and M among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Java (and J#) were supported in the past. Microsoft provides a free version of Visual Studio called the Community edition that supports plugins and is available at no cost. [4]

**Architecture**

Visual Studio does not support any programming language, solution or tool intrinsically; instead, it allows the plugging of functionality coded as a VS Package. When installed, the functionality is available as a Service. The IDE provides three services: SVs Solution, which provides the ability to enumerate projects and solutions; SVs UI Shell, which provides windowing and UI functionality (including tabs, toolbars and tool windows); and SVs Shell, which deals with registration of VS Packages. In addition, the IDE is also responsible for coordinating and enabling communication between services. All editors, designers, project types and other tools are implemented as VS Packages. Visual Studio uses COM to access the VS Packages. The Visual Studio SDK also includes the Managed Package Framework (MPF), which is a set of managed wrappers around the COM-interfaces that allow the Packages to be written in any CLI compliant language.[1] However, MPF does not provide all the functionality exposed by the Visual Studio COM interfaces. The services can then be consumed for creation of other packages, which add functionality to the Visual Studio IDE. [5]

Support for programming languages is added by using a specific VS Package called a Language Service. A language service defines various interfaces which the VS Package implementation can implement to add support for various functionalities. Functionalities that can be added this way include syntax coloring, statement completion, brace matching, parameter information tooltips, member lists and error markers for background compilation. If the interface is implemented, the functionality will be available for the language. Language services are implemented on a per-language basis. [6] The implementations can reuse code from the parser or the compiler for the language. Language services can be implemented either in native code or managed code. For native code, either the native COM interfaces or the Babel Framework (part of Visual Studio SDK) can be used. For managed code, the MPF includes wrappers for writing managed language services. [7]

Visual Studio does not include any source control support built in but it defines two alternative ways for source control systems to integrate with the IDE.A Source Control VS Package can provide its own customized user interface. In contrast, a source control plugin using the MSSCCI (Microsoft Source Code Control Interface) provides a set of functions that are used to implement various source control functionality, with a standard Visual Studio user interface. MSSCCI was first used to integrate Visual SourceSafe with Visual Studio 6.0 but was later opened up via the Visual Studio SDK. Visual Studio .NET 2002 used MSSCCI 1.1, and Visual Studio .NET 2003 used MSSCCI 1.2. Visual Studio 2005, 2008 and 2010 use MSSCCI Version 1.3, which adds support for rename and delete propagation as well as asynchronous opening. [7]

Visual Studio supports running multiple instances of the environment (each with its own set of VS Packages). The instances use different registry hives (see MSDN's definition of the term "registry hive" in the sense used here) to store their configuration state and are differentiated by their App Id (Application ID). [8] The instances are launched by an App Id-specific .exe that selects the App Id, sets the root hive and launches the IDE. VS Packages registered for one App Id are integrated with other VS Packages for that App Id. The various product editions of Visual Studio are created using the different App Ids. The Visual Studio Express edition products are installed with their own App Ids, but the Standard, Professional and Team Suite products share the same App Id. Consequently, one can install the Express editions side-by-side with other editions, unlike the other editions which update the same installation. The professional edition includes a superset of the VS Packages in the standard edition and the team suite includes a superset of the VS Packages in both other editions. The App Id system is leveraged by the Visual Studio Shell in Visual Studio 2008. [9]

**Features**

**Code editor**

Like any other IDE, it includes a code editor that supports syntax highlighting and code completion using IntelliSense for variables, functions, methods, loops and LINQ queries. IntelliSense is supported for the included languages, as well as for XML and for Cascading Style Sheets and JavaScript when developing web sites and web applications. Autocomplete suggestions appear in a modeless list box over the code editor window, in proximity of the editing cursor. In Visual Studio 2008 onwards, it can be made temporarily semi-transparent to see the code obstructed by tithe code editor is used for all supported languages. The Visual Studio code editor also supports setting bookmarks in code for quick navigation. Other navigational aids include collapsing code blocks and incremental search, in addition to normal text search and regex search. The code editor also includes a multi-item clipboard and a task list. [10]The code editor supports code snippets, which are saved templates for repetitive code and can be inserted into code and customized for the project being worked on. A management tool for code snippets is built in as well. These tools are surfaced as floating windows which can be set to automatically hide when unused or docked to the side of the screen.[10] The Visual Studio code editor also supports code refactoring including parameter reordering, variable and method renaming, interface extraction and encapsulation of class members inside properties, among others. [1]

Visual Studio features background compilation (also called incremental compilation). As code is being written, Visual Studio compiles it in the background in order to provide feedback about syntax and compilation errors, which are flagged with a red wavy underline. Warnings are marked with a green underline. Background compilation does not generate executable code, since it requires a different compiler than the one used to generate executable code. Background compilation was initially introduced with Microsoft Visual Basic but has now been expanded for all included languages.

**3.2 Web Application**

**ASP.NET**

ASP.NET is an open-source server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages. ASP.NET's successor is ASP.NET Core. It is a re-implementation of ASP.NET as a modular web framework, together with other frameworks like Entity Framework. The new framework uses the new open-source .NET Compiler Platform (codename "Roslyn") and is cross platform. ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages) have merged into a unified MVC 6. [2]

**Characteristics**

ASP.NET Web pages, known officially as Web Forms, are the main building blocks for application development in ASP.NET. [3] There are two basic methodologies for Web Forms, a web application format and a web site format. Web applications need to be compiled before deployment, while web sites structures allows the user to copy the files directly to the server without prior compilation. Web forms are contained in files with an ".aspx" extension; these files typically contain static (X) HTML markup or component markup. The component markup can include server-side Web Controls and User Controls that have been defined in the framework or the web page. For example, a textbox component can be defined on a page as <asp: textbox id='myid' runat='server'>, which is rendered into an html input box.

**3.3 C# programming language**

C# (pronounced as see sharp) is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure. C# is a general-purpose, object-oriented programming language. Its development team is led by Anders Hejlsberg. [1]The most recent version is C# 7.1, which was released in 2017 along with Visual Studio 2017 Update 3.

### **Typing**

C# supports strongly typed implicit variable declarations with the keyword var, and implicitly typed arrays with the keyword new [2] followed by a collection initializer. C# supports a strict [Boolean data type](https://en.wikipedia.org/wiki/Boolean_data_type), bool. Statements that take conditions, such as while and require an expression of a type that implements the true operator, such as the Boolean type. While C++ also has a Boolean type, it can be freely converted to and from integers, and expressions such as only that is convertible to bool, allowing to be an int, or a pointer. C# disallows this "integer meaning true or false" approach, on the grounds that forcing programmers to use expressions that return exactly can prevent certain types of programming mistakes such as (use of assignment = instead of equality ==, which while not an error in C or C++, will be caught by the compiler anyway).

C# is more than C++. The only implicit conversions by default are those that are considered safe, such as widening of integers. [5] This is enforced at compile-time, during, and, in some cases, at runtime. No implicit conversions occur between Booleans and integers, nor between enumeration members and integers (except for literal 0, which can be implicitly converted to any enumerated type). Any user-defined conversion must be explicitly marked as explicit or implicit, unlike C++ and conversion operators, which are both implicit by default. [7]

**3.4 Database**

**SQL server database 2014**

SQL Server Database (commonly referred to as SQL RDBMS or simply as SQL) is an object-relational database management system produced and marketed by SQL Maria DB. Larry Ellison and his two friends and former co-workers, Bob Miner and Ed Oates, started a consultancy called Software Development Laboratories (SDL) in 1977. SDL developed the original version of the Oracle software. The name Oracle comes from the code-name of a CIA-funded project Ellison had worked on while formerly employed by Apex. [8]

A [cloud database](https://en.wikipedia.org/wiki/Cloud_database) is a [database](https://en.wikipedia.org/wiki/Database) that runs on a [cloud computing](https://en.wikipedia.org/wiki/Cloud_computing) platform, and access to it is provided as a service. Managed database services take care of scalability, backup, and high availability of the database. Azure SQL Database is a managed database service which is different from AWS RDS which is a container service.Microsoft Azure SQL Database includes built-in intelligence that learns app patterns and adapts to maximize performance, reliability, and data protection. It was originally announced in 2009 and released in 2010. [9]

**Pricing**

Azure SQL Database is offered either as a Standalone database or Elastic database pool, and is priced in three tiers: Basic, Standard and Premium. Each tier offers different performance levels to accommodate a variety of workloads. The resources available for Standalone databases are expressed in terms of Database Transaction Units (DTUs) and for elastic pools in terms of elastic DTUs or eDTUs.[10] A DTU is defined as a blended measure of CPU, memory, and data I/O and transaction log I/O in a ratio determined by an OLTP benchmark workload designed to be typical of real-world OLTP workloads.Databases are available as Standalone databases or in database pools which allow multiple databases to share storage and compute resources.[10]