Online Tax Information System Asp.Net

Pothuluri Sravanthi
Student(M.Tech) , CSE,
Gokul Institue of Technology and Science,
Visakhapatnam, India.

P.Sandhya
Asst. Prof, CSE,
Gokul Institue of Technology and Science,
Visakhapatnam, India.

ABSTRACT:

This project proved good for me as it provided practical knowledge of not only programming in ASP.NET andC#.NET web based application and no some extent Windows Application and SQL Server, but also about all handling procedure related with “Geo Spatial Info Tech Solutions”. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future.

I INTRODUCTION:

Local bodies like municipalities need collect different kind of taxes from the citizens. Every citizen need to pay Water Bills and Electricity Bills to the particular authorities. Actually a citizen want to pay his municipal tax means he should go to municipal office and collect the necessary bill, pay there only. Same as water and electricity bills. This is not an easy job to get all these tax information in different authorities now days. For that purpose we are developing this application, which can collect the tax information from different authorities based on the house no’s and owner name by using Web Services and stores the data in our own database. For getting the information regarding taxes citizens need to register first. The bills are calculating on different modes like annual, half yearly, quarterly etc.

1.1 ORGANIZATION PROFILE:

Software Solutions is an IT solution provider for a dynamic environment where business and technology strategies converge. Their approach focuses on new ways of business combining IT innovation and adoption while also leveraging an organization’s current IT assets. Their work with large global corporations and new products or services and to implement prudent business and technology strategies in today’s environment.

1.2 PROBLEM IN EXISTING SYSTEM:

The existing system is a manual system. Here the employees needs to save the information in the form of excel sheets or Disk Drives.

There is no sharing is possible if the data is in the form of paper or Disk drives. The manual system gives us very less security for saving data; some data may be lost due to is management. It’s a limited system and fewer users friendly. Searching of particular information is very critical it takes lot of time. Gathering information of different sources is not an easy job, data will be mismanaged. Calculating different bills manually going to be as a process of mistake. There is no interface to provide various bills in the existing system. In the existing system citizen get the bills information manually, in various authorities like water board, electricity department, property tax department etc.

1.3 SOLUTION OF THESE PROBLEMS:

The development of this new system contains the following activities, which try to automate the entire process keeping in the view of database integration approach. User Friendliness is provided in the application with various controls provided by system Rich User Interface. The system makes the overall project management much easier and flexible. It can be accessed over the Intranet. The employee information can be stored in centralized database which can be maintained by the system.

This can give the good security for user information because data is not in client machine. Authentication is provided for this application only registered Users can access. There is no risk of data management at any level while the project development is under process. The automated system will provide to the employees for reliable services. The proposed system using web services, a web service can get the information from other sources also.

1.4 SYSTEM ANALYSIS:

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system.

Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

II ANALYSIS MODEL: SDLC METHDOLOGIES:

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, "A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system. A preliminary design is created for the new system. A first prototype of the new system is constructed from the preliminary design.

This is usually a scaled-down system, and represents an approximation of the characteristics of the final product. A second prototype is evolved by a fourfold procedure: Evaluating the first prototype in terms of its strengths, weakness, and risks. Defining the requirements of the second prototype. Planning an designing the second prototype. Constructing and testing the second prototype.

At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.

The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above. The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.

The final system is constructed, based on the refined prototype. The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

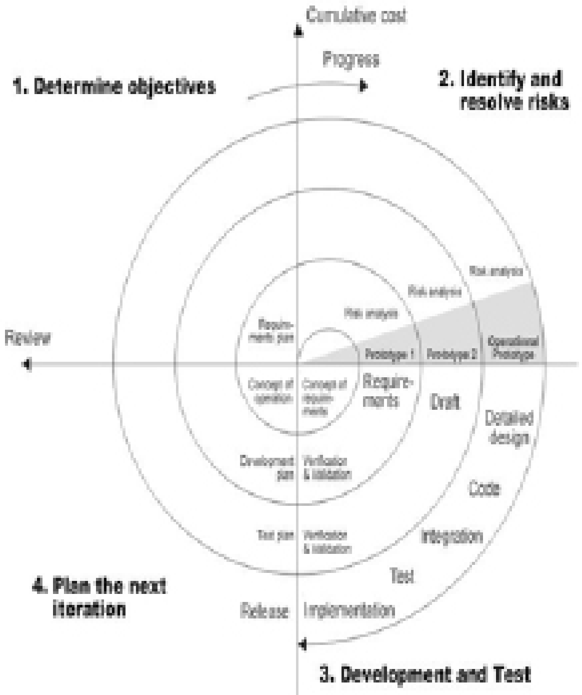


Fig 2.1 Spiral Model

III STUDY OF THE SYSTEM:

In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a browser interface. The GUI'S at the top level have been categorized as:

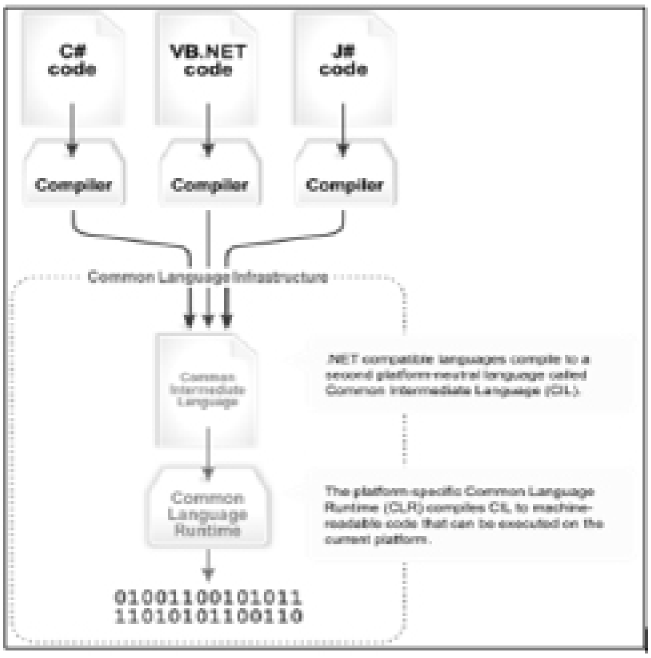
1.Administrative user interface.

2.The operational or generic user interface.

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The interfaces help the administrations with all the transactional states like Data insertion, Data deletion and Data updating along with the extensive data search capabilities.

The operational or generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.

ARCHITECTURE:


C# code
VB.NET code
J# code
Compiler
Compiler
Compiler
Common Language Infrastructure
Common Intermediate Language
.NET compatible languages compile to a second platform-neutral language called Common Intermediate Language (CIL).
Common Language Runtime
The platform-specific Common Language Runtime (CLR) compiles CIL to machine-readable code that can be executed on the current platform.
01001100101011
11010101100110

3.1 Visual overview of the Common Language Infrastructure (CLI)

3.1 COMMON LANGUAGE INFRASTRUCTURE:

The core aspects of the .NET framework lie within the Common Language Infrastructure, or CLI. The purpose of the CLI is to provide a language-neutral platform for application development and execution, including functions for exception handling, garbage collection, security, and interoperability. Microsoft's implementation of the CLI is called the Common Language Runtime or CLR.

3.2 ASSEMBLIES:

The intermediate CIL code is housed in .NET assemblies. As mandated by specification, assemblies are stored in the Portable Executable (PE) format, common on the Windows platform for all DLL and EXE files. The assembly consists of one or more files, one of which must contain the manifest, which has the metadata for the assembly. The complete name of an assembly (not to be confused with the filename on disk) contains its simple text name, version number, culture, and public key token.

The public key token is a unique hash generated when the assembly is compiled, thus two assemblies with the same public key token are guaranteed to be identical from the point of view of the framework. A private key can also be specified known only to the creator of the assembly and can be used for strong naming and to guarantee that the assembly is from the same author when a new version of the assembly is compiled (required to add an assembly to the Global Assembly Cache).

3.4 METADATA:

All CLI is self-describing through .NET metadata. The CLR checks the metadata to ensure that the correct method is called. Metadata is usually generated by language compilers but developers can create their own metadata through custom attributes. Metadata contains information about the assembly, and is also used to implement the reflective programming capabilities of .NET Framework.

3.5 SECURITY:

.NET has its own security mechanism with two general features: Code Access Security (CAS), and validation and verification. Code Access Security is based on evidence that is associated with a specific assembly. Typically the evidence is the source of the assembly (whether it is installed on the local machine or has been downloaded from the intranet or Internet). Code Access Security uses evidence to determine the permissions granted to the code.

Other code can demand that calling code is granted a specified permission. The demand causes the CLR to perform a call stack walk: every assembly of each method in the call stack is checked for the required permission; if any assembly is not granted the permission a security exception is thrown.

3.6 VERSIONS:

Microsoft started development on the .NET Framework in the late 1990s originally under the name of Next Generation Windows Services (NGWS). By late 2000 the first beta versions of .NET 1.0 were released.

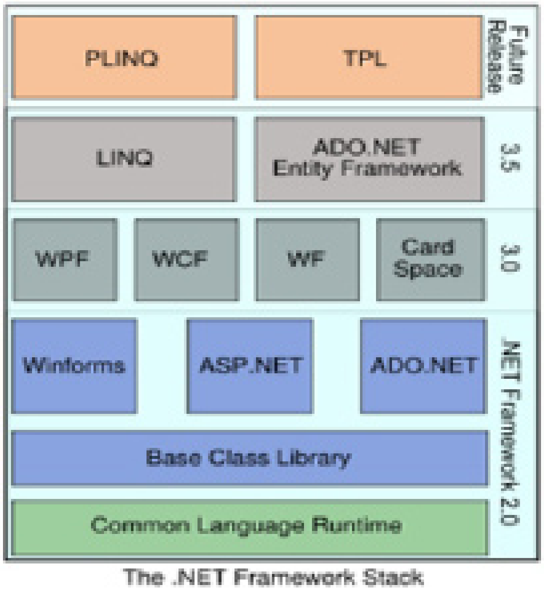
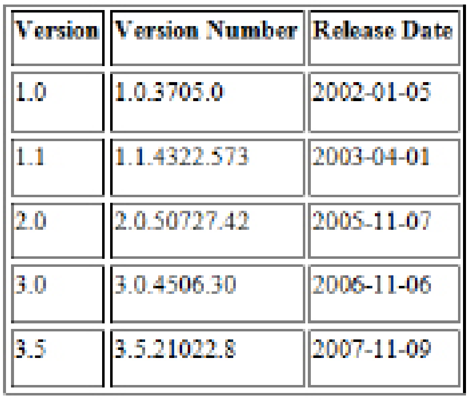


Fig: 3.2 .NET Framework stack.



Version	Version Number	Release Date
1.0	1.0.3705.0	2002-01-05
1.1	1.1.4322.573	2003-04-01
2.0	2.0.50727.42	2005-11-07
3.0	3.0.4506.30	2006-11-06
3.5	3.5.21022.8	2007-11-09

Fig: 3.3 Version table

IV SERVER APPLICATION DEVELOPMENT:

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server. The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code.

4.1 SERVER-SIDE MANAGED CODE:

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications.

However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web services use IIS and ASP.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET Framework.

XML Web services, an important evolution in Web-based technology, are distributed, server-side application components similar to common Web sites. However, unlike Web-based applications, XML Web services components have no UI and are not targeted for browsers such as Internet Explorer and Netscape Navigator. Instead, XML Web services consist of reusable software components designed to be consumed by other applications, such as traditional client applications, Web-based applications, or even other XML Web services. As a result, XML Web services technology is rapidly moving application development and deployment into the highly distributed environment of the Internet.

If you have used earlier versions of ASP technology, you will immediately notice the improvements that ASP.NET and Web Forms offers. For example, you can develop Web Forms pages in any language that supports the .NET Framework. In addition, your code no longer needs to share the same file with your HTTP text (although it can continue to do so if you prefer). Web Forms pages execute in native machine language because, like any other managed application, they take full advantage of the runtime. In contrast, unmanaged ASP pages are always scripted and interpreted. ASP.NET pages are faster, more functional, and easier to develop than unmanaged ASP pages because they interact with the runtime like any managed application.

The .NET Framework also provides a collection of classes and tools to aid in development and consumption of XML Web services applications. XML Web services are built on standards such as SOAP (a remote procedure-call protocol), XML (an extensible data format), and WSDL (the Web Services Description Language). The .NET Framework is built on these standards to promote interoperability with non-Microsoft solutions.

If you develop and publish your own XML Web service, the .NET Framework provides a set of classes that conform to all the underlying communication standards, such as SOAP, WSDL, and XML. Using those classes enables you to focus on the logic of your service, without concerning yourself with the communications infrastructure required by distributed software development.

Finally, like Web Forms pages in the managed environment, your XML Web service will run with the speed of native machine language using the scalable communication of IIS.

4.2 UNIFIED MODELING LANGUAGE DIAGRAMS:

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

User Model View:

1.This view represents the system from the users perspective.

2.The analysis representation describes a usage scenario from the end-users perspective.

SERVER SIDE VALIDATION:

Some checks cannot be applied at client side. Server side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server side checks imposed is:

Server side constraint has been imposed to check for the validity of primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results into a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values. User is intimating through appropriate messages about the successful operations or exceptions occurring at server side. Various Access Control Mechanisms have been built so that one user may not agitate upon another.

Access permissions to various types of users are controlled according to the organizational structure. Only permitted users can log on to the system and can have access according to their category. User- name, passwords and permissions are controlled o the server side.Using server side validation, constraints on several restricted operations are imposed.

V CONCLUSION:

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in ASP.NET andC#.NET web based application and no some extent Windows Application and SQL Server, but also about all handling procedure related with "Geo Spatial Info Tech Solutions". It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future.

The project is identified by the merits of the system offered to the user. The merits of this project are as follows:

- It's a web-enabled project.

- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.

- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updation so that the user cannot enter the invalid data, which can create problems at later date.

LIMITATION:

- The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.

- Training for simple computer operations is necessary for the users working on the system.

VI REFERENCES:

- FOR .NET INSTALLATION

www.support.mircosoft.com

- FOR DEPLOYMENT AND PACKING ON SERVER

www.developer.com
www.15seconds.com

- FOR SQL

www.msdn.microsoft.com

- FOR ASP.NET

Asp.Net 3.5 Unleashed

www.msdn.microsoft.com/net/quickstart/aspplus/default.com

www.asp.net

www.fmexpense.com/quickstart/aspplus/default.com

www.asptoday.com

www.aspfree.com

www.4guysfromrolla.com/index.aspx

- Software Engineering (Roger's Pressman)