J2EE vs Microsoft.NET

A Comparison of two platforms for component-based development of web applications

➢ Developer’s point of view

By
Asad Iqbal - ail09002@student.mdh.se
Naeem Ullah - nuh09001@student.mdh.se
August 2010

Supervisor/Examiner:
Frank Lüders
frank.luders@mdh.se
ABSTRACT

Paper is interested for anyone such as software developers, students, designers and companies who want to choose best platform.

Paper provides powerful comparison between .NET web application using ASP.NET and Java application using JSF technology for component based development. The choice of suitable platform for future is not easy, both has its strength and weaknesses. Web applications implemented within specified requirements using component technology such as EJB and Class Library. This application helps to compare both platforms against each other and provide support to target audience to take decision while choosing platform. Although both .NET and J2EE have covers a great deal of platform fields, paper will especially focus on components, programming languages, tools and technologies. At the end, we discussed our point of view while working on both applications using component and provide which platform is easier and user friendly for target audience.

Keywords: Comparison, .NET, J2EE, Component, EJB, developer’s point of view
Sammanfattning


Keywords:  Comparison, .NET, J2EE, Component, developer point of view
## Contents

Abstract ........................................................................................................................................................... 2

1. INTRODUCTION .................................................................................................................................. 6

2. COMPONENTS ..................................................................................................................................... 7
   2.1. What Is Component? ................................................................. 7
   2.2. Interfaces ................................................................................................. 9
   2.3. Component Versioning ........................................................................... 10
   2.4. Reuse Of Components: ........................................................................... 10

3. MICROSOFT .NET AND ASSEMBLIES: .......................................................................................... 12
   3.1. Common Language Infrastructure (CLI): ........................................ 12
   3.2. Common Language Runtime: ............................................................. 12
   3.3. What is .NET ............................................................................................. 12
   3.4. .NET Technology: ..................................................................................... 13
   3.5. .NET Component: .................................................................................... 15
       3.5.1. Assembly: ........................................................................................... 15
       3.5.2. Metadata: ........................................................................................... 17

4. ENTERPRISE Java BEANS COMPONENTS .................................................................................... 19
   4.1. Enterprise Java Beans (EJB) ................................................................. 19
   4.2. EJB Technology: ..................................................................................... 20
   4.3. Types of Enterprise JavaBeans Components ........................................ 22
       4.3.1. Session Beans: ................................................................................ 23
       4.3.2. Entity Beans: ..................................................................................... 25
       4.3.3. Message Driven Beans: .................................................................... 28

5. EXAMPLE APPLICATION OVERVIEW.......................................................................................... 30
   5.1. Project Methodology ............................................................................. 30
   5.2. Goals ......................................................................................................... 30
   5.3. Website specifications: ........................................................................... 30
       5.3.1. Overview ........................................................................................... 30
       5.3.2. Client website overview ................................................................. 31
       5.3.3. Administrator website overview ..................................................... 36

6. EXAMPLE APPLICATION IMPLEMENTATION ............................................................................. 40
   6.1. Environment ............................................................................................. 41
6.2. Database Designing ............................................................................................................................ 42

6.3. .NET Implementation .......................................................................................................................... 43
   6.3.1. .NET Component Creation ............................................................................................................. 45
   6.3.2. Master Pages .................................................................................................................................... 46
   6.3.3. Welcome page .................................................................................................................................. 47
   6.3.4. Registration Page ............................................................................................................................. 47
   6.3.5. User Account Page .......................................................................................................................... 48
   6.3.6. Administrator Ticket Management Page ........................................................................................ 48

6.4. J2EE Implementation ............................................................................................................................ 48
   6.4.1. EJB Components .............................................................................................................................. 50
   6.4.2. Faceletes Templates ......................................................................................................................... 51
   6.4.3. Welcome page .................................................................................................................................. 51
   6.4.4. Registration Page ............................................................................................................................. 52
   6.4.5. User Account Page .......................................................................................................................... 52
   6.4.6. Administrator Ticket Management Page ........................................................................................ 52

7. COMPARISON ........................................................................................................................................ 54
   7.1. Tools .................................................................................................................................................... 54
   7.2. Web Technologies .............................................................................................................................. 55
   7.3. Programming Languages .................................................................................................................. 57
   7.4. Components ........................................................................................................................................ 57

8. CONCLUSION ........................................................................................................................................... 59

REFERENCES .............................................................................................................................................. 60
1. INTRODUCTION

Many questions are arising in developer’s mind, which technology is easy to use and quick to learn for development that makes their future bright. Many young developers, who want to start their career with development, are getting frustration to choose better platform for their future. Lists of products from different companies are in the way of developers and all are claiming quality product. It’s really very difficult for developer to take a decision in favor of any platform and start his career. If we observe market condition then Microsoft and Sun are high in the market peak in technology world. Both are competing with each other with .NET and J2EE platform and improving their products day by day. Both platforms are getting mature, have many similarities and providing a large number of feature lists. These similarities between two platform creating interest in both side of developer’s life and giving chance to them to compare both platform using different technologies provided by Microsoft and Sun. Visual studio from Microsoft and Netbeans tool from Sun is providing high level of features that are creating easiness and user friendly environment for developer. Thesis is focus on comparing both platforms by implementing web application using software tools and suggest solutions for other developers so they can choose best of one from each. Thesis is creating interest for developer to compare both technologies component. The EJB (Enterprise Java Beans) technology is getting more popularity in the market and .NET technology is fairly new but both are similar to each other in many ways. Both technologies use for to develop components. Now question arise which technology is good to implement for certain tasks. We implemented both of technologies by experimentally, evaluated and provide solutions to the target audience through theoretically by finding strength and weaknesses. We developed two kind of web application using EJB and .NET for our experiment to know how technologies work well in reality using Microsoft Visual Studio and Netbeans software tools.
2. COMPONENTS

2.1. What Is Component?

Before going to anywhere else, let us define the component according to Oxford Learners Dictionary. This dictionary defines the component like this,

"A component is any part of which something is made"

According to Bosch [1] definition the component is

"A software component is a unit of composition with explicitly specified provided, required and configuration interfaces and quality attributes."

Now according to the above Bosch component definition, a component is a unit, piece or part of software, which provided three types of interfaces Provided, Required and Configuration. These three types of interfaces must be express clearly. These interfaces will be explained later on. According to definition, a component also have a quality attributes. These quality attributes must also be clearly expressed and defined.

Now let us define the component and Objects (OOPs). According to the definition of component and objects, they are not similar to each other. Szyperski [2] define the component and object like this

**Component:** Component is the basic independent deployment unit along with the third party composition and there is no persistence state for a component.

**Object:** Object has a unique identity and also a unit of instantiation. Unlike to component, object has state and it may be a persistence state. The behavior and state of an object are encapsulated. Object has the capability to encapsulate its behavior and state. Now what does the above component definition means?

A component must be able to deploy independently from its environment and also from other components. It must be a separate and independent unit of software, which encapsulate its feature.

Third party composition means that no one can be able to access the component construction details. It means that a component must be self contained and it must also be able to communicate with other third party components. Furthermore, what a component actually needs as an input to perform its functions and what a component provided as an output. These specifications must be clear for components. This means that interfaces must be clearly and fully defined for components.

No persistence state means that when a component is trigged by a call to perform its task, the component will not perform any task until the next call is trigged. Between two calls the component will be unable to remember anything. The component will forget everything whatever it was intended to perform.
There is no need of multiple copies of a component in a system [2]. This statement of Szyperski is useful for non-distributed application or system. But for a distributed system redundancy is useful for some situations. Like when high availability is the quality requirement, then multiple copies of the components in a system is a possible solution to achieve these kind requirements. Put multiple copies of the same component on different site can increase the availability.

As object has state and the life time of these states are the entire runtime of the application. Object is instantiated and object has a unique ID as well as for a specific object, its state is also unique. Objects are instantiated from classes or from an existing object. A component can accommodate one or more classes. Szyperski says that it is also possible to put one class in a component but he think that class cannot be classified as a component.

As from the definition of Szyperski it is clear that component and object are not similar to each other. As we know that a component must contain one or more classes, so it will have an object during runtime. This somewhat goes against from the definition of Szyperski. He says that there is no state for a component (Components are stateless) and we also know that component contain classes. So from the definition of an object, it might have a state. For example if a component is used in a web server application then it must have a state.

According to Bosch [1], he thinks that component should be considering larger then a class. He says that Szyperski define the component very close to a class (A class in object oriented programming).

Now according to Bas, Clements and Kazman [3], they defined the component like this, a component can be a commercial product, database, library, an object, a process and more. Actually they define the component in a more comprehensive way. From this definition it is clear that they are agreed and at the same time they are disagree with Bosch and Szyperski.

Szyperski [1] describe four types of components White Box, Black Box, Glass Box and Gray Box.

White Box: This kind of component describes every thing inside in it. So one can check the implementation and also edit and modify it.

Black Box: This kind of component did not disclose its implementation part. It only provides its interface to use the component features.

Glass Box: In Glass Box component, one can check the entire implementation part but did not allow to modifying the implementation part.

Gray Box: In Gray Box component, one can check some limited parts of the Implementation but not the entire implementation.

According to Shown and Garlar [4], they define two groups of components, one is Primitive and the other one is Composite components.

Primitive Components: These kinds of components are implemented in a conventional programming language.
Composite Components: These kinds of components are defined for configuration using notation types. They are totally independent of any programming language.

2.2. Interfaces

The developers should think about when developing components that components should not depend on each other, they could easily integrate with each other and use each other services they provided. Interfaces are used to gain these kinds of services [1]. Actually the interface provides access to the service provided by a component. Two components can communicate with each other through contract. Interfaces are used to define the contracts. Component provides different kind of services, for each of these services there are pre and post conditions. Contracts are used to contain the pre and post condition information [5]. Actually the provided and required information are included in a contract.

As mentioned above there are three types of interfaces provided, required and configuration interfaces [1]. These three types of interfaces are used to define what a component actually require as an input to perform its functions, what a component provide as an output and what kind of information is needed for adjustment. These interfaces should be defined for components, they are assumed to be reuse somehow as well as to communicate with each other. As mentioned above contracts are specified for these interfaces. The component coupling can be decrease by using these interfaces.

For component adjustment, the configuration interfaces are used. It is the most difficult job when selecting or developing component to know all the possible features of the component. It is very difficult for the developers to build a right component for a system if they do not know about the system internally. The component developers should think about the variation level in the component development time that it should fit with different systems and client needs. In this way the component should be adjust in other system also. Multiple interfaces are provided to fulfill the needs of different clients and systems.

When communicating with the components the most useful are the interfaces, as mentioned above. These interfaces are actually the access points to the components features. One problem is encountered when components are interacting with each other and the way to handle this interaction. The developers must have to follow the component specified interfaces, when dealing with the components. Problems must be occurring with different versions of interfaces. To solve this kind of problem there are two ways to solve this kind of problems. One way is to maintain the compatibility and the other one is to use Immutable interfaces. The developers should try to maintain the compatibility issue with other older interfaces. Whenever the components communicate there must be a version control. The second method is that whenever a component interface is introduced, it should be fix and unchangeable. This approach is called immutable interface specifications. The interface version problem is solving with this way. Using immutable interface approach, the developers will be unable to add additional features for the new system and components. When two components communicate with each other by sending references, the receiving components simple accept the call if the components support the interface or decline call if it does not support the interface. The Microsoft's COM objects using the immutable interface approach.
2.3. Component Versioning

Unique name or number is used for versioning a component, which tell us the component development state. Different version of the component represent that there is something different inside the component, which some time create compatibility issue. Many techniques are used for component versioning. Release date of the component, sequence number and some time development state of the component are used for versioning. From these versioning techniques the developers can easily point out which version is new and which version is old. Alpha, Beta and Public are used for versioning, which represents the development state of the component. [6]

Whenever the code inside the component is changed and the developers want to deploy the component, the version should also be updated. When the changes inside the component affect the compatibility issue, then the major number in version should be changed. Major number represents compatibility issue. If the changes inside the component are small or insignificant, small number in version should be changed, which represent the difference from the previous version.

2.4. Reuse Of Components:

Reusable components are those components which have the ability to use in other products or system with a later version [1]. Reuse is the use of existing component in a context domain which is new. This is the common way now a day's. The second and the most important use of the reusable components are in the product line development. The third uses of the reusable components are, in later versions product, in different product from the same company or in other new organizations or companies. But this last use is not so popular. The developers should think about the following points when developing the components.

Components domain should be analyze deeply.

Component will use in a context.

The client or origination requirement about how will be the component reusable.

Small components are more reusable than large components because small components contain a small domain or only a small concept. Small components are easily adjusting then large component. So the adjustment efforts will be decrease with the use of small components. The reuses of component therefore involve the cost. However it takes time to find the right components and analyze its features, to check that it is possible to use it in the required system. Whenever implementing the components, the developers should think about very carefully how much reusable will be the components. Cost is involved when using the reusable component but this is the not the only thing as mentioned above. According to Szyperski[2], if the component is more reusable then it will be less useful. When implementing the component the developers should consider the three reusability level according to Bosch[1]. The first and the lowest level of reusability is "System Version". In this level, the components should be only reusable within a single system and different versions related to that single system, unusable with other systems. "Software Product Line” is the second level. In this level the component reusability is related to a range of different products with different companies. "Third Party Components” is the last and third level. This kind of components are call commercial of the shelf.
components. The reusability of such components is not only related to a single company but many other companies can use them as well.
3. MICROSOFT .NET AND ASSEMBLIES:

3.1. Common Language Infrastructure (CLI):

Common language infrastructure is the core of the .NET technology developed by Microsoft. CLI allows the developers to develop the application in different kinds of high level languages and allow the execution on different types of runtime system and environment. [7]

3.2. Common Language Runtime:

The applications can be developed in a variety of high level language. To manage the execution of these different kinds of high level language, Microsoft use the engine which is called common language runtime. This engine converts and compiles the source code of different high level languages to a common intermediate language and it is called Microsoft Intermediate Language (MSIL) [8]. The MSIL is converted into machine code during the application execution. For this conversion the Microsoft use the Just-In Time compiler and final output after the conversion is a Portable Execution file. The CLR using the Just In Time compilation to convert the code to a code which is independent of any language and hardware. So it’s the major advantage of the Microsoft .NET framework to allow the developers to write the code on their own choice. The other advantages are memory management, garbage collection, security, performance improvement, thread execution, exception handling, code safety verification and many more.

3.3. What is .NET

The popularity and the importance of the internet was growing up in 1995, Microsoft realized that they have to change their operating system and also to adjust the internet in a better way [9]. After this decision the Microsoft Window operating system has changed forever. Also they realized that the programming model (DNA) is not sufficiently enough for the upcoming demands. The distributed internet architecture (DNA) programming model architecture was based on the three layer architecture along with the com technology. For component development the DNA programming model was too complex and difficult. For advance component development the Microsoft introduced the .NET.

Many students and developer are confused about .NET, so let's explain .NET. Actually .NET is neither a programming language nor an operating system, but according to Meyer [10], it's only an abstraction to OS. One can easily integrate the .NET environment to an operating system. Microsoft developed the C# programming language and designed in such a way that it will work properly in .NET environment but C# is not the part of .NET, it only uses its environment for execution. Different high level programming languages are used to write programs for .NET using the .NET environment for execution and compilation [9]. .NET provides many services to .NET application or program whenever it communicates with the operating system resources. .NET environment contains a huge library, which provides different services to its application and programs.
Microsoft designed the .NET in such a way that it can compile the source code of any high level programming language to a common intermediate language (IL) as mentioned above. When the developers compile the .NET source code of an application, program or component, it is converted to MSIL (Microsoft Intermediate Language) code. During the execution the CLR interpret the MSIL code to machine code (Native code). CLR is the main control unit of .NET. Independently to programming languages the .NET defines the MSIL code in a common way according to the rules. The MSIL code is same for different programming languages used for .NET environment. In this way, virtually .NET is independent to programming languages and as well as to some extend a platform independent. To obtain platform independence, one can use the .NET environment in other platform also.

3.4. .NET Technology.

Common language runtime (CLR) is the heart and core of the .NET technology [9]. The CLR using JIT compiler (Just In Time Compiler) to convert and compile the MSIL code to machine code. This property of .NET is much similar to Java virtual machine (JVM). The following figure shows the .NET architecture.

![Figure 3.1 .NET Architecture.](image)
Now when the developers developed a .NET application in any language which support the .NET environment and now the developer wish to compile and execute the application in .NET environment. So let's explain the .NET program execution. First the developer will compile the program code. The traditional way as when the developer compile the program code, the developers get directly the runnable program which contain the machine code, but this is not the compilation method here in .NET. When the developer compile the program code, which will be run able within .NET environment, after the compilation the output will be one or more MSIL code files [Rob]. Actually in .NET the program code is not converted to machine code until the program code is called to run. CLR consists a number of JIT compilers, which are able to run in the .NET environment needs the CLR, is called Managed Code. Unlike to JVM, the CRL actually does not run the program code but is actually compile the program code to MSIL code. The following figure illustrates the compilation and execution of the .NET application.

![Figure 3.2 .NET Program Execution Model](image)

The MSIL code and Java byte code are both intermediate code, but the MSIL code is compiled just in time while the Java byte code is interpreted. Although the Java byte code can also be compile just in time. In .NET environment, the CLR only compiles the needed parts of the .NET application not the whole application, which save time when application is started. If the application needs another part, the CLR compile it at that time. The executable parts are then cached because if the application needs
it later on again, then there is no need to recompile it again. This means that only those parts are compiled to machine codes once which are called. Having these properties the .NET application are running as fast as the traditional executable. Due to JIT compilation, the .NET application execution will be not as faster as the traditional executable but in .NET environment the CLR compilation have many advantages. The CLR actually optimized the executable code for the processor type (Platform) and for the application, which hosting the CLR. CLR also checks the MSIL code before compiling it, in this way the security level is increased which prevents the illegal code to execute and run on the platform.

During execution the only use of the CLR's JIT compilers are that it use the Native Image Generator (Ngen.exe) tool within .NET environment. This tool is used to create binary images (Naïve Images) from the compiled MSIL code in assemblies. The Ngen.exe is used for performance improvement because during start up, the JIT compilers are not used but if it needed it will be activated. Global assembly cache contained a specific cache called native image cache. All the native images created from the MSIL code are cached in this cache. So during run time there is no need to use the JIT compiler to compile again the program code. The .NET technology will re-compile the native code automatically if there is any change in the system security.

Cross language integration and security are the two important features of CLR. The CLR can also manage profiling services, versioning deployment support, thread execution, debugging, memory management and many more. One of the .NET features is that if the memory is not in use anymore, it will be free automatically and it is called garbage collector [9].

3.5. .NET Component:
3.5.1. Assembly:

The basic unit for deployment and versioning in .NET is Assembly. Assembly is the deployment unit of EXE or DLL file. .NET also provides a version control for component (Assembly), individual evaluation of component (Assembly) and also control different version of the same component (Assembly). .NET also informs us if there is any incompatible component (Assembly) is used. [6]

Assemblies are used to developed the .NET application but it itself are not an application [9]. All the classes, which are used in a .NET application, are placed in one or more then one assemblies. Assemblies are not components, but these are used to store all the .NET components [9, 11]. The following figure illustrates the structure and contents of an assembly.
There are different ways to group the contents of an assembly and store the assembly rather than a single ordinary file. Assembly can contain one file or can be divided into multiple files. The following figure illustrates a single file assembly.

Figure 3.4 Single File Assemblies [14].

The following figure illustrates multiple files assembly.
The fact is that files are not used to store an assembly because files are not used to store the dynamic assemblies. If the assembly is grouped in multiple files then a main file must be there to define the entry point and also describe the other file of an assembly. Same directory and same hard drive must be used to locate the assembly otherwise if the assembly is dispersed in different directory or hard drive then it will be unable to work. If the resource file is also the part an assembly then it must also be locate in the same place where the assembly is. However the assembly is group in one or more files, the CLR is responsible to handle the assembly files as a single unit. The CLR consider the whole assembly as a single entity.

Two file extensions are used to represent an assembly, one is .exe and the other is .dll. Same method is used to structure these two files types but there is an additional start up sequence for .exe file, with help of this start up sequence the CLR will know how to start the execution [11]. For installation of these files, it can only be copy in the destination directory of an application. There is no traditional way to installation. An assembly must be registered before it work and it describe itself.

In .NET there are two types of assemblies, Private and Shared assembly.

Private assemblies are those, which are placed in a private location of an application and the application has a direct access to these assemblies. In private assembly there are negligible problems about the naming conflicts and versioning because each application using their own compatible assemblies without sharing with other applications.

Shared assemblies on the other hand can cause a lot of versioning problem because the shared assemblies are placed in a common global location and every application can access and use the same shared assemblies. That’s why the shared assembly must have a strong unique name and the most important is the version number. Special tools are used to install such type of assemblies to its destination and when the assembly is placed in its destination, it is checked automatically.

3.5.2. Metadata:

As already mentioned, assembly is used to store the .NET components. So each component and the assembly itself have own metadata [11, 9]. In previous Microsoft component technologies (COM, COM+), the metadata is placed in a separate location, which later on creates the problems during up gradation and in installations. Components registration is must before using it in these technologies. Now components and assembly have their own metadata, so there is no need to register the components. As already mentioned, assembly describes itself and metadata contain all information.
about a specific component when using it. There is a special post in metadata, which save the component versions. Due to this property, there is no component versioning problem, when components are updated or used in a new application. Assembly metadata contain information about each file which is part of an assembly. In figure 3.5, an assembly is grouped in multiple files and if among these files one file is changed, updated or replaced, and then a failure will be occur during loading the assembly because the information in the new file and in metadata does not match with each other.

Metadata about an assembly is stored in a Manifest file. .exe file or .dll file is used to store the manifest file of an assembly [9]. If needed optionally the manifest file can be store in a separate file. But if there is only one physical file for an assembly then the manifest file must be store in same physical file (assembly). The manifest file contains the following metadata information an assembly [11].

- **Identity**: Here the assembly is identifying to others. The information stored here are an assembly name, public key, culture and its version information.
- **Referenced Assemblies List**: Here the information about the dependent assemblies can be found, if an assembly is depends on others. Such type of information is used to avoid the versioning problem because it checks the right assembly version which is needed.
- **Request Set of Permission**: Here information about the permissions requires for an assembly to run successfully is listed. Permissions like accessing a file for reading or writing.
- **Exported Types**: It is used to include types from modules. When communicate with components of COM technology such type of information are used.

Whenever a request is send to CLR, to run a specific assembly, the CLR confirms the version number that it actually matches the sender requirements [9, 11]. If the requirements are not match then the assembly will be run. .NET has a powerful versioning system. Due to this system, component with different versions can be parallel run and they will not be interfering. This is called *side by side execution* in .NET. Due to this feature the developers can not think about the backward compatibility because they can work on both versions, old and new versions. Also "DLL-hell" problem is solved in this way. However problem can be occurs, if a component old and new version wants access at the same time to a file or resource, then a problem may be occur. To handle such situation, the component can be design in such a way that it can handle the situation if it’s old and new version access the same resource at the same time.
4. ENTERPRISE Java BEANS COMPONENTS

4.1. Enterprise Java Beans (EJB)

Sun micro system defines enterprise Java beans (EJB) is an industry standard tool used to develop
distributed components on server side. Java programming language is used to developed EJB
components [12]. But EJB is totally different from Java Beans. As the beans is the common word in
both technologies, but EJB is totally different from Java Beans. Actually Java Beans are classes in
Java language, which have set and get methods. Java Beans contains core features, methods, events
and properties. Java Beans are reusable components in Java language and used to develop Java
application and mostly used to develop client side component based applications. While EJB's are
used for development and deployment of client server component based application. TO run Java
Beans there is no need of specific runtime environment but to run EJB's a specific server application is
needed.

Now let's see how the EJB and Java Beans are accommodated in three tier architecture [13]. As we
know that three tier architecture have three layers, Presentation layer, Business logic layer and
database storage layer. To implement the presentation layer, Java Beans are mostly used to handle
operations on client side, while EJB does not fit here in presentation layer because it is used to develop
components on server side. SO EJB's are used to impalement the business logic of an application,
which perform the server side operation, not the client side operation. It is clear that EJB's are not used
to implement graphical components. Java Beans are not only used to develop the graphical
components but can also used to develop EJB components. So Java Beans can be used in both
presentation and business logic layer.

As already mentioned that EJB’S are used to implement components on server side [12], and these
components can be deployed in distributed environments also. These components must be deployed
into an EJB container, which can be the portion of a server application. Due to Java all the features of
Java is also inherited in EJB's, like high security, portability, stability and many more Java features.
For example, if an Java application a thread is killed then the application will not be stop. Java has a
huge library, which is fully tested and the most important is that Java is platform independent. When
implement the EJB's, it uses one or more objects of Java. So these components are more then Java
objects. Same component interface must be always used and does not matter in which way the EJB is
developed. But the most important thing is that EJB specification must be followed by both EJB and
interface of the EJB components, which expose the methods included in EJB's. What ever EJB
containers are used, the uniformity of EJB's will be handled by these the EJB container with the help
of expose methods.

The EJB container provided many services. These services are called middle ware services such as
transaction management, clustering, networking, security, resource pooling, RMI services and many
more. Due to these services provided by the EJB container, the developers can not think about to
implement these functionality during application implementation because these middle ware services
are provided by the EJB container. So the developers can save a lot of money and time as well while
using this third party EJB container and the most important advantage is that these third party products are fully tested. As already mentioned that EJB is an industry standard used to develop distributed components, which gives the easy portability advantage also. Due to this advantage components are easy portable between different EJB containers. It's now easy for developers to implement a secure, scalable and portable components and the most important thing is that there is no need to write complex framework because EJB's are industry standard. The following table shows some free of cast EJB containers.

<table>
<thead>
<tr>
<th>Container/Application server</th>
<th>Company</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>JBoss</td>
<td>JBoss Group</td>
<td><a href="http://www.jboss.org/">http://www.jboss.org/</a></td>
</tr>
<tr>
<td>OpenEJB</td>
<td>OpenEJB</td>
<td><a href="http://openejb.sourceforge.NET/">http://openejb.sourceforge.NET/</a></td>
</tr>
<tr>
<td>JOnAS</td>
<td>EVIDIAN</td>
<td><a href="http://www.evidian.com/jonas">http://www.evidian.com/jonas</a></td>
</tr>
</tbody>
</table>

But there are also some EJB containers which are not free of cost when using for business use. The following table shows some EJB containers which have cost (fee) when using.

<table>
<thead>
<tr>
<th>Container/Application server</th>
<th>Company</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere</td>
<td>IBM</td>
<td><a href="http://www3.ibm.com/software/info1/webSphere/">http://www3.ibm.com/software/info1/webSphere/</a></td>
</tr>
<tr>
<td>WebOTX</td>
<td>NEC</td>
<td><a href="http://www.sw.nec.co.jp/middle/WebOTX-e/">http://www.sw.nec.co.jp/middle/WebOTX-e/</a></td>
</tr>
<tr>
<td>WebLogic Server</td>
<td>bea</td>
<td><a href="http://edocs.bea.com">http://edocs.bea.com</a></td>
</tr>
<tr>
<td>Pramati Server</td>
<td>Pramati</td>
<td><a href="http://www.pramati.com/">http://www.pramati.com/</a></td>
</tr>
<tr>
<td>Borland Enterprise Server</td>
<td>Borland</td>
<td><a href="http://www.borland.se/besappserver">http://www.borland.se/besappserver</a></td>
</tr>
</tbody>
</table>

If the client applications wish to access the EJB container, it will be servlet, applet or other EJB. IF the client applications access an EJB then this call create a chain of calls from where one call of EJB calls another and so on like a series of chain. It is possible due to this chain series calls that a complex task can be divided into several parts. That's why this behavior is so extensible and also powerful. It is called "Location Transparency". So location transparency means, there will be no problem, where the components reside on the network. The components can be either reside locally or some where else on the network. Due to this property, if the original destination server is down or crash for some reason then the call will be smoothly redirect to another server. The EJB provides auto redirection and client application does not notice any delay or problem because of this transparently redirection.

CORBA and Microsoft .NET are the alternative technologies, used to develop and deploy components. Actually the EJB technology adopts the CORBA concepts and still many application servers using CORBA technology.

### 4.2. EJB Technology:

The EJB’s are divided into three different types of beans [12]. All these beans have different properties and features. Due to these three different types of beans, it is very hard for new developers to fully understand the EJB technology and use these beans in an efficient way in different situations. It seems
disadvantage of EJB technology at the beginning of learning the EJB technology. So it the beginning its very difficult for the developers to decide which beans should be use in which situation and this may leads a bad usage of beans. However the EJB advantages are more important then these disadvantages like flexibility, location transparency, security and many more.

EJB’s are categorized as:

**Session Beans**

Session Beans performs some business tasks or process for its client, so that the client can execute business tasks inside the serve. Session bean is used for storing the objects temporary. Which is very useful when implementing the business login, like credit card authorization services, accessing data from a database etc? Session bean is not shared among the clients. Session bean have one client at a time and the client cannot use the previous session bean. Session bean data is not stored in a database. Session bean is further divided into two types. These two types of session beans are stateful session bean and stateless session bean.

**Entity Beans**

Entity beans are used for storing the data objects in a permanent storage or persistent. Entity beans are used to model the business data. To perform business transactions the session beans use entity beans. For uniquely representation of data objects, primary key is used. Some form of relation is used to connect the entity beans objects to other objects.

**Message Driven Beans**

The functionality of the session bean and message driven bean are similar but the message driven bean can only invoke the methods by sending messages to these beans. The best example of message driven bean is a bean that receives messages from credit card authorization services. Like event listener, the message driven bean acts as a message listener. Message driven bean have the ability to send and receive messages asynchronously.

These will be discussing in detail later on.

Many businesses were involved at the development time of EJB technology with different kind of requirements. These businesses are using different kinds of technologies with different kinds of requirements and different kinds of distributed system depending on the requirements. Keeping in mind these all requirement, the EJB’s are divided into three types. These three types Entity Beans, Session Beans and Message Driven Beans have the capability to support maximum different kinds of requirements.

Basically RMI-IIOP protocol is used to communicate with EJB. RMI-IIOP is the recommended protocol now days to use but one can also use the Java's remote method invocation (RMI), it was used before RMI-IIOP protocol. Java.rmi package is used by the RMI while Java.rmi and Javax.rmi packages are used by RMI-IIOP protocol because it obeys the CORBA specification. Messages are used to communicate with this bean. Message driven beans will be explain in details later on.
The following figure shows the architecture overview of EJB application and its communication with different types of protocols and clients.

Figure 4.1 EJB Architecture and its client interaction

4.3. Types of Enterprise JavaBeans Components

As already that there are three types of beans in EJB components model. These three types of beans are:

- Session Beans
- Entity Beans
- Message Driven Beans

Now let's discuss these beans in more details.
4.3.1. Session Beans:

Session beans perform some business tasks or process for its client like credit card authorization, database access, some kind of calculation etc, so that the client can execute business tasks inside the server. Session beans are used to model the business logic for an application. This bean is used to store the objects temporary and the life time of these beans are not so long [12]. Session beans interact with several other beans and services. Session beans are not shared among the clients. Session beans objects are available in only single interaction of client and server. Session bean have one client at a time and the client cannot use the previous session bean objects. Session bean data is not stored in a database. If the server hosting an EJB container goes down for a while or crashed, the session beans data will also be lost and also if the client performing some tasks, it will also be lost.

Session bean is further divided into two types. These two types of session beans are stateful and stateless session beans.

- **Stateful session beans:**

  When the session state is not lost during multiple client requests, this kind of session bean is called stateful session bean. Stateful session beans share method information and so other information during multiple client requests.

- **Stateless session beans:**

  When the session state is not retained for a particular client, this kind of session bean is called stateless session bean. Stateless session beans did not share information during multiple client requests. When the client request is finished, the session state is not retained for another request or method.

When a stateless session bean is call then it cannot remember its previous state for a particular client. Stateless session beans instances are equivalent among themselves. So there is no problem for a client to use which instance. The client can use any instance from these instances. A pool is used inside an EJB container to contain these instances. From here instances are swapped among clients. Many clients can be serving from few beans using such a pool. The following figure illustrates how the stateless session bean is called from a pool.
Unlike to stateless session beans, the stateful session beans can remember its previous states for a particular client. The client can access its previous state and can also continue its previous work from where the client leaves it. Pool can be used to contain the stateful session beans, but it is very costly and the EJB container has to perform extra work. The persistence storage is used to save the states of stateful session beans and this method is called passivation. The following figure shows the statefull session beans passivation.

To restore the state from the memory, this method is called activation. The following figure shows the statefull session bean activation.
4.3.2. **Entity Beans:**

Entity beans are actually data objects, which are used to model the business data e.g. credit card, a product, a stock, an order, an employee etc. To perform business transaction the session beans use entity beans to obtain the business goals. So entity beans are used for storing the data objects in a permanent storage or persistent [12], for example a database. There are several ways to accomplish the persistence in entity beans but it up to vendor's containers, which method they want to use. These methods are Java object mapping, serializing, using database object etc. The following figure illustrates how to create an entity bean object and how the persistence uses this object.
As entity beans are data objects and the data represented by these objects are handled very easily because the data are durable in these objects. Entity beans are very useful when using in an application or software. Data can be easily grouped by using these beans objects and to handle activities like comparing the data, etc., methods could be added to these entity bean objects. The application server which contains the EJB container provides the middleware services like, transaction, security, relationship, and networking. Special cache is used to maintain these data, which increases the application or software performance. Entity beans can be container managed persistence or bean managed persistence.

- **Container Managed Persistence (CMP)**

If the application server generates code for inserting, updating, or deleting, then the entity beans are container managed.
• **Bean Managed Persistence (BMP)**

If the developers write code for inserting, updating or deleting, the entity beans are bean managed.

These all features cannot be achieved by only using a simple ordinary database, but these are possible with entity beans. Entity beans use a primary key to represent the data uniquely and also some form of relations are used to connect entity beans objects with other objects.

Entity beans and session beans are totally opposite to each other. Entity beans are used to store the permanently in a persistence storage while the session beans on the other side store the data temporary because these are not persistence objects. Session beans use the entity beans to perform some operations and activities related to database but they are not at all persistence objects.

Entity beans are used to store the actual business data like credit card, product, order, and stock or employee information while session beans are used to perform tasks or processes for its client, which are related to the actual data. Entity beans are not used to implement the business logic for its client like credit card authorization services. So session beans are used to model the business login for its client application. Entity beans are not used to model complex processes or tasks for its clients.

As from the above discussion it is clear that entity bean is a part of database and the same time it also a part of a component. One can say that it is a hybrid component. There are two ways to define an entity beans because of a close relation with the database. First one is *entity bean instance* and the second one is entity bean data.

• **Entity Bean Instance:**

Through this entity, data is view from memory into database and if there is any change or updation in memory entity bean instance then database will be updated automatically. Entity bean class is used to instantiate it.

• **Entity Bean Data:**

Physical data are enclosed in this entity, which is later on saved in a database like credit card, product or employee data etc

Some entity beans features:

• **Survive failures of entity beans:** The life time of entity beans are more then client session because it represent the data permanently.

• **Entity Instance are view into a database:** This means that if there is any change in the in memory bean instance, the modification will be automatically done in database also.

• **Same business data can be represented by several entity bean instances:** this allow that same entity bean instance can be share among many clients.

• **A pool is used to keep entity beans instances:** To represent same type of data in different instances of the entity bean, they can be pool in an EJB container.

• **Entity beans use two ways for persistence:** These are Bean managed persistence and Container managed persistence.
• Entity beans can be created and removed also:
• Without using the entity beans, the data can be modified of the entity bean.

4.3.3. Message Driven Beans:

In version 2.0, the EJB technology adds a new bean, called Message driven beans [12]. The functionality of the session beans and message driven beans are almost similar but the message driven beans are stateless that can only invoke the methods by sending messages to these beans. The best example of message driven bean is a bean that receive message from credit card authorization services. Like event listener, the message driven bean acts as a message listener and these messages can be of any type like object messages, byte messages etc. This beans checks fully the message contents and also confirm and control that the message belong to this specific bean. There is no local, home or remote interface for message driven beans. These beans have the ability to send and receive messages asynchronously.

The RMI-IIOP is the recommended protocol used to communicate with EJB components, but there are some situations like if performance and reliability are the high importance requirements and the client needs a support for multiple senders and receivers at the same time. Then in these situations RMI-IIOP protocol is not enough to work.

Keeping in mind the above three requirements, EJB technology introduce the message driven beans. RMI-IIOP protocol is used to communicate with the entity and session beans components but to communicate with massage driven beans, the client use messages. For messaging a special kind of server called "middleman" is used between the client and the desire component hosting on a server. The following figure illustrate how massaging work between client and the desire message driven bean.

![Remote Method Invocations](image)

![Messaging](image)

Figure 4.6 RMI-IIOP and middleman messaging.

Actually the middleman is used to receive calls from one or more clients. Then the middleman server forwards these calls to servers (One or more servers). When the client sends a message call, then there is no need to wait for the message driven bean to process the call and send back the response to client. The client can also do some other tasks between the call processing. When the bean finished his work
then the client can be also notify optionally and this kind of programming is called asynchronous programming.

Lets discuss how the message driven beans handle the above three requirements.

- **Performance:** When a client made a call to message driven beans, then there is no need to wait for message driven bean to finish the task and response back. Between the call made and bean response back, between this times the client can perform some other tasks. Due to this time optimization the performance increases.

- **Reliability:** The message oriented middleware products play the role of the middleman. These products have the capability to deliver the message to the destination server, if the server is down for a time being or completely crashed. When the server starts working back, it will receive the messages. So these product guarantees to deliver the messages to destination servers, while the RMI-IIOP protocol thrown an exception if destination server is down.

- **Multiple Senders and Receiver Support:** Message oriented middleware products are used for middleman. These products have the capability to receive multiple message from several senders and the forward these message to several destination servers. Mostly of these products support n=ary communications.

If message driven beans have the above important advantages the, so it is not used instead of entity beans and session beans. It affects the performance because of the message oriented middleware products and also the message driven beans are stateless. These are the disadvantages of the message driven beans.
5. EXAMPLE APPLICATION OVERVIEW

5.1. Project Methodology

Project Methodology plays very important role in project development. It is a way of providing different solutions for planning, organizing and managing resources according to the project needs which fulfill the project goals and objectives. There are many methodologies that help to produce the better quality for the project but these methodologies vary on project to project. For this project iterative software method preferred [15]. This methodology helps the project to solve many development problem regarding planning, design, early functionalities, and requirement changes. Problems resolve in early stages of the project which save much more time and customer satisfied by playing active role in every step. This approach is help in project to reduce chances of risk, program complexity and save the time in very easy way. This methodology helps a lot during development, requirement defines in structured way and save much more time by using short cycles. Mostly project have always been late in companies just because new requirements comes time by time which creates many problems in design and development phase but this approach helps in the project to make progress in small steps and remove mistakes which make more improvement for this project. Iterative modal approach provides many ways especially when requirements need to be change so it’s better approach for these projects to improve them in future implementations. Throughout the both projects, different terminologies resolved the complexity of projects and provide easy ways for completion of the projects.

5.2. Goals

Main goal of the project is compare two different technologies which are developed in different languages using component technologies and provide developer point of view which conclude throughout the project. Another main purpose of the project is to suggest better platform for development to the developer. To achieve these goals, chooses first platform for development is .NET that uses the ASP (Active Server Pages) and second platform is Java technology which uses xhtml (extensible HyperText Markup Language) pages. Now a days, developers and specially entrepreneurs are facing many difficulties to find best platform for their future career. By working with both applications, find out many solutions for developers to analyze best platform for their future.

5.3. Website specifications:

5.3.1. Overview

Website is mainly developed for the organization which main purpose is to provide different kind of travelling services to the customer where customer can book or un-book ticket by exploring the website. Website handles different allies companies such as bus, train and airline companies where customer can choose traveling option for his/her travels. This website provides travelling services for national and international locations. These locations are specific within the allies companies.
Organization is providing traveling services for almost all major cities. Customer can choose city where he wants to depart and where he wants to arrive. Customer should register before book the ticket. Figure 5.1() modal describe the overall information about the system. Website is providing different functionalities to the administrator where administrator can handle overall booking information. Administrator is responsible for the all changes in the website. Within website, administrator can issues different tickets and responsible for management of the ticket. Administrator can book or un-book ticket for any customer. Website provides functionalities to the administrator to manage company information and vehicle information. At the end, customer can get feedback about their ticket by email or by login in his own account.

5.3.2. Client website overview

Client website developed specially for customers where customer can search ticket, book ticket, un-book ticket and can register to the website. A non register customer can only check availability of the tickets for different locations but cannot books ticket until not register to the website. Website is very user friendly and easy to use for the customer.

In first page, website has welcome page which have all information about website see figure 5.2. On the top right side an image consist of company name and company logo which must displays in every page of the website. Welcome page have top and footer navigation which must also display in each of the page. Top and footer navigations are providing same navigation functionalities but have different layout design. Top header image, travel guide feature, new & promotion feature and features tour will show only on welcome page. These different features provide different companies information regarding services. Welcome page also have search feature which is the main objective of the page. Guest and existing user can check ticket availability by using this feature but existing user can also book or un-book ticket. Search feature have different vehicle types, depart & arrival option, date from & date to option, ticket persons, different ticket types for the user. To book a ticket, user will select first vehicle type with given options such as flight, train and bus then user will enter city and country for departure and arrival for different locations. In next step, user will enter date from and date to after selecting from calendar then user can select number of tickets for travelling up to range 0 to 7 and finally user will select ticket type from given options such as economy, economy plus and business class. These all fields are mandatory and if any one missing then page will not move to the next step and will redirect to the same page with error message regarding empty field. To see the search feature functionalities with filled fields and error message, see figure 5.3 (search feature with fill fields and with one error message). When user will click on search flight button, website will check mandatory fields. After success, website will check ticket availability from the database and will show result in searchresult page, see figure 5.3.

searchresult page is the page, where all information regarding ticket availability exist. Once user has clicked the search flight button then user can check all tickets regarding their search which are available in the database. Now user can select ticket in existing list of different records. In searchresult page, Top heading shows number of records found then table coming from the database which is showing the all records to the user. First column shows the departure information regarding vehicle with column name departs. Second column shows the arrival information with column name regarding
date, time, city, and country information. Third column shows the vehicle information where user can easily see the company name, vehicle name, vehicle type, and vehicle small information. Last three columns show the different ticket types such as economy class ticket, economy plus ticket and business class ticket where user can see ticket information. All information related to the tickets available in columns such as ticket price and number of ticket available. Now user is able to select any ticket regarding their search and can click on continue button for further process. Once user clicks on continue button, then website will check that either user selected any option or not. In not selection
case, website will redirect in same page and will show an error message to the user. In selection case, selected ticket will add temporary in session and page will redirect to the selectedtickets page. Guest and customer users both can add tickets in the website. User can add multiple tickets to their account in each booking process so in selectedtickets page user can see his all tickets which user added in one booking process, see figure 5.4.

![Figure 5.3: Prototype of the Search Result Page](image)

selectedtickets page is a page where guest and existing user can see selected tickets. If user searches multiple tickets in each booking time so user can view all selections in selectedtickets page. This page also provides functionality to the user to book and un-book ticket. In top right side, number of ticket will represent in digit form same in left top side, where heading with items will show same number of tickets selected by user. Then user is able to see his selection tickets in detail view in table form. Table first row shows the ticket number with departure city, country and arrival city, country. Other rows shows the ticket information such as company name, vehicle name, vehicle type, departs and arrival information, ticket type, ticket persons, ticket rate and at the end total amount of the tickets. Now user can book ticket by clicking book ticket link and can also have option to search more tickets by clicking more booking link. When user will click on more booking link, user will redirect to the main page and can search for more tickets, see figure 5.2. When user will click book ticket link then website will check either it’s a guest user or existing user from the database. For guest user, page will redirect to the registration page where user can register for new account and for existing user, user can login to the website, see figure 5.5.
Registration page is a page, where guest user can create new account with form information such as first name, last name, email, phone, mobile, address, post code, city, country, username, password and retype-password. Registration page also provide functionalities to the existing user to login the website. When guest user will enter field data and click on register user button for new registration then firstly website will check some mandatory fields such as firstname, lastname, username,
password and retype-password field. In case of empty any field, website will redirect to the same page with error message for specific field. After success of filled field, website will check, either password match with retype-password or not, in case of mismatch password, website will redirect to the same page with error message and in success case, website then will check from database either username already exist or not. If user already exist in system then website will redirect to the same page and must show error message. After all success cases, website will create new account for the user and will redirect to useraccount page, see figure 5.6 and all information regarding registration will save in database. Now user can book ticket by clicking book ticket button. Registration page also provide user login functionality, so in first table form, user can login page. To perform this action, user will enter username, password and must click on login button. Website will check firstly, fields are empty or not and in case of empty field, page will redirect to the same page with error message then secondly website will check validity of username and password from the database. In case of invalid username, website must redirect to the same website along with error message and in valid username case, website must redirect to the useraccount page see figure 5.6.

![Travel Operator](image)

**Figure 5.6: Prototype of the User Account Page**

After login to the website, now user can book added ticket by clicking book ticket button in selectedtickets page see figure 5.4 and also can explore his old booking history in his account by clicking myaccount link in navigation where user can also un-book ticket that are not still booked by administrator which will discuss later, to see the user account login page, see figure 5.6. After login success page, now user able to book ticket in selectedticket page where user can book ticket by clicking book ticket button. Website will check availability of ticket from the database. In success case, all temporary added session tickets will move to the user account in database and page will redirect to the success page see figure 5.7.
Success page purpose is to give confirmation of book ticket to the user. Within success page, page has back button link so by clicking this button, page will redirect to the main page where user can search for more tickets.

As previously discussed some about user account page where user can see all old and new history of book and un-book tickets see figure 5.6. Every registered user has its own booking history which user can see this information in useraccount page that is coming from the database. In top left of the page, heading is showing the number of tickets. In next, table is showing the all information about all tickets by user. In first column of the table showing the vehicle information regarding their each ticket, second is showing the booking start date where third column showing when booking end date.

Booking end date is the date; in which user can un-book the ticket during this time. Fourth column is showing the departure city and country of the ticket and same fifth column is showing arrival city and country of the ticket. Next column is presenting different selected ticket types such as economy, economy plus and business class. Next column is showing selected number of tickets by user. Then next to columns are representing the vehicle start and end dates. Next column is about status of the ticket. This column is very important for the user perspective where user can see ticket status. This column has two statuses. If status is pending then user must wait for the final decision from the administrator where administrator will decide that either ticket should be booked or not and if status is booked, then user can easily understand that there ticket is booked form administrator. Last column provides the functionality to the user to cancel the ticket if user doesn’t want it. User can cancel only those tickets which are with pending status. Myaccount page also consist of book more link button, by clicking this link button, page will redirect to the main page where user can search for more tickets.

5.3.3. Administrator website overview

Administrator site provides different functionalities to the administrator user for all changes in the website. Only site manager have rights to access administrator site. Administrator can book or un-book ticket for any customer. Administrator site is responsible for four major roles to administrator such as booking management, company management, vehicle management and ticket management. To access the website, administrator user will login by giving username and password in registration page, see figure 5.5. Once administrator user will login to the site, a welcome booking management screen will display on the site, see figure 5.8.
Administrator welcome page consist of some of previously client site features such as company logo on top left side, top menu and footer menu which will display in each of the page. A new administrator menu, after the top menu, will start showing in each of the page. This menu consists of booking management link, company management link, vehicle management link and ticket management link. Welcome page is providing booking management information where administrator can see all list of customer transactions regarding ticket booking. Booking management showing number of records in top left heading and booking management information in table where all old and new customers booking history is available. In first and second column, table showing user name and vehicle name. In next two columns, table showing booking start and end dates. In next two columns table showing departure and arrival information, table also showing ticket type, number of ticket booked by customer, vehicle start and end time and booking status for each customers. Last column is providing functionality to the administrator to confirm or un-book ticket for any customer. When administrator will click on confirm link then status will change from pending to booked and ticket will booked for this customer. If administrator wants to cancel ticket for any customer then administrator will click on cancel link button then ticket will remove from the database and page will redirect to the same page.
Second major feature of administrator site is company management. List of all allies companies are available in this feature. Administrator can view, add, edit and delete different allies companies. To access the page, administrator will click on company management link in given administrative menu navigation list then page will redirect to the company management page, see figure 5.9. Page consists of heading with number of records, insert new company link and company information table. In table, administrator can see company information such as name and detail. Table also provide administrator user to view, edit and delete functionalities with same name links so administrator can view, insert, edit and delete companies information by clicking on that. When user will click on delete button, selected company will delete from database and page will redirect to the same page.

![Figure 5.9: Prototype of the Company Management page](image)

Third feature of the administrator website is vehicle management. All list of vehicles regarding allies companies are showing there. When administrator will click on vehicle management from administrative menu list, page will redirect to the vehicle management page, see figure 5.10. This page consists of records heading, insert new vehicle link button and table. Table is showing company name, vehicle name and detail. Table also providing links of view, edit and delete. When user will click on delete button, record will delete from database and page will redirect to the same page. Administrator can also achieve functionalities of view, insert and edit vehicle information by clicking the links.

The last feature is ticket management. The purpose of the ticket management is to provide functionalities to the administrator to add new tickets in the system. These tickets are specific with allies companies and company’s vehicles. Once tickets added to the system then customer can search for these tickets from client website. To add new tickets in system, administrator will click on ticket management link in given administrative menu list and page will redirect to the ticket management page, see figure 5.11. This page consists of number of records heading, insert new tickets link and table with ticket information. Table has different columns which are providing different ticket information. First column provide vehicle name and next three columns are providing different ticket types such as economy, economy plus and business with total number of tickets and their rate. Table also has departure and arrival locations information and vehicle start and end dates. Administrator can view, edit and delete tickets by clicking on links which exist in table. To add new tickets, administrator will click on insert new tickets link then page will redirect to the insertion page where administrator can add new tickets. To edit, view and delete the existing information, administrator will click on particular link and can achieve their functionality.
At the end, administrator can logout by clicking logout link in top menu. By clicking the logout link, session will set to null and page will redirect to the main page of the client website.

Figure 5.10: Prototype of the Vehicle Management page

Figure 5.11: Prototype of the Ticket Management page
6. EXAMPLE APPLICATION IMPLEMENTATION

We used same design and architecture for both applications see figure 6.1. We divide our application in four major components. These components are logically different according to their functionalities such as booking component, authentication component, search component and data component. Booking component provides functionalities to the customer and administrator to book or un-book the ticket. Authentication component holds the all information of registration and login to the website. Customer can login and register to the client website and administrator can also login to the administrator website. Search component is responsible for all search information regarding ticket booking in the client website. Using this component, all searching functionalities can retrieve in the client website. Last component is data component which is most important component for the application. This component is responsible for the retrieve and saving all information regarding customer or administrator. All changes in administrator and customer side regarding retrieving and saving of data provided by this component. All other components are connected to data component to retrieve information. Finally website responsible for the use these component in different purposes, see figure 6.1.

Figure 6.1: Prototype of the Component Diagram
6.1. Environment

Main idea is developing two different applications in different platform to compare both technologies. To find out the accurate comparison, both applications developed in same machine. To see the hardware specification for machine, see the table 5.1.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
<td>Desktop</td>
</tr>
<tr>
<td>Computer brand</td>
<td>Intel</td>
</tr>
<tr>
<td>Memory (Ram)</td>
<td>2 GB</td>
</tr>
<tr>
<td>Processor</td>
<td>Core 2 Due 2.8 GHz</td>
</tr>
<tr>
<td>Cache</td>
<td>4 MB</td>
</tr>
<tr>
<td>Hard drive</td>
<td>150 GB</td>
</tr>
</tbody>
</table>

Table 5.1: Hardware Environment

For developing both websites in same machine, different software install during the development. These software performed different purposes according to the requirement. A list of software is available in table 5.2 used during development.

<table>
<thead>
<tr>
<th>Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows XP Professional Version 2002 with service pack 3</td>
<td>Operating system used for the experiment of the both website.</td>
</tr>
<tr>
<td>Microsoft Visual Studio .NET professional 2010</td>
<td>An environment for development of .NET application.</td>
</tr>
<tr>
<td>.NET Framework v 4.0</td>
<td>A package to run the target machine and OS.</td>
</tr>
<tr>
<td>Microsoft SQL Server 2008</td>
<td>Software which use to create the database for the applications.</td>
</tr>
<tr>
<td>IIS</td>
<td>A server engine which enable framework to host website in high performance and reliable way.</td>
</tr>
<tr>
<td>NetBeans IDE 6.8</td>
<td>An application environment for developing Java applications.</td>
</tr>
<tr>
<td>Ejb 3.1</td>
<td>An enterprise Java beans which provide server side components architecture for the J2EE.</td>
</tr>
<tr>
<td>Glassfish 3.0</td>
<td>Server engine which gives completely support for deployment of the J2EE application.</td>
</tr>
<tr>
<td>MySQL 5.1</td>
<td>A database platform for use to create database.</td>
</tr>
</tbody>
</table>

Table 5.1: Software Environment

We choose windows XP for development of both applications. Many reason to choose this operating system such are:

- Worked very fine with .NET environment.
- Provides universal hardware support and plug and play features.
- User friendly operating system while installing different software and configurations as compare to other operating systems.
- Fast and quick response is main key features of this operating system.
- Hardware driver support
6.2. Database Designing

We build same database design for both applications according to the requirements, see figure 6.2. To build database for .NET project, we use Microsoft SQL Server 2008 which is built in exists in visual studio package. We build our database using visual studio that provide visual database creation tool to build database within the server explorer window. To build database for Java project, we use MySql 5.1 database. MySql is open source software that can easily download from MySql official website.

![Figure 6.2: Prototype of the Database Design](image)

We create different tables for database according to design plan. Companies table is responsible for the saving of all allies companies information. Table created with id, name and detail columns where Id is unique and cannot be duplicated. Vehicles table stores all vehicle data that belongs to the specific company by company id. This table created with vehicle id, name, type, detail and company id where company id has relation with company table by id. Diagram is also showing vehicle service table which purpose is to manage customer tickets in the system. This table created with id, vehicle id, economy, economy plus, business, economy rate, economy plus rate, business rate, country from, city from, city to, vehicle start and end dates columns. Payment table will save all information regarding payment after ticket success by administrator. Payment table created with id, booking id, amount, payment date and status columns which have relationship with booking table. Booking table is responsible for the booking management in the database. Booking table created with
id, customer id, status, date, ticket type and total booked tickets columns. Booking table have three relations with different tables. First relationship has with customer then second with payment and third with booking service table according to the requirements. Booking services table purpose is to save different services regarding booking. This table has one relation with booking table by booking id and second with vehicle services table by vehicle service id. This table also contains booking start and end dates. Customer table is responsible for storage of different user’s data such as customer and administrator. Customer table also provide website credential with username and password. This table created with different user information columns such as id, first name, last name, username, password, email, mobile, address, postcode, city, country and phone where id is unique and cannot be duplicated. Customer table has relationship with booking table.

We used different stored procedures while our development. These stored procedures are also same in both applications according to their functionalities. To see list of stored procedure, see figure 6.2.

### 6.3. .NET Implementation

To build .NET application in visual studio, firstly we installed visual studio in our system. Visual studio is complete package of component based tools which helps to build high performance applications. Visual studio is very easy and user friendly installation. You must have administrator credential before the install of visual studio in your system. We choose visual studio 2010 professional for our project. One of our team members already had experience in previous visual studio versions but not with new one.

After installation, we explore the new interface of the visual studio, which have new look, easy interfaces and menus. Visual studio 2010 has many new features and improvement as compare to previous version. Microsoft has their official websites for developers which help a lot to the developer to learn the ASP.NET and different languages. These website also provide different tutorials in article, bookish, content and videos form for the beginners. All information is regarding installation, development and new features available. Both we learn different bookish and video tutorials and were ready to develop application using ASP.NET and C#. After running the visual studio first time in web development environment we choose C# language for our project. To see graphical interface for visual studio 2010, see figure 6.3.

Visual studio interface has different windows which used for different purposes. Toolbox window is showing in left side, which holds the list of controls that we can drag & drop to our page. There are many sets of controls available in toolbox depends on designer. To use a simple button for our web page, we can drag & drop it to the designer page so there all attributes start showing in properties window which is displaying in right bottom side of the page, see figure 6.3. We can see list alphabetically or categorized by attributes. By using this attributes of different controls, developer can change control layout, id name, size, height, width, color, css-class, text etc according to their needs. Ids attribute showing the identity of each control which developer can access source code for their purpose. Visual studio also has server explore window which allows to the developer to view the various services which are available in particular server such as database services, message queues and other services. We can also use drag & drop facility in server explorer window by dragging any service to page. This window allow developer to drag database table to the page which will
automatically create database connection with server and data adapter object that helps to retrieve the data from table. Visual studio have also solution explorer window that is showing in right top side of the page, see figure 6.3. Solution explorer window shows the list of project in tree view and their references, components and other contents. There are also more windows like team explorer, class view, bookmark, object browser etc which developers can use for different purposes according to our needs. If any window closed, developer can open it from view menu. We get lot of information regarding learning ASP.NET, C# and sql server in [16] [17] [18] [19]. By studying these, now we can solve our problems regarding development at any point.

Table 6.3: Prototype of the Visual Studio and Master page

To start our application, firstly we create a visual studio blank solution by clicking file>new>project>other project types>visual studio solutions in desired location. Solution helps to contain different kind of projects in same solution which helps to create web, window or console based applications.
6.3.1. .NET Component Creation

Creating a component in .NET using visual studio is very simple. We can build component in very easy steps. Firstly we select File->Project->Visual C#->Class Library and then we gave component name and selected desired location and click OK. We can also perform same steps by write clicking on our solution and then select add and New project from menu. After clicking on ok button, Solution explorer provides properties, references and Class1.cs. Now component is ready to use. We can add method and properties to the classes and can call it by adding reference to the main website. For our project we divide our application in four major components according to their functionalities which we explain below one by one regarding their creation, calling method and their usage in application.[20]

Data Component:

To create data component we follows .NET component creation steps and give name “DataComponent” in project. Data component is responsible to provide CRUD functions for all tables in our main website and other components. To achieve CRUD methods we write click by mouse in DataComponent project and select Add->New Item->Visual C# Items->Dataset.

Dataset is an object which uses to store temporary data table in application. Dataset load data in local memory cache of the application so even application is disconnected from the database we can work with it. All changes can be re-tracked when application re-connected to the database. Dataset can be typed or untyped. We used typed dataset in our application which is easy to program. We selected our dataset name “DS_TravelExpert” and click Add. Now “DS_TravelExpert.xsd” dataset created in data component project. Dataset provides designer and four different kinds of classes. We don’t care about other classes but we will work with database designer. “The Dataset Designer is a set of visual tools for creating and editing typed datasets and the individual items that make up datasets” [22].

We drag and drop all tables one by one in designer from server explorer window using database. Connection with database created automatically and saved in configuration file. All data tables, columns and relations automatically created see figure 6.4. Dataset automatically created CRUD (Create, Read, Update, Delete) function for each table in designer class. To explore designer class, double click on ‘TravelExpert.Designer.cs’ where you can see all declarations of the class. We can add more functions or make changes in existing CRUD functions according to the requirements. Now our booking component created successfully and ready to use. To use this component in our other projects we will create library file (dll) for this project by right clicking from mouse on DataComponent and select Build from menu. After success of build process, a dll file will created in bin\Debug directory. We can use this component dll file in any provide by adding its reference.

Search Component

Search component provides the all searching functionalities in the website such as ticket search by customer. To create this component, we follow same .NET component creation steps. We gave name to this component project as SearchComponent and select desired location and clicked ok button. Component created successfully with regarding properties, references and Class1.cs. We don’t need Class1.cs so we delete it from the project. We add dataset by write clicking from mouse and select Add->New Items->Dataset. We gave name “DS_Search” to the dataset. We already have stored
procedure for ticket searching. To access this stored procedure we add table adapter in our dataset by write clicking in dataset designer and selected add->tableAdapter. Wizards windows start showing in dataset designer where we selected our connection and “searchTickets” stored procured. We gave name “getSearchTickets” and finished the wizard. getSearchTicket method is start showing in search ticket table adapter. We build the dll file by write clicking on SearchComponent project and select build. Now we can use it in other projects.

**Booking Component**

Booking component provides the all booking functionalities in this component. By using this component, Customer or administrator can book or un-book ticket and there are some more functions regarding booking information. We gave name to this component project as “BookingComponent” and add dataset by write clicking from mouse and select Add->New Items->Dataset. We gave name “DS_Booking” to the dataset. We add table adapter in our dataset by write clicking in dataset designer and selected add->tableAdapter. Wizard windows start showing in dataset designer where we selected our connection and different stored procedures one by one. We gave name for each function and finished the wizard. All function are now start showing in booking table adapter. We build the dll file by write clicking on BookingComponent project and select build. Now this dll file is ready to use in other projects.

**Authentication Component**

Authentication component responsible for registration and provide access to the customer and administrator to login the website. To create this component, we follow same .NET component creation steps and gave the name “AuthenticationComponent” to the component and select desired location and clicked ok button. Add dataset by write clicking from mouse and select Add->New Items->Dataset. We gave name “DS_Authentication” to the dataset. We add table adapter in our dataset by write clicking in dataset designer and selected add->tableAdapter. Wizard windows start showing in dataset designer where we selected our connection and add different stored procedures one by one regarding authentication. We gave name for each function and finished the wizard. All function are now start showing in Authentication table adapter. We build the dll file by write clicking on AuthenticationComponent project and select build. Now this dll file is ready to use in other projects.

All components created successfully, now they are ready to use in our main application. To create our main Application, we write click on our solution and then select add->New project->Visual C#->ASP.NET Web Application from menu. We add all component one by one in web application by write clicking on Bin folder and select Add Reference->Projects-> list of components. Now all component dll files added in Bin folder so we can use them in web pages.

**6.3.2. Master Pages**

We create two master pages in our websites for client and administrator website layout support. Master pages provides constant layout for different pages. It’s shared controls and markup for multiple pages and provide template for target pages. We create master pages with purpose of reuse them in for client and administered website pages. First master page hold the top company name and logo image, top menu and footer which includes the footer menu. See figure 5.2, 5.3, 6.3. We design our website
using div html tag. We use unordered list for top and footer menus. In menus, we used ASP.NET
linkbutton that helps to navigate one page to another. First master page designed for client site layout.
Second master page designed for administrator pages layout. Second master page uses first master
page contents and plus administrator menu. We used ASP.NET linkbutton control in administrator
menu for navigation from one page to another in second master page. Now these master pages are
ready to use so we can use them in our client and administrator websites pages.

6.3.3. Welcome page

We started our first page by creating welcome page. We add web form page by writing clicking on
project and selecting add new item. We enable the “select master page” option and then selected first
master page. Now page have all master page contents which do not need to recreate. We inserted some
basic company information like header, travel guide information, news & promotions, tours
information and ticket searching tool. Ticket searching tool consist of some client and server side
controls. We inserted ASP radio button control list for vehicle type, textbox controls for date &
location, drop down list control for number of persons & ticket type in searching and finally image
submit button for searching the ticket in searching tool. We add Ajax water mark in text boxes and
Ajax calendar for date picking which helps to make easy interface for user, see figure 5.2. When user
will click on search flight image button, server will check all mandatory fields. In case of any missi-
ng field, page will display an error on page and only search tool part of the page will updated because we
use Ajax update panel in our page. Update panel control is an Ajax control which helps to refresh the
partial update for contents of the page. In case of success of all mandatory fields, server code will
search ticket by using search component from database using inserted data and save it in session and
redirect to the next page for ticket result.

6.3.4. Registration Page

Registration page helps the user to register or login to the website. We add web form page and
selected first master page. We insert two html tables, first for existing user and second for new user
registration. We drag & drop two textboxes and one button into the table where fist textbox labeled for
username and second labeled with password using ASP.NET standard label controls. When user will
click on login button, server side will check mandatory fields and user existence in database using
authentication component. After success, if user is an admin user then it will redirect to the
administrator website, see figure 5.8 and if it is not admin user then page will redirect to the client
website using session, see figure 5.4 where user can easily booked the ticket. For second table, we
drag & drop several textboxes for different form field data and labeled them with ASP label control
and one submit button for registration, see figure 5.5. When user will click on registration button,
server side code will check all mandatory fields and password mismatch. After success, server code
will check that either user already exist in database or not using authentication component. If it is
already exist then page will show error message in ASP.NET label text otherwise page will redirect to
the client website.
6.3.5. User Account Page

We create user account page using first master page. User account page provide facility to user to view ticket history and un-book tickets with pending statuses, see figure 5.6. Page consists of ASP.NET label, gridview and link button controls. We drag & drop gridview and link button to the page and label the link button as Book more. Gridview is an ASP.NET data source control which helps to handle automatically representing, paging, editing, sorting and deleting data operations. Gridview bound with fields and also helps to generate dynamic columns. Read more about Gridview control [23]. We programmatically generate all columns in gridview using booking component. Gridview have cancelation link button which purpose is to un-book the ticket. When user will click on un-book link, server will delete the selected record from database using booking component. Book more link button provide the navigation to the main page so by clicking this, user will redirect to the main page.

6.3.6. Administrator Ticket Management Page

We used second master page for ticket management page. Ticket management page created for administrator website where admin user can add and update tickets for customer user, see figure 5.11. Page consists of master page contents, ASP.NET labels, linkbutton, multiview, views, objectdatasource, detailviews and gridview controls. Top left label is showing the number of records in gridview. We divided our page in three views. First view is showing list of records, second view used for insertion and third view used for representing single record view. To perform this action, we drag & drop one ASP.NET multiview control in page and three view controls in multiview control. Now multiview divided in three views. We also inserted three objectdatasources controls into the page from data toolbox window and configure those using data component operations. Object data source is a middle-tier object that used for retrieval and update capabilities of data. In first view, we inserted gridview control from data toolbox window and set it to active view using programmatically. We used gridview data control for showing number of records, editing, deleting and viewing data operations. We configure first objectdatasource for gridview operations and enable its paging, deletion editing and selection. Now administrator user can see list of records, view record, edit record and delete record. In second view, we drag & drop detail view control and configure it using second objectdatasource and changed its default mode to insert. When user will click on insert new ticket linkbutton then second view will active on the page and admin user can insert record using detail view different fields. Finally in third view, we drag & drop another detail view control from data toolbox window to representing the single record and configure it using third objectdatasource. Now admin user can view selected record from gridview by clicking on view linkbutton.

6.4. J2EE Implementation

We used same architecture for Java application. Both we don't know Java and never work before with NetBean. We explore sun Microsystems website and learn some basic tutorials regarding installation. Netbean also have their official site where different tutorial available for beginners. We go through some of important tutorials and start application by installation of NetBeans. NetBeans an open source integrated development environment that support the developer to create web, desktop and mobile applications using the Java platform. After installation, we explore the interface which was very easy and user friendly. We already developed our one of application in visual studio so we easily
understood NetBeans interface. Netbeans have different windows regarding their purposes such as projects, files, services, navigator, palette, properties windows, see figure 6.4. Project window purpose to manage different projects where files window responsible for manage directory structure of the source project. Services window provides different database, servers, web services etc to the user. Property windows display the properties of selected control or component. Palate windows contains the html, html forms, jsf/swing and Java components controls. We can directly drag & drop any of control in design or source view by using this window. Netbeans also have output window where we have inner windows like debug, glassfish domain and debugger console where we can see our list of errors, stop/play glassfish server and enter query in debugger console for output. Central window used for code where developer can write their Java/jsf or any other code regarding their application requirements. A list of menu buttons showing in top of the screen. By clicking play image button, we can run our application. Other buttons used for different purposes such as running application in debug mode, run profile project, clean and build project. We explore all windows and now we are ready to start our web application in NetBeans.

Table 6.4: Prototype of the NetBeans Interface
To take a start for our application, firstly we created Java EE enterprise application by clicking file>new project>Java EE>Enterprise Application. Java enterprise application is set modules which can be web application, EJBs and client applications see for more [24] [25]. We provide name “ProjectJava” and provide desired project location to the application. In next step, we chose GlassFish v3 Domain for server, Java EE 6 for Java EE version, enable the checkbox with text “Enable Context and Dependency Injection” and finally provide name “TravelExpertWebsite-war” for web application module and finish it from wizard window.

6.4.1. EJB Components

We used same component architecture that we had in .NET application. We can build our component in Java in very easy steps. To create component, we select file>new project>Java EE>EJB Module and provide EJB module name and add it to enterprise application and finish it. Component automatically will start showing in enterprise project application, see figure 6.4. When EJB Module will create, we can explore it by double click from projects window. Component will create some built-in directories and files such as source packages, test packages, libraries, test libraries files, configuration files etc. We will use these directories and files in our project while working. Now component is ready to use so we can add component in any project by giving its reference.[26]

Data Component

We used EJB component steps to create ejb module and gave name “DataComponent” in project. We create two packages under source package directory where first package storing data access logic and second storing business logic information. We generate all database entities in data access package by choosing new->others->persistence->entity classes from databases class where we create database connection with MySQL and select all tables from database and generate entities. These entities have built-in named queries and class properties. In second business logic package, we store all CRUD functions for these entities which also generated automatically. We write click on business login package and select new->others->Java EE->Session Beans for Entity classes. In next step, we choose all entities classes and finish the wizard. All entities façade classes automatically created with their CRUD functions. Now this component is ready to use for other projects.[27]

Search Component

We used the same EJB component steps for creation and give name “SearchComponent” to this component. Then we create SearchTicketPkg under source package directory. We create session bean class by write clicking on SearchTicketPkg and selected new->others->Java EE->Session Bean and named this class with SearchClass. This class has built-in some necessary import packages, class constructor. We create function with name getSearchResult which purpose is to return list of ticket records. We used preparecall function to call the stored procedure form MySQL database. now this component is ready to use in other projects.

Booking Component

We create booking component by using same steps and create a package under source package with name “BookingPkg”. We create a session bean class with name “BookingCompClass”. This class use
different stored procedure regarding booking functionalities and create different functions for the
component.

**Authentication Component**

We created our last component with name AuthenticationComponent. Under the source package we
create create another package with name “UserAuth”. A session bean class created with name
“userAuthAdapter” created. Under this class, we create different functions regarding authentication
such as checking the user existence, verify the particular user, etc.

We created successfully all packages, now we will use them in our main application. As we already
created our web application during Java enterprise application creation process. This web application
has some built-in directories and files for developer user.

6.4.2. **Facelets Templates**

We use the same concept of .NET master pages and created two facelets templates in our websites for
client and administrator website layout support by write clicking on main website and select new-
{others->}JavaServer Faces->Facelete Template. Faceltes template use to avoid repetition in facelets
pages. Templates used to share the same basic layout and general look in all client multiple pages.
Template file use two most common tags. One is content tag which uses to appear for all clients to
enter the content directly. Second ui:insert tage which use to replaced in client file. In firste facelete
template, we copy the code of first .NET master page and replace menu navigations with html tags.
This master page uses for client website with very basic layout like top logo, top menu and footer. See
figure 5.2, 5.3, 6.3. we entered content and ui:insert tags for client pages and saved the page. Second
template created for administrator website. We copy all code from first facelete template and paste it
in here. We entered the same administrator menu navigation that we used in second .NET master page
and replace it using html tags, see figure 5.8, 5.9, 5.11. Both templates are now ready for customer and
administrator pages. Next will creates all pages using these templates.

6.4.3. **Welcome page**

Next we create facelats template client page using first facelet template by write clicking on main
website and select new->others->JavaServer Faces->Facelete Template Client. Provide the name
“index” to the page and choose first template from wizard and finish it. A index.xhtml page created
successfully including two tags ui:composition and ui:define. We replaced the .NET welcome page
code and replace ASP.NET controls with jsf controls. JSF do not provide ajax function so we use
JQuery tool to watermark the textbox and calendar for date picking, see figure 5.2.

Create another package under source package directory with name “Presentation” in our web
application. This package specially created for jsf presentation classes. JSF pages will use these
presentation classes for different purposes.

We replace ASP.NET linkbutton control with jsf commandbutton control. Created JSF managed bean
class with name “SearchTicket” for welcome page under the presentation package. Creates some
properties and methods in this class for welcome page and assign these properties and method to the
welcome page. When user will click on search flight button from interface, website will check all mandatory fields. In case of any missing field, page will redirect to the main page with missing field error in label. In success case, website will search ticket by using search component from database using inserted data and save it in session and redirect to the next page for ticket result.

6.4.4. Registration Page

We create page using first template and paste ASP.NET registration page code in under ui:define tag. Replace all ASP.NET controls with JSF controls. Create a class managed bean class under presentation package in main website for registration page. Programmed necessary properties and methods in registration managed bean class. All properties bind to the jsf textbox controls and assign methods to command buttons control under registration page. For registration page, see figure 5.5. When user will click on login button, server side will check mandatory fields and user existence in database using authentication component. After success, if user is an admin user then it will redirect to the administrator website, see figure 5.8 and if it is not admin user then page will redirect to the client website using session, see figure 5.4 where user can easily booked the ticket. When user will click on registration button, server side code will check all mandatory fields and password mismatch. After success, server code will check that either user already exist in database or not using authentication component. If it is already exist then page will show error message in jsf label text otherwise page will redirect to the client website.

6.4.5. User Account Page

We create user account page using first facelete template and paste contents from ASP.NET user account page. After pasting the contents, we replaced ASP.NET data gridview control with jsf dataTable control and book more linkbutton with jsf commandlink control. DataTable is a powerful formatting table which used to iterate the collection of array of values. Table itself have columns, header, footer and give accessibility to jsf controls to perform their functionality in it. Table can associate to the backing bean so we can obtain data from backing managed bean classes and display it into table. We created a managed bean class for user account page and necessary methods. We bound list of user ticket records to the dataTable for display and un-book link under a column. When user will click on un-book link, server will delete the selected record from database using booking component, see figure 5.6.

6.4.6. Administrator Ticket Management Page

We used second facelete template for ticket management page and paste all contents from ASP.NET ticket management page. After pasting contents, we replaced ASP.NET controls with jsf controls and ASP.NET data gridview control with jsf dataTable control. Created managed bean class and their properties and methods with purpose to achieve deletion, insertion, updating and viewing functionality. Datatable Bounds with ticket management method using bean class. Enter view, delete and edit commandlink controls and bound them using bean methods with action attribute. When user will click on view commandlink button, method will call using bean and page will redirect to the ticket view page and will show selected record. Same with edit button but when user will click on delete
button, selected record will be deleted and page will redirect to the same page using managed bean class method, see figure 5.11.
7. COMPARISON

Comparison is common concept in almost every developer’s life, importantly when developers compare .NET and Java platforms. Comparing of both platforms using component technology is not easy because both have advantages and disadvantages. Many new developers are coming from both platform sides and taking interest in these technologies. One main reason to comparison for both technologies is that both are booming in the current market. To understand both technologies, we firstly draw analogies between .NET and J2EE technologies, see table 7.1.

<table>
<thead>
<tr>
<th>Features</th>
<th>J2EE</th>
<th>Microsoft.NET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendors</td>
<td>30+</td>
<td>Microsoft</td>
</tr>
<tr>
<td>OS</td>
<td>Mac, Unix, Linux, Windows, Others</td>
<td>Windows</td>
</tr>
<tr>
<td>Interpreter</td>
<td>JRE</td>
<td>CLR</td>
</tr>
<tr>
<td>Dynamic Web Pages</td>
<td>JSF</td>
<td>ASP.NET</td>
</tr>
<tr>
<td>Packaging</td>
<td>.jar, .war</td>
<td>.exe, .dll</td>
</tr>
<tr>
<td>Middle Tier Component</td>
<td>EJB</td>
<td>.NET Managed component</td>
</tr>
<tr>
<td>Database Access</td>
<td>JDBC, SQL/J</td>
<td>ADO.NET</td>
</tr>
<tr>
<td>SOAP, WSDL, UDDI</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

7.1. Tools

Visual studio is fully compatible with all windows operating systems and with few requirements, it’s very easy to install. Many useful tools are already exists within the package including Microsoft SQL Server, SharePoint, Report designer and UML modeling tools. In few clicks, it’s ready to use. Software startup time is very quick and flexible interface. Visual studio provides many built-in controls which save much more time while using it. Visual studio provides better debugger and can easily debug application while Netbeans debugger is very slow as compare to visual studio. We waste our much more time while debugging our application in NetBeans. Using multiple break points in Netbean debugger, runtime performance is slow down the application work. No need to deploy or undeploy the application using debugging option in visual studio. We drag and drop many major objects in designer even complete database table that automatically bind with data controls. Visual studio support many languages that releases by earlier version. Visual studio still has excellent code completion while using methods names and parameters.

Netbeans is a free product that can freely download from sun website. Netbeans compilation is much better and time saving. Netbeans have better feature for source code editor where it highlight the variable that used in different locations. So variable can easily identify while writing the code. Errors and warnings are separately showing, we can resolve these warning by double click. Automatically display list of packages using intelligence when writing non added package method so we can add package by clicking helping icon from source code editor. Profiler is a good addition in Netbeans. While writing code we can choose best algorithm by experimenting it by checking memory and processing in profiler. Netbeans have facility to save local file history and provide source control.
integration with cvs and svn which is now available in netbean so we don’t need to install third party tool. Database explorer is use to work almost with every database. Java have interesting plug-in system that provide support many other languages. Its open source and can install in different platforms with VM.

While working on visual studio and Netbeans, we explore both tools but cannot learn all features due to short time. We learn many things and now we can work on both tools in future at any time. Both tools have some advantages over each other but we feel to work free on visual studio much better than Netbeans. No doubt there are many similarities but visual studio is rock.

7.2. Web Technologies

We develop our web application using ASP.NET and JSF technologies. Both technologies are next generation for web development frameworks and use to build server side user interfaces. ASP.NET is a part of .NET platform that provides the ASP (active server pages) technology while JSF is a part of J2EE that provide XHTML pages. Both technologies are event driven mythology and component technology of software developing. When we work in ASP.NET component, we use controls because components called controls in ASP.NET but JSF provide Component for naming Components.

- **ASP.NET**
  - **TextBox:** Textbox is a server side control that provides facility user to enter data including numbers, text and dates. Control has different TextMode, by default it’s a single line but we can change it to multi line or password.
  - **Label:** label is use to display data in web form. We can change programmatically its text when required.
  - **Button:** There are different kinds of button available in ASP.NET such as button, linkbutton, imagebutton, htmlbutton, htmlinputbutton, htmlinputimage and imagemap for different purposes. Buttons used to post page to the server when specific event is triggered.
  - **MultiView:** This control helps to present data in different views on the same page. Control provides multiple panels, separate views and different user choice views on the same page.
  - **ObjectDataSource:** This control manages the middle tier data and provides facilities for retrieval and updatation of data. This control bound that data with different other controls such as GridView, DetailView so using these controls we can edit or display data using middle tier business object. Very time saving and useful control that provides very easy designer, sorting, paging, updating, caching, and filtering and conflict detection capabilities to the user to manage data in simple way.
  - **GridView:** Gridview is very strong and well known control in .NET. It uses to display data in tabular form. Gridview bind the data using sqlDataSource or ObjectDataSource. Provides many built-in capabilities such as sorting, deleting, updatating, paging, row selection, dynamically set properties, themes, styling, multiple key fields and a lot are more. AutoGenerateColumns capability creates different fields using data source which no need to code more.
  - **Detail View:** detail view shows the single row of data in tabular form. It has less capabilities compare to gridview but it’s mostly used with gridview control for master-detail purposes.
This control also bind data from data sources and provide different built-in capabilities such as paging, inserting, deleting, updating, dynamically access, themes and styling etc.

- **JSF**
  - **TextField:** Provides facility user to enter data in string or integer form. Name can generate automatically or we can give it. This component required when input data required in form.
  - **OutputText:** it is use to display output data in the page.
  - **Command button:** this button provide similar to the html input submit tag. It use to submit the form. Control use for calling Java method. We can specify this Java method in action attribute of the button. If we set immediate attribute to true then we can call function directly without any setter method.
  - **Datatable:** it is use to show data in tabular form. We can put different jsf html component inside such as textbox, command button, selectmanylistbox, selectoneradio etc. we can set header and footer of the datatable. Datatable itera tes the collection of data automatically by associating backing bean.

After working on both technologies, we find out, both are good but ASP.NET is more mature technology. It is provide large range of controls and very easy to use them. No need to write extra codes. ASP.NET also provides many templates as compare to JSF such as style sheets, mobile form, web forms, master pages, class diagram, reports etc. JSF has some similar templates but still working on templates.

ASP.NET has very good feature called smart tag. This tag is display list of menu in designer view so developer can customize very quickly. For example, we drag and drop gridview in our .NET project. After that a list of menu start displaying with different features such as data source, insert, delete, update, sorting, paging etc and we customize it's with few clicks. A list of properties also start showing with control. JSF also provide smart tag feature in some sense but we need to right click from mouse and end with wizard. As compare to ASP.NET, it is not intelligence feature in JSF.

Both provide code behind capability. In ASP.NET, code behind class automatically linked with .ASPx page unless developer doesn’t change its name manually but in JSF scenario, this linked with configuration file. In JSF its call backing bean which not inherit from any interfaces or classes, they are just regular classes.

JSF has major advantage that large companies like Sun, Oracle, Redhat, IBM and research universities, open source communities etc support JSF.

JSF worked on cross platform which means, JSF page can be view in different platforms including internet explorer.

ASP.NET easy to learn where as jsf is difficult to work in start and have many bugs till now. There are many video tutorials available for learning ASP.NET that is well documented compare to jsf. More forums, blogs and official site provide easy solutions for developer to fix the problem. ASP.NET have better support regarding learning and understanding the technology.
ASP.NET support many different languages such as C#, vb.NET, c++, j# etc while jsf supports only Java.

Both provide configuration files and we can say that both are similar in some way because both use to set application parameters with some setting and patterns.

### 7.3. Programming Languages

It’s very difficult to compare both languages because both are very similar and very near to the nature of developer.

Java is old language developed by sun Microsystems that provide large and dynamic user base. Java has more learning contents and much more specialization in different computer courses. C# is younger language then Java, developed by Microsoft. Microsoft remove many problems regarding runtime environment and shortcoming that was exist in many old languages like Java, Pascal etc and provide more user friendly program, analysis and fast data mining.

Both languages are strongly, statically and explicitly typed.

Both languages are curly brace and use garbage collections. Both languages syntaxes are very similar because both related with curly brace in c and c++ family but we feel difficulty when things were different to each others. It is easy to learn both languages if developer already have experience of c or c++. If developer has experience of any of Java or C# language then he can easily learn another one with few changes.

There are many simple and primitive types available in C# compare to Java such as UInt16, UInt32, Uint64 and Decimal. C# provides more features, keywords and grammar than Java language.

C# provides pointer arithmetic and value types in programming language whereas Java doesn’t allow them.

Java has advantage to run in any platform while C# runs only windows environment.

Both support try-catch-finally exceptions but C# doesn’t support checked exceptions that are very important for developer’s point of view.

After coding with both languages, our more interest develops in the favor of C# because it is the language to learn seriously.

### 7.4. Components

We build components in both technologies in our experiments. We used assemblies in .net and EJB in java project.
<table>
<thead>
<tr>
<th>Feature</th>
<th>.Net Component</th>
<th>EJB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform</td>
<td>Windows</td>
<td>Multiple</td>
</tr>
<tr>
<td>State Management</td>
<td>Stateless</td>
<td>Statefull &amp; stateless</td>
</tr>
<tr>
<td>Packaging</td>
<td>Single(.dll, .exe), Distribution(.cab), window installer(.msi)</td>
<td>.jar</td>
</tr>
<tr>
<td>Features</td>
<td>deployment, automatically version control, reuse, activation scoping, and security permissions.</td>
<td>Deployment, reuse, manual version control, security permission.</td>
</tr>
<tr>
<td>Runtime Environment</td>
<td>CLR</td>
<td>JVM</td>
</tr>
<tr>
<td>Run Multiple version simultaneously</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Portable</td>
<td>Windows</td>
<td>Multiple</td>
</tr>
<tr>
<td>Composition</td>
<td>Multiple classes using specific Language</td>
<td>Multiple classes using java language</td>
</tr>
<tr>
<td>Language Support</td>
<td>Multiple</td>
<td>Java</td>
</tr>
<tr>
<td>Intermediate Code</td>
<td>IL Code</td>
<td>Byte Code</td>
</tr>
</tbody>
</table>
8. CONCLUSION

Microsoft provides more foundation and user friendly environment to the developer in many areas as compare to J2EE. No doubt, both have many similarities and each technology deal with various tasks in its own ways. Both support to develop component development either local or distributed but EJB have three types of the components which are difficult to understand for a beginner developer. During the project, we compared both technologies firstly with theoretically and then practically and find out advantages that lead to the .NET for a developer’s point of view, advantages are:

- Visual studio is more powerful and user friendly IDE.
- .NET supports multiple programming languages.
- ASP.NET has more features for web development.
- Easy to work and understand .NET component development.
- Less source code.
- Time saving.
- Documentation available in well formed.

In other hand, J2EE also have some advantages:

- The bigger advantage of J2EE is portability and it can run in many platforms including Windows, Sun, IBM, Unix
- Java is old language so it’s quite mature.
- Many free, open source and third party components are available.

In this report, we tried to compare both platforms using component development for developer point of view. Theoretically both platform are almost similar and provide same concepts but practically we feel free to work with .NET platform.

While working with both platforms, our all experienced was very good. We both had no previous experience with both platforms so we learned many new things.

In future, we can compare both technologies while implementing MVC and LINQ technologies. JSF is trying to competing ASP.NET so we can again expand our research area in these fields. We also need to compare deeper research on both platform components.
REFERENCES


[16] Joe Mayo, Microsoft Visual Studio 2010


[18] “ASP.NET 4 and Visual Studio 2010 Web Development Beta 2 Overview”


[22] Sahil Malik, Pro ADO.NET 2.0

[23] GridView for ASP.NET AJAX, info@ComponentOne.com, http://www.componentone.com


[26] Rima Patel Sriganesh, Gerald Brose, Micah Silverman, Mastering Enterprise JavaBeans 3.0