Politecnico Di Milano

Facoltà di Ingegneria dell'Informazione

POLO REGIONALE DI COMO

Master of Science in Computer Engineering

Automated Fault Detection Tool – Interface Application

Supervisor: Prof. Fabio Salice

Master Graduation Thesis by: Iqbal, Azhar

Student Id. Number: 736418

Table of Contents

ABSTRACT

AFD (Automated Fault Detection Tool) Testing tool in existing state is a command-based, mono – interface, with absence of results and history management of the performed tests. SUT (system under test) is a board consiste of multiple electronic components. A set of Tests executed on board and derived their results. To start testing procedure a CTM (component tests matrix) is strictly required. In more compact spectrum, AFD Tool is a .dll file contains a dozen of method which encapsulate the core logic for performing the Testing. The thesis work's purpose is to create an application which should be web-based, concurrent, interactive, effective and efficient storage mechanism of performed tests in a fashionable and effective way, much commonly in a relational database. The challenging parts of this work are: to use the .dll (*unmanaged code*) in managed code like C#, managing the input/output of .dll methods which all are based on Text files, Managing the outputs files' format, which are always specific formatted text files and need to be parsed on an individual pattern, stimulation of the incremental iteration of the testing and design the meaningful and appealing interfaces.

The approach for the implementation, we use the standard and convenient development environment like .net in our case. Upon the successful implementation of the web interface named as AFDIA (Automated Fault Detection Tool – Interface Application), it will be really easy and user-friendly to perform testing, view the result in rich GUI-based interface, explore the history of the performed tests and save the resulted data into relational database format, which is a standard approach to save information in meaningful, relational oriented and for fast extraction of stored data. In conclusion, AFDIA will help the tester to view the results of Tests in rich GUI format, storage of the result data for viewing history purposes and overall will be benefited to reduce the time consumption, cost and effort instead which we invest in case of command-based system.

sommario

Lo strumento di test AFD (Automated Fault Detection Tool) allo stato attuale è uno strumento a riga di commando, mono-interfaccia, con assenza di gestione dei risultati e dello storico dei test eseguiti. SUT (sistema in prova) è un consiglio consiste di molteplici componenti elettronici. Una serie di test eseguiti a bordo e loro derivati risultati. Per avviare la procedura di verifica di un marchio comunitario (componente della matrice test) è strettamente necessario. Dal punto di vista software, AFD è un file .dll che contiene una dozzina di metodi che incapsulano la logica di base per eseguire il test. Lo scopo della tesi è realizzare una applicazione web-based, concorrente, interattivo ed efficiente per gestire, eseguire e salvare il risultato dei test eseguiti in un modo semplice e distribuito, facendo uso di un database relazionale. Le parti significative di questo lavoro sono: utilizzo in modo adeguato la .dll attraverso un applicativo realizzato in C#, gestione dell'input/output dei metodi della .dll (il meccanismo di comunicazione è basato su file di testo), gestione del formato dei file di output (sempre file di testo specificamente formattati e che hanno bisogno di essere interpretati su uno schema individuale), gestione del meccanismo di scelta incrementale del test, e progettazione di interfacce significative e si facile uso e comprensione .

Come strumento per l'implementazione, è stato utilizzato l'ambiente di sviluppo .net. Attraverso l'implementazione dell'interfaccia web - chiamata AFDIA (Automated Fault Detection Tool – Interface Application)- l'esecuzione e la gestione dei test è semplificata e di facile utilizzo così come risulta semplice avere un riscontro sullo stato della analisi sugli apparati; in particolare, vedere i risultati (interfaccia basata su una GUI), esplorare la storia dei test eseguiti e salvare i dati risultanti. In conclusione, AFDIA aiuta l'ingegnere che si occupa dei test a vedere e analizzare i risultati del test, immagazzinare i dati risultanti allo scopo di vedere lo storico. AFDIA, inoltre, riduce tempo, costo e sforzo nella operazione di analisi degli apparati rispetto al caso dell'attuale sistema a riga di comando.

ACKNOELEDGEMENT

I am thankful from the core of my heart to **Prof. Fabio Salice** for his continuous
Support, supervision and guidance that helped me to learn the basics of Research and development.

And also thanks to **Luca Amati**, who’s effortless and rigorous guidelines and help make it possible to achieve the desired results.

List of Figures:

List of Tables:

INTRODUCTION

In the present thesis, we are working on developing a richly interactive web based user interface for the AFD Tool. AFD (Automated Fault Detection) Tool is an automatic tool for the identification of a failure in a complex system. The Tool exploits a general purpose mathematical methodology based on Bayes Belief Network. The AFD Tool implemented for Cisco, targets as system under analysis a digital board for networking applications.

What is AFD Tool?

The Automatic Fault Detective Tool is an automatic tool for the identification of a failure in a complex system. The Tool exploits a general purpose mathematical methodology based on Bayes Belief Network. The AFD Tool implemented for Cisco targets as system under analysis a digital board for networking applications. AFD Tool is currently workable trough dos-based Command line instructions; output of all provided operations is Observable in text file format. There is no GUI (Graphical User Interface) for the Tester to interact with the Tool conveniently and effectively. There is a strong absence to manage the test results for further exploration and for the history purposes. These are the major pitfalls or frankly speaking the major weaknesses of the Tool. By more summarizing, it is valid to say that it is a code Library which contains the logic of derived mathematical equations used for Fault detection of complex Cisco systems, and this specific researched and implemented logic is invoke able by simple programming functions or methods under a programming paradigm. The programming environment selected for the implementation of this research was Linux, and programming language was Linux-C. The complete functionality of the AFD Tool as today is available in total around a dozen C language traditional methods. Whereas the detail of each method, its pre and post conditions, that's all will be described in a dedicated session of this document.

As in the above paragraph provided information is a rapid glimpse about the current status of the AFD Tool. And now this is pretty easy to figure out the purpose and need of developing a solid Presentation layer upon these existing code libraries (.a or .dll). This presentation layer should be responsible to overcome all these pitfalls or discrepancies which have been identified in the current state of the Tool.   In more detail description that would be fantastic to classify the solid functional components of the Presentation Layer. So in this regards we have to divide our PL (presentation logic) into different components. In the following text we are going to see the different main purposed components and how they communicate each other to achieve the final Goal of this work.

What is AFDIA?

Before going further, let's rename our *"Presentation Layer"* of the AFD Tool with a more meaningful name, can be called AFDT-IA (Automated Fault Detection Interface Application), so from now on we will always use this purposed name for all references. Back to main stream discussion, AFDT-IA implementation is composed of the following main functional components:

1- AFDT-IA Domain: component responsible for Database transactions during one testing session of the Tool. The more detailed description of this component can be seen in a separate dedicated segment.

2- AFDT-IA Parser: component responsible for text file parsing. As to call the underlying .Dll methods, there is always need to prepare a text-based input file before calling any specific library method and there is also need to parse the resulted text-based file, which contains the output of the specific called methods. This output can contains Errors, warnings or normal expected output results. The more detailed description of this component can be seen in a separate dedicated segment.

3- AFDT-IA Presentation: component responsible for:
   - Preparing Web Pages contain business logic and communicating with other components of the AFDT-IA to cope up with the final results.
   - Call the .Dll methods that is most important and vital task, that is encompasses within the Presentation block, the reason is we have Business logic in the same component. So to call these functional is all about a sequence. The more detailed description of this component can be seen in a separate dedicated segment.

4- AFDT_IA Test: component responsible for Unit Testing of implemented business logic, test the wrapper method which can confirm the validity of the Library methods of the tools. More openly saying, this component is not for deploying, but just to test the major Code snippets.

The ultimate result of the activity is, to come up with an autonomous, effective and efficient Interface of the AFD Tool. Which is web-based, multi-threaded, user friendly, save test results in relational databases and enrich with rich graphical presentation of the output result.

AFD TOOL

This Chapter is solely dedicated to AFD Tool. It describes the general overview of the tool, its functionalities and some schematic insight. The Automatic Fault Detective Tool is an automatic tool for the identification of a failure in a complex system. The Tool exploits a general purpose mathematical methodology based on Bayes Belief Network. The AFD Tool implemented for Cisco targets as system under analysis a digital board for networking applications.

The system is described with a high-level simplified model as a group of Components. Those items represent an abstraction of the real electronic components contained in the board. Components are the nodes of a topological graph, where edges represent an abstraction of the interconnections implemented in the board. The set of all Components contained in a system is referred to as Component Set.

On every system it is possible to run a set of diagnostic operations. Those operations are grouped and they are executed together in a sequential fashion. Each group of operations (and not the single ones) is called Test and it is uniquely identified by a name. For each system a defined group of Tests is available, and it is referred to as Available Test Set (ATS). The expert reduces the complexity of the analysis be selecting a subset of Tests among the available ones, and they are the only tests that can be executed by the test operator. This subset is referred to as Used Test Set (UTS). For each Test, the expert (test designer) defines a specific expected result. The Test is considered failed when the expected results differ from the effective results obtained by running the operations on the system under analysis. The Test is passed otherwise.

Each pair (Component, Test) is related to a Coverage level, representing the degree at which a Test is able to detect a failure on a Component. Coverage are extracted by a coarse label set, namely High (H), Medium (M), Low (L), none (-). The first label describes a relation when a test stimulates largely a component; on the other hand, when the relation is absent there is no stimulation at the entire Component when the Test is run. The information of the coverage levels is summarized in the so called Component-Test Matrix (CTM).

This matrix has dimensions |C| X |T|, where |C| X |T| represents the cardinalities of the component and Test Sets. This matrix is automatically transformed into an equivalent Bayes Belief Network (BBN). All Components and Tests are mapped into the BBN as nodes representing a Boolean random variable, associated to a probability value. For Component nodes, probability range 0 to 1 represents a component that is faulty or non-faulty (the higher the probability node value, the higher the probability of the Component to be faulty-free). For Test nodes, probability range 0 to 1 represents a test that is going to fail or to pass (the higher the probability node value, the higher the probability of the Test to pass).

The BBN is exploited by a numerical engine to extract the probability of a Component to be faulty or not. The Components are sorted in an ascending order and the Components with the lower probability to be faulty-free are referred to as Faulty Candidates. The Bayes Belief Network updates those probabilities values inserting new information about the outcome of new Tests that the test operator decides to run. The AFD Tool contains an Artificial Intelligence engine that is able to sort, at any moment, the tests in UTS that have not been executed yet. A metric function is used to identify the most Significant, among them to be executed first. As it is shown in this picture 1.1, this approach is designed to reduce test session's execution time and improve the failure diagnosis process efficiency.

All faults are observable: this means that whenever a failure occurs in a Product, the misbehavior affects the results of (some) operations and in the Test Set at least one Test fails. For each system, a subset of the UTS is run entirely on the system to detect the presence of a failure. This set of tests is referred to as Initial Test Set (ITS).Entirely means that all tests contained in the ITS are executed as the first step for the analysis of a board. Initial Test Set is defined ad-hoc by the Test Designer or it is automatically extracted from the system description by an extension of the AFD Tool.

The Tool has been implemented on a multi-layered class hierarchy (Figure 0.1). Only the highest layer is available to interface with, in order to implement the User Interface.

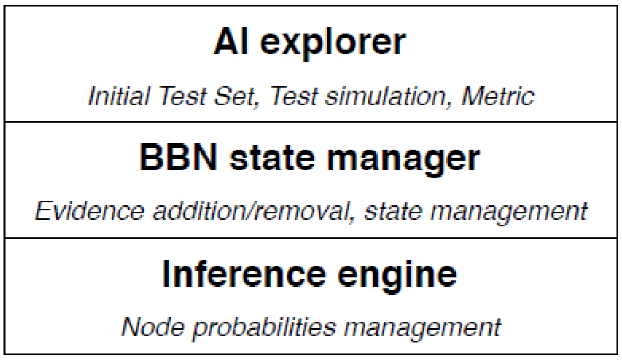

AI explorer
Initial Test Set, Test simulation, Metric
BBN state manager
Evidence addition/removal, state management
Inference engine
Node probabilities management

Figure 2.1: Multi-Layered AFD Tool architecture

FSM (Finite Stat Machine) AFD Tool

Now, it is time to see the functionalities those should be implemented through the AFDIA in order to comply with the actual core requirements. This can be best explained by a Finite stat machine below.

Note: we will use the same FSM Model to describe our AFDIA in detail and see the mapping of each stat into Application User interfaces.

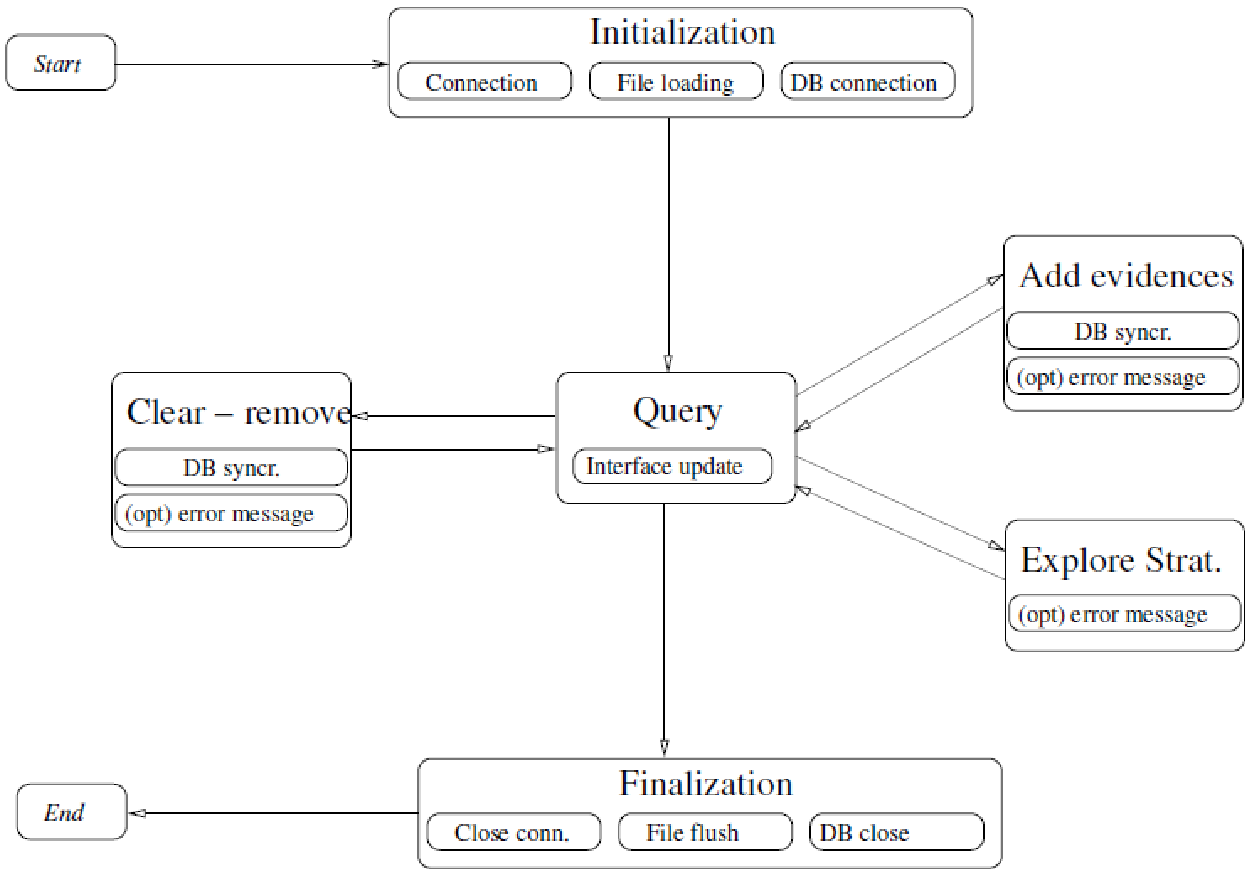


Figure 2.2: Finite Stat Machine Model of the AFD Application

Here we are going to explain each stat description in bullets.

2.1 Initialization:

The initialization stat is composed of three steps that need to be completed sequentially in order to reach the Idle Stat of the FSM.

2.1 .1 Connection

The AFD Server is always up and considered to be working correct when AFDIA starts. AFDIA try to connect with the AFD Server and quit immediately when AFD Server is not available due to any reason. AFDIA in return must inform the user about the specific error trough an explicit message and prevent any further operation.

2.1 .2 File Loading

AFDIA has to inform AFD Server Application about the required files required to initialize the AFD Tool. In particular AFDIA must transmit CTM Model file to AFD Server, if this is not available yet. Without the posting of this Model file you cannot proceed further with your testing and this stat will be considered as incomplete. For file loading there is two main mechanisms one is user would upload a new CTM Model from his computer and transfer it to the server trough AFDIA, the second is user will choose a model from the existing available models on the server.

2.1 .3 Database Connections

The AFD Server is supposed to query the Database containing the information of the test sequences that can be executed on the board under test. This connection is to be correctly activated before exiting the Initialization state; otherwise, an error message is to be returned and the application is to be close. During the Initialization phase, the database will be queried to retrieve the unique ID of current session. A new session entry is to be added to the Database.

2.1 .4 AFD Wrapper Method(s) Initialization

Once the connection to the AFD Server is open, source files are available, database test connection is active, now it is possible to call the AFD Wrapper Initialization methods. They can return a no-error code, and the application moves to the Query (idle) state. Other operations related to the implementation details could (must) be performed here before making the application operative. Otherwise, specific error code (and messages) must be logged and the application must quit, signaling an explicit error message to the User.

2.2 Query

Query state is the idle state of the FSM. This state can be reached from the initialization state; it can also be reached from the *Add*, *Explore* and *clear* operations stated in the FSM, and representing the real interaction from the User and AFD Tool.

2.2 .1 Test Visualization

AFDIA trough the AFD Server can query the AFD Wrapper method to get the information of the Test nodes. Particularly user is able verify in a table-based or equivalent form, the list of tests for which the outcome passed/failed known. The user should be able to visualize the static reference information (test name and test ID) and dynamic state-specific information (test outcome). The user should be able to sort the information in multiple ways.

Initialization Test set Visualization: The AFD Wrapper will contain a method used to obtain a list of tests that the operator is required to execute in any case at the beginning of a test session. This method returns a list of tests that depend only on the source file containing the CTM, and it is to be called once (repeated calls will return exactly the same values). The visualization of this list is to be similar with the visualization of the other tests.
It is up to the implementer to decide how to organize the visualization of this information in the AFD User Interface.

2.2 .2 Component Visualization

AFDIA trough the AFD Server can query the AFD Wrapper method to get the information of the Component nodes. Particularly user is able verify in a table-based or equivalent form, the list of components contained in the system under analysis. The user should be able to visualize the reference information (component name and component ID) and dynamic state-specific information (faulty candidate probability) and to sort information in multiple ways. The user should be able to sort the information in multiple ways.

2.2 .3 Strategy Visualization

AFD Server should recognize as strategies all tests not performed yet, or groups of them. A strategy identify a test or a group of tests that the AFD Tool suggests to the user (test operator) in order to shorten test session time and to increase the efficiency of the fault localization. In a preliminary phase, and in the current implemented version of the AFD Wrapper, only single tests can be seen as strategies. They can be accessed with specific query methods available in the AFD Wrapper.

The user should be able to visualize all possible strategies, in a compact fashion (strategy name, strategy ID) or in an extended fashion (list of tests composing the strategy, numeric values of the metric used to sort the strategy efficiency). The table-based visualization is mandatory but other strategies are also possible. The user should be able to shorten the visualized list of strategies in an efficient way, in order to concentrate only on the most promising ones.

2.3 Add Evidences

The user will be presented with a list of available tests. On this list, all tests that have not been executed yet will be available for selection, and a unique “Execute” command should be issued. The selection mode definition is not constrained. A proposed mechanism is an exclusive check-box, for instance Select Initial Test Set and Select Strategy. In the latter case, the user should be able to pick up a specific strategy on a list or a combo-box.

Test Execution: The procedure to be followed in order to execute tests is shown in Figure 2.3. The test database contains the outcome of all available tests. A future implementation will execute the test and will insert the test outcome into the database only on-demand, i.e. when a test execution is required. For a prototype version, this operation is not required. Thus the AFD Server will query the test database to obtain the outcome of the execution of the required test or of the required group of tests. Once the query result is available, the AFD Server will call the specific methods of the AFD Wrapper in order to synchronize the AFD Tool application with the new information.

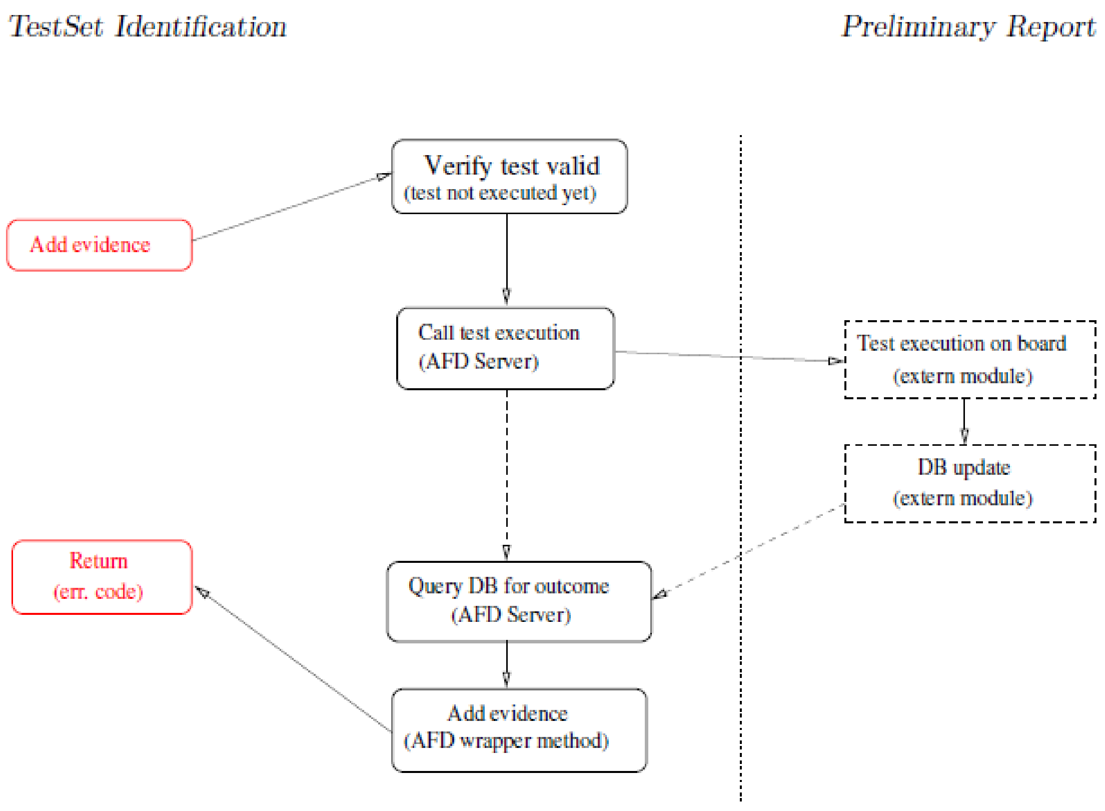

TestSet Identification
Preliminary Report
Verify test valid (test not executed yet)
Add evidence
Call test execution (AFD Server)
Test execution on board (extern module)
DB update (extern module)
Return (err. code)
Query DB for outcome (AFD Server)
Add evidence (AFD wrapper method)

Figure 2.3: Add Evidence Process (AFD Server and Test DB)

The AFD Tool will return to the Query idle state when the add evidence operations have been completed.

2.4 Explore Strategies

In order to decide the optimal test sequencing for diagnosis, the AFD Tool provides the user with an automatic exploration methodology. This exploration ration maximizes the residual information that can be obtained by each test (or by a group of tests) that have not been executed yet. This information is transformed in a Strategy ranking that is proposed to the user according to 2.2.3. The user should be able to select an exploration method of the remaining tests, which will produce a set of possible Strategies. Those exploration methods are, for instance, select next test method, select next 2-tests method, select next k-tests method, at the moment of writing, only the first exploration policy is available.
The user will be able to explore the possible Strategy set starting from the Query state. This exploration will be executed with the specific methods available in the AFD Wrapper. This operation can be executed an undetermined number of times but the result would not change until Add or Remove states have not been successfully traversed at least once.

The execution of the Exploration operation does not affect the content of the test database. The execution of this exploration could be implemented in an automatic policy after any Add or Remove operation, or it could be demanded to the user with an explicit call. The AFD Tool will return to the Query idle state when the explore strategy operations have been completed.

2.5 Remove/Clean Evidences

The AFD Server provides the user with a mechanism to remove the outcome of a test or a group of tests previously inserted. This possibility is required for usability but it has to be strictly regulated for (back-ward) compatibility reasons. In particular, two situations could occur in a typical remove/clear scenario.

Specific Evidence Removal: Given the visualization mode described in 2.2.1, the user will be able to remove the insertion of the outcome evidence relative to a test or a specific group of tests. The AFD Server will be responsible for updating the test database content, modified with insertion procedure described in 2.3, in order to keep the consistency of the test database.
Complete evidence removal: Given the visualization mode described in 2.2.1, the user will be able to remove the insertion of any available outcome evidence. The AFD Server will be responsible for updating the test database content, modified with insertion procedure described in 2.3, in order to keep the consistency of the test database.

AFD Tool Wrapper Methods & their Utilization

The interaction of the AFDIA with the AFD Tool is performed using a set of functions. Those functions represent the states of the AFD Tool, as depicted in the Finite State Machine of the AFD Tool.

Introduction: we recall quickly the four important states of the FSM of AFD Tool:

1. *Initialization:* the CTM Model loaded, along with the components and test lists, all probabilities are cleared and set to initial values, all tests outcomes are cleared and set to value UNKOWN.

2. *Strategies exploration:* Tests to be executed are evaluated by the AFD Tool and ranked. Different policies of exploration are available like single, Multiple and K-Step. A list of possible tests to be executed and an evaluation matric returned.

3. *Test Outcome insertion and Removal:* When new tests are executed and their outcomes (PASS or FAIL) become available, they are fed to the system. If this is necessary test outcomes previously inserted can be also removed. With these operations, components and test probabilities are updated.

4. *Finalization* the system model is unloaded, all structures are cleared. The tool is ready for another initialization (or to be closed).

Usually, the user application starts a test session with step 1. Then steps 2 and 3 are repeated alternatively, executing the tests selected by step 2 with external equipment and passing the PASS/FAIL outcome returned by such external equipment to AFD in step 3. All test sessions are concluded with step 4.

AFD Tool Methods (Input/output mechanism)

Passing Parameter: Each function requires a text file containing its input parameters. If the function name is XYZ (), then the input file name must be XYZ_input.txt. These input files should be in C:/TempAFDTool, if these files do not exist, then code could crash. Default locations for these files are hard coded in AFD_common.h as AFD_TEXTFILES_PATH macro. This default path is changeable but in case of change a compilation is needed of the project, because at the end we are using .Dll, which is a compiled file, so this information must be noted into the Dll. If input file are absent, the function will not raise an exception. A specific error will be reported in the output _le, with a specific error code (see later).

Then the function will return without performing any further operation, nor raising any exception. Do note that if the output _le cannot be created, a message is sent to STDERR; also in this case, the function will return without performing any further operation, nor raising any exception. All parameters must be written in the input file as a comma-separated list of strings. A comma is required also after the last parameter. This list of string must be written in the first line of the file.

Example: parameter1, parameter2, parameter3... parameter Last, {EOF}

Where {EOF} represents the End Of File character, all parameters are strictly ordered, as specified for each function. It is up to the user application to ensure that the content of the input file is correct. If a function requires no input, the file is still required; otherwise an error will be reported. However, the file is simply opened and closed, but its content is not read.

Return value: Each function generates a text file containing its output. If the function name is XYZ (), the output file name must be XYZ_output.txt. The output file is always overwritten and its previous or existing content will be discarded. These input files should be in C:/TempAFDTool, if these files do not exist, then code could crash. Default locations for these files are hard coded in AFD_common.h as AFD_TEXTFILES_PATH macro.

The output file is always created, before trying to access the input file. Thus, the output file will contain an error if the input file cannot be created. In this case, the function will return without performing any further operation, nor raising any exception. Do note that if the output file cannot be created, a message is sent to STDERR; also in this case, the function will return without performing any further operation, nor raising any exception.

Output files are arranged in two lines:

- ✓ The first line will contain a positive integer value representing the error code. Error codes are reported in AFD common.h. When a function returns with no error, the AFD ERROR NONE code (value 0) is returned. Otherwise, a positive value will report another specific error condition.

- ✓ The second line (or more) will contain the output of the function, and its formatting depends on the specific called function. In any case, all values are returned as comma-separated values list.

- **Initialization**

Void AFD_BBN_Library_Init (), this function is the first function be called to initialize the Library. It loads the system model (BNE file), and reset all evidences (test outcomes). It is used also to re-initialize the AFD Tool but, in this case, a call to AFD BBN Finalize () is required first.

Input: The function requires a single parameter, which is the name of the BNE model to be loaded. An error occurs if input file is missing in the default path, cannot open file or BNE model file contains a non-valid or ill-formatted model.

Output: The function only returns error code and, in case, the error message associated with the error.

Void AFD_BBN_Library_getSize (), this function is used to retrieve the dimension of the loaded model, for example the number of components and tests.

Input: The function requires no input.

Output: The function only returns error code and, in case, the error message associated with the error. If there is no error, the function returns two-element list of strings, containing the number of components and the number of tests.

- **Get Probabilities Methods**

Void AFD_BBN_Library_getComponents_All (), this function returns a list containing all information about the components of the system.

Input: The function requires no input.

Output: The function returns the error code and, in case, the error message associated with the error. If no error occurred, the function returns a list of strings: the first element of the list is an integer, containing the number of components (CNUM) information returned; then, for each component, three values are listed: the component name, the component a-priori probability indicated in the BNE model, the component probability (this last values can be retrieved also using AFD BBN Library getComponents Fast ()). All information is ordered according to components indices (from 0 to CNUM-1). The total number of values contained in this list is 1+3*CNUM.

Void AFD_BBN_Library_getComponents_Fast (), this function returns a list containing only probabilities about the components of the system.

Input: The function requires no input.

Output: The function returns the error code and, in case, the error message associated with the error. If no error occurred, the function returns a list of strings: the first element of the list is an integer, containing the number of components (CNUM) probabilities returned; the following values are double, containing the probabilities ordered according to components indices (from 0 to CNUM-1). The total number of values contained in this list is 1+CNUM.

Void AFD_BBN_Library_getTests_All (), this function returns a list containing all information about the tests of the system.

Input: The function requires no input.

Output: The function returns the error code and, in case, the error message associated with the error. If no error occurred, the function returns a list of strings: the first element of the list is an integer, containing the number of tests (TNUM) information returned; then, for each test, three values are listed: the test name, the test status (Pass, Fail & Unknown), the test probability.

Void AFD BBN Library getTests_Fast (), this function returns a list containing only probabilities about the tests of the system.

Input: The function requires no input.

Output: The function returns the error code and, in case, the error message associated with the error. If no error occurred, the function returns a list of strings: the first element of the list is an integer, containing the number of tests (TNUM) probabilities returned; the following values are double, containing the probabilities ordered according to tests indices (from 0 to TNUM-1).The total number of values contained in this list is 1+TNUM.

- **Add/Remove Evidences Methods**

Void AFD_BBN_Library addEvidences () this function is used to insert the evidence of a newly executed test. Test status is changed from UNKNOWN to PASS or FAIL. If a test was already in PASS or FAIL status, new evidences will overwrite old ones.

Input: This function requires the list of evidences, comma-separated. All Evidences about Pass will follow this pattern P:0, P:3,P:7, whereas all evidences about Fail will follow this pattern F:0,F:5,F:9. Both Pass and Fail evidences can be mixed and there is no restricting order of recording these evidences.

Output: The function returns the error code and, in case, the error message associated with the error.

Void AFD_BBN_Library_removeEvidences () this function is used to remove the evidence of a test previously inserted Test Status is changed from PASS or FAIL to Unknown status.

Input: This function requires the list (comma-separated) of test indices whose evidence is to be removed, comma-separated. For example

1,2,5,0,
Will remove evidences of tests 0,1,2,5 (no ordering is important).

Output: The function returns the error code and, in case, the error message associated with the error.

- **Explore Strategies**

Void AFD_BBN_Library_getStrategies_All (), this function evaluates all possible strategies (list of tests) to be executed, and it ranks optimal strategies according to a quantitative metric (higher value, better quality). The strategy can be evaluated using different policies: a list of implemented policies is indicated in AFD Strategies.h (enum type AFD STRATEGY TYPE).

Input: A single element list is required, containing an integer. An error occurs if none or more elements are passed in this list. The integer MUST be a valid code, representing an implemented strategy (enum type AFD STRATEGY TYPE). It is an error to pass AFD STRATEGY NONE, corresponding to the value 0. The number of tests (TSNUM) contained in each strategy depends on the exploration policy selected.

Output: The function returns the error code and, in case, the error message associated with the error. If no error occurs, the function returns a list comma-separated list. The first value of the list is an integer, containing the number of strategies (SNUM) elaborated and returned. Then, each strategy contains a list of TSNUM test indices, followed by the quality metric value (double). Eventually, the returned list will contain 1+SNUM*(TSNUM+1)

- **Finalize AFD Tool**

Void AFD_BBN_Library_Finalize (), This function is to be called when the library is no more required, or the AFD Tool is to be re-initialized with another call to AFD BBN Library Init(), since it frees dynamically allocated structures and re-initializes their contents

Input: The function requires no input.

Output: The function only returns error code and, in case, the error message associated with the error.

AFDIA – Interface Application

This chapter is titled with the name of our actual implemented project; the content of this chapter will start with a brief introductive overview of AFDIA, then Architecture of AFDIA its importance and functional detail, Tool and technologies have used for the development and finishing with the DB schemas with its thorough detail.

Overview: AFDIA is a web based, concurrent, interactive, user friendly and with the capability of managing the history of performed tests in the database. It can be seen as a Presentation Layer upon the AFD Tool. It communicates with the AFD Tool's Dll methods, invokes their programming modules, parsed their resulted files, and stored the performed tests results in relation database. In all it is a website application which provides all the features described above.

AFDIA architecturally divided into three parts, in the bullets below you can find an abstract overview of these major components

- **AFDOnline:** this website titled with AFDOnline, which contains a number of interactive web pages with which Test Engineer can interact. Examples include uploading a CTM Model, View All Tests, View All Components, and Remove Evidences etc.

- **Test Administrator:** this website titled with AFDIA - TestAdministrator, demonstrate some very useful functionalities. In fact when a Test Engineer request to Test Administrator (a role who record the evidences) for evidence insertion which he/she noted from the Cisco Hardware Test machines. So in that case the entire request from AFDOnline came to AFDIA – TestAdministrator. This application is physically located on a separate place as the Test Administrator can be and more likely is on a far located place. I would like to explain this usage by a real life example. Let's say our Test Engineer is working in Milano, he uploads a CTM Model trough the AFDOnline, and now he sent request to Test Administrator to know the outcomes of specific tests. Promptly In that case, he is sending some requests to AFDIA – TestAdministrator website. Where all the requests will be entertained.

- **AFDIA Web Server:** this website titled with AFDIA - WebServices, contains few web Methods, which used by both AFDOnline application and AFDIA – TestAdministrator to share the data and consistency purposes.

AFDIA Architecture, its importance and functional detail: purpose of this segment is to go into the deep overview of AFDIA's Architecture its important and functional detail.

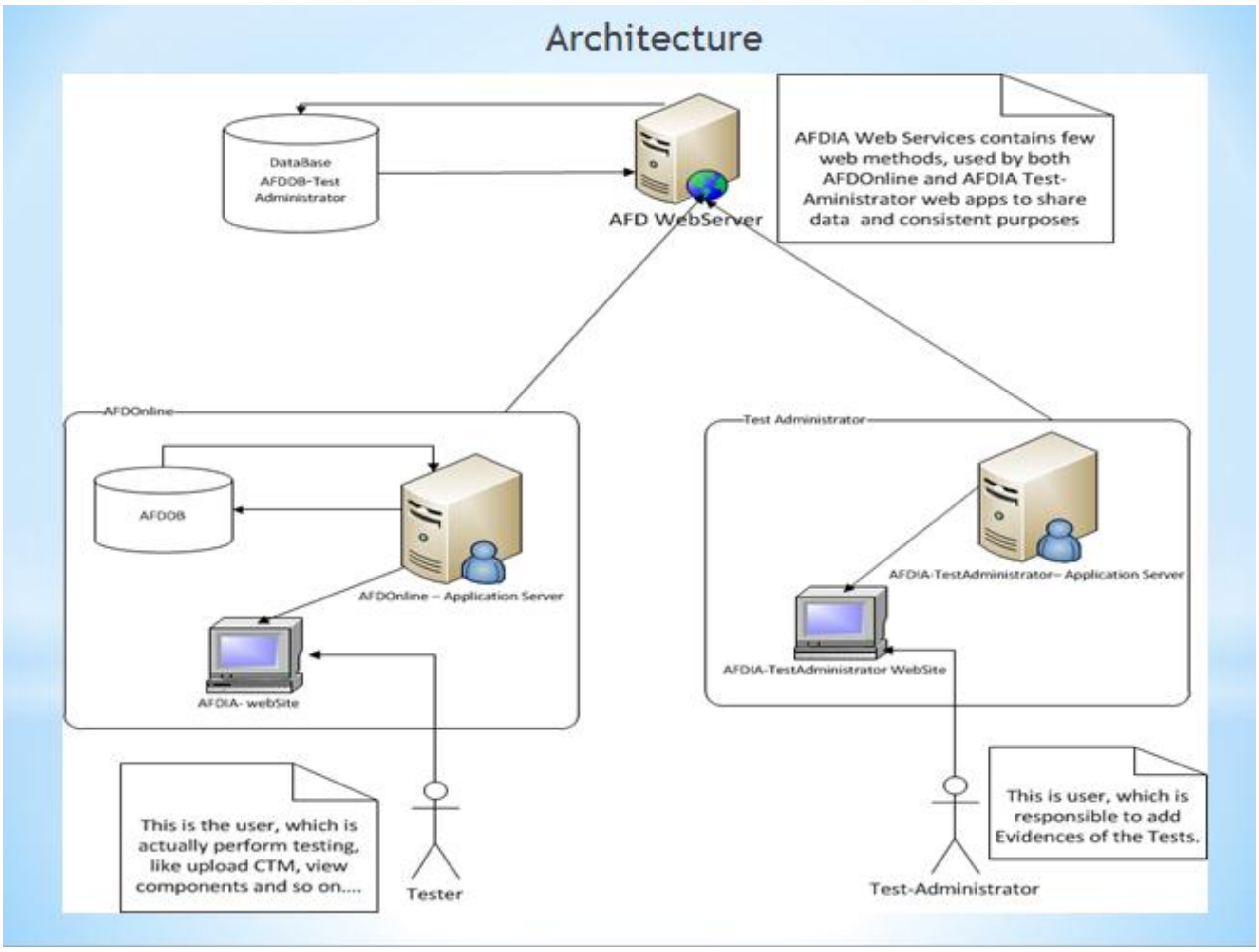

Architecture
DataBase AFDDB-Test Administrator
AFD WebServer
AFDIA Web Services contains few web methods, used by both AFDOnline and AFDIA Test-Aministrator web apps to share data and consistent purposes
AFDOnline
AFDDB
AFDOnline – Application Server
AFDIA- webSite
Test Administrator
AFDIA-TestAdministrator– Application Server
AFDIA-TestAdministrator WebSite
This is the user, which is actually perform testing, like upload CTM, view components and so on....
Tester
Test-Administrator
This is user, which is responsible to add Evidences of the Tests.

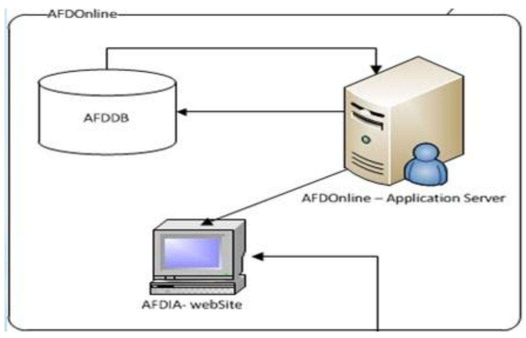
Figure 3.1: AFDIA Component and Communicative Architecture

Architectural Roles: There is the complete list of architectural rules those you can notice in the figure above 3.1, detail explanation of each role that how a particular role is playing to achieve the overall process's objective. Here is the abstract list:

- **AFDOnline**
- **Test Administrator**
- **AFD WebServer**
- **Tester**
- **Test Administrator**

AFDOnline: AFDOnline is a Web Based Application implemented via dotNet framework. AFDOnline – Application Server is a web server which contained all the business Logic code, Database access code and would host our web site for the users outside from internet world. AFDOnline has its own Data Store, which is named as AFDBD. This database contains all the information about performed tests, all components, strategies, all uploaded models and available boards.

AFDOnline, is the same application from where we access AFDTool Dll methods, which we described in the AFDTool chapter. These are around ten methods implemented in C, and dispose all the functionality of AFD Tool. As we know C Dll are unmanaged Code. And our .Net framework runs under Managed environment, so we have to find how we can call AFDTool methods (unmanaged Code) from our .Net Application AFDOnline (Managed environment). Follow this link to understand difference between Managed and Unmanaged code and how to call unmanaged code from Managed environment like .Net http://msdn.microsoft.com/en-us/library/ms973872.aspx. Although in the figure 3.1, you can see abstract view of the AFDOnline component like this:



Categorically, here you cannot see AFDTool (C Dll) as a separate functional component. So just for the clear and mature understanding I am pasting another picture below which gives you a complete thorough view of real existing calling mechanism.

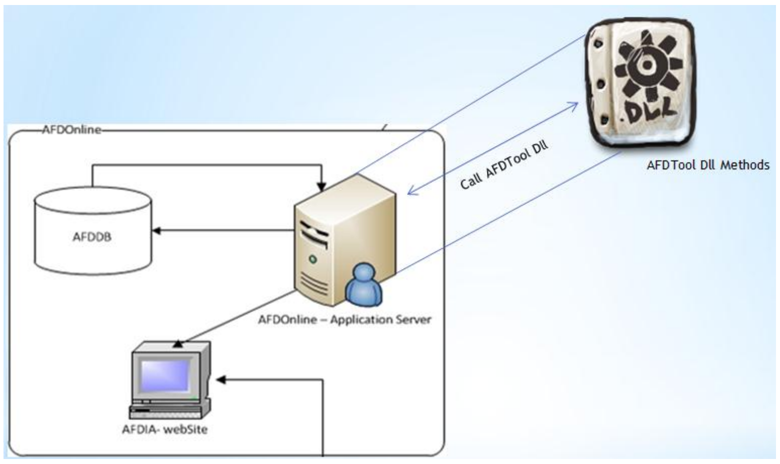


Figure 3.2: AFDOnline Communicative Architecture with AFD Tool Dll

Picture above giving you sense that AFDOnline – Application Server is utilizing some other functional components and this is AFDTool Dll which contain definition of all provided methods.

As we have encountered two main important aspects of AFDOnline, first host AFDOnline web Application by which users can access the web application over internet and second is communication with the AFD Tool Dll on every request when user demand it to view.

Last interaction of AFDOnline is with AFD WebServer (a web Application that expose web Services for Evidence Insertion management, and to keep Communication Asynchronous between AFDOnline and Test Administrator). In Live application utilization, there is one web page interface (called AddEvidence.aspx) which demonstrate to user all the tests whose results are yet not known. User can select some tests and sent these tests to Test Administrator (a role in testing process, who add results of performed tests) to know their current status by calling some Web Services exposed by AFD WebServer.

Test Administrator: Test Administrator in real, is a role in Cisco Testing Process. This block represent another web application, named as like its user's role just for viable visibility of its functionality. This application is also a web application implemented via dotNet framework and its contains few web pages. Its main interaction is with AFD WebServer application for consuming some of its exposed web services.

To understand quickly the Role of Test Administrator, we should start from the AFDOnline web site application. Assume we start AFDOnline application, upload a new CTM Model, view all tests, components etc. and then we navigate to AddEvidences.aspx Page. This page is dedicated to send tests request to another application from where a Test Administrator will add his evidences by exploring the result from Hardware Testers machines. This application is called Test Administrator web app. For recording the tests result, Test Administrator will call the web services exposed by AFD Web Server. AFD WebServer (is a web Application that expose web Services for Evidence Insertion management, and to keep Communication Asynchronous between AFDOnline and Test Administrator).

One important thing to notice that, Test Administrator has no Local Data Store. its reason is obvious that all the requested tests is stored in the Data Store of AFD WebServer. So when Administrator will login to the Test Administration application, the Test Administrator app will call the web Service to Load all the tests requested till last moment. Administrator will add the evidences and then app will send these results to the AFD WebServer using web service. And AFDOnline application can be updated its test results by utilizing the web service method.

In nutshell, from architectural point of view this block contain an web app that would be utilized by Administrator to record the test evidences explored from the Hardware tester machine. This app has no local data store, no any communication with AFD Tool Dll and for test result management just use web Services exposed by AFD WebServer.

AFD WebServer: AFD WebServer is a web application which exposed few web services or web methods that would be utilized by AFDOnline and Test Administrator web apps. This application is represented as a bridge between AFDOnline and Test Administrator. Its main interaction is with both these two apps. This app is implemented via dotNet as others two. It has not any web interface because this application will be used just by other apps to manage the Test Evidence insertions. So technically it should exposed something communicable by other web apps.

For a comprehensive understanding and to feel the true need of AFD WebServer, lets sketch a scenario of overall flow. User start AFDOnline application, upload a new CTM Model or select an existing one, view all tests, view components and perform as available commands user want. Then navigate to AddEvidences.aspx Page. This page is dedicated to send tests request to another application from where a Test Administrator will add his evidences by exploring the result from Hardware Testers machines. This application is called Test Administrator web app. For recording the tests result, Test Administrator will call the web services exposed by **AFD Web Server**. So here it lies in the communication between two apps.

In nutshell, this is a bridge between AFDOnline and Test Administrator. It has its own Database store, it communicate with other apps trough web services mechanism and it has not any interaction with AFD Tool Dll.

Tester: This is tester (a human not any machine or system), who is responsible for performing testing. Tester is the most important part of Cisco testing process. His task is to interact with the AFDOnline completely. Tester can utilize all of the available features. Tester does not know about the other applications taking part in the testing process, for him AFDOnline web site is the whole testing world.

Let’s take an example to be more familiar with tasks of Tester. Tester would run the AFDOnline, upload a new CTM model from local machine or select an already existing CTM model. Tester view all Testing results, view components and may perform a series of steps. Tester may request to send tests with unknown results to Test Administrator. Than upon results arrivals, tester can view it. This is the usual phenomena of being a Tester in AFDIA.

Test Administrator: Test Administrator (a human not any machine or web System) have the same functional status as Tester had. Test Administrator is the person within Cisco who would record the evidences of requested tests (requested from Tester trough AFDOnline) after exploring the results from Hardware testing machine. Test Administrator and Tester are not obligatory to be at same office geographically.

Let’s take an example to be more familiar with tasks of Test Administrator. Test Administrator would run the Test Administrator web app, Test Administrator may record evidences for all tests (with unknown results) sent by the Tester for evidence insertion. Test Administrator must record all these evidences or results by exploring the hardware testing machines. Than upon sending results to Tester successfully its role would be totally scoped.

Tools/Technologies Used in Development

The scope of this segment will put light upon Tool/Technologies used in the development phase of AFDIA – Interface application. It is important part of the chapter because here you can find Tools, Technologies and methodologies which has been used to implement the architecture described in the previous portion of the same chapter. Commenting in this segment would be at abstract level, we will address at the level of architectural components (AFDOnline, AFD WebServer and Test Administrator): Example can be like which kind of Database management system we are using in AFDOnline, how we query the data, by using simple SQL statement or any specified feature enriched ORM (Object Relational Mapper) and how we manage the Database access layer in the application layered architecture.

So, it is one dive deep into each web application taking part in the testing process to complete the overall objective of the system. The explanation method that can be more intuitive is that we would see each major block of the figure 3.1, comment it with used technologies respectively. We have three major building blocks.

- AFDOnline
- Test Administrator
- AFD WebServer

<u>AFDOnline</u>: let's take a quick review of block diagram given below.

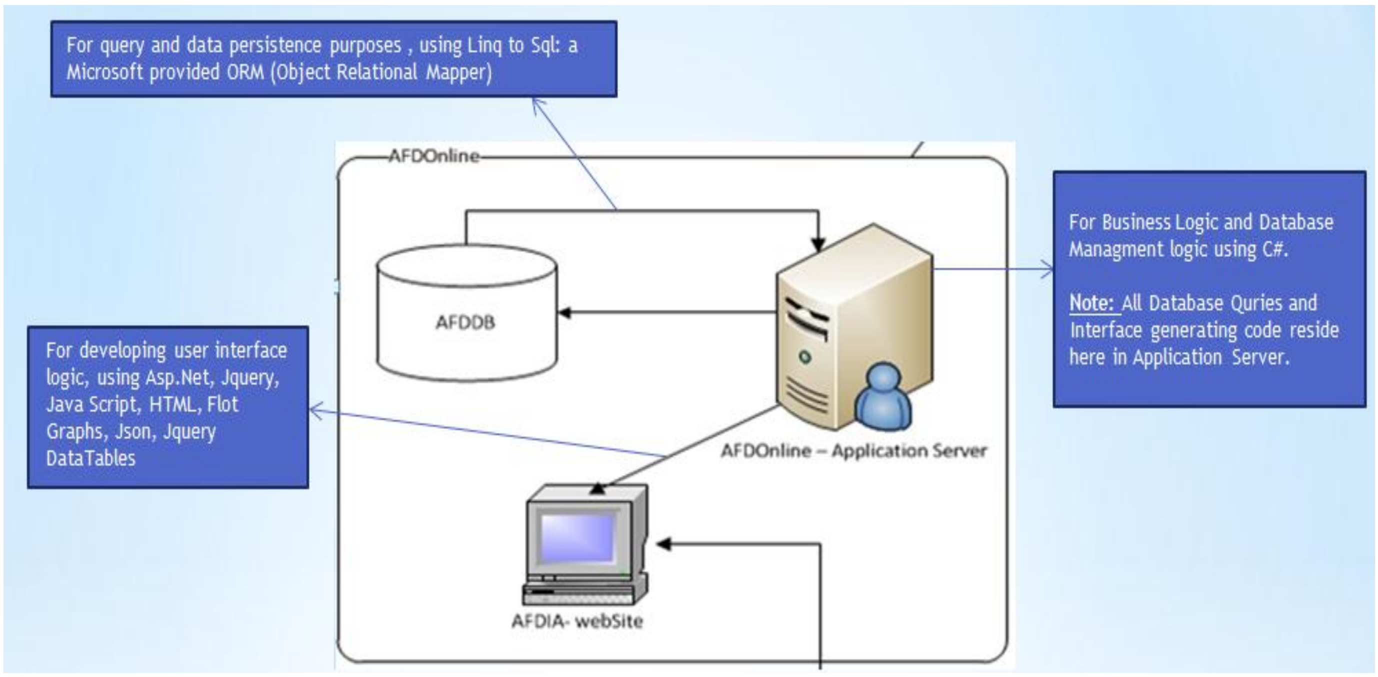

For query and data persistence purposes , using Linq to Sql: a Microsoft provided ORM (Object Relational Mapper)
AFDOnline
AFDDB
For Business Logic and Database Managment logic using C#.
Note: All Database Quries and Interface generating code reside here in Application Server.
For developing user interface logic, using Asp.Net, Jquery, Java Script, HTML, Flot Graphs, Json, Jquery DataTables
AFDOnline – Application Server
AFDIA- webSite

Figure 3.3: AFDOnline used Technologies/Tools

DBMS: MS SQL Server 2008 is being used as a DBMS.

Data Access Technology: Linq to SQL(ORM - Object Relational Framework) is used for Data Manipulation and data creation. It is a Microsoft provided framework for managing database efficiently and effectively with less effort and low-cost.

Database Logic: for database logic, AFDOnline used Linq (Language intergraded Query). Linq is an Data Source independent query language, it is an ideal language who don't care about the target data source it can be relational database, XMLs, Entities and a long range of list.

Business Logic: C# being selected for writing business logic.

Presentation Logic: ASP.net, HTML, JavaScript and Jquery are used for generating presentation logic.

<u>**Test Administrator**</u>**:** let's take a quick review of block diagram given below.

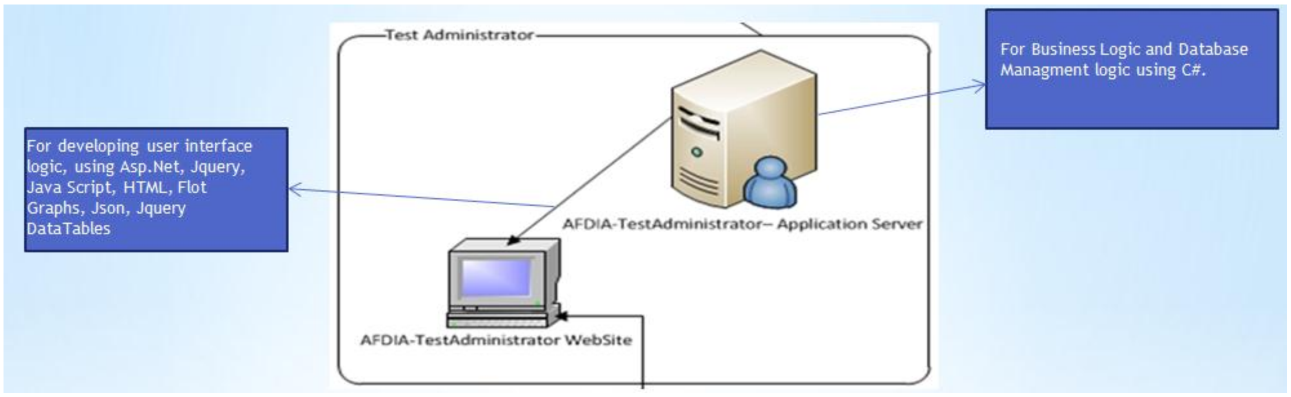

Test Administrator
For Business Logic and Database Managment logic using C#.
For developing user interface logic, using Asp.Net, Jquery, Java Script, HTML, Flot Graphs, Json, Jquery DataTables
AFDIA-TestAdministrator– Application Server
AFDIA-TestAdministrator WebSite

Figure 3.4: AFDIA Test Administrator used Technologies/Tools

DBMS: There is no database in this application, so no need any DBMS. As Test Administrator web application utilize web services to communicate its completed tasks.

Data Access Technology: None.

Database Logic: None.

Business Logic: C# being selected for writing business logic.

Presentation Logic: ASP.net, HTML, JavaScript and Jquery are used for generating presentation logic.

AFD WebServer: let's take a quick review of block diagram given below.

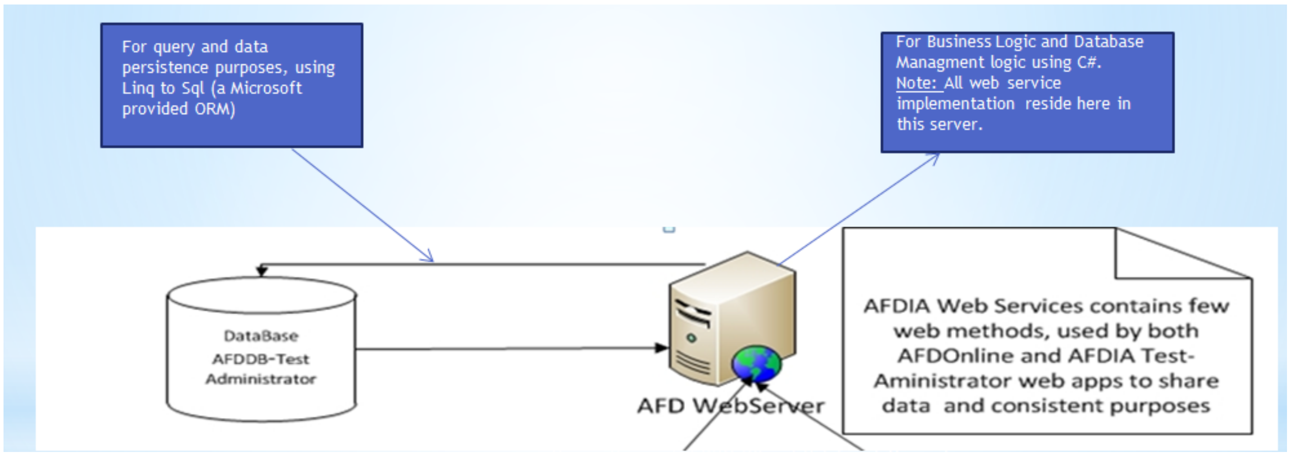

For query and data persistence purposes, using Linq to Sql (a Microsoft provided ORM)
For Business Logic and Database Managment logic using C#. Note: All web service implementation reside here in this server.
DataBase AFDDB-Test Administrator
AFD WebServer
AFDIA Web Services contains few web methods, used by both AFDOnline and AFDIA Test-Aministrator web apps to share data and consistent purposes

Figure 3.5: AFDIA WebServer used Technologies/Tools

DBMS: MS SQL Server 2008 is being used as a DBMS.

Data Access Technology: Linq to SQL(ORM - Object Relational Framework) is used for Data Manipulation and data creation. It is a Microsoft provided framework for managing database efficiently and effectively with less effort and low-cost.

Database Logic: for database logic, AFDOnline used Linq (Language intergraded Query). Linq is an Data Source independent query language, it is an ideal language who don't care about the target data source it can be relational database, XMLs, Entities and a long range of list.

Business Logic: C# being selected for writing business logic.

Presentation Logic: None because this project just expose web services used by other web applications.

Database Schemas

The term schema refers to the overall design of the database. It is a logical database description and is drawn as a chart of the type of data that are used it gives the name of entity and relationship between them e-g information display system that gives info about arrival and departure of train and aircraft at stations. The schema will remain the same though the values displayed in the system will change from time to time. The term subschema refers to the same view but for the data item types and record types which are used in a particular application or by particular user.

This AFDDB database is a significative subset of the real database available in the Cisco testing framework the AFD Tool is going to be integrated in. This database is to be used as a stub in order to provide a realistic user experience for the testing phase of the AFD User Interface prototype.

In AFDIA – Interface application we have two databases AFDDB and AFDDB – Test Administrator. The roles and importance of these Databases has been described above in this chapter that why we created these databases. Now I will move into the detail discussion of these dbs. I will name all the tables in each database and their need in the specific context.

AFDDB: AFDDB only queried by AFDOnline. It is managed by Linq to Sql. An ORM (object Relational Mapper) provided by Microsoft. This database has six database tables those are relational among them. This database store the information about available Board on which we can make tests, CTM models detail, Performed Test results, available strategies, all components and a Session table. Figure below shows all available tables in this db.

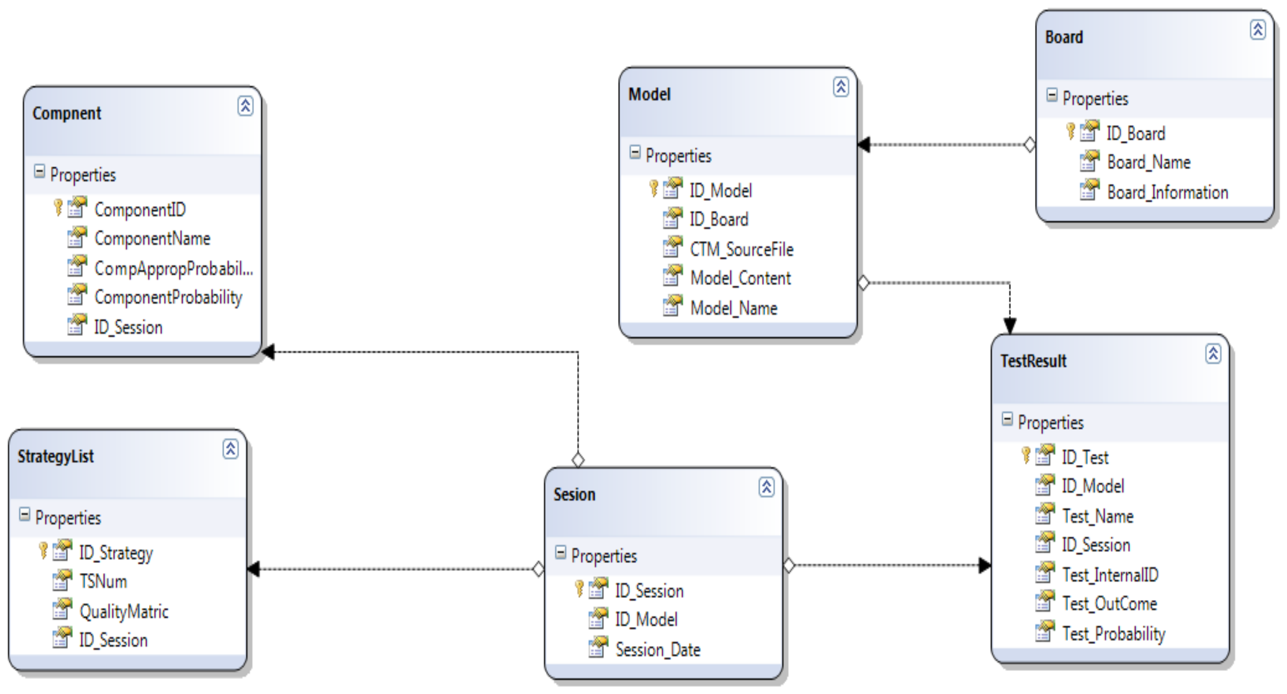


Figure 3.6: AFDDB schema

Let's see each database object in detail, its fields; it's relational with other table(s) and its Access level (read or writes).

Board: This table contains information about the board under test. This information is irrelevant with respect to the AFD Tool. It is important because it creates a unique ID reference (primary key) for any board that can be tested. Further- more, this ID is used as foreign key in Model tables. Permission for this table is read-only from the point of view of the AFDOnline. So as this table is read-only the data must be inserted by DBA or anyone who has authority to feed this information. ***Board_Name*** is the field represents the Name of the board and ***Board_Information*** is used to store the detail information about the Board.

Model: This table contains an entry for each available CTM model (source file). Each entry is to be associated uniquely with an entry in the board model. Also, the file path to access the file is to be specified. Permission for this table is read-only during the execution of a testing session of the AFD Tool. However, permission is read-write because the User Interface should allow the user to upload a new source file into the system.

Field Name	Detail
ID_Model	Unique Id to identify the model in the table
ID_Board	Foreign Key that refer to the Board associated with the Model
CTM_SourceFile	Store the relative path of the CTM Model text file
Model_Content	It contains the actual content of the CTM Model
Model_Name	Represents the name of the Model file

Table 3.2: Scheme table Dbo.Model

TestResult: This table contains the information about all tests that are available in the Cisco testing framework. This table denes a unique ID for each entry, but this is not necessary the same as the internal ID of the AFD Tool; this ID is used as foreign key in other tables. Each test is related to a board. Each test is identified with a string name, which should be the same in the CTM source le. The Unique ID is the Cisco unique code to identify a specific test.

Filed Name	Detail
ID_Test	Unique Id to identify test in the table
ID_Model	Foreign Key that refer to the Model associated with the test
Test_Name	Store Name of the test
ID_Session	Foreign Key that refer to the Session associated with the test
Test_InternalID	It represents the internal ID of the test
Test_Outcome	It store the test result
Test_Probability	It store the probability value of a test range from 0 to 1

Table 3.3: Scheme table Dbo.TestResult

Sesion: This table contains a unique entry for each testing session. A testing session records any valid traversal of the Initialization state of the AFD FSM (Figure 2.2). During this traversal, the AFDOnline will insert a new entry in the table, whether the session will be successfully opened or not. The session will be related to a source file and a (unique) time-stamp. The returned value of the session ID field will be used to call the specific initialization method in the AFD Wrapper to custom the output, warning and error messages the AFDTool could return. Moreover, this session ID will be used for the database update described for the Add evidence state. Permission for this table is read-write from the point of view of the AFDOnline.

Filed Name	Detail
ID_Session	Unique Id to identify Session in the table
ID_Model	Foreign Key that refer to the Model associated with the Session
Session_Date	Store the timestamp of the Session

Table 3.4: Scheme table Dbo.Sesion

Component: This table contains the information about all components that are available in the Cisco testing framework. This table denes a unique ID for each entry, but this is not necessary the same as the internal ID of the AFD Tool. Each component is related to a session. Each component is identified with a string name. The UniqueID is the Cisco unique code to identify a specific test.

Filed Name	Detail
ComponentID	Unique Id to identify Component in the table
ComponentName	Name of the Component
CompApropProbability	Represent the appropriate probability of a component
ComponentProbability	This represent the actual resulted probability
ID_Session	Foreign Key that refer to the Session associated with the component testing

Table 3.5: Scheme table Dbo.Component

AFDDBAdministrator: AFDDBAdministrator only queried by AFDIA-WebServices. It is managed by Linq to Sql. An ORM (object Relational Mapper) provided by Microsoft. This database has two database tables. This database stores the information about the requested tests that need to be resulted by Test Administrator. So, both AFDOnline and AFDIA – TestAdministrator interact with this database trough web services. Figure below shows all available tables in this db.

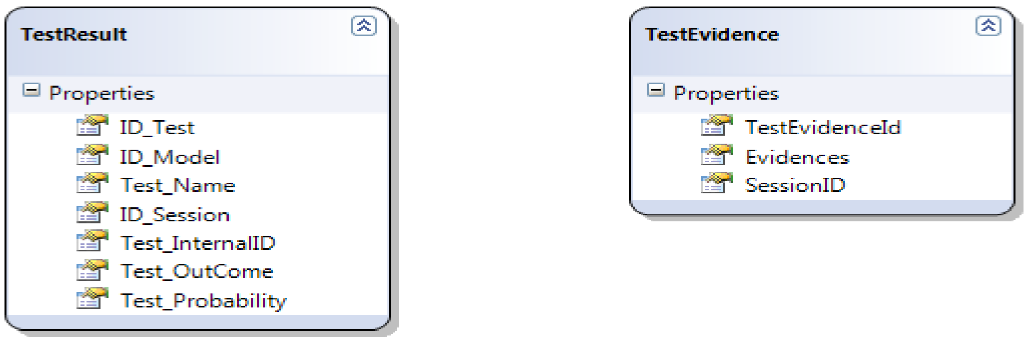


Figure 3.7: schema Database AFDDBAdministrator

TestResult: This table contains the information about all tests that are available in the Cisco testing framework. This table denes a unique ID for each entry, but this is not necessary the same as the internal ID of the AFD Tool; this ID is used as foreign key in other tables. Each test is related to a board. Each test is identified with a string name, which should be the same in the CTM source le. The UniqueID is the Cisco unique code to identify a specific test.

Filed Name	Detail
ID_Test	Unique Id to identify test in the table
ID_Model	Foreign Key that refer to the Model associated with the test
Test_Name	Store Name of the test
ID_Session	Foreign Key that refer to the Session associated with the test
Test_InternalID	It represents the internal ID of the test
Test_Outcome	It store the test result
Test_Probability	It store the probability value of a test range from 0 to 1

Table 3.6: Scheme table Dbo.TestResult

TestEvidence: This table contains the information about the recorded evidences by the Test Administrator. This is the most important information from Evidence management point of view. When test administrator would record the evidence, AFDIA – TestAdministrator will call the web services exposed by the AFDIA – WebServer to send the evidences list. Called web service will insert the data into the TestEvidence table. Then AFDOnline will query to the same table for getting the result of the recently added evidence trough the web service exposed by the AFDIA – WebServer.

Filed Name	Detail
TestEvidenceId	Unique Id to identify TestEvidence record in the table
Evidences	Represent the recorded evidence string by the test administrator
SessionID	Reference key of the Session, under which User request for the Evidence insertion. This is the same SessionID which AFDOnline created at the start of the testing procedure for a particular session

Table 3.7: Scheme table Dbo.TestEvidences

UML Diagrams & websites Structures

This chapter will cover architectural modeling and web structural aspects of the AFDIA. Before diving into the detail, let me explain what architectural modeling meant is and what structural meant for. In architectural modeling part: we would define some UML (unified Modeling Language) diagrams and in web structural part: we would define IDM: Interactive Descriptive Model of ours website pages. We can make progress in our discussion by explaining the terms UML and IDM.

UML: unified Modeling language is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of a software system. It captures decisions and understanding about systems that must be constructed. The UML captures information about the static structure and dynamic behavior of a system. The static structure defines the kind of objects important to a system and to its implementation, as well as the relationship among the objects. The dynamic behavior defines the history of objects over time and communication among objects to accomplish goals.

One way to organize the UML diagrams is by using views. A *view* is a collection of diagrams that describe a similar aspect of the project. I very often use a set of three distinct yet complementary views that are called the Static View, Dynamic View, and Functional View. Figure 11 illustrates the complementary nature of the three views and the diagrams that make up each view.

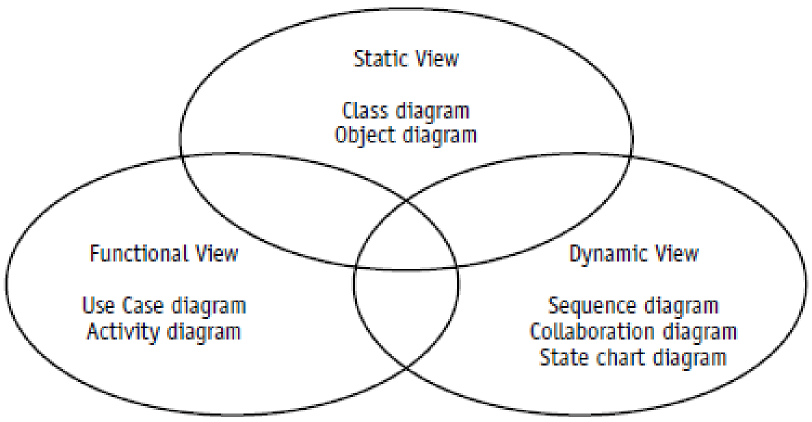


Figure: 4.1: Three complementary views or set of UML diagrams

Functional View: In the Functional View, both the Use Case diagram and the Activity diagram are included. The reason to keep them together is because they both model how the system is supposed to work. Figure 12 shows that the *Use Case diagram* defines the functions that the system must provide. The functions are expressed first as goals. But then the goals are fleshed out in a narrative to describe what each Use Case is expected to do to achieve each goal.

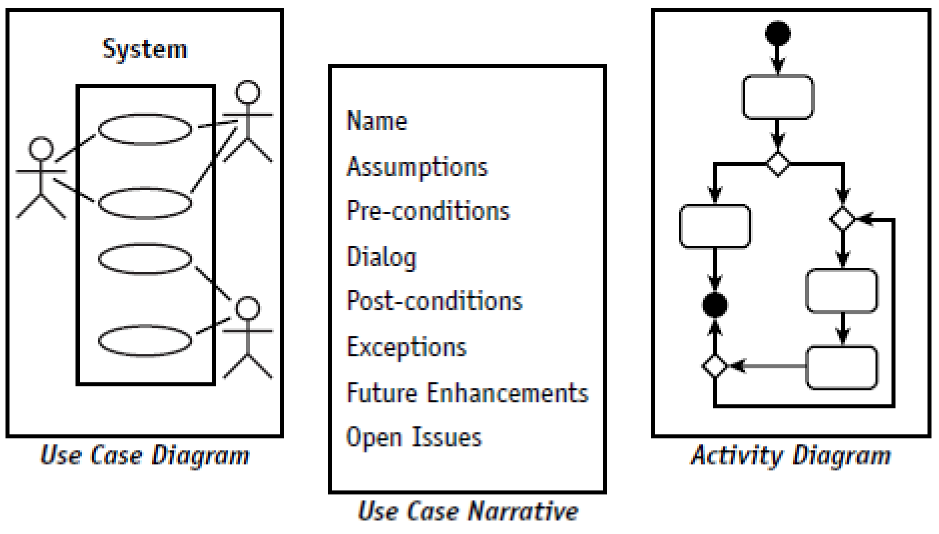


Figure: 4.2 Elements of Functional View

- The **Use Case diagram** describes the features that the users expect the system to provide.
- The **Activity diagram** describes processes including sequential tasks, conditional logic, and concurrency. This diagram is like a flowchart, but it has been enhanced for use with object modeling.

Static View: The Static View includes those diagrams that provide a snapshot of the elements of the system but don't tell you how the elements will behave. It is very much like a blueprint. Blueprints are comprehensive, but they only show what remains stationary, hence the term *Static View.*

- The **Class diagram** is the primary static diagram. It is the foundation for modeling the rules about types of objects (classes), the source for code generation, and the target for reverse engineering.
- The **Object diagram** illustrates facts in the form of objects to model examples and test data. The Object diagram can be used to test or simply to understand a class diagram.

Dynamic View: includes the diagrams that reveal how objects interact with one another in response to the environment. It includes the *Sequence and Collaboration diagrams.* Those collectively are referred to as interaction diagrams. They are specifically designed to describe how objects talk to each other. It also includes the *Statechart diagram,* which shows how and why an object changes over time in response to the environment.

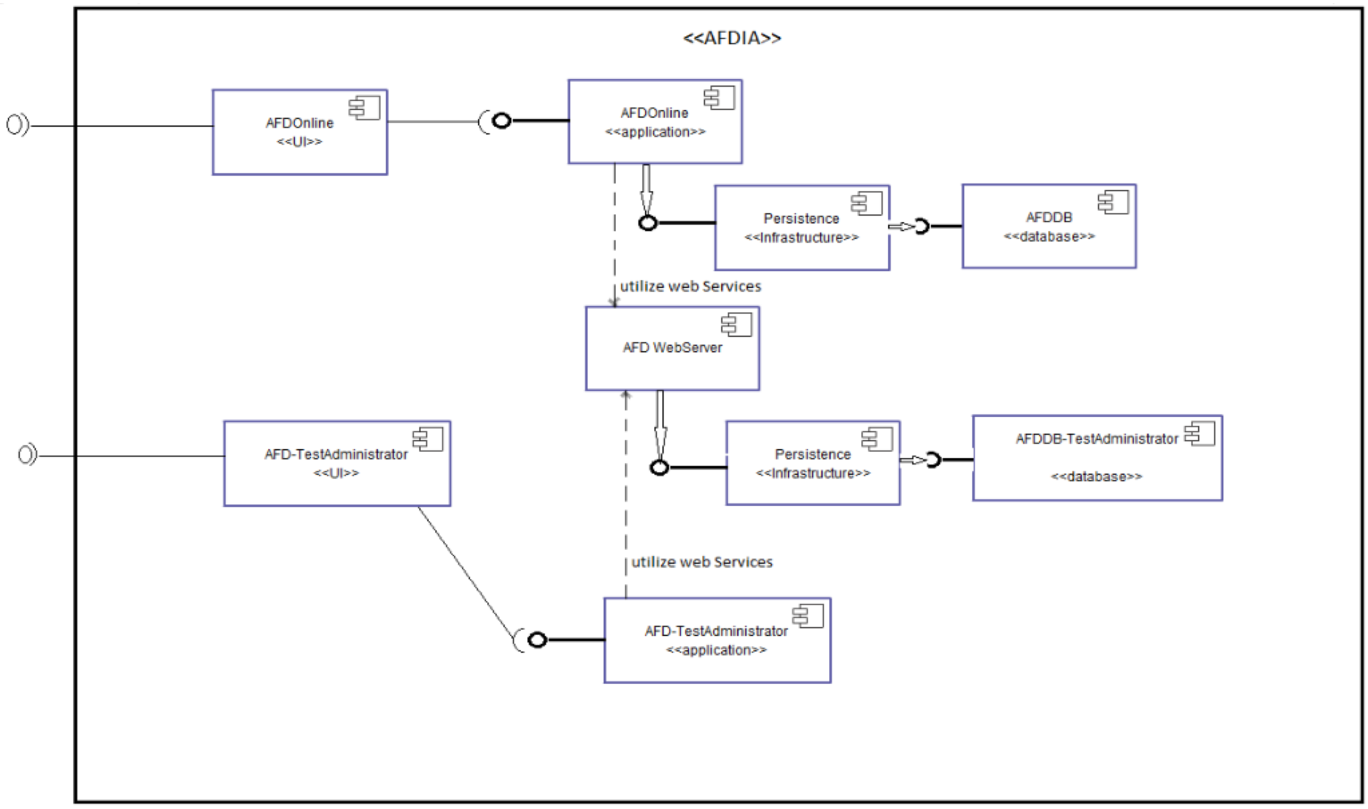
- For modeling object interactions, the Dynamic View includes the **Sequence and Collaboration diagrams**.
- The **State chart diagram** provides a look at how an object reacts to external stimuli and manages internal changes.

After a brief overview of UML, it is important to discover which diagrams among them we are going to elaborate for ours AFDIA? The Answer is Use Case Diagram (*function view*), Activity Diagram (*function view*), Sequence Diagram (*Dynamic view*) and Deployment Diagram. The selection of these diagrams is not random but based on rational judgment of keeping the most important aspects to be mentioned. In the following, there will be all about the UML; describing AFDIA through diagrams.

AFDIA – Component Diagram: Component diagrams are different in terms of nature and behavior. Component diagrams are used to model physical aspects of a system. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams are used to represent the whole.

So the purpose of the component diagram can be summarized as:

- Visualize the components of a system.
- Construct executable by using forward and reverse engineering.
- Describe the organization and relationships of the components.



<<AFDIA>>
AFDOnline <<UI>>
AFDOnline <<application>>
Persistence <<Infrastructure>>
AFDDB <<database>>
utilize web Services
AFD WebServer
AFD-TestAdministrator <<UI>>
Persistence <<Infrastructure>>
AFDDB-TestAdministrator <<database>>
utilize web Services
AFD-TestAdministrator <<application>>

Figure 4.3: Component Diagram AFDIA

Above in picture you can see the component diagram, now I want to put little comments on its construction. We will start from the left of AFDIA component wise. There are two UI components AFDOnline<<UI>> and AFD-TestAdministrator<<UI>>, they both are dependent on two application components (components that exploits the core functionality) respectively AFDOnline<<application>> and AFD-TestAdministrator<<application>>.

AFDOnline<<application>> component further depend upon the Persistence component (a component interact with AFDDB<<database>> component). And second dependency of AFDOnline<<application>> component is upon AFD WebServer<<WebServices>> component.

AFD-TestAdministrator<<application>> component further depend upon AFD WebServer<<WebServices>> component to utilize its exposed services for complete its some tasks.

AFD WebServer<<WebServices>> component further depend upon the Persistence component (a component interact with AFDDB-TestAdministrator<<database>> component).

These all components interact with each other to complete the overall objective of the system. And you see a boundary line around the components that depicts the whole components as a single components that can be called as AFDIA<<component>>. And the two interfaces that are available through the boundary are basically access points to these applications and obviously you can connect via Internet browser to these access points.

Use Case Diagram:
Use Case Diagram describes the features that User expects from System to provide. Its major concern is to describe ***what*** the system should do ***"not how to do"***. There are two levels of use case Diagram: ***system level or Abstract level*** use case diagram and ***Detail Level*** use case Diagram.

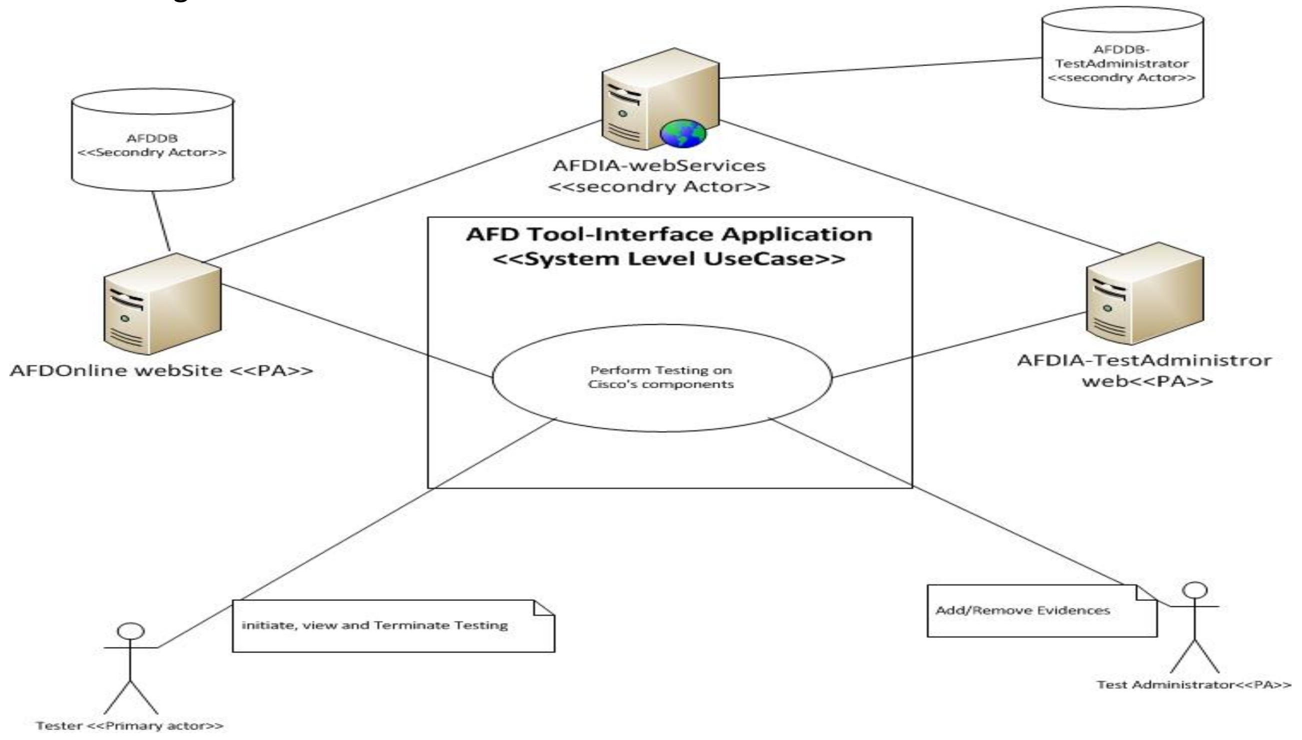

AFDB-TestAdministrator <<secondry Actor>>
AFDDB <<Secondry Actor>>
AFDIA-webServices <<secondry Actor>>
AFD Tool-Interface Application <<System Level UseCase>>
AFDOnline webSite <<PA>>
Perform Testing on Cisco's components
AFDIA-TestAdministror web<<PA>>
initiate, view and Terminate Testing
Add/Remove Evidences
Tester <<Primary actor>>
Test Administrator<<PA>>

Figure 4.4: System Level Use Case Diagram of AFDIA

<u>**System Level Use Case Diagram:**</u> defines abstract or high level view of overall system. In the above figure 13: depicts the AFDIA system level use case diagram. As you can see the name of the use case is "Perform testing on Cisco's component". This name itself describes the whole story that would occur in the software application. Another term which is also very essential to understand is ***Actor;*** actor is a user or external system with which a system being modeled interacts. There are two types of actors **Primary actors** and **Secondary Actors.** Primary actors are those actors who directly interact with the system and Secondary actors are those actors who interact with the System through Primary actors. Let's have a look on the Primary and secondary actors in AFDIA.

Primary Actors:
Tester: is a human actor, who would interact with the system by initiating, viewing and terminating the testing process.
Test – Administrator: is a human actor, who would interact with the system by Adding/Removing Evidences.

AFDOnline website: is a system actor that is responsible to entertain all the requests generated by the *Tester*. To accomplish the generated request by the *Tester*, *AFDOnline Website* would may utilize or contact with the *AFDIA – WebServices.*
AFDIA – TestAdministrator website: is a system actor that is responsible to entertain all the requests generated by the *Tester- Administrator*. To accomplish the generated request by the *Tester- Administrator*, *AFDIA – TestAdministrator website* would may utilize or contact with the *AFDIA – WebServices.*

Secondary Actors:
AFDIA – WebServices: is a secondary system actor that is responsible to provide the services to two primary actors *AFDOnline Website* and *AFDIA – TestAdministrator Website* to accomplish their tasks.
In the figure above there are two others mentioned secondary actors namely *AFDDB* and *AFDDB – TestAdministrator*.

As the Abstract level UC diagram shows the very high level understanding of the system, it can give us a silent idea of the system, but to know exactly in detail the role of each actor in the system and more eventually the sequence of the story, we must have to dive into the detail level of the use case diagram. Next, you can see a detail level use case diagram of AFDIA in **figure 14**. Just an important thing to note: in detail level use case diagram all the actors including both primary and secondary must have to be equal in numbers and same as role compared with Abstract level UC diagram. It is wrong to add a new actor or try to remove an existing actor at this level which appeared in Abstract level use case diagram.

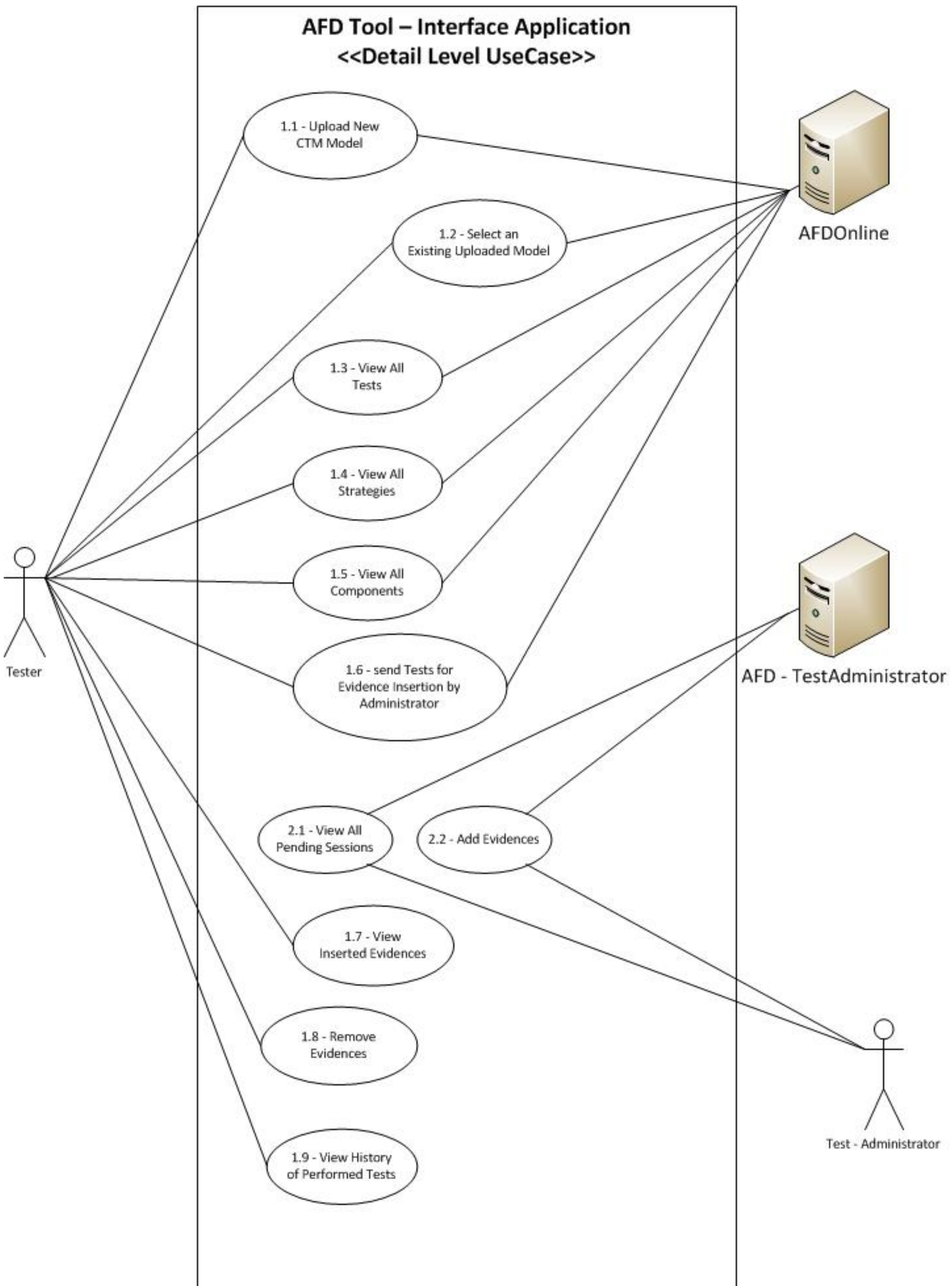

AFD Tool – Interface Application
<<Detail Level UseCase>>
1.1 - Upload New CTM Model
1.2 - Select an Existing Uploaded Model
1.3 - View All Tests
1.4 - View All Strategies
1.5 - View All Components
1.6 - send Tests for Evidence Insertion by Administrator
2.1 - View All Pending Sessions
2.2 - Add Evidences
1.7 - View Inserted Evidences
1.8 - Remove Evidences
1.9 - View History of Performed Tests
AFDOnline
AFD - TestAdministrator
Tester
Test - Administrator

Figure 4.5: Detail Level Use Case Diagram of AFDIA

<u>Detail Level Use Case Diagram:</u> in the depicted figure above you can have a look on the detail level use case diagram of AFDIA. By exploring keenly you can figure out the following points. I) the number of actors are same both in number and roles II) But the bigger use case which we saw in case of system level UC diagram has been broken down into many small autonomous and self-descriptive use cases, and now they are presenting almost clear and understandable look of interaction and sequences of occurrences in the whole story of the system.
Next, we are moving to describe our ultimate and fruitful part of the use cases descriptions. That is description of each individual use case in the detail level use case diagram. For example we will pick up first use case from the diagram, like 1.1 "upload a CTM Model" and then by sequence description of each UC.

The Structure of the description will contain the following components/parts:

- **Use Case name:** the name of the use case.
- **Brief Description:** some text describes briefly the overall function of UC.
- **Level:** like Enterprise level UC or Corporate level UC or Unit level UC.
- **Primary Actor:** Actors those directly communicate with the UC.
- **Stakeholders and interests:** All bodies who will get benefited with the success.
- **Precondition:** The steps must be performed before calling the use case functionally.
- **Minimal Guarantee:** describe the minimal affect in the system even in case of failure or incompletion.
- **Success Guarantee:** describe the overall effective occurrences in the system in case of successful completion of the use case.
- **Main Success Scenario:** describe the complete success scenario steps.
- **Exceptions:** describe the exceptions that make cause the failure or incompletion of the UC.

Provided above is a template for explaining each use case by selecting from the Detail level use case diagram. One thing is important to remember that we will use the same name of use cases in chapter 5, which is about implementation detail of AFDIA. In our AFDIA DL-UC Diagram we have total 11 use cases: from ***1.1 to 1.9*** and **from 2.1 to 2.2.**

- **<u>Use Case name</u>:** *[1.1] Upload New CTM Model*
- **Brief Description:** Tester will upload a new CTM Model (a text base file contains a specified formatted meaningful data) by his/her Testing machine. If process goes well and successful then Tester will receive a success message from the Server and will be able to go on in the testing process. In case of any error/failure, Server will generate an exception and let the Tester either to try another time to upload Model or quit from the testing process.
- **Level:** Application Level UC.

- **Primary Actor:** There are two primary actors interacting with the UC [1.1]
    I. **Tester:** Tester would try to upload New CTM Model on the Server.
    II. **AFDOnline website:** is a system that let the Tester to upload file on the server. It is a traditional web application, accessible through any standard internet browser.
- **Stakeholders and interests:** Tester is one who can further progress in the testing. Cisco, as a company will get benefited by the completion of the UC [1.1].
- **Precondition:**
    I. Tester should have a computer.
    II. Enabled with internet/intranet.
    III. Must have installed Standard web Browser.
    IV. Login to AFDIA.
    V. Navigate to Upload New Model Web page.
- **Minimal Guarantee:** if Tester will fail to upload new model, he has option to retry or can select an existing model resides on the Server.
- **Success Guarantee:** Upon successful uploading of Model, Tester would be able to proceed further in the testing process.
- **Main Success Scenario:**
    I. Tester has uploaded the CTM Model on the server.
    II. Upon Successful completion of uploading, Server will allow the Tester to move on in the Testing process.
- **Exceptions:**
    I. Internet/Intranet is unavailable.
    II. Machine is crashed or faulty.
    III. Tester does not have a CTM Model locally for uploading.

- <u>**Use Case name**</u>: *[1.2] Selected an existing uploaded Model*
- **Brief Description:** Tester have the option to select an already existing uploaded model on the server. By selecting a model, Tester would be able to proceed further in the testing process. If there is not any available model on the server, then only one option keep left that is Upload a new CTM model on the server for proceeding in the testing process.
- **Level:** Application Level UC.
- **Primary Actor:** There are two primary actors interacting with the UC [1.2]
    I. **Tester:** Tester would try to select an existing model on the server.
    II. **AFDOnline website:** is a system that let the Tester to show the entire available Model on the server. It is a traditional web application, accessible through any standard internet browser.
- **Stakeholders and interests:** Tester is one who can further progress in the testing. Cisco, as a company will get benefited by the completion of the UC [1.2].

- **Precondition:**
  I. Tester should have a computer.
  II. Enabled with internet/intranet.
  III. Must have installed Standard web Browser.
  IV. Login to AFDIA.
  V. Navigate to Select an Existing Model Web page.
- **Minimal Guarantee:** if Tester will fail to find any existing uploaded model, he has option to upload new CTM Model and can further proceed.
- **Success Guarantee:** Upon successful selection of Model, Tester would be able to proceed further in the testing process.
- **Main Success Scenario:**
  I. Tester has selected successfully an existing CTM Model on the server.
  II. Upon Successful selection process, Server will allow the Tester to move on in the Testing process.
- **Exceptions:**
  I. Internet/Intranet is unavailable.
  II. Machine is crashed or faulty.
  III. Tester does not have a CTM Model locally for uploading and there is no model even on the server.

- **<u>Use Case name</u>:** *[1.3] View All Tests*
- **Brief Description:** Tester would be able to see all the Tests results.
- **Level:** Application Level UC.
- **Primary Actor:** There are two primary actors interacting with the UC [1.3]
  I. **Tester:** Tester would explore the Test Results.
  II. **AFDOnline website:** is a system that let the Tester to show the entire tests results. It is a traditional web application, accessible through any standard internet browser.
- **Stakeholders and interests:** Tester is one who can further progress and navigate on different other functionalities in the testing process. Cisco, as a company will get benefited by the completion of the UC [1.3].
- **Precondition:**
  I. Already selected an existing model or already uploaded a new CTM Model.
  II. Navigate to the View All Test web page.
- **Minimal Guarantee:** Tester at least will explore the Test results as this use case is just about to explore the result so if preconditions went well then the minimal guarantee is actual goal of this use case.
- **Success Guarantee:** Upon successful completion of the use case, Tester would be able to explore the tests and can proceed further in the testing process.
- **Main Success Scenario:**
  I. Tester has selected an existing CTM Model on the server or uploaded a new CTM Model.

II. Upon Successful selection process or upload process.
III. User would be able to explore the Tests results.

- **Exceptions:**
  I. Internet/Intranet is unavailable.
  II. Machine is crashed or faulty.
  III. Tester does not have a CTM Model locally for uploading and there is no model even on the server.

- <u>**Use Case name**</u>**:** *[1.4] View All Strategies*
- **Brief Description:** Tester would be able to see all the Available Test Strategies.
- **Level:** Application Level UC.
- **Primary Actor:** There are two primary actors interacting with the UC [1.4]
  I. **Tester:** Tester would explore the Available Test Strategies.
  II. **AFDOnline website:** is a system that let the Tester to show the entire Strategies. It is a traditional web application, accessible through any standard internet browser.
- **Stakeholders and interests:** Tester is one who can further progress and navigate on different other functionalities in the testing process. Cisco, as a company will get benefited by the completion of the UC [1.4].
- **Precondition:**
  I. Already selected an existing model or already uploaded a new CTM Model.
  II. Navigate to the View All Test Strategies web page.
- **Minimal Guarantee:** Tester at least will explore the Test Strategies as this use case is just about to explore the Test strategies so if preconditions went well then the minimal guarantee is actual goal of this use case.
- **Success Guarantee:** Upon successful completion of the use case, Tester would be able to explore the all available test Strategies and can proceed further in the testing process.
- **Main Success Scenario:**
  I. Tester has selected an existing CTM Model on the server or uploaded a new CTM Model.
  II. Upon Successful selection process or upload process.
  III. User would be able to explore the all available test Strategies.
- **Exceptions:**
  I. Internet/Intranet is unavailable.
  II. Machine is crashed or faulty.
  III. Tester does not have a CTM Model locally for uploading and there is no model even on the server.

- **<u>Use Case name</u>:** *[1.5] View All Components*
- **Brief Description:** Tester would be able to see all components associated with board.
- **Level:** Application Level UC.
- **Primary Actor:** There are two primary actors interacting with the UC [1.5]
  - I. **Tester:** Tester would explore all components.
  - II. **AFDOnline website:** is a system that let the Tester to show the entire components list. It is a traditional web application, accessible through any standard internet browser.
- **Stakeholders and interests:** Tester is one who can further progress and navigate on different other functionalities in the testing process. Cisco, as a company will get benefited by the completion of the UC [1.5].
- **Precondition:**
  - I. Already selected an existing model or already uploaded a new CTM Model.
  - II. Navigate to the View All component web page.
- **Minimal Guarantee:** Tester at least will explore all components as this use case is just about to explore the result so if preconditions went well then the minimal guarantee is actual goal of this use case.
- **Success Guarantee:** Upon successful completion of the use case, Tester would be able to explore all components and can proceed further in the testing process.
- **Main Success Scenario:**
  - I. Tester has selected an existing CTM Model on the server or uploaded a new CTM Model.
  - II. Upon Successful selection process or upload process.
  - III. User would be able to explore all components.
- **Exceptions:**
  - I. Internet/Intranet is unavailable.
  - II. Machine is crashed or faulty.
  - III. Tester does not have a CTM Model locally for uploading and there is no model even on the server.

- **<u>Use Case name</u>:** *[1.6] send Tests for Evidence Insertion by Administrator*
- **Brief Description:** Tester will send tests to the Administrator for the Evidences insertion as the test administrator is the actor who will insert evidence.
- **Level:** Application Level UC.
- **Primary Actor:** There are two primary actors interacting with the UC [1.6]
  - I. **Tester:** Tester will send tests for the evidence insertion.
  - II. **AFDOnline website:** is a system that let the Tester to send the tests for evidence insertion with the help of a web page. It is a traditional web application, accessible through any standard internet browser.
- **Stakeholders and interests:** Tester is one who can further progress and navigate on different other functionalities in the testing process. Cisco, as a company will get benefited by the completion of the UC [1.6].
- **Precondition:**

I. Already selected an existing model or already uploaded a new CTM Model.
II. Navigate to the Add Evidence web page.
III. Tester will send tests to whom, he want to see the inserted evidences by the test administrator.

- **Success Guarantee:** Upon successful sending of the tests for evidence insertion, Tester would be able to see the inserted evidence by the administrator.
- **Main Success Scenario:**

IV. Tester has selected an existing CTM Model on the server or uploaded a new CTM Model.
V. Upon Successful selection process or upload process.
VI. Tester would select tests for evidence insertion and send them to administrator.

- **Exceptions:**

IV. Internet/Intranet is unavailable.
V. Machine is crashed or faulty.
VI. Tester does not have a CTM Model locally for uploading and there is no model even on the server.

- <u>**Use Case name**</u>**:** *[1.7] View Inserted Evidences*
- **Brief Description:** Tester would explore the test results, recorded by the test administrator. For tester these results are read-only.
- **Level:** Application Level UC.
- **Primary Actor:** There are two primary actors interacting with the UC [1.7]

III. **Tester:** Tester will explore the recorded tests results.
IV. **AFDOnline website:** is a system that let the Tester to view the test results. It is a traditional web application, accessible through any standard internet browser.

- **Stakeholders and interests:** Tester is one who can further progress and navigate on different other functionalities in the testing process. Cisco, as a company will get benefited by the completion of the UC [1.7].
- **Precondition:**

III. Already selected an existing model or already uploaded a new CTM Model.
IV. Navigate to the Add Evidence web page.
IV. Tester will explore the results inserted by the test administrator.

- **Success Guarantee:** Upon successful sending of the tests exploration, Tester would be able to keep going to perform other successive operations.

- **Main Success Scenario:**

VII. Tester has selected an existing CTM Model on the server or uploaded a new CTM Model.
VIII. Upon Successful selection process or upload process.

IX. Tester would explore the test results returned by the test administrator after processing.

- **Exceptions:**

VII. Internet/Intranet is unavailable.
VIII. Machine is crashed or faulty.
IX. Tester does not have a CTM Model locally for uploading and there is no model even on the server.

- **<u>Use Case name</u>:** *[1.8] Remove/Clean Evidences*
- **Brief Description:** Tester would remove or clean the inserted evidences, which was recorded by the test administrator.
- **Level:** Application Level UC.
- **Primary Actor:** There are two primary actors interacting with the UC [1.8]

V. **Tester:** Tester will Remove or clean the evidences.
VI. **AFDOnline website:** is a system that let the Tester to Remove/clean the recorded evidences. It is a traditional web application, accessible through any standard internet browser.

- **Stakeholders and interests:** Tester is one who can further progress and navigate on different other functionalities in the testing process. Cisco, as a company will get benefited by the completion of the UC [1.8].
- **Precondition:**

V. Already selected an existing model or already uploaded a new CTM Model.
VI. Navigate to the Remove Evidence web page.
V. Tester will Remove/clean results inserted by the test administrator.

- **Success Guarantee:** Upon successful Removal of tests results, Tester would be able to keep going to perform other successive operations.

- **Main Success Scenario:**

X. Tester has selected an existing CTM Model on the server or uploaded a new CTM Model.
XI. Upon Successful selection process or upload process.
XII. Tester would remove the test results inserted by the test administrator.

- **Exceptions:**

X. Internet/Intranet is unavailable.
XI. Machine is crashed or faulty.
XII. Tester does not have a CTM Model locally for uploading and there is no model even on the server.

- **Use Case name:** *[1.9] View History of Performed Tests*
- **Brief Description:** Tester would traverse the performed tests, all the information should be available in the database for the history purposes and Tester can view it, as history of performed tests.
- **Level:** Application Level UC.
- **Primary Actor:** There are two primary actors interacting with the UC [1.9]
  - VII. **Tester:** Tester who can watch the history of performed tests.
  - VIII. **AFDOnline website:** is a system that let the Tester to traverse the performed tests history. It is a traditional web application, accessible through any standard internet browser.
- **Stakeholders and interests:** Tester is one who can further progress and navigate on different other functionalities in the testing process. Cisco, as a company will get benefited by the completion of the UC [1.9].
- **Precondition:**
  - VII. Already selected an existing model or already uploaded a new CTM Model.
  - VIII. Already performed all the necessary operation that results could be kept in the database for the further pointing from history management point of view.
- **Success Guarantee:** Upon successful viewing of the history, Tester would be able to keep going to perform other successive operations.

- **Main Success Scenario:**
  - XIII. Tester will choose a session ID from a provided web page where all the session Id will be listed in a grid (all the sessions that has been performed in past)
  - XIV. Tester would choose one session id and continue to view all of its stored results

- **Exceptions:**
  - XIII. Internet/Intranet is unavailable.
  - XIV. Machine is crashed or faulty.
  - XV. There is nothing store in the history of testing database, so you are the first one to perform operations.

- **Use Case name:** *[2.1] View All Pending Sessions*
- **Brief Description:** Test Administrator will see all the pending requests sent by the Tester for the evidence insertion. Test Administrator would choose any session id and proceed for the evidence recorded. After selection a session id, system will transfer to administrator to a new web page, where administrator can insert the evidences.

- **Level:** Application Level UC.
- **Primary Actor:** There are two primary actors interacting with the UC [2.1]
    IX. **Test - Administrator:** Test Administrator will see and select any sent Session id for the evidence insertion.
    X. **AFD - TestAdministrator:** is a system that let the Administrator to add evidences. It is a traditional web application, accessible through any standard internet browser.
- **Stakeholders and interests:** Administrator is one who can further progress and navigate on different other functionalities in the testing process. Cisco, as a company will get benefited by the completion of the UC [2.1].
- **Precondition:**
    IX. Tester will send session id for the evidence insertion by the administrator
- **Success Guarantee:** Upon selecting a session Id, Administrator will add/recorded the test results.

- **Main Success Scenario:**
    XV. Administrator will choose a session ID from a provided web page where all the session Id will be listed in a grid (all the sessions that has been performed in past)
    XVI. Administrator will be moved to evidences insertion page.

- **Exceptions:**
    XVI. Internet/Intranet is unavailable.
    XVII. Machine is crashed or faulty.

- <u>**Use Case name**</u>**:** *[2.2] Add Evidences*
- **Brief Description:** Test Administrator will record the test results.
- **Level:** Application Level UC.
- **Primary Actor:** There are two primary actors interacting with the UC [2.1]
    XI. **Test - Administrator:** Test Administrator would add results by select any sent Session id for the evidence insertion.
    XII. **AFD - TestAdministrator:** is a system that let the Administrator to add evidences. It is a traditional web application, accessible through any standard internet browser.
- **Stakeholders and interests:** Administrator is one who can further progress and navigate on different other functionalities in the testing process. Cisco, as a company will get benefited by the completion of the UC [2.1].
- **Precondition:**
    X. Administrator will choose any sent session id for the evidence insertion.

- **Success Guarantee:** Upon selecting a session Id, Administrator will add/recorded the test results.

- **Main Success Scenario:**

XVII. Administrator will choose a session ID from a provided web page where all the session Id will be listed in a grid (all the sessions that has been performed in past)

XVIII. Administrator will add the concrete results by the exploring the hardware machine results.

- **Exceptions:**

XVIII. Internet/Intranet is unavailable.

XIX. Machine is crashed or faulty.

AFDIA Implementation Detail

Overview: the purpose of this chapter is to put insight into the implementation detail of each web page of all three applications AFDOnline, AFDIA WebServer and TestAdministrator. The detail will include the following information about all the pages in all three applications: **Page Name**, **Page Main Functionality**, **Possible Movements to/from other pages**, **Interaction with the AFD Database (Table Name)**, **Interaction with the Dll (Wrapper Method Name)** and **Detail Description of the Page.**

AFDOnline

Page Name: Index.aspx
Page Main Functionality:
This is the first page while running the AFD tool, the content of this page includes two Links button, which allows user to move on two successive pages.
Possible Movements to/from other pages: AvailableBNEModels.aspx, UploadModel.aspx
Interaction with the AFD Database (Table Name(s)): None
Interaction with the Dll (Wrapper Method Name): None
Screen Shot:

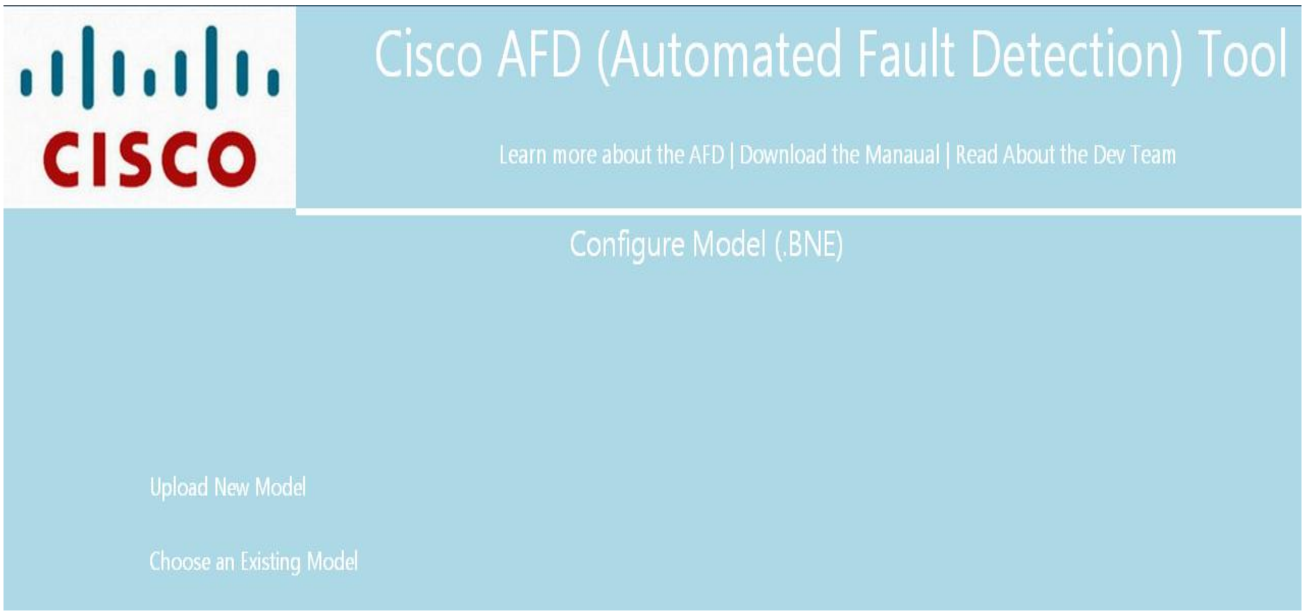

CISCO
Cisco AFD (Automated Fault Detection) Tool
Learn more about the AFD | Download the Manaual | Read About the Dev Team
Configure Model (.BNE)
Upload New Model
Choose an Existing Model

Figure 5.1: webpage Index.aspx

Detail Description of the Page: this page as shown above have two links "Update New Model" and "Choose an Existing Model", these links will allows the user to choose a CTM Model. User will be transfer to an appropriate page as he will choose an option for providing Model to the AFD Tool.

By considering the (FSM Model of AFD Tool: Figure 2.2), if we explore the Initialization state of the Tool, we can say that we are performing second sub-state (File Uploading) with this page. Let me be clear that this page will not complete the File uploading completely as we will move on to two other pages optionally to complete the CTM Model uploading.

Interaction with the DB while navigating on this Page is nothing; this is more formally just a page to choose a way to proceed with the selecting a CTM Model.

Interaction with the DLL wrapper methods is also nothing, because we can call AFD_BBN_Library_Init () method after the successful uploading or selection of a Model, as on this page we are just dealing to choose a way to refer a model.

Page Name: UploadModel.aspx
Page Main Functionality:

This Page will be used to upload a new Model into the AFD Server from the local system.
After the successful uploading of the Model, we will call the AFD_BBL_Library_Init () to complete the Initialization state of the AFD Tool.

Possible Movements to/from other pages: AvailableBNEModels.aspx, Test.aspx
Interaction with the AFD Database (Table Name(s)): dbo.Session------dbo.Test open point
Interaction with the Dll (Wrapper Method Name): AFD_BBL_Library_Init ()
Screen Shot:

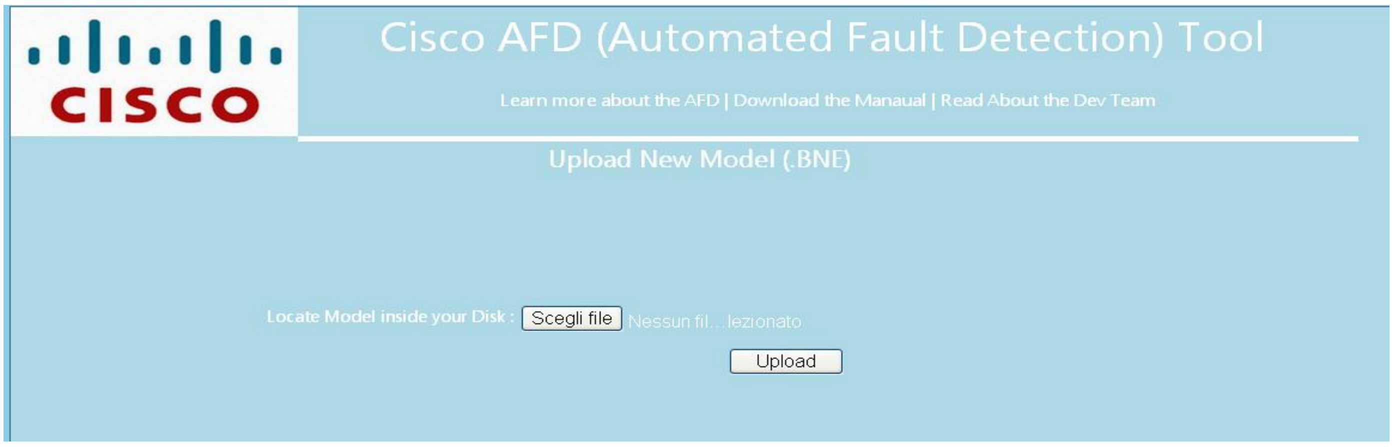

CISCO
Cisco AFD (Automated Fault Detection) Tool
Learn more about the AFD | Download the Manaual | Read About the Dev Team
Upload New Model (.BNE)
Locate Model inside your Disk :
Scegli file
Nessun fil... lezionato
Upload

Figure 5.2: webpage UploadModel.aspx

Detail Description of the Page: this page will be appear after the first traverse from the index.aspx page when user have selected an option to upload a new CTM model instead of an existing one onto the AFD Server's available Models list, button with the text "Scegli file" will be shown up with a File Dialog box to let the user to locate a .bne file from the user's system.

After the file selection, Upload button's action will serialized the selected Model (a file with .bne extension) and upload it onto the AFD Server.

By uploading the file, the Interface Application immediately opens AFD DB Connection, with the completion of activity. Let's recall the Initialization State of the FSM Model of the AFD Application. (Ref: FSM Model of AFD Tool – AFD Tool GUI Specs – page 6 Fig: 1.2)

At this state of the Interface Application , we have already connect to the AFD Tool, we done with CTM Model selection and Uploading and DB connection is open and alive for the further transactions. These were three activities to complete the initialization state, and we are done with these by Leaving that Page (UploadModel.aspx).

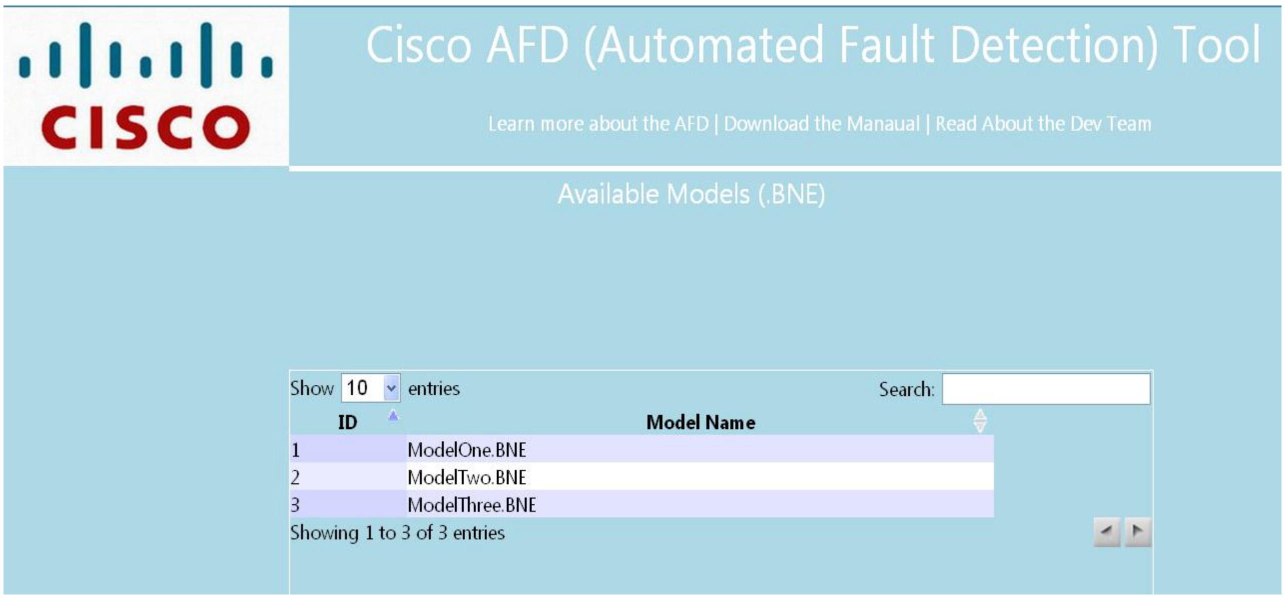
In Addition, AFDIA will assign a "Unique session Id" to track history of testing of each specific user. AFDIA will insert a new record in the session table of the AFDDB.

Page Name: AvailableBNEModels.aspx
Page Main Functionality:

This Page will be used to select an existing Model already reside onto the AFD Server. After the successful selection of the Model, we will call the AFD_BBL_Library_Init () to complete the Initialization state of the AFD Tool.

Possible Movements to/from other pages: UploadModel.aspx, Test.aspx
Interaction with the AFD Database (Table Name(s)): dbo.Model, dbo.session
Interaction with the Dll (Wrapper Method Name): AFD_BBL_Library_Init ()

Screen Shot:


Cisco AFD (Automated Fault Detection) Tool

Learn more about the AFD | Download the Manaual | Read About the Dev Team

Available Models (.BNE)

Show 10 entries Search:

ID	Model Name
1	ModelOne.BNE
2	ModelTwo.BNE
3	ModelThree.BNE

Showing 1 to 3 of 3 entries

Figure 5.3: webpage AvailableBNEModels.aspx

Detail Description of the Page: this page will be appearing after the first traverse from the index.aspx page when user have selected an option to choose an existing CTM model.

This Page contains a Grid which shows all available .bne Models on the AFD Server, AFDIA will query to AFDDB to bring all available .bne models on the AFD Server and then bind the result to the Grid control. user would select any available model by clicking onto a Row [the decision that, whether this page should contain a Button, that let the tool to proceed after the Row selection OR just By the Clicking upon the Row application should Proceed] in any way, Application will get the information of the selected Model and call the AFD_BBN_Library_Init () wrapper method. Before calling the wrapper method, AFDIA will query on the Model table, get the CTM source file from the AFDDB and Parser component of AFDIA will be invoked to write data in the input text file.

Let's recall the Initialization State of the FSM Model of the AFD Application. (Ref: FSM Model of AFD Tool – AFD Tool GUI Specs – page 6 Fig: 1.2)

At this state of the Interface Application, we have already connected to the AFD Tool, we done with CTM Model selection, Uploading and DB connection is open and alive for the further transactions. These were three activities to complete the initialization state, and we are done with these by Leaving that Page (AvailableBNEModels.aspx).In Addition, AFDIA will assign a "Unique session Id" to track history of testing of each specific user. AFDIA will insert a new record in the session table of the AFDDB.

Page Name: Test.aspx

Page Main Functionality:

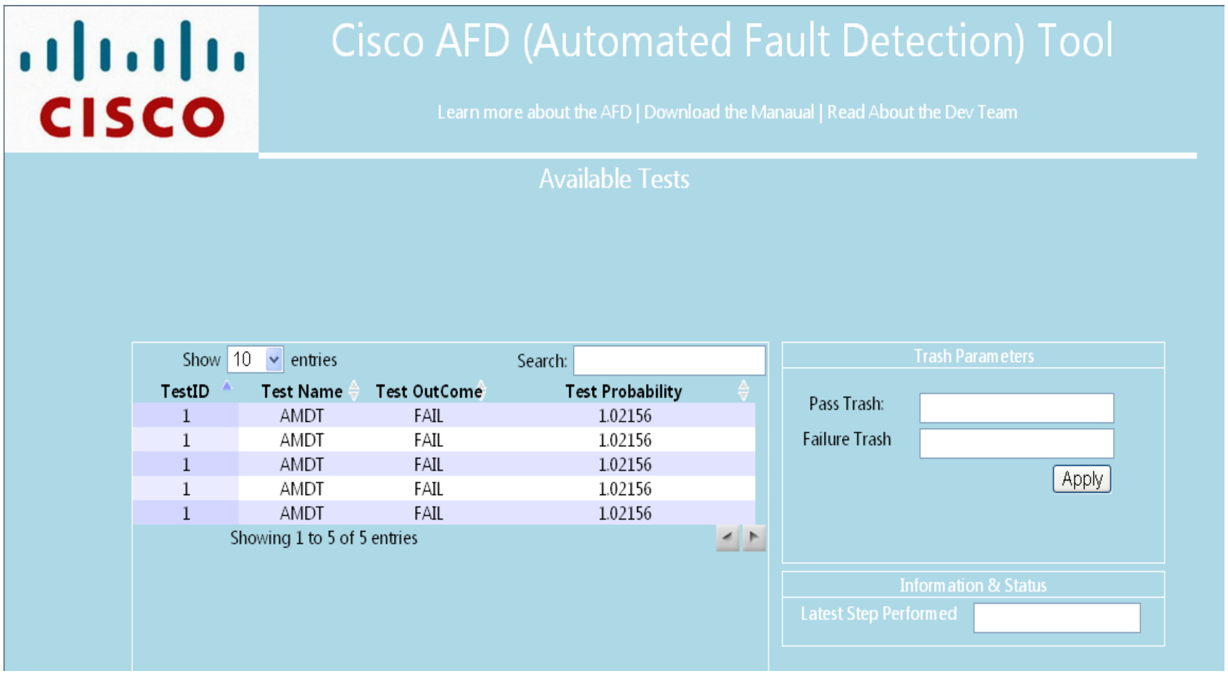
This Page will be used to make a query to AFD Server for fetching list of Tests. After receiving the list of tests, AFDIA will present all this information in a Grid, where user can explore these tests by applying sorting etc.

Possible Movements to/from other pages: Component.aspx, Strategy.aspx, History.aspx

Interaction with the AFD Database (Table Name(s)): None?

Interaction with the Dll (Wrapper Method Name): AFD_BBN_Library_getTests_All ()

Screen Shot:



TestID	Test Name	Test OutCome	Test Probability
1	AMDT	FAIL	1.02156
1	AMDT	FAIL	1.02156
1	AMDT	FAIL	1.02156
1	AMDT	FAIL	1.02156
1	AMDT	FAIL	1.02156

Showing 1 to 5 of 5 entries

Figure 5.4: webpage Test.aspx

Detail Description of the Page: AFDIA will reach this page, after completing the Initialization state. On the Page Load method, AFDIA will call a Wrapper Method AFD_BBN_Library_getTests_All ().

The output of this method will result an AFD_BBN_Library_getTests_All_output.txt text file which contain the list of strings: test name, test status and test probability. AFDIA will parse the output file and present it into the Grid, showing above in a picture.

There is group box name Trash parameters: contain two textboxes, Pass trash and failure trash; these two values will be used to set the trash value to differentiate among test. The entered values will be used to sort the test probability. Therefore entered value must be less than 1. Another group box named Information and status, contain a read-only text box, which shows the latest step performed information. That is more concern about the history of the testing session.

Page Name: Component.aspx

Page Main Functionality:

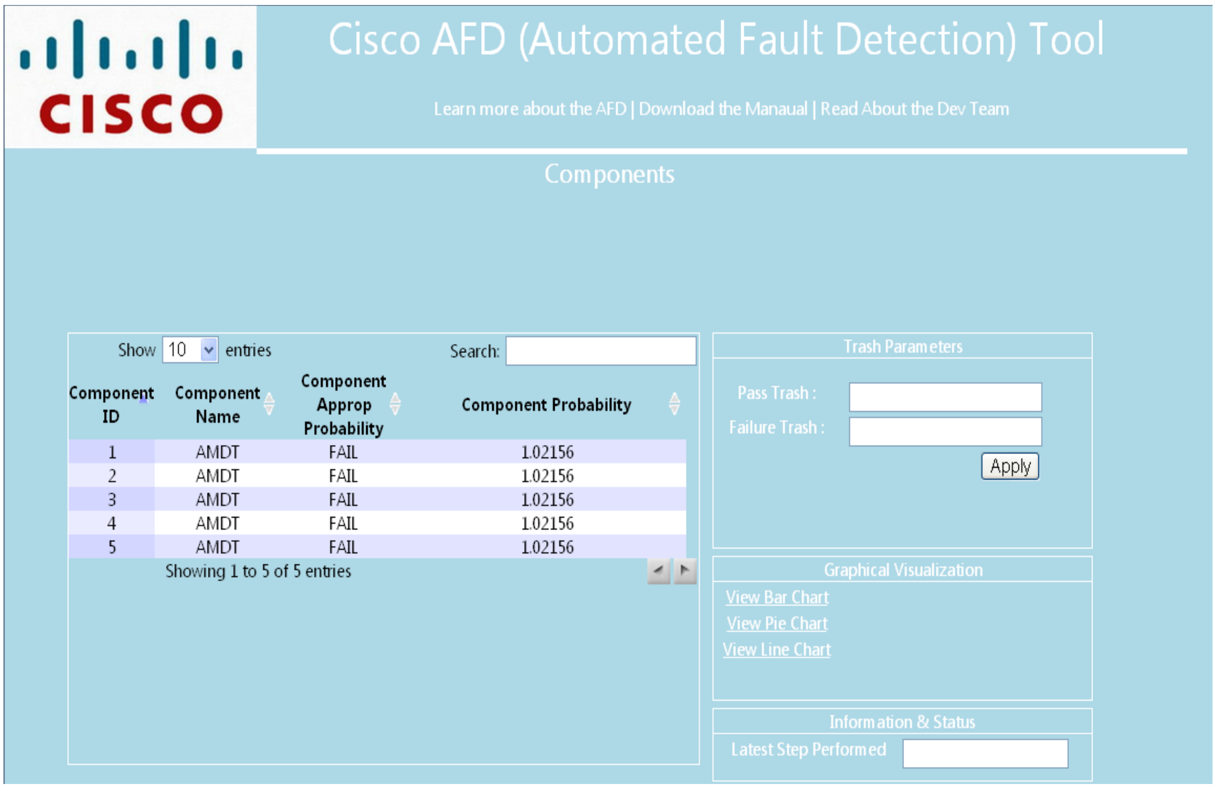
This Page will be used to make a query to AFD Server for fetching information of component Nodes. After receiving the list of component nodes, AFDIA will present all this information in a Grid, where user can explore these components by applying sorting in multiple ways etc.

Possible Movements to/from other pages: Test.aspx, Strategy.aspx, History.aspx

Interaction with the AFD Database (Table Name(s)): None

Interaction with the Dll (Wrapper Method Name): AFD_BBN_Library_Components_All ()

Screen shot:



Cisco AFD (Automated Fault Detection) Tool

Learn more about the AFD | Download the Manaual | Read About the Dev Team

Components

Show 10 entries Search:

Component ID	Component Name	Component Approp Probability	Component Probability
1	AMDT	FAIL	1.02156
2	AMDT	FAIL	1.02156
3	AMDT	FAIL	1.02156
4	AMDT	FAIL	1.02156
5	AMDT	FAIL	1.02156

Showing 1 to 5 of 5 entries

Trash Parameters

Pass Trash :

Failure Trash :

Apply

Graphical Visualization

View Bar Chart

View Pie Chart

View Line Chart

Information & Status

Latest Step Performed

Figure 5.5: webpage Component.aspx

Detail Description of the Page: AFDIA will reach this page, after completing the Initialization state. On the Page Load method, AFDIA will call a Wrapper Method AFD_BBN_Library_Components_All ().

The output of this method will result an AFD_BBN_Library_Components_All () _output.txt text file which contain the list of strings: component name, component a-priori probability and component probability retrieved from wrapper method. AFDIA will parse the output file and present it into the Grid, showing above in a picture.

There is group box name Trash parameters: contain two textboxes, Pass trash and failure trash; these two values will be used to set the trash value to differentiate among components. The entered values will be used to sort the component probability. Therefore entered value must be less than 1. The differentiation can be managed by different Row Colors. Group box named with Graphical Visualization, let the user to choose the type of chart to see the components and their probabilities. The type of charts can be add/deleted as suggested by the supervisor.

Another group box named Information and status, contain a read-only text box, which shows the latest step performed information. That is more concern about the history of the testing session.

Page Name: Strategy.aspx

Page Main Functionality:

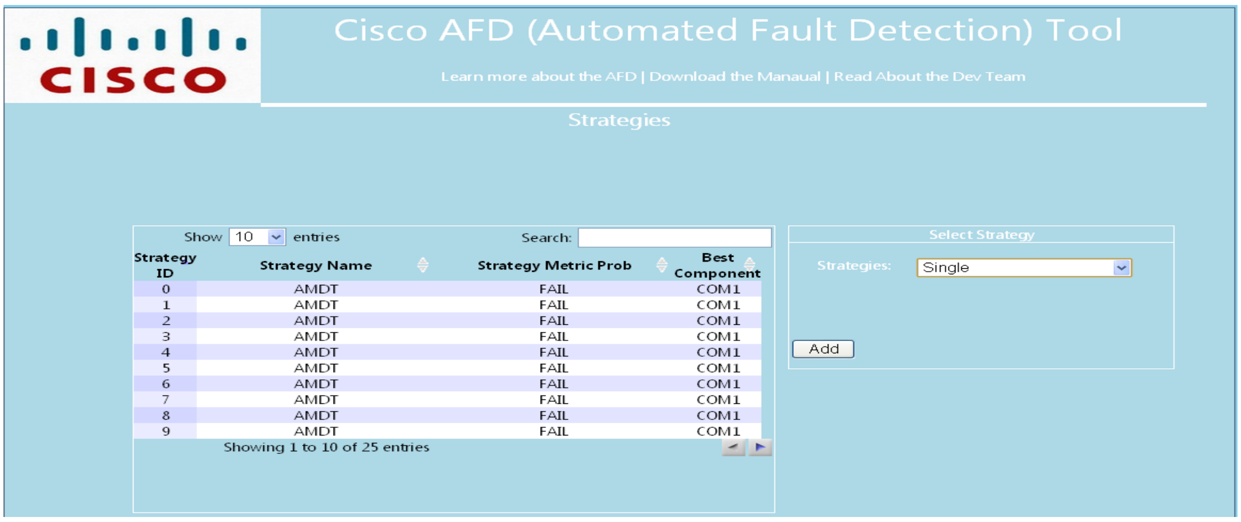
> This Page will be used to make a query to AFD Server for fetching all possible strategies (set of tests) to be executed. After receiving the list of component nodes, AFDIA will present all this information in a Grid, where user can explore these strategies by applying sorting in multiple ways etc.

Possible Movements to/from other pages: Test.aspx, Components.aspx, History.aspx

Interaction with the AFD Database (Table Name(s)): None?

Interaction with the Dll (Wrapper Method Name): AFD_BBN_Library_getStrategies_All ()

Screen shot:


Cisco AFD (Automated Fault Detection) Tool
Learn more about the AFD | Download the Manaual | Read About the Dev Team
Strategies
Show 10 entries
Search:
Select Strategy
Strategies: Single
Add

Strategy ID	Strategy Name	Strategy Metric Prob	Best Component
0	AMDT	FAIL	COM1
1	AMDT	FAIL	COM1
2	AMDT	FAIL	COM1
3	AMDT	FAIL	COM1
4	AMDT	FAIL	COM1
5	AMDT	FAIL	COM1
6	AMDT	FAIL	COM1
7	AMDT	FAIL	COM1
8	AMDT	FAIL	COM1
9	AMDT	FAIL	COM1

Showing 1 to 10 of 25 entries

Figure 5.6: webpage Strategy.aspx

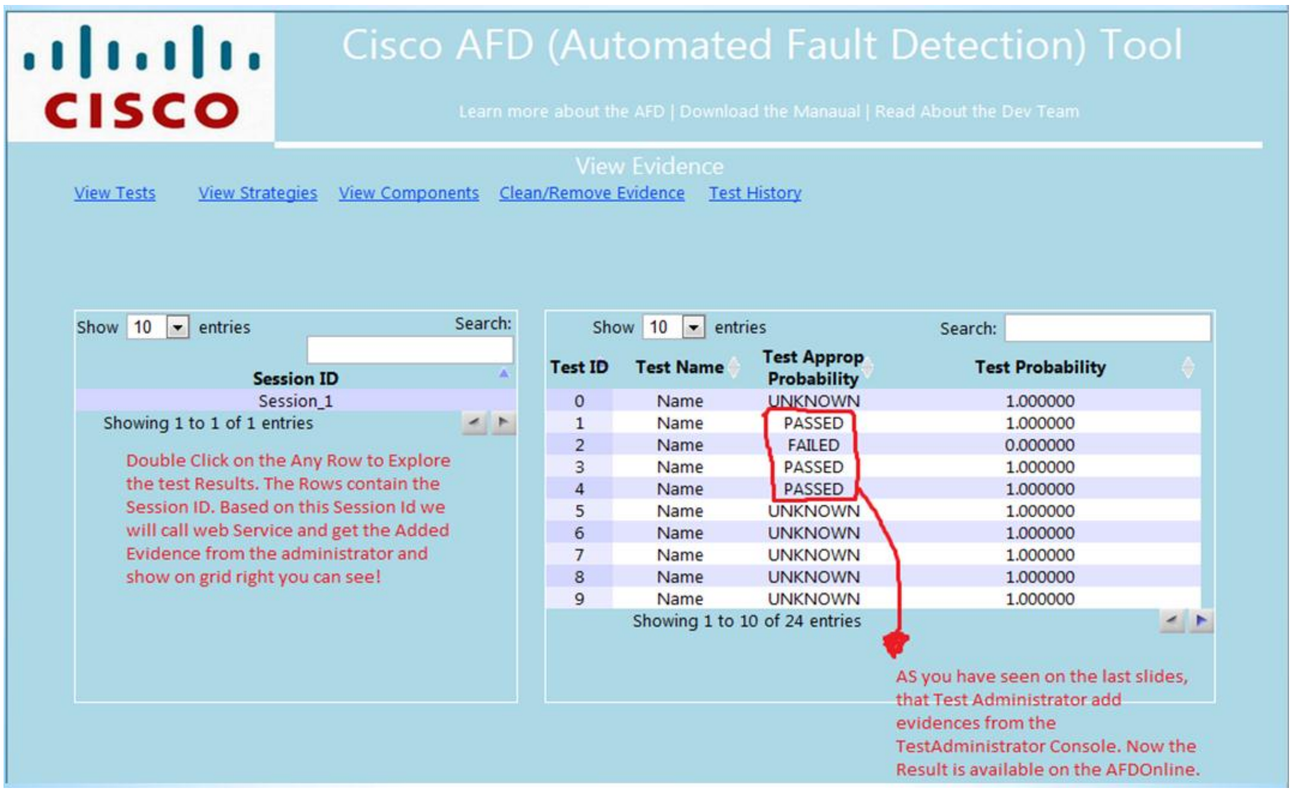
Detail Description of the Page: AFDIA will reach this page, after completing the Initialization state. On the Page Load method, AFDIA will call a Wrapper Method AFD_BBN_Library_getStrategies_All ().

The output of this method will result an AFD_BBN_Library_getStrategies_All () _output.txt text file which contain the list of tests and its ranked optimal strategies. AFDIA will parse the output file and present it into the Grid, showing above in a picture. Group box named with select Strategy, Currently as AFD Tool provide only single test strategy, so we can go on only with single test. But in implementation, other two options are also available, which can be modified upon supervisor instructions.

User can select a specific strategy from the Grid, by single click on the Row and the Click on "ADD" button in the group box.

Page Name: AddEvidence.aspx
Page Main Functionality:
User will visit this Page, to send the sessions id for evidences insertions. Than AFDOnline will call the web service exposed by the AFD WebServer and send the selected session ids to the AFD database. Then upon the successful calling the web service, these session ids will be saved into a database table from where this information will be used by AFDIA Test Administrator web application for evidence insertion.
Possible Movements to/from other pages: Remove_CleanEvidence.aspx**,**Test.aspx, Components.aspx, History.aspx and strategies.aspx
Interaction with the AFD Database (Table Name(s)): dbo.Test, dbo.Sesion
Interaction with the Dll (Wrapper Method Name):
AFD_BBN_Library_addEvidences (), AFD_BBN_Library_getTests_All ().

Screen shot:



Cisco AFD (Automated Fault Detection) Tool

Learn more about the AFD | Download the Manaual | Read About the Dev Team

View Evidence

View Tests View Strategies View Components Clean/Remove Evidence Test History

Show 10 entries Search:

Session ID
Session_1

Showing 1 to 1 of 1 entries

Double Click on the Any Row to Explore the test Results. The Rows contain the Session ID. Based on this Session Id we will call web Service and get the Added Evidence from the administrator and show on grid right you can see!

Show 10 entries Search:

Test ID	Test Name	Test Approp Probability	Test Probability
0	Name	UNKNOWN	1.000000
1	Name	PASSED	1.000000
2	Name	FAILED	0.000000
3	Name	PASSED	1.000000
4	Name	PASSED	1.000000
5	Name	UNKNOWN	1.000000
6	Name	UNKNOWN	1.000000
7	Name	UNKNOWN	1.000000
8	Name	UNKNOWN	1.000000
9	Name	UNKNOWN	1.000000

Showing 1 to 10 of 24 entries

AS you have seen on the last slides, that Test Administrator add evidences from the TestAdministrator Console. Now the Result is available on the AFDOnline.

Figure 5.7: webpage AddEvidence.aspx

Detail Description of the Page:

This Page will present all sessions Id in the left grid which you can see above in the picture. And the grid on the right will be empty upon arrival on this page. And that will be populated when you click on the any selected session id. All the sessions id will be presented in the Grid only and only if its corresponding tests results had been inserted by the test administrator.

1) When the user will traverse to this page. On this arrival AFDOnline immediately call a web services exposed by the AFD WebServer, this web service will notify my AFDOnline that followings are the session ids against them Test Administrator has record the results.
2) AFDOnline will map all these session ids to the Grid, you can see above in the picture on the left.
3) Tester can select any of the following session id.
4) Upon selection of a particular session id, AFDOnline will query the test table in the database and return and map all the test results on the grid right to the session id's grid.

Tester can select and see all the session ids corresponding results by just double clicking on the session id in the grid.

Page Name: Remove_CleanEvidence.aspx

Page Main Functionality:

User will visit this Page, to Remove/Clean evidences. Than AFDIA will call the method AFD_BBN_Library_removeEvidences () to remove the evidences when user will click on the Remove button, provided on the interface. A Refresh of all visualization contents is required. For this AFDIA will call the AFD_BBN_Library_Finalize ().

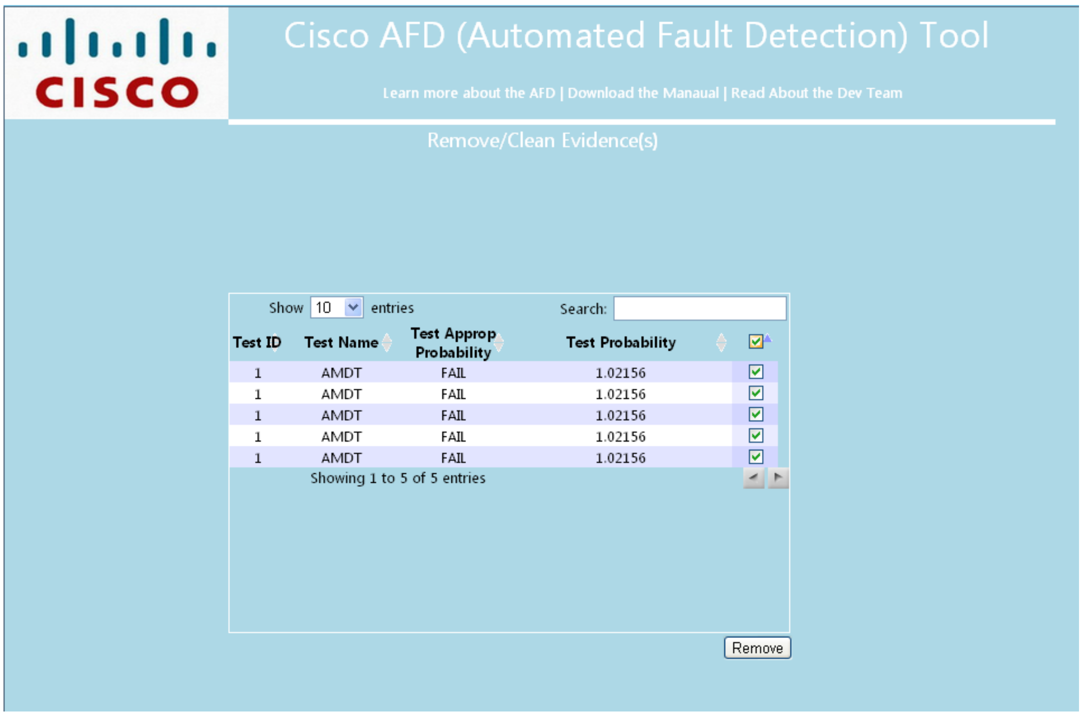
Possible Movements to/from other pages: Remove_CleanEvidence.aspx**,** Test.aspx, Components.aspx, History.aspx

Interaction with the AFD Database (Table Name(s)): dbo.Test

Interaction with the Dll (Wrapper Method Name):

AFD_BBN_Library_removeEvidences (), AFD_BBN_Library_Finalize ()

Screen Shot:


Cisco AFD (Automated Fault Detection) Tool
Learn more about the AFD | Download the Manaual | Read About the Dev Team
Remove/Clean Evidence(s)
Show 10 entries
Search:
Remove

Test ID	Test Name	Test Approp Probability	Test Probability	✓
1	AMDT	FAIL	1.02156	✓
1	AMDT	FAIL	1.02156	✓
1	AMDT	FAIL	1.02156	✓
1	AMDT	FAIL	1.02156	✓
1	AMDT	FAIL	1.02156	✓

Showing 1 to 5 of 5 entries

Figure 5.8: webpage Remove_CleanEvidence.aspx

Detail Description of the Page: this page is reserve to remove/clean evidences by calling the wrapper method. When user will come on this page, AFDIA will query test table to retrieve all tests. And then present in the grid. User can select one or more than one test to perform this operation.

When user will click on the Remove button after selection, an input file will be prepared by the parser component of the AFDIA. Then a call is performed to the wrapper method to remove the evidences and synchronize the AFD tool with the DB.

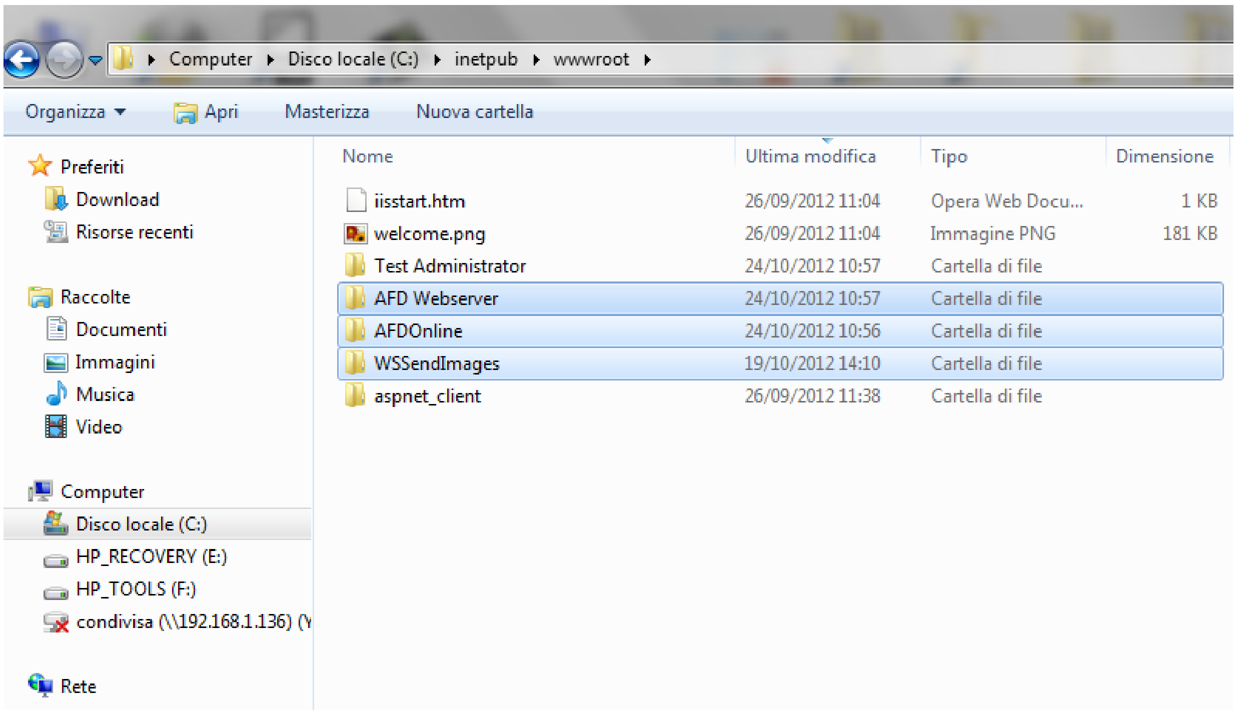
Deployment

Software deployment is all of the activities that make a software system available for use. A newly created program may work fine on your computer, but that doesn't mean it is really ready for others to use. There are many extra program features you probably hadn't needed for you, but ought to provide if the program will be used by others. So it is necessary to put all our web applications on a web server that public them for users. We are deploying AFDOnline, AFD Webserver and Test Administrator on IIS (Internet Information Services), IIS is a web server developed by Microsoft. After deployment on IIS every web application being assigned unique URL (Unified Resource Locator) addresses for accessing trough internet/Intranet.

As we have not any formal web server to deploy. Let's assume localhost as web server. But later, it can be a formal dedicated Server. Here is complete procedure (in bullets) of deployment on IIS.

Deployment of AFDOnline (Website)

- Create c:\inetpub\wwwroot, if not exist
- Create three folders in c:\inetpub\wwwroot namely AFDOnline, Test Administrator and AFD Webserver. We created three because we need to publish all three solutions.



Nome	Ultima modifica	Tipo	Dimensione
iisstart.htm	26/09/2012 11:04	Opera Web Docu...	1 KB
welcome.png	26/09/2012 11:04	Immagine PNG	181 KB
Test Administrator	24/10/2012 10:57	Cartella di file	
AFD Webserver	24/10/2012 10:57	Cartella di file	
AFDOnline	24/10/2012 10:56	Cartella di file	
WSSendImages	19/10/2012 14:10	Cartella di file	
aspnet_client	26/09/2012 11:38	Cartella di file	

Figure 6.1: inetpub/wwwroot/FoldersForDeployment

- Open Visual Studio 2010 as Administrator

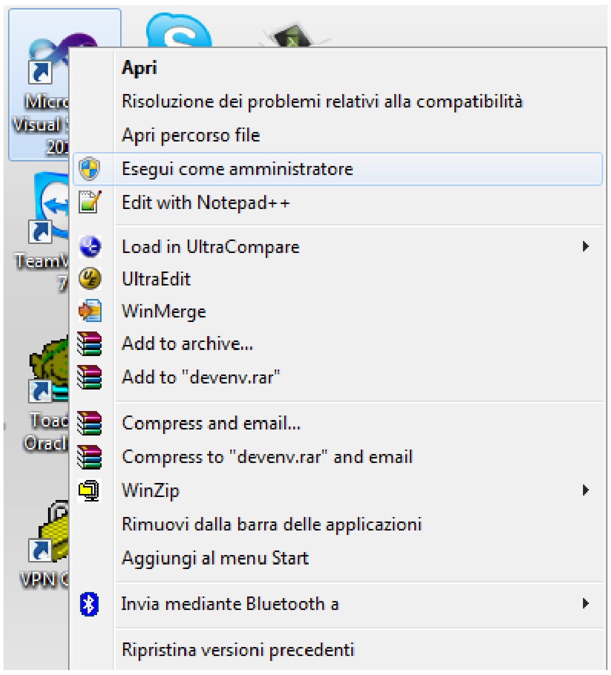
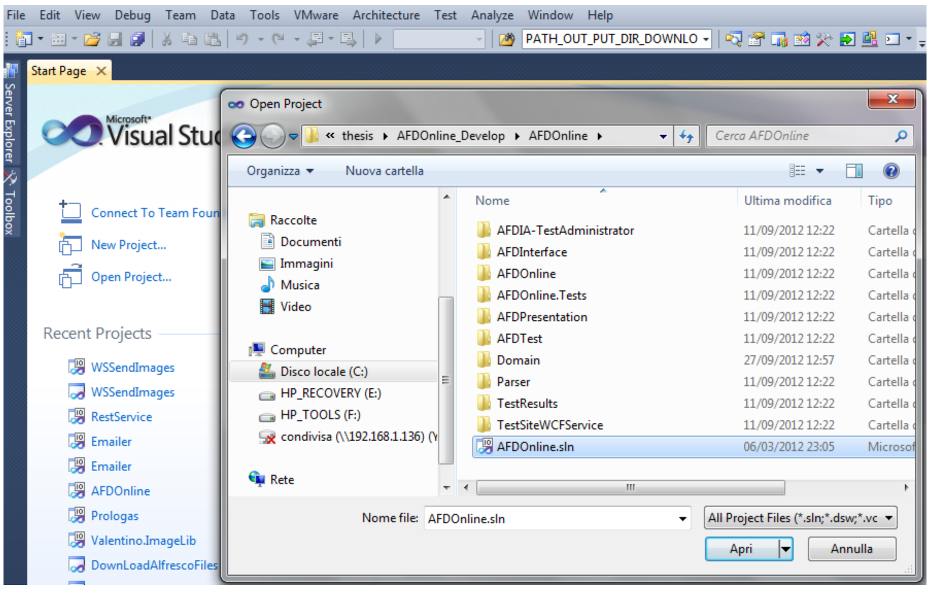

Apri
Risoluzione dei problemi relativi alla compatibilità
Apri percorso file
Esegui come amministratore
Edit with Notepad++
Load in UltraCompare
UltraEdit
WinMerge
Add to archive...
Add to "devenv.rar"
Compress and email...
Compress to "devenv.rar" and email
WinZip
Rimuovi dalla barra delle applicazioni
Aggiungi al menu Start
Invia mediante Bluetooth a
Ripristina versioni precedenti

Figure 6.2: Run VS as administrator

- Open project locating on local disk's path



Nome	Ultima modifica	Tipo
AFDIA-TestAdministrator	11/09/2012 12:22	Cartella
AFDInterface	11/09/2012 12:22	Cartella
AFDOnline	11/09/2012 12:22	Cartella
AFDOnline.Tests	11/09/2012 12:22	Cartella
AFDPresentation	11/09/2012 12:22	Cartella
AFDTest	11/09/2012 12:22	Cartella
Domain	27/09/2012 12:57	Cartella
Parser	11/09/2012 12:22	Cartella
TestResults	11/09/2012 12:22	Cartella
TestSiteWCFService	11/09/2012 12:22	Cartella
AFDOnline.sln	06/03/2012 23:05	Microsof

Nome file: AFDOnline.sln
All Project Files (*.sln;*.dsw;*.vc

Figure 6.3: Project on Local Disk

- Right click on the project which you want to publish, will appear a pop up containing multiple options. There should be one option called “Publish”

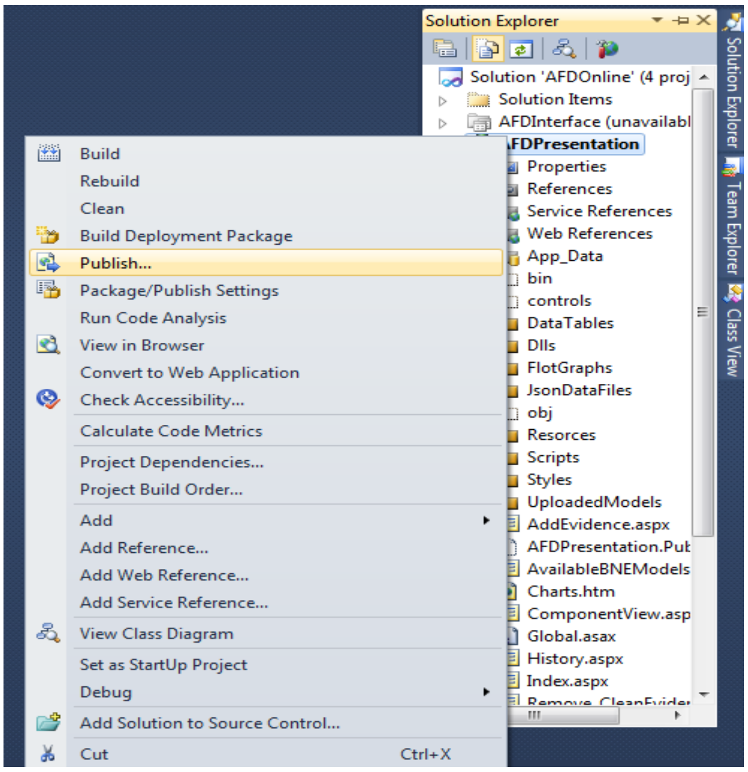
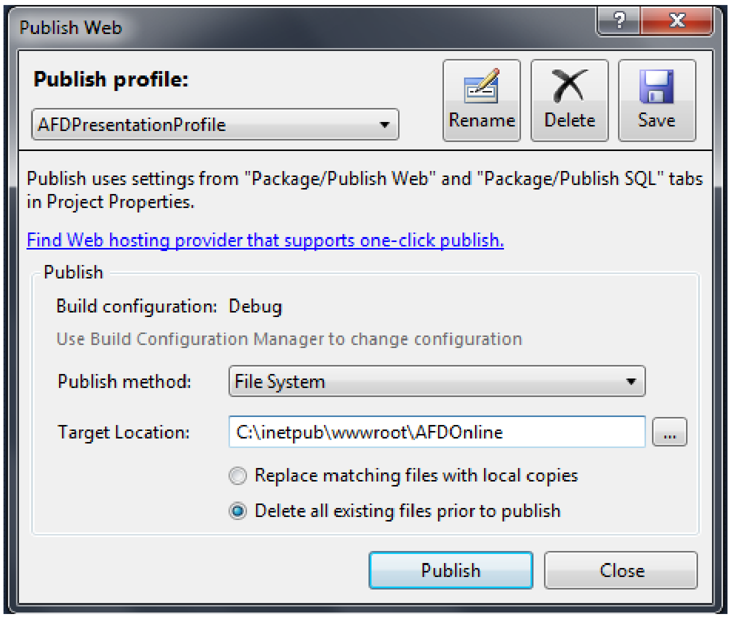


Figure 6.4: Publish WebSite

- Following window will appear, select Publish method as “FileSystem” and set the target location where you want published site. In our case it is C:\inetpub\wwwroot\AFDOnline. After these setting click on the Publish button.



Publish profile: AFDPresentationProfile

Publish uses settings from "Package/Publish Web" and "Package/Publish SQL" tabs in Project Properties.

Find Web hosting provider that supports one-click publish.

Publish

Build configuration: Debug

Use Build Configuration Manager to change configuration

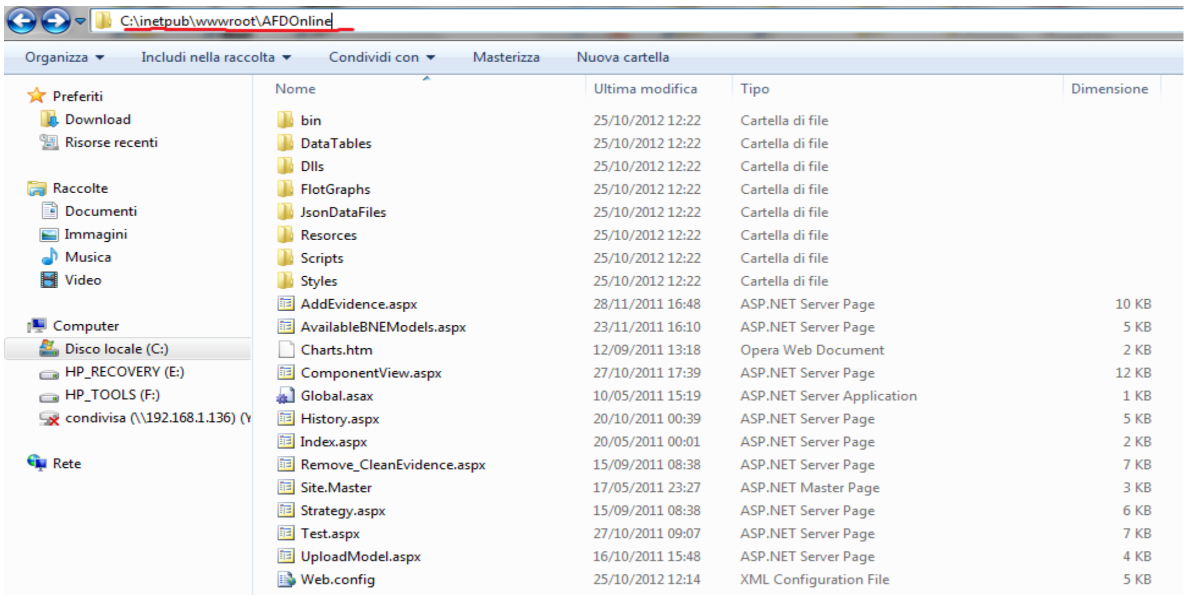
Publish method: File System

Target Location: C:\inetpub\wwwroot\AFDOnline

Replace matching files with local copies [✓ Delete all existing files prior to publish]

Figure 6.5: Publish Web Profile

- Upon successful publishing, notice on the specified path we have our published web site.


C:\inetpub\wwwroot\AFDOnline

Organizza ▾ Includi nella raccolta ▾ Condividi con ▾ Masterizza Nuova cartella

Preferiti, Download, Risorse recenti, Raccolte, Documenti, Immagini, Musica, Video, Computer, Disco locale (C:), HP_RECOVERY (E:), HP_TOOLS (F:), condivisa (\\192.168.1.136) (Y, Rete

Nome	Ultima modifica	Tipo	Dimensione
bin	25/10/2012 12:22	Cartella di file	
DataTables	25/10/2012 12:22	Cartella di file	
Dlls	25/10/2012 12:22	Cartella di file	
FlotGraphs	25/10/2012 12:22	Cartella di file	
JsonDataFiles	25/10/2012 12:22	Cartella di file	
Resorces	25/10/2012 12:22	Cartella di file	
Scripts	25/10/2012 12:22	Cartella di file	
Styles	25/10/2012 12:22	Cartella di file	
AddEvidence.aspx	28/11/2011 16:48	ASP.NET Server Page	10 KB
AvailableBNEModels.aspx	23/11/2011 16:10	ASP.NET Server Page	5 KB
Charts.htm	12/09/2011 13:18	Opera Web Document	2 KB
ComponentView.aspx	27/10/2011 17:39	ASP.NET Server Page	12 KB
Global.asax	10/05/2011 15:19	ASP.NET Server Application	1 KB
History.aspx	20/10/2011 00:39	ASP.NET Server Page	5 KB
Index.aspx	20/05/2011 00:01	ASP.NET Server Page	2 KB
Remove_CleanEvidence.aspx	15/09/2011 08:38	ASP.NET Server Page	7 KB
Site.Master	17/05/2011 23:27	ASP.NET Master Page	3 KB
Strategy.aspx	15/09/2011 08:38	ASP.NET Server Page	6 KB
Test.aspx	27/10/2011 09:07	ASP.NET Server Page	7 KB
UploadModel.aspx	16/10/2011 15:48	ASP.NET Server Page	4 KB
Web.config	25/10/2012 12:14	XML Configuration File	5 KB

Figure 6.6: Published Web Content

- At this point we are ready to put our published site on IIS Web Server. To do this first step is to open IIS Manager which reside <u>Control Panel/Administrative Tools/Internet Information Services (IIS) Manager</u>. By opening, following window would appear.

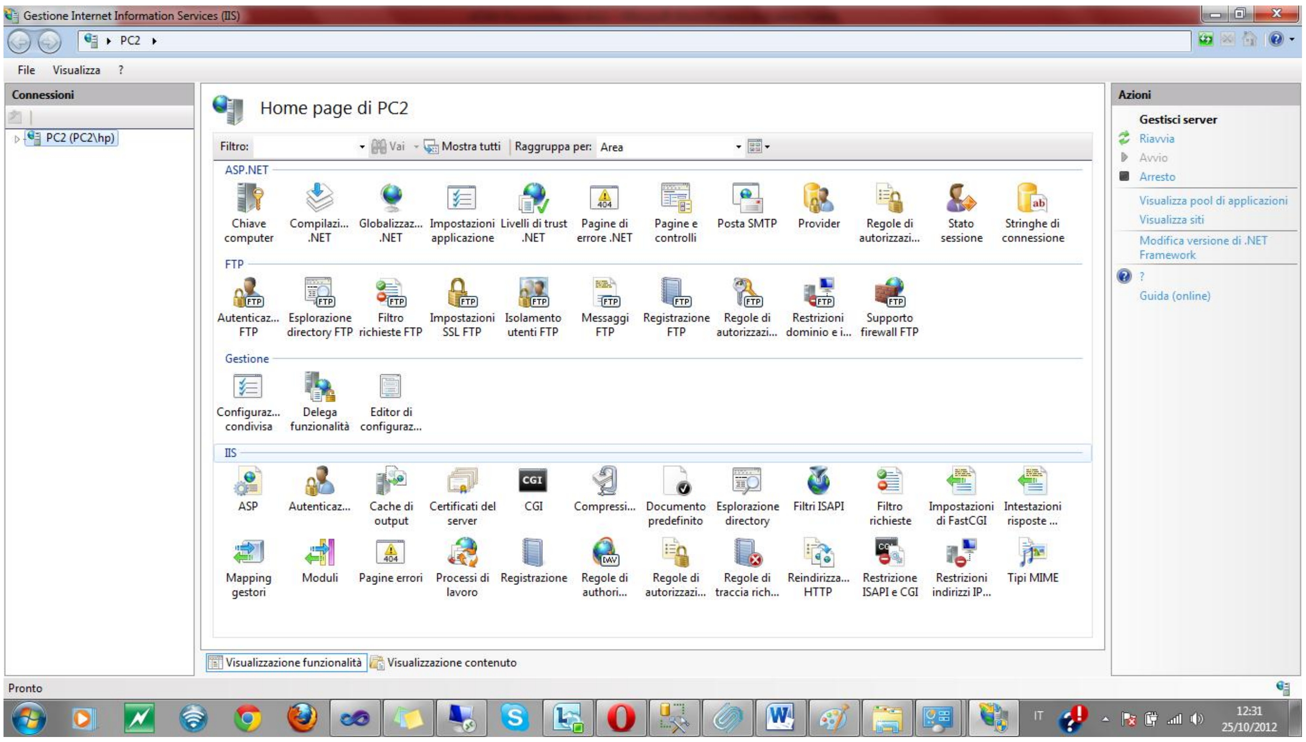


Figure 6.7: IIS Manager

- In Connections pane (at extreme left) of the IIS manager console there is one virtual server, by expanding it we would see one folder called Web Sites. All web sites those needs to publish over IIS must present here. We will add a new web site called AFDOnline and associate the published site which produced by VS 2010.

Adding new web site:

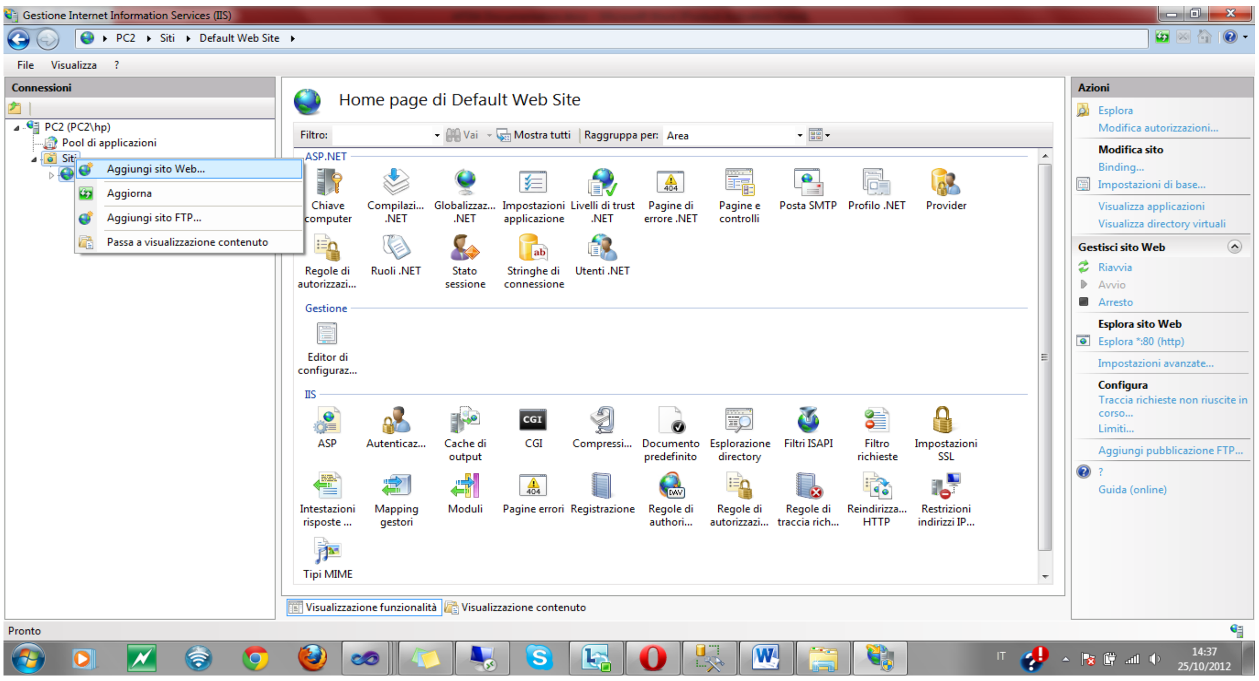
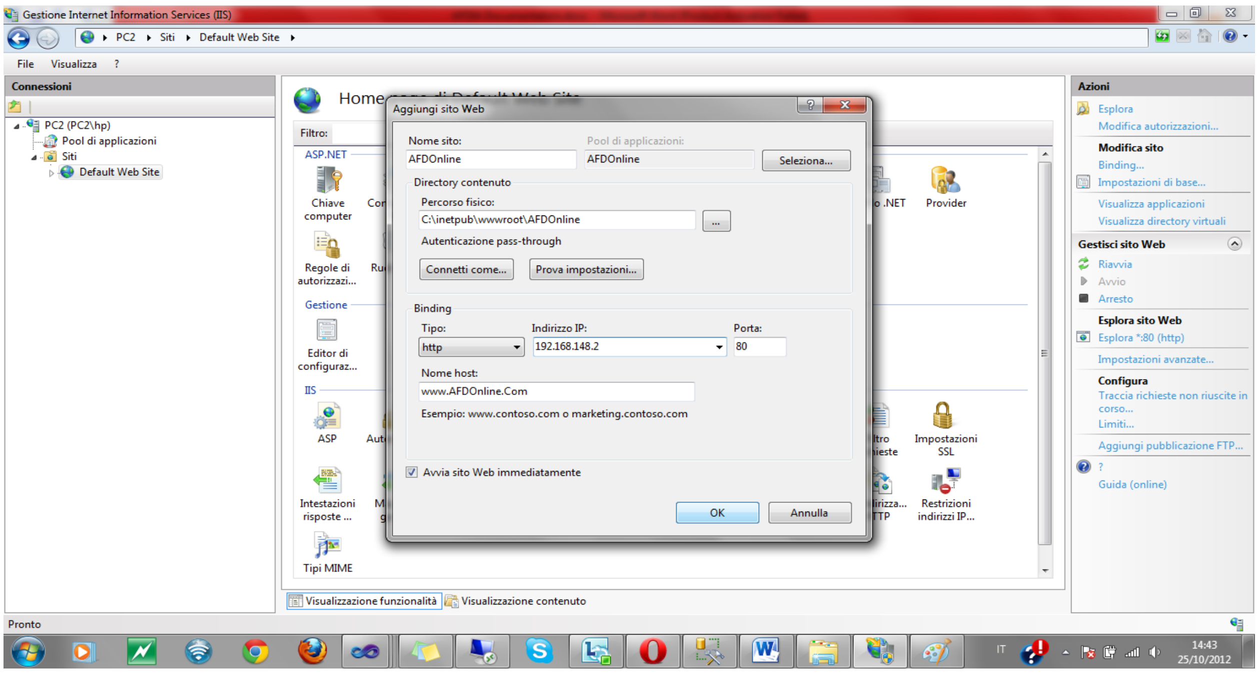


Figure 6.8: IIS Manager- Add New Web Site

Associate Published site (published by VS 2010) with the newly added Site:



Nome sito: AFDOnline
Pool di applicazioni: AFDOnline
Directory contenuto
Percorso fisico: C:\inetpub\wwwroot\AFDOnline
Autenticazione pass-through
Binding
Tipo: http
Indirizzo IP: 192.168.148.2
Porta: 80
Nome host: www.AFDOnline.Com
Esempio: www.contoso.com o marketing.contoso.com
☑ Avvia sito Web immediatamente

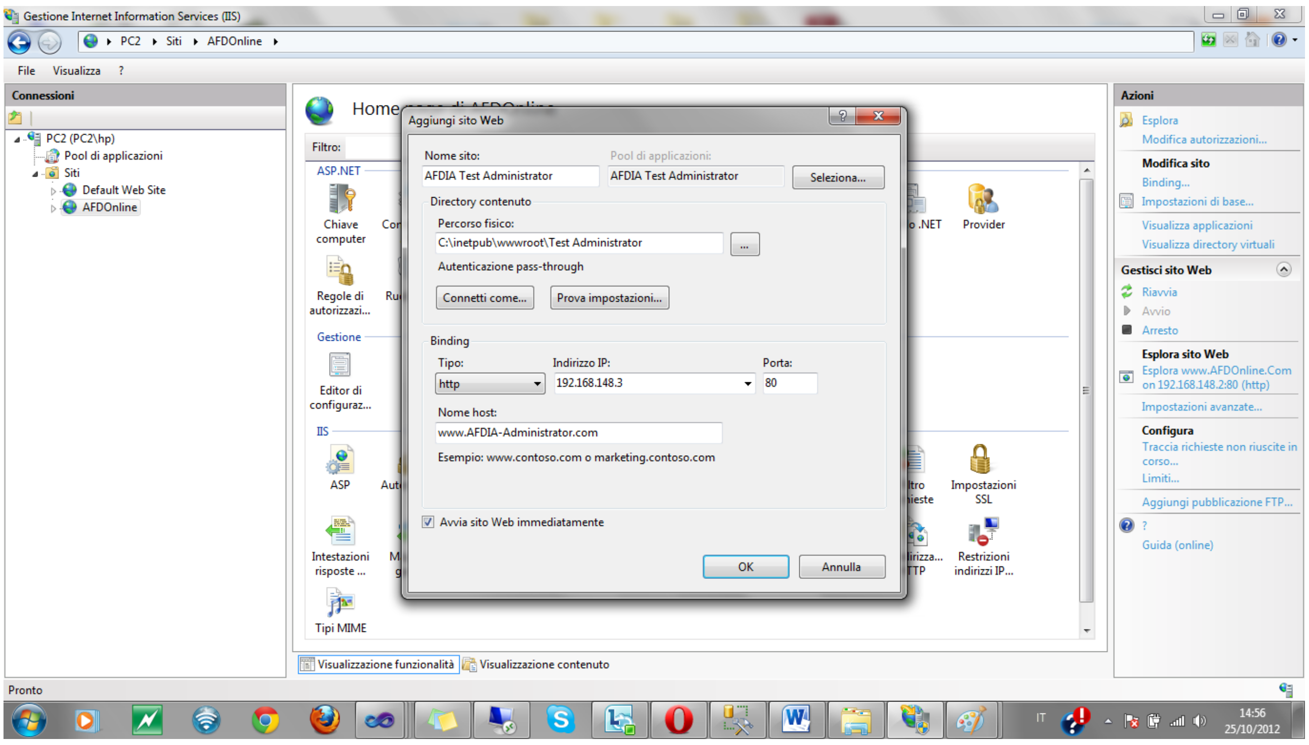
Figure 6.8: IIS Manager- Configure Added WebSite

That’s it about publishing AFDOnline on IIS Server, now this server can be access by calling www.AFDOnline.com only on Local host.

Deployment of AFD Webserver & Test Administrator (Website)

- As the publishing and deployment procedure are same in the case of AFD Webserver and Test Administrator. So, I will not repeat this work. But would like present a final look of all published sites on IIS.

Test Administrator deployment:


Aggiungi sito Web

Nome sito: AFDIA Test Administrator

Pool di applicazioni: AFDIA Test Administrator

Directory contenuto

Percorso fisico: C:\inetpub\wwwroot\Test Administrator

Autenticazione pass-through

Binding

Tipo: http

Indirizzo IP: 192.168.148.3

Porta: 80

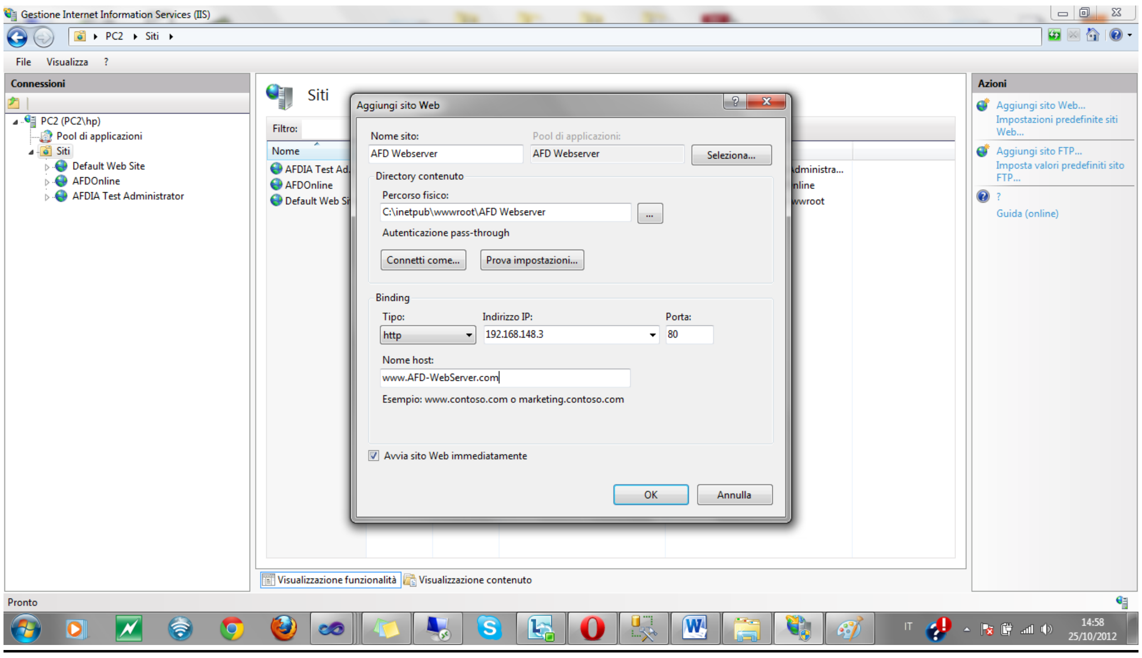
Nome host: www.AFDIA-Administrator.com

Esempio: www.contoso.com o marketing.contoso.com

☑ Avvia sito Web immediatamente

Figure 6.9: IIS Manager- Configure Test Administrator WebSite

AFD Webserver deployment:



Aggiungi sito Web

Nome sito: AFD Webserver

Pool di applicazioni: AFD Webserver — Seleziona...

Directory contenuto

Percorso fisico: C:\inetpub\wwwroot\AFD Webserver

Autenticazione pass-through

Connetti come... Prova impostazioni...

Binding

Tipo:	Indirizzo IP:	Porta:
http	192.168.148.3	80

Nome host: www.AFD-WebServer.com

Esempio: www.contoso.com o marketing.contoso.com

☑ Avvia sito Web immediatamente

OK Annulla

Figure 6.10: IIS Manager- Configure AFD Webserver WebSite

Code

Project: AFD WebServer

Interface of web Service: scAddEvidence:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using Doamin;

namespace AddEvidenceWedService
{
  // NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name "IService1" in both
code and config file together.
  [ServiceContract]
  public interface scAddEvidence
  {

    [OperationContract]
    string  GetTestStatus(string sessionID);


    [OperationContract]
    void SendTestToAdministrator(List<dcAddEvidence> tests);

    [OperationContract]
    List<string> GetListofSessiosIDs();

    [OperationContract]
    List<TestResult> GetListOfTestsBySessionID(string sessionID);

    [OperationContract]
    void AddEvidenceIntoDB(string sessionID, string evidenceString);


  }
```

Data Transfer object of web Service: dcAddEvidence:

```
  // Use a data contract as illustrated in the sample below to add composite types to service operations.
  [DataContract]
  public class dcAddEvidence
  {


      private int _TestID;
      [DataMember]
      public int TestID
      {
        get { return _TestID; }
```

```
            set { _TestID = value; }  }
        private string _TestOutCome;
        [DataMember]
        public string TestOutCome
        {
            get { return _TestOutCome; }
            set { _TestOutCome = value; }
        }

        private string _TestProbability;
        [DataMember]
        public string TestProbability
        {
            get { return _TestProbability; }
            set { _TestProbability = value; }
        }


        private string _TetstName;
        [DataMember]
        public string TetstName
        {
            get { return _TetstName; }
            set { _TetstName = value; }
        }


        private string _ModelID;
        [DataMember]
        public string ModelID
        {
            get { return _ModelID; }
            set { _ModelID = value; }
        }


        private string _SessionID;
        [DataMember]
        public string SessionID
        {
            get { return _SessionID; }
            set { _SessionID = value; }
        }


        private string _TestInternalID;
        [DataMember]
        public string TetsInternalID
        {
            get { return _TestInternalID; }
            set { _TestInternalID = value; }
        }
    }

}
```

Service Implementation class, implements scAddEvidence: AddEvidenceServiceImplentation

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using System.Data.Linq;
using Doamin;

namespace AddEvidenceWedService
{
   // NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "Service1" in code, svc
and config file together.
   public class AddEvidenceServiceImplementation : scAddEvidence
   {
      private AFDDBAdministratorDataContext _AfdDataContext;
      private AFDDBAdministratorDataContext AfdDataContext
      {
         get
         {
            return _AfdDataContext;
         }
         set
         {
            _AfdDataContext = value;
         }
      }

      public AddEvidenceServiceImplementation()
      {
         _AfdDataContext = new AFDDBAdministratorDataContext();

      }

      #region   public bool SendTestToAdministrator(List<dcAddEvidence> tests)

      public void SendTestToAdministrator(List<dcAddEvidence> testsDetail)
      {


         for (int i = 0; i < testsDetail.ToList().Count; i++)
         {

            if (testsDetail[i] != null)
            {

               var item = new TestResult();

               item.ID_Model = testsDetail[i].ModelID;
               item.ID_Session = testsDetail[i].SessionID;
               item.ID_Test = testsDetail[i].TestID;
               item.Test_InternalID = testsDetail[i].TetsInternalID;
               item.Test_Name = testsDetail[i].TetstName;
               item.Test_OutCome = testsDetail[i].TestOutCome;
            item.Test_Probability = testsDetail[i].TestProbability;

            _AfdDataContext.TestResults.InsertOnSubmit(item);
          }


        }

        _AfdDataContext.SubmitChanges();


      }
      #endregion

      #region public List<string> GetListofSessiosIDs()
      public List<string> GetListofSessiosIDs()
      {
        return AfdDataContext.TestResults.Select(p => p.ID_Session).Distinct().ToList();
      }

      #endregion

      #region    public List<TestResult> GetListOfTestsBySessionID(string sessionID)
      public List<TestResult> GetListOfTestsBySessionID(string sessionID)
      {
        return AfdDataContext.TestResults.Where(i => i.ID_Session == sessionID).ToList();

      }

      #endregion

      #region   public void AddEvidenceIntoDB(string sessionID, string evidenceString)

      public void AddEvidenceIntoDB(string sessionID, string evidenceString)
      {

        if (AfdDataContext.TestEvidences.Where(i => i.SessionID == sessionID).Count() > 0)
        {
          // update the existing record
          TestEvidence testEvidence = AfdDataContext.TestEvidences.Single(i => i.SessionID == sessionID);
          testEvidence.Evidences = evidenceString;

        }
        else
        {
          // Register the new one
          TestEvidence item = new TestEvidence();
          item.SessionID = sessionID;
          item.Evidences = evidenceString;
          AfdDataContext.TestEvidences.InsertOnSubmit(item);
        }

        AfdDataContext.SubmitChanges();
      }

      #endregion

      #region   public string GetTestStatus(string sessionID)

      public string GetTestStatus(string sessionID)
```

```
        {
          return AfdDataContext.TestEvidences.Where(i => i.SessionID == sessionID).Select(i => i.Evidences).SingleOrDefault();
        }

        #endregion
    }
}
```

Project: AFDIA Test Administrator

Web Page: AllPendingTests.aspx [Markup]

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeBehind="AddEvidences.aspx.cs" Inherits="TestSite._Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">

 <link rel="Stylesheet" href ="DataTables/css/demo_table.css" type ="text/css" />
    <script type ="text/javascript" src="DataTables/js/jquery.js"></script>
    <script type ="text/javascript" src="DataTables/js/jquery.dataTables.js"></script>


    <script type ="text/javascript" language ="javascript">

        var FileInputString = "";

      $(document).ready(function () {

          $('#tblTests').dataTable({
                "oLanguage": { "oView": "" },
                "iDisplyLength": 10,
                "aaSorting": [[0, "asc"]]
            });
        });


        $('#tblTests tbody tr').click(function () {
            alert('e');
        });




        function buttonupdateTestStatuses() {


            var tab = document.getElementById("tblTests"); // table with id tbl1
            var elems = tab.getElementsByTagName("input");
            var len = elems.length/3;


            for (var i = 0; i < len; i++)
             {

                var rbPass = "radiobuttonPass_" + i;
                var rbFail = "radiobuttonFail_" + i;
                var rbUnknown = "radiobuttonUnknown_" + i;

                var objrbPassID = document.getElementById(rbPass.toString());
                var objrbFailID = document.getElementById(rbFail.toString());
                var objrbUnknownID = document.getElementById(rbUnknown.toString());



                if (objrbPassID.checked) {


                    var testIDKey = "testid_ " + i;
                    var objtestID = document.getElementById(testIDKey.toString());
                    FileInputString += "P:" + objtestID.innerHTML.toString().trim() + ",";

                    var HiddenValue = document.getElementById("<%= hiddenFieldFileData.ClientID %>");
                    HiddenValue.value = FileInputString;


                }
                else if (objrbFailID.checked) {

                    var testIDKey = "testid_ " + i;
                    var objtestID = document.getElementById(testIDKey.toString());
                    FileInputString += "F:" + objtestID.innerHTML.toString().trim() + ",";

                    var HiddenValue = document.getElementById("<%= hiddenFieldFileData.ClientID %>");
                    HiddenValue.value = FileInputString;
                }
                else if (objrbUnknownID.checked) {

                    var testIDKey = "testid_ " + i;
                    var objtestID = document.getElementById(testIDKey.toString());
                    FileInputString += "Unknown:" + objtestID.innerHTML.toString().trim() + ",";

                    var HiddenValue = document.getElementById("<%= hiddenFieldFileData.ClientID %>");
                    HiddenValue.value = FileInputString;
                }


            }

            if (confirm('Press OK! if You are sure about your Evidences selection, upon successful inserttion the control will transfer
to Page: View_Pending_Session_List '))
            {
                return true;
            }
            else {
                // reset the FileintputString in case, when user select "NO" from the confirmation popUp
                // this is because Either Yes OR No, in both cases, this variable has been papulated
                // SO, it must be Cleaned when user select NO
                FileInputString = "";
               return false;
             }
        }



      </script>
```

```
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
  <div id="divPage" style =" height:650px; width :100%; background-color :lightblue;   ">

    <div style = " color :White;  width:100%; text-align: center; ">
     <span style="color: #FFFFFF; font-size:large;  "> Add Evidences </span></div>

     <div style = " width : 100%">

       <div style = " margin-left :50px; width:179px; float :left;">
        <asp:LinkButton ID="linkbuttonViewTests" runat="server" onclick="linkbuttonViewTests_Click"
            >View Pending Sessions List</asp:LinkButton>
       </div>


     </div>


     <div style =" margin-top:100px; ">



     <div style = " border: thin solid #FFFFFF; margin-left :200px;  float:left ; margin-top :25px;  height:auto; width:650px;
text-align :center;" >


    <asp:Repeater ID="repeaterTests" runat ="server">

        <HeaderTemplate>
          <table id="tblTests" cellpadding ="0" width ="650" cellspacing ="0" border ="0" class = "disply">
           <thead>
            <tr>
              <th style =" width:80px;" >Session ID</th>
              <th style =" width:50px;" >Test ID</th>
              <th style =" width:100px;" >Test Name</th>
              <th style =" width:100px;" >Test Approp Probability</th>
              <th >Test Probability</th>
              <th  style =" width:80px;">Pass </th>
              <th  style =" width:80px;">Fail </th>
              <th  style =" width:80px;">Unknown </th>
            </tr>
           </thead>
          <tbody>
        </HeaderTemplate>


        <ItemTemplate>
         <tr>

            <td> <%# Eval("ID_Session")%> </td>
            <td id="testid_ <%# Eval("Test_InternalID")%>" > <%# Eval("Test_InternalID")%> </td>
            <td id="testname_ <%# Eval("Test_InternalID")%>"> <%# Eval("Test_Name")%></td>
            <td id="testoutcome_ <%# Eval("Test_InternalID")%>"> <%# Eval("Test_OutCome")%> </td>
            <td id="testProbability_ <%# Eval("Test_InternalID")%>"> <%# Eval("Test_Probability")%></td>
            <td ><input id= "radiobuttonPass_<%# Eval("Test_InternalID")%>" type="radio" name = "radiobutton_<%#
Eval("Test_InternalID")%>"  ></td>
            <td ><input id= "radiobuttonFail_<%# Eval("Test_InternalID")%>" type="radio" name ="radiobutton_<%#
Eval("Test_InternalID")%>"  ></td>
```

```
              <td ><input id= "radiobuttonUnknown_<%# Eval("Test_InternalID")%>" type="radio" name = "radiobutton_<%#
Eval("Test_InternalID")%>"  ></td>
            </tr>
          </ItemTemplate>


          <FooterTemplate>
           </tbody>
           </table>
          </FooterTemplate>

        </asp:Repeater>

      </div>


          </div>

           <div style =" margin-left:640px;">
          <asp:button   id ="buttonupdateTestStatuses"  runat ="server"  Text  ="update Tests Status"
                      OnClientClick  = "return buttonupdateTestStatuses();" width="130"
                    onclick="buttonupdateTestStatuses_Click" />
      </div>

      <div style =" margin-left:200px; width :500px;">
        <asp:Label ID="labelMessage" runat ="server" ForeColor ="White" Font-Size ="Large"  ></asp:Label>
      </div>

      <div id="hiddenFieldDiv"  style = "  display :none;"  >
          <asp:HiddenField ID="hiddenFieldFileData" Value = "" runat="server" />
      </div>

  </div>

</asp:Content>
```

Web Page: AllPendingTests.aspx.cs [Code-Behind]

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using TestSite.AFDIAWebServiceReference;

namespace TestSite
{
   public partial class AllPendingTests : System.Web.UI.Page
   {


      protected void Page_Load(object sender, EventArgs e)
      {
         binGrdifromDBData();
      }



private void binGrdifromDBData()
```

```
        {

            /*call wcf service for get the list of all Sessions ID*/
            scAddEvidenceClient client = new scAddEvidenceClient();
            client.Open();
            var list = client.GetListofSessiosIDs();
            client.Close();


            /* bind that list with the DataGrid */
            List<bindableList> bdablelist = new List<bindableList>(list.Length);
            if (list.Length  != 0)
            {


                for (int i = 0; i < list.Length; i++)
                {
                    var item = new bindableList();

                    item.ID_Session = list[i];
                    bdablelist.Add(item);
                }


                repeaterAllBNEModels.DataSource = bdablelist;
                repeaterAllBNEModels.DataBind();
            }
            else
            {

                repeaterAllBNEModels.DataSource = bdablelist;
                repeaterAllBNEModels.DataBind();
            }

        }

        protected void linkbuttonViewTests_Click(object sender, EventArgs e)
        {

        }

        protected void linkbuttonViewAllPendingChanges_Click(object sender, EventArgs e)
        {
            Server.Transfer("AllPendingTests.aspx");
        }


    }

    class bindableList {

        private string _sessionid;
        public string ID_Session
        {
            get { return _sessionid; }
            set { _sessionid = value; }
        }
    }
}
```

Web Page: AddEvidences.aspx [Markup]

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
   CodeBehind="AddEvidences.aspx.cs" Inherits="TestSite._Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">

 <link rel="Stylesheet" href ="DataTables/css/demo_table.css" type ="text/css" />
   <script type ="text/javascript" src="DataTables/js/jquery.js"></script>
   <script type ="text/javascript" src="DataTables/js/jquery.dataTables.js"></script>


   <script type ="text/javascript" language ="javascript">

      var FileInputString = "";

     $(document).ready(function () {

         $('#tblTests').dataTable({
              "oLanguage": { "oView": "" },
              "iDisplyLength": 10,
              "aaSorting": [[0, "asc"]]
           });
      });


      $('#tblTests tbody tr').click(function () {
         alert('e');
      });





      function buttonupdateTestStatuses() {


         var tab = document.getElementById("tblTests"); // table with id tbl1
         var elems = tab.getElementsByTagName("input");
         var len = elems.length/3;


         for (var i = 0; i < len; i++)
          {

            var rbPass = "radiobuttonPass_" + i;
            var rbFail = "radiobuttonFail_" + i;
            var rbUnknown = "radiobuttonUnknown_" + i;

            var objrbPassID = document.getElementById(rbPass.toString());
            var objrbFailID = document.getElementById(rbFail.toString());
            var objrbUnknownID = document.getElementById(rbUnknown.toString());



            if (objrbPassID.checked) {
```

```
            var testIDKey = "testid_ " + i;
            var objtestID = document.getElementById(testIDKey.toString());
            FileInputString += "P:" + objtestID.innerHTML.toString().trim() + ",";

            var HiddenValue = document.getElementById("<%= hiddenFieldFileData.ClientID %>");
            HiddenValue.value = FileInputString;


          }
          else if (objrbFailID.checked) {

            var testIDKey = "testid_ " + i;
            var objtestID = document.getElementById(testIDKey.toString());
            FileInputString += "F:" + objtestID.innerHTML.toString().trim() + ",";

            var HiddenValue = document.getElementById("<%= hiddenFieldFileData.ClientID %>");
            HiddenValue.value = FileInputString;
          }
          else if (objrbUnknownID.checked) {

            var testIDKey = "testid_ " + i;
            var objtestID = document.getElementById(testIDKey.toString());
            FileInputString += "Unknown:" + objtestID.innerHTML.toString().trim() + ",";

            var HiddenValue = document.getElementById("<%= hiddenFieldFileData.ClientID %>");
            HiddenValue.value = FileInputString;
          }


        }

        if (confirm('Press OK! if You are sure about your Evidences selection, upon successful inserttion the control will transfer
to Page: View_Pending_Session_List '))
        {
          return true;
        }
        else {
          // reset the FileintputString in case, when user select "NO" from the confirmation popUp
          // this is because Either Yes OR No, in both cases, this variable has been papulated
          // SO, it must be Cleaned when user select NO
          FileInputString = "";
         return false;
        }
      }




    </script>

</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
  <div id="divPage" style =" height:650px; width :100%; background-color :lightblue;   ">

    <div style = " color :White;  width:100%; text-align: center; ">
     <span style="color: #FFFFFF; font-size:large;  "> Add Evidences </span></div>

     <div style = " width : 100%">

      <div style = " margin-left :50px; width:179px; float :left;">
```

```
        <asp:LinkButton ID="linkbuttonViewTests" runat="server" onclick="linkbuttonViewTests_Click"
            >View Pending Sessions List</asp:LinkButton>
        </div>


      </div>


      <div style =" margin-top:100px; ">



      <div style = " border: thin solid #FFFFFF; margin-left :200px;  float:left ; margin-top :25px;  height:auto; width:650px;
text-align :center;" >


      <asp:Repeater ID="repeaterTests" runat ="server">

         <HeaderTemplate>
           <table id="tblTests" cellpadding ="0" width ="650" cellspacing ="0" border ="0" class = "disply">
             <thead>
               <tr>
                  <th style =" width:80px;" >Session ID</th>
                  <th style =" width:50px;" >Test ID</th>
                  <th style =" width:100px;" >Test Name</th>
                  <th style =" width:100px;" >Test Approp Probability</th>
                  <th >Test Probability</th>
                  <th  style =" width:80px;">Pass </th>
                  <th  style =" width:80px;">Fail </th>
                  <th  style =" width:80px;">Unknown </th>
               </tr>
             </thead>
           <tbody>
         </HeaderTemplate>


         <ItemTemplate>
          <tr>

               <td> <%# Eval("ID_Session")%> </td>
               <td id="testid_ <%# Eval("Test_InternalID")%>" > <%# Eval("Test_InternalID")%> </td>
               <td id="testname_ <%# Eval("Test_InternalID")%>"> <%# Eval("Test_Name")%></td>
               <td id="testoutcome_ <%# Eval("Test_InternalID")%>"> <%# Eval("Test_OutCome")%> </td>
               <td id="testProbability_ <%# Eval("Test_InternalID")%>"> <%# Eval("Test_Probability")%></td>
               <td ><input id= "radiobuttonPass_<%# Eval("Test_InternalID")%>" type="radio" name = "radiobutton_<%#
Eval("Test_InternalID")%>"  ></td>
               <td ><input id= "radiobuttonFail_<%# Eval("Test_InternalID")%>" type="radio" name ="radiobutton_<%#
Eval("Test_InternalID")%>"  ></td>
               <td ><input id= "radiobuttonUnknown_<%# Eval("Test_InternalID")%>" type="radio" name = "radiobutton_<%#
Eval("Test_InternalID")%>"  ></td>
           </tr>
         </ItemTemplate>


         <FooterTemplate>
          </tbody>
          </table>
         </FooterTemplate>

        </asp:Repeater>
      </div>


         </div>

           <div style =" margin-left:640px;">
         <asp:button   id ="buttonupdateTestStatuses"  runat ="server"  Text  ="update Tests Status"
                     OnClientClick  = "return buttonupdateTestStatuses();" width="130"
                 onclick="buttonupdateTestStatuses_Click" />
      </div>

      <div style =" margin-left:200px; width :500px;">
        <asp:Label ID="labelMessage" runat ="server" ForeColor ="White" Font-Size ="Large"  ></asp:Label>
      </div>

      <div id="hiddenFieldDiv"  style = "  display :none;"  >
         <asp:HiddenField ID="hiddenFieldFileData" Value = "" runat="server" />
      </div>

  </div>

</asp:Content>
```

Web Page: AddEvidences.aspx.cs [Code-behind]

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using TestSite.AFDIAWebServiceReference;


namespace TestSite
{
    public partial class _Default : System.Web.UI.Page
    {


        string selectedSessionId = "";

        protected void Page_Load(object sender, EventArgs e)
        {

            if (Request["SelectedSessionID"] != null)
                selectedSessionId = Request["SelectedSessionID"].ToString();
             bindDatawithGridfromDB();
        }

        private void bindDatawithGridfromDB()
        {
           /*call wcf service for get the list of all Sessions ID*/
           scAddEvidenceClient client = new scAddEvidenceClient();
           client.Open();
           var list = client.GetListOfTestsBySessionID(selectedSessionId);
           client.Close();
```

```
            /* bind that list with the DataGrid */
            List<TestBindableEntity> testBindAbleEntity = new List<TestBindableEntity>();


            for (int i = 0; i < list.ToList().Count; i++)
            {
                var item = new TestBindableEntity();
                item.ID_Test = list[i].ID_Test.ToString();
                item.Test_Name = list[i].Test_Name;
                item.Test_OutCome = list[i].Test_OutCome;
                item.Test_Probability = list[i].Test_Probability;
                item.ID_Session = list[i].ID_Session;
                item.Test_InternalID = list[i].Test_InternalID;

                testBindAbleEntity.Add(item);
            }

            repeaterTests.DataSource = testBindAbleEntity;
            repeaterTests.DataBind();

        }


        protected void linkbuttonViewTests_Click(object sender, EventArgs e)
        {
            Server.Transfer("AllPendingTests.aspx");
        }

        protected void buttonupdateTestStatuses_Click(object sender, EventArgs e)
        {

            try
            {
                /*call wcf service for set the evidence into the DB*/
                scAddEvidenceClient client = new scAddEvidenceClient();
                client.Open();
                client.AddEvidenceIntoDB(selectedSessionId, hiddenFieldFileData.Value.ToString());
                client.Close();

                Server.Transfer("AllPendingTests.aspx");
                labelMessage.Text = "Evidences has been updated in the Database! For Add More Evidence Please go to All Pending
tests page.";
            }
            catch (Exception exp)
            {
            }



        }
    }




class TestBindableEntity
    {
        private string _SessionID;
        public string ID_Session
        {
            get { return _SessionID; }
            set { _SessionID = value; }
        }

        private string _TestID;
        public string ID_Test
        {
            get { return _TestID; }
            set { _TestID = value; }
        }

        private string _TestName;
        public string Test_Name
        {
            get { return _TestName; }
            set { _TestName = value; }
        }


        private string _TestOutCome;
        public string Test_OutCome
        {
            get { return _TestOutCome; }
            set { _TestOutCome = value; }
        }


        private string _TestProbability;
        public string Test_Probability
        {
            get { return _TestProbability; }
            set { _TestProbability = value; }
        }

        private string _Test_InternalID;
        public string Test_InternalID
        {
            get { return _Test_InternalID; }
            set { _Test_InternalID = value; }
        }



    }


}
```

Project: AFDIAOnline

Web Page: Index.aspx [Markup]

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
   CodeBehind="Index.aspx.cs" Inherits="AFDPresentation._Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">

  <script type ="text/javascript" language ="javascript" >

      function linkbuttonUploadNewModel_OnClientClick() {

          alert("I am in linkbuttonUploadNewModel_OnClientClick Event");
          return false;

      }

      function linkbuttonChooseExistingModel_OnClientClick() {

          alert("I am in linkbuttonChooseExistingModel_OnClientClick Event");
          return false;
      }

  </script>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">

    <div id="divPage" style =" height:650px; width :100%; background-color :lightblue;   ">

        <div style = " color :White;  width:100%; text-align: center; ">
         <span style="color: #FFFFFF; font-size:large;  "> Configure Model (.BNE)</span></div>


        <div style = " margin-top :100px; height:100px; width:100%;">

           <div style =" margin-left :100px;  width:500px; ">
              <asp:LinkButton ID="linkbuttonUploadNewModel"  ForeColor ="White"

                  Font-Underline="false"  runat="server"
                  onclick="linkbuttonUploadNewModel_Click" > Upload New Model</asp:LinkButton>

           </div>

           <div style =" margin-left :100px;  width:500px; margin-top:20px;  ">
              <asp:LinkButton ID="linkbuttonChooseExistingModel"   ForeColor ="White"
                  OnClientClick="linkbuttonChooseExistingModel_OnClientClick"
                  Font-Underline="false" runat="server"
                  onclick="linkbuttonChooseExistingModel_Click">  Choose an Existing Model</asp:LinkButton>

           </div>
         </div>
    </div>
</asp:Content>
```

Web Page: Index.aspx.cs [Code-behind]

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace AFDPresentation
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            Session["HistoryFlag"] = "false";
        }

        protected void linkbuttonUploadNewModel_Click(object sender, EventArgs e)
        {

            Response.Redirect("UploadModel.aspx");

        }

        protected void linkbuttonChooseExistingModel_Click(object sender, EventArgs e)
        {

            Response.Redirect("AvailableBNEModels.aspx");

        }
    }
}
```

Web Page: UploadModel.aspx [Markup]

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeBehind="UploadModel.aspx.cs" Inherits="AFDPresentation.UploadModel" %>




<%@ Register Assembly="EeekSoft.Web.PopupWin" Namespace="EeekSoft.Web" TagPrefix="cc1" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">

 <link href="Styles/StyleSheet.css" type="text/css" rel="Stylesheet" />

<script type = "text/javascript" language = "javascript" >


    function ShowPopup() {
        alert("Upon Successful upload of the CTM Model, Tool will be transfered to Test Pages! ");
    }

</script>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">

    <div id="divPage" style =" height:650px; width :100%; background-color :lightblue;   ">
```

```
      <div style = " color :White;  width:100%; text-align: center; ">
       <span style="color: #FFFFFF; font-size:large;  "> Upload New Model (.BNE) </span></div>

        <div style = "  margin-left:910px;  height:34px; width:50px;">
             <asp:ImageButton ID="imageButtonHelpPopUp" runat="server"  Height = "50px"
                 Width = "50px" ImageUrl="~/Resorces/Images/help-Cir.jpg"
                 onclick="imageButtonHelpPopUp_Click"   />
        </div>



      <div style = " margin-top :100px; height:34px; width:100%;">

             <div id="uploadFileLabel" style =" margin-left:100px; width:250px; margin-top:2px;  float :left; " dir="rtl"  >
                 <asp:Label ID="labelUploadModel" runat="server" ForeColor ="White"  Text=": Locate Model inside your
Disk"></asp:Label>
                 </div>


             <div id="divUploadFileControl" style =" margin-left:5px; width:250px; float :left; "   >
                 <asp:FileUpload ID="fileUploadBNEModel" Width ="200"  runat="server" ForeColor ="White"  />
             </div>

            </div>

              <div id="divButtonUpload" style =" margin-left:500px; width:80px;  "   >
               <asp:Button ID="buttonUploadModel" runat ="server" Width="80" Text ="Upload"
                     onclick="buttonUploadModel_Click" />
              </div>
              <div id="div" style =" margin-left:200px; width:500px;  "   >
                 <asp:Label ID="LabelFileUploadstatus" runat="server"></asp:Label>
              </div>

                 <div id="div1" style =" margin-left:500px; width:80px;  "   >
               <asp:Button ID="buttonInitilize"  Visible = "false"  runat ="server" Width="80" Text ="Init" OnClientClick =
"ShowPopup()" onclick="buttonInitilize_Click"
                     />
              </div>

               <div>
                <cc1:PopupWin id="popupWin" runat="server" visible="False"
                    colorstyle="Blue" width="300px" height="150px"  DragDrop ="true" DockMode ="BottomRight"
                    windowscroll="False" windowsize="300, 200" style="top: 0px; left: 0px" />
                 </div>



   </div>
</asp:Content>
```

Web Page: UploadModel.aspx.cs [Code-behind]

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Data.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;
using Domain;
using AFDPresentation.Dlls;
using Parser;




namespace AFDPresentation
{
    public partial class UploadModel : System.Web.UI.Page
    {

        private AFDOnlineDBDataContext _AfdDataContext;
        private AFDOnlineDBDataContext  AfdDataContext
        {
            get
            {
                return _AfdDataContext;
            }
            set
            {
                _AfdDataContext = value;
            }
        }


        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                setupPopUp();
            }

        }

       public UploadModel()
        {
            _AfdDataContext = new AFDOnlineDBDataContext();
        }

        protected void buttonUploadModel_Click(object sender, EventArgs e)
        {
            if (fileUploadBNEModel.HasFile)
            {
                try
                {
                    string filename = Path.GetFileName(fileUploadBNEModel.FileName);
                    fileUploadBNEModel.SaveAs(Server.MapPath("~/UploadedModels/" + filename ));
                    LabelFileUploadstatus.Text = "Upload Status: File Uploaded! Please Press Init button Below to Initliza the
AFDTool";



                    Model  newModel = new Model();

                    var modelRecords = from models in AfdDataContext.Models select models;
                    if (modelRecords.Count() > 0)
                    {
                       var previouscount = (from moels in AfdDataContext.Models select moels.ID_Model).Count();
                       var currentcount = previouscount + 1000;
                       currentcount = currentcount + 1;
                       newModel.ID_Model = "Mod_" + Convert.ToString(currentcount);
                       newModel.ID_Board = (from bords in AfdDataContext.Boards
                                   select bords.ID_Board).First().ToString() ;
                    newModel.CTM_SourceFile = "~/UploadedModels/" + filename;
                    }

                    else
                    {
                     newModel.ID_Model = "Mod_1001";
                     newModel.ID_Board = (from bords in AfdDataContext.Boards
                                   select bords.ID_Board).First().ToString() ;
                    newModel.CTM_SourceFile = "~/UploadedModels/" + filename;

                    }


                   var filepath =
"C:\\MyDisk\\Data\\thesis\\AFDOnline_Develop\\AFDOnline\\AFDPresentation\\UploadedModels\\"+ filename;

                   newModel.Model_Content = "";//ReadContentFromUploadedFile(filepath);
                   newModel.Model_Name = filename;
                    AfdDataContext.Models.InsertOnSubmit(newModel);




                    // Insert a new record into a Session Table and Save the session Id in ASP.Net Seesion variable for others
references

                    Sesion newSession = new Sesion();


                    var sessionRecords = from sessions in AfdDataContext.Sesions select sessions;

                    if (sessionRecords.Count() > 0)
                    {
                       var previouscount = (from sessions in AfdDataContext.Sesions select sessions.ID_Session).Count();
                       var currentcount = previouscount + 1;

                       newSession.ID_Session = "Session_" + Convert.ToString(currentcount);
                       newSession.ID_Model = newModel.ID_Model;
                       newSession.Session_Date = DateTime.Now;
                    }

                    else
                    {
                       newSession.ID_Session = "Session_1";
                       newSession.ID_Model  = newModel.ID_Model;
                       newSession.Session_Date = DateTime.Now;

                    }

                    // save Seesion Id and Model Id into Seesion Variables
                    Session["SessionID"] = newSession.ID_Session;
                    Session["ModelID"] = newModel.ID_Model;


                    // submit changes to the Server
                    AfdDataContext.Sesions.InsertOnSubmit(newSession);


                    // Submit Chages into the DataBase
                    AfdDataContext.SubmitChanges();



                    //MoveFileToCDrive(newModel.Model_Content, filename);
                    //SetFilePathIntoInitFile(filename);
                    buttonInitilize.Visible = true;
                }
                catch (Exception ex)
                {
                    LabelFileUploadstatus.Text = "Upload Status: File Cannot be Uploaded: The Following Exception Occur" +
ex.Message;
                }
            }
        }

        private void  SetFilePathIntoInitFile(string filename)
         {

              TextWriter tsw = new StreamWriter(@"C:\TempAFDTool\AFD_BBN_Library_Init_input.txt");
              tsw.Flush();
              tsw.WriteLine("C:\\TempAFDTool\\Model\\" + filename + ",");
              tsw.Close();
         }


        private void MoveFileToCDrive(string fileModelContent, string fileName)
        {


            StreamWriter sw;
            sw = File.CreateText("C:\\TempAFDTool\\Model\\" + fileName);
            sw.WriteLine(fileModelContent.Trim());
            sw.Close();

        }

        protected void buttonInitilize_Click(object sender, EventArgs e)
        {

            UnsafeNativeMethods.AFD_BBN_Library_Init();
            var parser = new fileParser("C:\\TempAFDTool\\AFD_BBN_Library_Init_output.txt");
            var outDto =  parser.ParseInitOutFile();

            if (!outDto.result)
            {
                LabelFileUploadstatus.Text = outDto.ErrorMessage;
            }
            else
            {
                Server.Transfer("Test.aspx");
            }

        }

        private string  ReadContentFromUploadedFile(string filPath)
        {

            var fileContent = "";

            try
            {

                // Create an instance of StreamReader to read from a file.
                // The using statement also closes the StreamReader.
                using (StreamReader sr = new StreamReader(filPath))
                {
                    String line;
                    // Read and display lines from the file until the end of
                    // the file is reached.
                    while ((line = sr.ReadLine()) != null)
                    {
                        fileContent = fileContent + line;
                    }
                }

            }
            catch (Exception exp)
            {
                LabelFileUploadstatus.Text = "Opps! System Counter Problem while Reading Content of CTM Model file!";
            }

            return fileContent;
        }

        protected void imageButtonHelpPopUp_Click(object sender, ImageClickEventArgs e)
        {
            setupPopUp();
        }

        private void setupPopUp()
        {
            popupWin.ActionType = EeekSoft.Web.PopupAction.MessageWindow;




            //Set popup and window texts
            popupWin.Title = "Instructions for Instilization";
            popupWin.Message = "</br></br></br><b>You are At Initilization Phase of AFD Tool...Click Me To know More!!!</b>";
            popupWin.Text = " </br> - Upload  a CTM Model </br> - Upon Successful upload, Init Button will be Visible </br> -
Press Init Button </br> - it will Initilize AFD Tool";

            //Change color style
            popupWin.ColorStyle = EeekSoft.Web.PopupColorStyle.Blue;



            //' Change timing
            popupWin.HideAfter = 20000;
            popupWin.ShowAfter = 500;
```

```
            //Show popup (after page is loaded)
            popupWin.Visible = true;
        }
    }
}
```

Web Page: Test.aspx [Markup]

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeBehind="Test.aspx.cs" Inherits="AFDPresentation.Test" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">

<link rel="Stylesheet" href ="DataTables/css/demo_table.css" type ="text/css" />
    <script type ="text/javascript" src="DataTables/js/jquery.js"></script>
    <script type ="text/javascript" src="DataTables/js/jquery.dataTables.js"></script>

    <script type ="text/javascript">

        $(document).ready(function () {

            $('#tblAllTests').dataTable({
                "oLanguage": { "oView": "Select a Model(.BNE)" },
                "iDisplyLength": 10,
                "aaSorting": [[0, "asc"]],

                "sAjaxSource": "../../JsonDataFiles/jdTest.txt"

            });
        });

    </script>
    <style type="text/css">
        .style3
        {
            color: #FFFFFF;
        }
    </style>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">

    <div id="divPage" style =" height:650px; width :100%; background-color :lightblue;   ">

        <div style = " color :White;  width:100%; text-align: center; ">
         <span style="color: #FFFFFF; font-size:large;  ">  Available Tests </span></div>

         <div style = " width : 100%">
            <div style = " margin-left :50px; width:100px; float:left;">
                <asp:LinkButton ID="linkbuttonViewStrategies" runat="server"
                    onclick="linkbuttonViewStrategies_Click">View Strategies</asp:LinkButton>
            </div>

            <div style = " margin-left :5px; width:115px; float :left;">
             <asp:LinkButton ID="linkbuttonViewComponents" runat="server"
                    onclick="linkbuttonViewComponents_Click">View Components</asp:LinkButton>
            </div>
```

```
          <div style = " margin-left :5px; width:115px; float :left;">
         <asp:LinkButton ID="linkbuttonaddEvidence" runat="server" onclick="linkbuttonaddEvidence_Click"
              >Add Evidence</asp:LinkButton>
        </div>

          <div style = " margin-left :5px; width:152px; float :left;">
         <asp:LinkButton ID="linkbuttonRemoveEvidence" runat="server" onclick="linkbuttonRemoveEvidence_Click"
              >Clean/Remove Evidence</asp:LinkButton>
        </div>

          <div style = " margin-left :5px; width:152px; float :left;">
         <asp:LinkButton ID="linkbuttonViewHistory" runat="server" onclick="linkbuttonViewHistory_Click"
              >Test History</asp:LinkButton>
        </div>
      </div>


     <div style =" margin-top:100px; ">

          <div style = " border: thin solid #FFFFFF; margin-left :100px; float:left ;  height:auto; width:500px;  text-align :center;"
>




            <table id="tblAllTests" width ="490" cellpadding ="0" cellspacing ="0" border ="0" class = "disply">
              <thead>
                <tr>
                   <th >TestID</th>

                   <th style =" width:100px;" >Test OutCome</th>
                   <th >Test Probability</th>
                 </tr>
              </thead>
            <tbody>



          <tr>
              <td></td>

              <td> </td>
              <td> </td>
           </tr>





          </tbody>
          </table>





      </div>

              <div style =" float:left; height: 313px;  ">
     <div style = "   border: thin solid #FFFFFF; margin-left :10px;  height:150px; width:300px;  text-align :center;" >
```

```
                <div style =" width:300px; border-bottom-style: solid; border-bottom-width: thin; border-bottom-color:
#FFFFFF;"
            class="style3">
            Trash Parameters</div>

                 <div style =" width:300px;  margin-top :10px;  ">
                  <div style="width: 100px; margin-top:5px;   float: left;">
                     <asp:Label ID="Label1" runat="server" Text="Pass Trash:" Width="69px"></asp:Label>
                  </div>
                  <div style=" margin-top:5px; width: 168px; float:left;  ">
                     <asp:TextBox ID="textboxPassTrash" runat="server" Width="150px" ></asp:TextBox></div>
                 </div>

                 <div style =" margin-top:15px; ">
                  <div style="float: left; width: 100px;  float:left; ">
                     <asp:Label ID="Label2" runat="server" Text="Failure Trash"></asp:Label></div>
                  <div style="   width: 168px; float:left;  ">
                     <asp:TextBox ID="textboxFailureTrash" runat="server" Width="150px"></asp:TextBox></div>

                 </div>

               <div id="divButtonApply" style =" margin-left :170px; margin-top :10px;">
                <asp:Button ID="buttonApplyTrash" Text ="Apply" runat ="server" />

               </div>




        </div>

        <div style = "  border: thin solid #FFFFFF; margin-left :10px; margin-top :5px; height:50px; width:300px;  text-align
:center;" >

          <div style =" width:300px; border-bottom-style: solid; border-bottom-width: thin; border-bottom-color: #FFFFFF;"
            class="style3">
            Information & Status </div>


                    <div style=" width: 150px; margin-top:2px; float:left;    ">
                     <asp:Label ID="labelLatestStepPerformed" ForeColor ="White" runat ="server" Text ="Latest Step
Performed" ></asp:Label>
                    </div>

                     <div style=" width: 100px;  margin-top:2px; float:left;    ">
                      <asp:TextBox ID="textboxLatestStepPerformed" ReadOnly ="true"  runat ="server" Width =
"129px"></asp:TextBox>
                     </div>



        </div>
      </div>

        </div>

    </div>
</asp:Content>
```

Web Page: Test.aspx.cs [Code-behind]

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using AFDPresentation.Dlls;
using Parser;
using Domain;
using Domain.Entities;
using System.IO;
using AFDPresentation.AddEvidenceService;


namespace AFDPresentation
{
    public partial class Test : System.Web.UI.Page
    {
        private AFDOnlineDBDataContext _AfdDataContext;
        private AFDOnlineDBDataContext AfdDataContext
        {
            get
            {
                return _AfdDataContext;
            }
            set
            {
                _AfdDataContext = value;
            }
        }

        public Test()
        {
            _AfdDataContext = new AFDOnlineDBDataContext();
        }

        String SessionID = "";
        protected void Page_Load(object sender, EventArgs e)
        {

            if (!IsPostBack)
            {
                if (Session["HistoryFlag"].ToString() == "true")
                {
                    SessionID = Session["SessionID"].ToString();
                    GetDataFromDBAndBind();
                }
                else if (Request["modelID"] != null)
                {
                    var selectedModelID = Request["modelID"].ToString();
                    CallWrapperMethod(selectedModelID);
                }
               else
            {
                CallWrapperMethod();
            }
            }
        }

        private void GetDataFromDBAndBind()
        {




          var results = (from tests in AfdDataContext.TestResults
                  where tests.ID_Session == SessionID
                  select tests).ToList();


          List<TestOutcome> list = new List<TestOutcome>(results.Count);

          for (int i = 0; i < results.Count; i++)
          {
            TestOutcome item = new TestOutcome();
            item.RowID = i.ToString();
            item.TestID = results[i].ID_Test.ToString() ;
            item.TestName = results[i].Test_Name;
            item.TestOutCome = results[i].Test_OutCome;
            item.TestProbability = results[i].Test_Probability;

            list.Add(item);

          }


          FillJson(list);

        }


        private void CallWrapperMethod(string modelID)
        {
          //SetFilePathOfCTmModelIntoInitFile( (from models in  AfdDataContext.Models where models.ID_Model ==  modelID
select models.Model_Name).ToString()   );


          UnsafeNativeMethods.AFD_BBN_Library_Init();


          UnsafeNativeMethods.AFD_BBN_Library_getTests_All();
          var parser = new fileParser("C:\\TempAFDTool\\AFD_BBN_Library_getTests_All_output.txt");
          var result = parser.parseTest();

          FillJson(result);
          InsertDataIntoDB(result);

        }
        private void CallWrapperMethod()
        {

          UnsafeNativeMethods.AFD_BBN_Library_getTests_All();
          var parser = new fileParser("C:\\TempAFDTool\\AFD_BBN_Library_getTests_All_output.txt");
          var result = parser.parseTest();
```

```
            FillJson(result);
            InsertDataIntoDB(result);

        }

        private void SetFilePathOfCTmModelIntoInitFile(string modelFileName)
        {
            TextWriter tsw = new StreamWriter(@"C:\TempAFDTool\AFD_BBN_Library_Init_input.txt");
            tsw.Flush();
            tsw.WriteLine("C:/TempAFDTool/Model/".Replace('/','\\') + modelFileName + ",");
            tsw.Close();
        }


        private void FillJson(List<Domain.Entities.TestOutcome> result)
        {
            string dataString = "";

            for (int i = 0; i < result.Count; i++)
            {
                var TestID = "\"" + result[i].TestID + "\" ";

                var testoutcome = "\"" + result[i].TestOutCome + "\" ";
                var testProbabality = "\"" + result[i].TestProbability + "\" ";
                dataString += "[" + TestID + "," + testoutcome + "," + testProbabality + "],";


            }

            String line;
            using (StreamReader sr = new
    StreamReader("D:\\MyDisk\\Data\\thesis\\AFDOnline_Develop\\AFDOnline\\AFDPresentation\\JsonDataFiles\\jtdata.txt"))
            {
                line = sr.ReadToEnd();
                sr.Close();
            }


            string res = line.Replace("{0}", result.Count.ToString()).Replace("{1}", result.Count.ToString());
            string finaldata = res.Replace("{2}", dataString.Remove(dataString.Length - 1));

            using (System.IO.StreamWriter file = new
    System.IO.StreamWriter("D:\\MyDisk\\Data\\thesis\\AFDOnline_Develop\\AFDOnline\\AFDPresentation\\JsonDataFiles\\jdT
    est.txt"))
            {
                file.WriteLine(finaldata);
                file.Close();
            }



        }


        private void InsertDataIntoDB(List<TestOutcome>  result )
        {
              // insert data into Test Table

            // declare data contract for sending to web server
            dcAddEvidence[] listTest = new dcAddEvidence[result.ToList().Count];
            Domain.TestResult test;
            for (int i = 0; i < result.Count; i++)
            {
                test = new Domain.TestResult();
                test.Test_InternalID = i.ToString();
                test.ID_Session = Session["SessionID"].ToString();
                test.ID_Model = Session["ModelID"].ToString();
                test.Test_Name = result[i].TestName;
                test.Test_OutCome = result[i].TestOutCome;
                test.Test_Probability = result[i].TestProbability;

                listTest[i] = new dcAddEvidence();
                listTest[i].TetsInternalID  = i.ToString();
                listTest[i].SessionID = Session["SessionID"].ToString();
                listTest[i].ModelID = Session["ModelID"].ToString();
                listTest[i].TetstName =  result[i].TestName;
                listTest[i].TestOutCome = result[i].TestOutCome;
                listTest[i].TestProbability = result[i].TestProbability;
                listTest[i].TestID  = i;



                AfdDataContext.TestResults.InsertOnSubmit(test);

            }


            AfdDataContext.SubmitChanges();

            // send all the Test output data to Test Admininstrtor




            scAddEvidenceClient client = new scAddEvidenceClient();
            client.Open();
            client.SendTestToAdministrator(listTest);

            client.Close();



        }

        protected void linkbuttonViewStrategies_Click(object sender, EventArgs e)
        {
            Server.Transfer("Strategy.aspx");
        }

        protected void linkbuttonViewComponents_Click(object sender, EventArgs e)
        {
            Server.Transfer("ComponentView.aspx");
        }

        protected void linkbuttonaddEvidence_Click(object sender, EventArgs e)
        {
            Server.Transfer("AddEvidence.aspx");
        }
```

```
        protected void linkbuttonRemoveEvidence_Click(object sender, EventArgs e)
        {
            Server.Transfer("Remove_CleanEvidence.aspx");
        }

        protected void linkbuttonViewHistory_Click(object sender, EventArgs e)
        {
            Server.Transfer("History.aspx");
        }


    }
}
```

Web Page: Strategy.aspx [Markup]

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeBehind="Strategy.aspx.cs" Inherits="AFDPresentation.Strategy" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">

<link rel="Stylesheet" href ="DataTables/css/demo_table.css" type ="text/css" />
    <script type ="text/javascript" src="DataTables/js/jquery.js"></script>
    <script type ="text/javascript" src="DataTables/js/jquery.dataTables.js"></script>

    <script type ="text/javascript">

        $(document).ready(function () {

            $('#tblStrategies').dataTable({
                "oLanguage": { "oView": "Select a Model(.BNE)" },
                "iDisplyLength": 10,
                "aaSorting": [[0, "asc"]],
                "sAjaxSource": "../../JsonDataFiles/jdStrategies.txt"
            });
        });

    </script>
    <style type="text/css">
        .style3
        {
            color: #FFFFFF;
        }
    </style>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">

    <div id="divPage" style =" height:650px; width :100%; background-color :lightblue;   ">

        <div style = " color :White;  width:100%; text-align: center; ">
         <span style="color: #FFFFFF; font-size:large;  "> Strategies </span></div>


          <div style = " width : 100%">
            <div style = " margin-left :50px; width:100px; float:left;">
                <asp:LinkButton ID="linkbuttonViewTest" runat="server" onclick="linkbuttonViewTest_Click"
                    >View Tests</asp:LinkButton>
            </div>
        <div style = " margin-left :5px; width:115px; float :left;">
         <asp:LinkButton ID="linkbuttonViewComponents" runat="server" onclick="linkbuttonViewComponents_Click"
             >View Components</asp:LinkButton>
        </div>


         <div style = " margin-left :5px; width:115px; float :left;">
         <asp:LinkButton ID="linkbuttonaddEvidence" runat="server" onclick="linkbuttonaddEvidence_Click"
             >Add Evidence</asp:LinkButton>
        </div>

         <div style = " margin-left :5px; width:152px; float :left;">
         <asp:LinkButton ID="linkbuttonRemoveEvidence" runat="server" onclick="linkbuttonRemoveEvidence_Click"
             >Clean/Remove Evidence</asp:LinkButton>
        </div>

          <div style = " margin-left :5px; width:152px; float :left;">
         <asp:LinkButton ID="linkbuttonViewHistory" runat="server" onclick="linkbuttonViewHistory_Click"
             >Test History</asp:LinkButton>
        </div>
      </div>

    <div style = " border: thin solid #FFFFFF; margin-left :100px; margin-top :100px; float:left ;  height:auto; width:500px;  text-
align :center;" >


          <table id="tblStrategies" width ="490" cellpadding ="0" cellspacing ="0" border ="0" class = "disply">
           <thead>
             <tr>
                <th style =" width:50px;">TSNum</th>
                <th style =" width:200px;" >Quality Metric Value</th>


              </tr>
           </thead>
          <tbody>





         <tr>
             <td> </td>
             <td> </td>

          </tr>





         </tbody>
         </table>




      </div>
      <div style = " float :left;  border: thin solid #FFFFFF; margin-left :10px; margin-top :100px; height:150px; width:300px;
text-align :center;" >

         <div style =" width:300px; border-bottom-style: solid; border-bottom-width: thin; border-bottom-color: #FFFFFF;"
           class="style3">
           Select Strategy </div>

                <div style =" width:300px;  margin-top :10px;  ">
                  <div style="width: 100px; margin-top:5px;   float: left;">
                    <asp:Label ID="Label1" runat="server" ForeColor ="White"  Text="Strategies:"
Width="69px"></asp:Label>
                  </div>
                  <div style=" margin-top:5px; width: 168px; float:left;  ">
                    <asp:DropDownList ID="dropdownlistStrategies" runat="server" Width="168px">

                        <asp:ListItem Value ="Single">Single</asp:ListItem>
                        <asp:ListItem Value ="Double">Double</asp:ListItem>
                        <asp:ListItem Value ="K-Step">K-Step</asp:ListItem>

                    </asp:DropDownList>
                 </div>

             </div>

               <div id="divButtonApply" style =" margin-left :2px; margin-top :100px; text-align:left;  ">
             <asp:Button ID="buttonAddStrategy" Text ="Add"  Width = "50" runat ="server" />

             </div>
   </div>
</asp:Content>
```

Web Page: Strategy.aspx.cs [Code-behind]

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Domain.Entities;
using AFDPresentation.Dlls;
using Parser;
using System.IO;
using Domain;

namespace AFDPresentation
{
    public partial class Strategy : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            CallWrapperMethod();
        }

        private void CallWrapperMethod()
        {
            UnsafeNativeMethods.AFD_BBN_Library_getStrategies_All();
            var parser = new fileParser("C:\\TempAFDTool\\AFD_BBN_Library_getStrategies_All_output.txt");
            var result = parser.parseStrategy();
            string dataString = "";

            for (int i = 0; i < result.Count; i++)
            {
                var Tsum = "\"" + result[i].TSNum + "\" ";
                var qualityMatric = "\"" + result[i].QualityMatric  + "\" ";

                dataString += "[" + Tsum + "," + qualityMatric + "],";


            }

            String line;
            using (StreamReader sr = new
StreamReader("C:\\MyDisk\\Data\\thesis\\AFDOnline_Develop\\AFDOnline\\AFDPresentation\\JsonDataFiles\\jtdata.txt"))
            {
                line = sr.ReadToEnd();
                sr.Close();
            }


            string res = line.Replace("{0}", result.Count.ToString()).Replace("{1}", result.Count.ToString());
            string finaldata = res.Replace("{2}", dataString.Remove(dataString.Length - 1));

            using (System.IO.StreamWriter file = new
System.IO.StreamWriter("C:\\MyDisk\\Data\\thesis\\AFDOnline_Develop\\AFDOnline\\AFDPresentation\\JsonDataFiles\\jdSt
rategies.txt"))
            {
                file.WriteLine(finaldata);
                file.Close();
            }


             AFDOnlineDBDataContext dc = new AFDOnlineDBDataContext();

            StrategyList strategy;

              for (int i = 0; i < result.Count; i++)
              {
                  strategy = new StrategyList();
                  strategy.TSNum  = result[i].TSNum ;
                  strategy.ID_Session = Session["SessionID"].ToString ();
                  strategy.QualityMatric  =  result[i].QualityMatric ;

                  dc.StrategyLists.InsertOnSubmit(strategy);

              }


              dc.SubmitChanges();


        }


        protected void linkbuttonViewTest_Click(object sender, EventArgs e)
        {
            Server.Transfer("Test.aspx");
        }
        protected void linkbuttonViewComponents_Click(object sender, EventArgs e)
        {
            Server.Transfer("ComponentView.aspx");
        }

        protected void linkbuttonaddEvidence_Click(object sender, EventArgs e)
        {
            Server.Transfer("AddEvidence.aspx");
        }

        protected void linkbuttonRemoveEvidence_Click(object sender, EventArgs e)
        {
            Server.Transfer("Remove_CleanEvidence.aspx");
        }

        protected void linkbuttonViewHistory_Click(object sender, EventArgs e)
        {
            Server.Transfer("History.aspx");
        }



    }
}
```

Web Page: Remove_CleanEvidenc.aspx [Markup]

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeBehind="Remove_CleanEvidence.aspx.cs" Inherits="AFDPresentation.Remove_CleanEvidence" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">

   <link rel="Stylesheet" href ="DataTables/css/demo_table.css" type ="text/css" />
   <script type ="text/javascript" src="DataTables/js/jquery.js"></script>
   <script type ="text/javascript" src="DataTables/js/jquery.dataTables.js"></script>

    <script type ="text/javascript">

     var FileInputString = "";

        $(document).ready(function () {

            $('#tblTests').dataTable({
                "oLanguage": { "oView": "Select a Tests" },
                "iDisplyLength": 10,
                "aaSorting": [[0, "asc"]]
            });
        });


        function onClientClick_Remove() {


            var tab = document.getElementById("tblTests"); // table with id tbl1
            var elems = tab.getElementsByTagName("input");
            var len = elems.length;
        for (var i = 0; i < len; i++) {

            var chkBoxId = "testCheckbox_" + i;
            var objTestCheckBoxID = document.getElementById(chkBoxId.toString());


            if (objTestCheckBoxID.checked.toString() == "true") {


                var testIDKey = "testid_" + i;
                var testResultOutComeKey = "testoutcome_" + i;

                var objtestResultOutCome = document.getElementById(testResultOutComeKey.toString());


                var objtestID = document.getElementById(testIDKey.toString());



                var res = objtestResultOutCome.innerHTML.toString();


                if (res == "PASS") {
                    FileInputString +=   objtestID.innerHTML.toString().trim() + ",";
                }
                else {
                    FileInputString +=  objtestID.innerHTML.toString().trim() + ",";
                }

                var HiddenValue = document.getElementById("<%= hiddenFieldFileData.ClientID %>");
                HiddenValue.value = FileInputString;

            }


        }

        return true;
    }

    </script>
    </asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">

    <div id="divPage" style =" height:650px; width :100%; background-color :lightblue;   ">

        <div style = " color :White;  width:100%; text-align: center; ">
         <span style="color: #FFFFFF; font-size:large;  "> Remove/Clean Evidence(s) </span></div>




         <div style = " width : 100%">

           <div style = " margin-left :50px; width:88px; float :left;">
            <asp:LinkButton ID="linkbuttonViewTests" runat="server" onclick="linkbuttonViewTests_Click"
                >View Tests</asp:LinkButton>
           </div>
           <div style = " margin-left :5px; width:100px; float:left;">
            <asp:LinkButton ID="linkbuttonViewStrategies" runat="server" onclick="linkbuttonViewStrategies_Click"
                >View Strategies</asp:LinkButton>
           </div>

           <div style = " margin-left :5px; width:115px; float :left;">
            <asp:LinkButton ID="linkbuttonViewComponents" runat="server" onclick="linkbuttonViewComponents_Click"
                >View Components</asp:LinkButton>
           </div>




            <div style = " margin-left :5px; width:152px; float :left;">
            <asp:LinkButton ID="linkbuttonAddEvidence" runat="server" onclick="linkbuttonAddEvidence_Click"
                >Add Evidence</asp:LinkButton>
           </div>

            <div style = " margin-left :5px; width:152px; float :left;">
            <asp:LinkButton ID="linkbuttonViewHistory" runat="server" onclick="linkbuttonViewHistory_Click"
                >Test History</asp:LinkButton>
           </div>
         </div>



         <div style =" margin-top:100px; ">



         <div style = " border: thin solid #FFFFFF; margin-left :200px;  float:left ; margin-top :25px;  height:auto; width:500px;
text-align :center;" >

           <asp:Repeater ID="repeaterTests" runat ="server">

            <HeaderTemplate>
            <table id="tblTests" cellpadding ="0" width ="490" cellspacing ="0" border ="0" class = "disply">
                <thead>
                <tr>
                    <th style =" width:50px;" >Test ID</th>
                    <th style =" width:100px;" >Test Name</th>
                    <th style =" width:100px;" >Test Approp Probability</th>
                    <th >Test Probability</th>
                    <th ><input type="checkbox" name="selected_test" id='checkboxTestSelectionHeader'> </th>
                </tr>
                </thead>
            <tbody>
            </HeaderTemplate>


            <ItemTemplate>
            <tr>
                <td id="testid_ <%# Eval("RowID")%>" > <%# Eval("TestID")%> </td>
                <td id="testname_ <%# Eval("RowID")%>"> <%# Eval("TestName")%></td>
                <td id="testoutcome_ <%# Eval("RowID")%>"> <%# Eval("TestOutCome")%> </td>
                <td id="testProbability_ <%# Eval("RowID")%>"> <%# Eval("TestProbability")%></td>
                <td ><input id= "testCheckbox_<%# Eval("RowID")%>" type="checkbox" name="selected_test" ></td>
            </tr>
            </ItemTemplate>
            <FooterTemplate>
            </tbody>
            </table>
            </FooterTemplate>

           </asp:Repeater>

         </div>


            </div>

            <div style =" margin-left:640px; width:50px;  ">
            <asp:Button ID="buttonRemove" runat="server" Text="Remove"
                  onclick="buttonRemove_Click"  OnClientClick = "javascript:onClientClick_Remove();"/>
         </div>

           <div id="hiddenFieldDiv"  style = "  display :none;"  >
            <asp:HiddenField ID="hiddenFieldFileData" Value = "" runat="server" />
         </div>
       </div>
</asp:Content>
```

Web Page: Remove_CleanEvidence.aspx.cs [Code-behind]

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Domain.Entities;
using AFDPresentation.Dlls;
using Parser;

namespace AFDPresentation
{
    public partial class Remove_CleanEvidence : System.Web.UI.Page
    {
        string hiddenFiledContent = "";
        protected void Page_Load(object sender, EventArgs e)
        {
            if (IsPostBack)
            {
                hiddenFiledContent = hiddenFieldFileData.Value;

            }
            BindGrid();
        }

        private void BindGrid()
        {

            UnsafeNativeMethods.AFD_BBN_Library_getTests_All();
            var parser = new fileParser("C:\\TempAFDTool\\AFD_BBN_Library_getTests_All_output.txt");
            var result = parser.parseTest();
            repeaterTests.DataSource = result.ToList();
            repeaterTests.DataBind();

        }

        protected void linkbuttonViewTests_Click(object sender, EventArgs e)
        {
            Server.Transfer("Test.aspx");
        }

        protected void linkbuttonViewStrategies_Click(object sender, EventArgs e)
        {
            Server.Transfer("Strategy.aspx");
        }

        protected void linkbuttonViewComponents_Click(object sender, EventArgs e)
        {
            Server.Transfer("ComponentView.aspx");
        }

        protected void linkbuttonAddEvidence_Click(object sender, EventArgs e)
        {
            Server.Transfer("AddEvidence.aspx");
        }

        protected void linkbuttonViewHistory_Click(object sender, EventArgs e)
        {
            Server.Transfer("History.aspx");
        }


        protected void buttonRemove_Click(object sender, EventArgs e)
        {
            // Write the string to a file.
            System.IO.StreamWriter file = new
System.IO.StreamWriter("C:\\TempAFDTool\\AFD_BBN_Library_RemoveEvidences_input.txt");
            file.WriteLine(hiddenFiledContent);
            file.Close();

            // call the AFD_LIBERARY.DLL
            //UnsafeNativeMethods.AFD_BBN_Library_removeEvidences();

        }
    }
}
```

Web Page: History.aspx.aspx [Markup]

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeBehind="History.aspx.cs" Inherits="AFDPresentation.History" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">

<link rel="Stylesheet" href ="DataTables/css/demo_table.css" type ="text/css" />
    <script type ="text/javascript" src="DataTables/js/jquery.js"></script>
    <script type ="text/javascript" src="DataTables/js/jquery.dataTables.js"></script>

    <script type ="text/javascript">
```

```
        $(document).ready(function () {

            $('#tblHistory').dataTable({
                "oLanguage": { "oView": "Select a Model(.BNE)" },
                "iDisplyLength": 10,
                "aaSorting": [[0, "asc"]],



            });
        });


        function showHistory(obj) {


          var HiddenValue = document.getElementById("<%= hiddenFieldFileData.ClientID %>");
          HiddenValue.value = obj.ch.toString();

         var answer = confirm ("Do you want to explore the Test Session with SessionID = [" + HiddenValue.value + "]");

         if (answer.toString() == "true")
           {

            __doPostBack();
          }



        }



    </script>
    <style type="text/css">
        .style3
        {
            color: #FFFFFF;
        }
    </style>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">

    <div id="divPage" style =" height:650px; width :100%; background-color :lightblue;   ">

        <div style = " color :White;  width:100%; text-align: center; ">
         <span style="color: #FFFFFF; font-size:large;  "> History </span></div>


         <div style = " width : 100%">
            <div style = " margin-left :50px; width:100px; float:left;">
                <asp:LinkButton ID="linkbuttonViewStrategies" runat="server"
                    onclick="linkbuttonViewStrategies_Click">View Strategies</asp:LinkButton>
            </div>

              <div style = " margin-left :50px; width:100px; float:left;">
                <asp:LinkButton ID="linkbuttonViewTest" runat="server" onclick="linkbuttonViewTest_Click"
                    >View Tests</asp:LinkButton>
            </div>
          <div style = " margin-left :5px; width:115px; float :left;">
           <asp:LinkButton ID="linkbuttonViewComponents" runat="server"
                onclick="linkbuttonViewComponents_Click">View Components</asp:LinkButton>
          </div>

           <div style = " margin-left :5px; width:115px; float :left;">
           <asp:LinkButton ID="linkbuttonaddEvidence" runat="server" onclick="linkbuttonaddEvidence_Click"
                >Add Evidence</asp:LinkButton>
          </div>



            <div style = " margin-left :5px; width:152px; float :left;">
           <asp:LinkButton ID="linkbutton1" runat="server" onclick="linkbuttonRemoveEvidence_Click"
                >Clean/Remove Evidence</asp:LinkButton>
          </div>
        </div>




       <div style = " margin-top :100px;">


        <div style = " border: thin solid #FFFFFF; margin-left :200px;  float:left ; margin-top :25px;  height:auto; width:500px;
text-align :center;" >

         <asp:Repeater ID="repeaterHistory" runat ="server">

           <HeaderTemplate>
             <table id="tblHistory" cellpadding ="0" width ="490" cellspacing ="0" border ="0" class = "disply">
              <thead>
               <tr>
                 <th style =" width:50px;" >Session ID</th>
                 <th style =" width:100px;" >Model ID</th>
                 <th style =" width:100px;" >Time Stamp </th>
               </tr>
              </thead>
             <tbody>
           </HeaderTemplate>


           <ItemTemplate>
            <tr  onclick = "javascript:showHistory(this)"; char ="<%# Eval("SessionID")%>"  >
                <td> <%# Eval("SessionID")%> </td>
                <td> <%# Eval("ModelID")%></td>
                <td> <%# Eval("SessionDateTime")%> </td>

            </tr>
           </ItemTemplate>



           <FooterTemplate>
            </tbody>
            </table>
           </FooterTemplate>
         </asp:Repeater>

        </div>

        </div>

         <div id="hiddenFieldDiv"  style = "  display :none;"  >
           <asp:HiddenField ID="hiddenFieldFileData" Value = "" runat="server" />
        </div>


    </div>
</asp:Content>
```

Web Page: History.aspx.aspx.cs [Code-behind]

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Domain.Entities;
using Domain;
using System.IO;
using AFDPresentation.Dlls;
using Parser;

namespace AFDPresentation
{
    public partial class History : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

            if (IsPostBack)
            {
              Session["SessionID"] = hiddenFieldFileData.Value;
               Session["HistoryFlag"] = "true";
               Server.Transfer("Test.aspx");
            }
            bindGrid();

        }

        private void bindGrid()
        {
            AFDOnlineDBDataContext dc = new AFDOnlineDBDataContext();

            var sessionList = (from sessionlist in dc.Sesions select sessionlist).ToList() ;

            List<EntityHistory> listHistory = new List<EntityHistory>(sessionList.Count);

            for (int i = 0; i < sessionList.Count; i++)
            {
                EntityHistory en = new EntityHistory();
                en.RowId = i.ToString();
                en.SessionID = sessionList[i].ID_Session;
                en.ModelID = sessionList[i].ID_Model;
                en.SessionDateTime = sessionList[i].Session_Date.ToString();
                listHistory.Add(en);
            }


            repeaterHistory.DataSource = listHistory;
            repeaterHistory.DataBind();

        }


        protected void linkbuttonViewTest_Click(object sender, EventArgs e)
        {
            Server.Transfer("Test.aspx");
        }

        protected void linkbuttonViewStrategies_Click(object sender, EventArgs e)
        {
            Server.Transfer("Strategy.aspx");
        }

        protected void linkbuttonViewComponents_Click(object sender, EventArgs e)
        {
            Server.Transfer("ComponentView.aspx");
        }

        protected void linkbuttonaddEvidence_Click(object sender, EventArgs e)
        {
            Server.Transfer("AddEvidence.aspx");
        }

        protected void linkbuttonRemoveEvidence_Click(object sender, EventArgs e)
        {
            Server.Transfer("Remove_CleanEvidence.aspx");
        }

    }
}
```

Web Page: ComponentView.aspx.aspx [Markup]

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeBehind="ComponentView.aspx.cs" Inherits="AFDPresentation.ComponentView" %>

<%@ Register Assembly="System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35"
    Namespace="System.Web.UI.DataVisualization.Charting" TagPrefix="asp" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">

   <link rel="Stylesheet" href ="DataTables/css/demo_table.css" type ="text/css" />
   <link rel="Stylesheet" href="Styles/StyleSheetMenu.css" type = "text/css" />
   <script type ="text/javascript" src="DataTables/js/jquery.js"></script>
   <script type ="text/javascript" src="DataTables/js/jquery.dataTables.js"></script>

    <script type ="text/javascript">

        $(document).ready(function () {

            $('#tblComponents').dataTable({
                "oLanguage": { "oView": "Select a Model(.BNE)" },
                "iDisplyLength": 10,
                "aaSorting": [[0, "asc"]],
                "sAjaxSource": "../../JsonDataFiles/jdComponents.txt"
            });
        });

        function OnClientClick_linkCloseChart() {

            document.getElementbyid('divChartComponent').style.visibility = 'hidden';
         }

        function ShowBarChart() {

            window.open("Charts.htm","Component Probability Chart", "width=500,height=500,toolbar=no,resizable=no")

        }
    </script>
    <style type="text/css">
        .style3
        {
            color: #FFFFFF;
        }
    </style>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">

    <div id="divPage"
        style =" height:600px; width :100%; background-color :lightblue;   ">

        <div style = " color :White;  width:100%; text-align: center; ">
         <span style="color: #FFFFFF; font-size:large;  "> Components </span></div>


         <div style = " width : 100%">
            <div style = " margin-left :50px; width:100px; float:left;">
                <asp:LinkButton ID="linkbuttonViewTest" runat="server" onclick="linkbuttonViewTest_Click"
                    >View Tests</asp:LinkButton>
            </div>

            <div style = " margin-left :5px; width:115px; float :left;">
             <asp:LinkButton ID="linkbuttonViewStrategies" runat="server" onclick="linkbuttonViewStrategies_Click"
                    >View Strategies</asp:LinkButton>
            </div>

              <div style = " margin-left :5px; width:115px; float :left;">
             <asp:LinkButton ID="linkbuttonaddEvidence" runat="server" onclick="linkbuttonaddEvidence_Click"
                    >Add Evidence</asp:LinkButton>
            </div>

              <div style = " margin-left :5px; width:152px; float :left;">
             <asp:LinkButton ID="linkbuttonRemoveEvidence" runat="server" onclick="linkbuttonRemoveEvidence_Click"
                    >Clean/Remove Evidence</asp:LinkButton>
            </div>

                <div style = " margin-left :5px; width:152px; float :left;">
             <asp:LinkButton ID="linkbuttonViewHistory" runat="server" onclick="linkbuttonViewHistory_Click"
                    >Test History</asp:LinkButton>
            </div>
         </div>
```

```
        <div style =" margin-top:100px; ">

        <div style = " border: thin solid #FFFFFF; margin-left :50px;  float:left ;  height:auto; width:500px;  text-align :left;" >


            <table id="tblComponents" cellpadding ="0" width ="490" cellspacing ="0" border ="0" class = "disply">
              <thead>
                <tr>

                  <th style =" width:200px; text-align:left; " >Component Name</th>
                  <th style =" width:50px;" >Component Approp Probability</th>
                  <th style =" width:50px;">Component Probability</th>
                 </tr>
              </thead>
            <tbody>





           <tr>

                <td> </td>
                <td> </td>
                <td> </td>
            </tr>





           </tbody>
           </table>





        </div>
        <div style =" float:left;   ">
       <div style = "   border: thin solid #FFFFFF; margin-left :10px;  height:150px; width:300px;  text-align :center;" >

           <div style =" width:300px; border-bottom-style: solid; border-bottom-width: thin; border-bottom-color: #FFFFFF;"
              class="style3">
              Trash Parameters</div>

                           <div style =" width:300px;  margin-top :10px;  ">
                             <div style="width: 100px; margin-top:5px;   float: left;">
                                <asp:Label ID="Label1" runat="server" ForeColor ="White"  Text="Pass Trash :"
Width="69px"></asp:Label>
                             </div>
                             <div style=" margin-top:5px; width: 168px; float:left;  ">
                                <asp:TextBox ID="textboxPassTrash" runat="server" Width="150px" ></asp:TextBox></div>
                            </div>

                            <div style =" margin-top:15px; ">
                             <div style="float: left; width: 100px;  float:left; ">
                                <asp:Label ID="Label2" runat="server" ForeColor ="White" Text="Failure Trash :"></asp:Label></div>
                             <div style="   width: 168px; float:left;  ">
                                <asp:TextBox ID="textboxFailureTrash" runat="server" Width="150px"></asp:TextBox></div>
```

```
                    </div>

                     <div id="divButtonApply" style =" margin-left :170px; margin-top :10px;">
                 <asp:Button ID="buttonApplyTrash" Text ="Apply" runat ="server" />

                </div>



        </div>

        <div style = "  border: thin solid #FFFFFF; margin-left :10px; margin-top :5px; height:125px; width:300px;  text-align
:center;" >

          <div style =" width:300px; border-bottom-style: solid; border-bottom-width: thin; border-bottom-color: #FFFFFF;"
            class="style3">
            Graphical Visualization</div>




                     <div style=" width: 100px; margin-top:2px;   ">
                       <asp:LinkButton ID="linkbuttonViewBarChart" ForeColor ="White"
                         Text ="View Bar Chart" runat ="server" onclick="linkbuttonViewBarChart_Click"
></asp:LinkButton></div>


                   <div style =" width:300px; margin-top :5px;">
                     <div style =" width:91px; float:left;">
                      <asp:Label ID="labelShowTitle" runat ="server" Text = "Show me First " ForeColor ="White" ></asp:Label>
                     </div>

                     <div style =" float:left; width:50px;">
                      <asp:TextBox ID="textboxNumberOfElements" runat ="server" Width = "50"  ></asp:TextBox>
                     </div>

                     <div style =" width:50px; float:left;">
                      <asp:Label ID="labelElements" runat ="server" Text = "Element" ForeColor ="White" ></asp:Label>
                     </div>

                   </div>

                   </br>

                   <div style =" width:300px; margin-top :5px; margin-left:2px;  ">


                     <div style ="  width:50px; float :left;">
                      <asp:TextBox ID="textboxProbLessGreaterThan" runat ="server" Width = "50px"  ></asp:TextBox>
                     </div>

                     <div style =" width:124px; float:left;">
                      <asp:Label ID="labelProbability" runat ="server" Text = "<= Probability <=" ForeColor ="White"
></asp:Label>
                     </div>

                      <div style =" float:left; width:50px;">
                      <asp:TextBox ID="textboxprobLessthan" runat ="server" Width = "50px"  ></asp:TextBox>
                     </div>
                    </div>




         </div>

          <div style = "  border: thin solid #FFFFFF; margin-left :10px; margin-top :5px; height:50px; width:300px;  text-align
:center;" >

            <div style =" width:300px; border-bottom-style: solid; border-bottom-width: thin; border-bottom-color: #FFFFFF;"
               class="style3">
               Information & Status </div>


                              <div style=" width: 150px; margin-top:2px; float:left;    ">
                               <asp:Label ID="labelLatestStepPerformed" ForeColor ="White" runat ="server" Text ="Latest Step
Performed" ></asp:Label>
                              </div>

                              <div style=" width: 100px;  margin-top:2px; float:left;    ">
                               <asp:TextBox ID="textboxLatestStepPerformed" ReadOnly ="true"  runat ="server" Width =
"129px"></asp:TextBox>
                              </div>



         </div>
      </div>

      </div>


    </div>

      <%--Chart Area--%>
      <div id="divChartComponent" style = "width:100%; height :550px;    background-color :lightblue;  ">

        <div style =" width :100px;">
          <div style =" margin-left:850px; width: 75px;">
             <asp:LinkButton ID="linkCloseChart" Visible = "false"  OnClientClick ="OnClientClick_linkCloseChart()"
runat="server">Close Graph</asp:LinkButton>
          </div>
        </div>


        <div  style =" width :100%; height :550px;">

       <asp:Chart ID="Chart1" runat="server" Visible = "false"  BorderlineColor="Black"
            BorderlineDashStyle="Solid" BackColor="#B6D6EC" BackGradientStyle="TopBottom"
            BackSecondaryColor="White" Height="550px" Width="960px">

               <Titles>
                 <asp:Title Name="Title1" Text="Component-Probability Chart"
                   Alignment="TopCenter" Font="Verdana, 12pt, style=Bold">
                 </asp:Title>
               </Titles>

               <Series>
```

```
                <asp:Series Name="Series1" CustomProperties="DrawingStyle=Cylinder,
                  MaxPixelPointWidth=50" ShadowOffset="2" IsValueShownAsLabel="True">
                </asp:Series>
              </Series>

              <ChartAreas>
                <asp:ChartArea Name="ChartArea1" BackGradientStyle="TopBottom"
                  BackSecondaryColor="#B6D6EC" BorderDashStyle="Solid" BorderWidth="2">
                  <AxisX>

                    <LabelStyle Angle="-90" />
                    <MajorGrid Enabled="False" />
                  </AxisX>
                </asp:ChartArea>
              </ChartAreas>

        </asp:Chart>

         </div>


      </div>


</asp:Content>
```

Web Page: ComponentView.aspx.aspx.cs [Code-behind]

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Domain.Entities;
using AFDPresentation.Dlls;
using Parser;
using System.IO;
using Domain;



namespace AFDPresentation
{
    public partial class ComponentView : System.Web.UI.Page
    {
        string SessionID = "";

        List<ComponentOutcome>  ResultGlobalVariable = new List<ComponentOutcome> ();


        protected void Page_Load(object sender, EventArgs e)
        {
          if (!IsPostBack)
           {
             if (Session["HistoryFlag"].ToString() == "true")
             {
               SessionID = Session["SessionID"].ToString();
               GetDataFromDBAndBind();
             }
        else
            CallWrapperMethod();
    }

}

private void CallWrapperMethod()
{
    UnsafeNativeMethods.AFD_BBN_Library_getComponents_All();
    var parser = new fileParser("C:\\TempAFDTool\\AFD_BBN_Library_getComponents_All_output.txt");
    var result = parser.parseComponent();
    // put data into Global Variable for Graph use
    ResultGlobalVariable = result;


    FillJson(result);
    InsertDataIntoDB(result);

}

 private void GetDataFromDBAndBind()
{

    AFDOnlineDBDataContext dc = new AFDOnlineDBDataContext();


    var results = (from components in dc.Compnents
                   where components.ID_Session == SessionID
                   select components).ToList();


    List<ComponentOutcome> list = new List<ComponentOutcome>(results.Count);

    for (int i = 0; i < results.Count; i++)
    {
      ComponentOutcome item = new ComponentOutcome ();
       item.ComponentName =  results[i].ComponentName;
       item.CompApropProbability  =  results[i].CompApropProbability;
       item.ComponentProbability  =  results[i].ComponentProbability;


      list.Add(item);

    }


    FillJson(list);

}

private void FillJson(List<ComponentOutcome> result)
{
    string dataString = "";

    for (int i = 0; i < result.Count; i++)
    {
        var comname = "\"" + result[i].ComponentName + "\" ";
        var comAprob = "\"" + result[i].CompApropProbability + "\" ";
        var compProb = "\"" + result[i].ComponentProbability  + "\" ";
```

```
            dataString += "[" + comname + "," + comAprob + "," + compProb + "],";


        }

        String line;
        using (StreamReader sr = new
StreamReader("C:\\MyDisk\\Data\\thesis\\AFDOnline_Develop\\AFDOnline\\AFDPresentation\\JsonDataFiles\\jtdata.txt"))
        {
            line = sr.ReadToEnd();
            sr.Close();
        }


        string res = line.Replace("{0}", result.Count.ToString()).Replace("{1}", result.Count.ToString());
        string finaldata = res.Replace("{2}", dataString.Remove(dataString.Length - 1));

        using (System.IO.StreamWriter file = new
System.IO.StreamWriter("C:\\MyDisk\\Data\\thesis\\AFDOnline_Develop\\AFDOnline\\AFDPresentation\\JsonDataFiles\\jdC
omponents.txt"))
        {
            file.WriteLine(finaldata);
            file.Close();
        }
    }

    private void InsertDataIntoDB(List<ComponentOutcome> result)
    {

        AFDOnlineDBDataContext dc = new AFDOnlineDBDataContext();



        for (int i = 0; i < result.Count; i++)
        {
          var component = new Compnent();
            component.ComponentName = result[i].ComponentName;
            component.ID_Session = Session["SessionID"].ToString();
            component.CompApropProbability = result[i].CompApropProbability;
            component.ComponentProbability = result[i].ComponentProbability;


            dc.Compnents.InsertOnSubmit(component);

        }




        dc.SubmitChanges();
    }


    protected void linkbuttonViewTest_Click(object sender, EventArgs e)
    {
        Server.Transfer("Test.aspx");
    }

    protected void linkbuttonViewStrategies_Click(object sender, EventArgs e)
    {
        Server.Transfer("Strategy.aspx");
    }

    protected void linkbuttonaddEvidence_Click(object sender, EventArgs e)
    {
        Server.Transfer("AddEvidence.aspx");
    }

    protected void linkbuttonRemoveEvidence_Click(object sender, EventArgs e)
    {
        Server.Transfer("Remove_CleanEvidence.aspx");
    }

    protected void linkbuttonViewHistory_Click(object sender, EventArgs e)
    {
        Server.Transfer("History.aspx");
    }

    protected void linkbuttonViewBarChart_Click(object sender, EventArgs e)
    {
        Chart1.Visible = true;
        linkCloseChart.Visible = true;
        AFDOnlineDBDataContext dc = new AFDOnlineDBDataContext();

        var FirstXRecord = textboxNumberOfElements.Text;
        var probLessthan = textboxprobLessthan.Text;
        var probGreaterthan = textboxProbLessGreaterThan.Text;

        var listOfComponents = new List<Compnent>();







        if (SessionID == "")
        {
            if (FirstXRecord != "" && probLessthan != "" && probGreaterthan != "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == Session["SessionID"].ToString() &&
                    Convert.ToDouble(i.ComponentProbability) <= Convert.ToDouble(probLessthan) &&
Convert.ToDouble(i.ComponentProbability) >=
Convert.ToDouble(probGreaterthan)).Take(Convert.ToInt16(FirstXRecord)).ToList();
            }
            else if (FirstXRecord != "" && probLessthan == "" && probGreaterthan != "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == Session["SessionID"].ToString() &&
                        Convert.ToDouble(i.ComponentProbability) >=
Convert.ToDouble(probGreaterthan)).Take(Convert.ToInt16(FirstXRecord)).ToList();
            }
            else if (FirstXRecord != "" && probLessthan != "" && probGreaterthan == "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == Session["SessionID"].ToString() &&
                    Convert.ToDouble(i.ComponentProbability) <=
Convert.ToDouble(probLessthan)).Take(Convert.ToInt16(FirstXRecord)).ToList();
            }
            else if (FirstXRecord != "" && probLessthan == "" && probGreaterthan == "")
            {
```

```
                listOfComponents = dc.Compnents.Where(i => i.ID_Session ==
Session["SessionID"]).Take(Convert.ToInt16(FirstXRecord)).ToList();
            }

            if (FirstXRecord == "" && probLessthan != "" && probGreaterthan != "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == Session["SessionID"].ToString() &&
                    Convert.ToDouble(i.ComponentProbability) <= Convert.ToDouble(probLessthan) &&
Convert.ToDouble(i.ComponentProbability) >= Convert.ToDouble(probGreaterthan)).ToList();
            }
            else if (FirstXRecord == "" && probLessthan == "" && probGreaterthan != "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == Session["SessionID"].ToString() &&
                        Convert.ToDouble(i.ComponentProbability) >= Convert.ToDouble(probGreaterthan)).ToList();
            }
            else if (FirstXRecord == "" && probLessthan != "" && probGreaterthan == "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == Session["SessionID"].ToString() &&
                          Convert.ToDouble(i.ComponentProbability) <= Convert.ToDouble(probLessthan)).ToList();
            }
            else if (FirstXRecord == "" && probLessthan == "" && probGreaterthan == "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == Session["SessionID"].ToString()).ToList();
            }


            Chart1.Series["Series1"].Points.DataBindXY((from components in listOfComponents select
components.ComponentName).ToList() , "ComponentName", (from components in listOfComponents select
components.ComponentProbability).ToList() , "ComponentProbability");
        }
        else
        {
            if (FirstXRecord != "" && probLessthan != "" && probGreaterthan != "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == SessionID &&
                    Convert.ToDouble(i.ComponentProbability) <= Convert.ToDouble(probLessthan) &&
Convert.ToDouble(i.ComponentProbability) >=
Convert.ToDouble(probGreaterthan)).Take(Convert.ToInt16(FirstXRecord)).ToList();
            }
            else if (FirstXRecord != "" && probLessthan == "" && probGreaterthan != "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == SessionID &&
                      Convert.ToDouble(i.ComponentProbability) >=
Convert.ToDouble(probGreaterthan)).Take(Convert.ToInt16(FirstXRecord)).ToList();
            }
            else if (FirstXRecord != "" && probLessthan != "" && probGreaterthan == "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == SessionID &&
                    Convert.ToDouble(i.ComponentProbability) <=
Convert.ToDouble(probLessthan)).Take(Convert.ToInt16(FirstXRecord)).ToList();
            }
            else if (FirstXRecord != "" && probLessthan == "" && probGreaterthan == "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == SessionID).ToList();
            }

            if (FirstXRecord == "" && probLessthan != "" && probGreaterthan != "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == SessionID &&
                    Convert.ToDouble(i.ComponentProbability) <= Convert.ToDouble(probLessthan) &&
Convert.ToDouble(i.ComponentProbability) >= Convert.ToDouble(probGreaterthan)).ToList();
            }
            else if (FirstXRecord == "" && probLessthan == "" && probGreaterthan != "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == SessionID &&
                        Convert.ToDouble(i.ComponentProbability) >= Convert.ToDouble(probGreaterthan)).ToList();
            }
            else if (FirstXRecord == "" && probLessthan != "" && probGreaterthan == "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == SessionID &&
                            Convert.ToDouble(i.ComponentProbability) <= Convert.ToDouble(probLessthan)).ToList();
            }
            else if (FirstXRecord == "" && probLessthan == "" && probGreaterthan == "")
            {
                listOfComponents = dc.Compnents.Where(i => i.ID_Session == SessionID).ToList();
            }

            Chart1.Series["Series1"].Points.DataBindXY(from components in listOfComponents select
components.ComponentName, "ComponentName", from components in listOfComponents select
components.ComponentProbability, "ComponentProbability");
          }
          Chart1.ChartAreas["ChartArea1"].AxisX.Interval = 1;

      }

      protected void linkbuttonViewPieChart_Click(object sender, EventArgs e)
      {

      }

      protected void linkbuttonViewLineChart_Click(object sender, EventArgs e)
      {

      }

      protected void Menu1_MenuItemClick(object sender, MenuEventArgs e)
      {

      }

      //private  List<ComponentOutcome> GetDataList()
      //{



      //    List<ComponentOutcome> list = new List<ComponentOutcome>(results.Count);

      //    for (int i = 0; i < ResultGlobalVariable.Count; i++)
      //    {
      //       ComponentOutcome item = new ComponentOutcome ();
      //        item.ComponentName =  results[i].ComponentName;
      //        item.CompApropProbability  =  results[i].CompApropProbability;
      //        item.ComponentProbability  =  results[i].ComponentProbability;


      //       list.Add(item);

      //    }
      //}
```

```
    }
}
```

Web Page: AvailableBNEModels.aspx.aspx [Markup]

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeBehind="AvailableBNEModels.aspx.cs" Inherits="AFDPresentation.AvailableBNEModels" %>
<%@ Register Assembly="EeekSoft.Web.PopupWin" Namespace="EeekSoft.Web" TagPrefix="cc1" %>


<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">

    <link rel="Stylesheet" href ="DataTables/css/demo_table.css" type ="text/css" />
    <script type ="text/javascript" src="DataTables/js/jquery.js"></script>
    <script type ="text/javascript" src="DataTables/js/jquery.dataTables.js"></script>

    <script type ="text/javascript">

        $(document).ready(function () {

            $('#tblAllBNEModels').dataTable({
                "oLanguage": { "oView": "Select a Model(.BNE)" },
                "iDisplyLength": 50,
                "aaSorting": [[0, "asc"]]
            });
        });


        $('#tblAllBNEModels tbody tr').click(function () {
            alert('e');
        });

        function RowClick(obj) {

            obj.style.backgroundColor = "white";
            window.location.replace('Test.aspx?modelID=' + obj.id.toString());

        }

    </script>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">

    <div id="divPage" style =" height:650px; width :100%; background-color :lightblue;   ">

        <div style = " color :White;  width:100%; text-align: center; ">
         <span style="color: #FFFFFF; font-size:large;  "> Available Models (.BNE)</span></div>


            <div style = " width : 100%">
            <div style = " margin-left :50px; width:168px; float:left;">
                <asp:LinkButton ID="linkbuttonUploadNewModel" runat="server" Visible = "false"
onclick="linkbuttonUploadNewModel_Click"
                   >Upload a New (CTM)Model</asp:LinkButton>
            </div>
            </div>
             <div style = "  margin-left:910px;  width:50px;">
                <asp:ImageButton ID="imageButtonHelpPopUp" runat="server"  Height = "50px"
```

```
          Width = "50px" ImageUrl="~/Resorces/Images/help-Cir.jpg" onclick="imageButtonHelpPopUp_Click"
           />
     </div>


    <div style = " border: thin solid #FFFFFF; margin-left :200px; margin-top :100px; height:auto; width:600px;  " >

      <asp:Repeater ID="repeaterAllBNEModels" runat ="server">

       <HeaderTemplate>
         <table id="tblAllBNEModels" width ="490" cellpadding ="0" cellspacing ="0" border ="0" class = "disply">
          <thead>
           <tr>
             <th style =" width :80px;">Model ID</th>
             <th style =" width :400px;">Board ID</th>
             <th style =" width :400px;">CTM SourceFile</th>
            </tr>
          </thead>
         <tbody>
       </HeaderTemplate>




       <ItemTemplate>
        <tr id = "<%# Eval("ID_Model")%> " ondblclick ="RowClick(this)">
           <td> <%# Eval("ID_Model")%> </td>
           <td> <%# Eval("ID_Board")%></td>
           <td> <%# Eval("CTM_SourceFile")%></td>
        </tr>
       </ItemTemplate>


       <FooterTemplate>
        </tbody>
        </table>
       </FooterTemplate>

      </asp:Repeater>

     </div>


     <div style = " margin-left : 725px; width :100px;">
       <asp:Button ID="buttonInitilize" runat="server" Text="Init"  Width = "80"
         onclick="buttonInitilize_Click"/>
     </div>

      <div style = " margin-left : 200px; width :600px; ">
       <asp:Label ID="labelStatus" runat="server"  Font-Bold = "true" Font-Underline = "true" Width = "600" />
     </div>

      <div>
           <cc1:PopupWin id="popupWin" runat="server" visible="False"
             colorstyle="Blue" width="300px" height="150px"  DragDrop ="true" DockMode ="BottomRight"
             windowscroll="False" windowsize="300, 200" style="top: 0px; left: 0px" />
          </div>
        </div>
    </div>
</asp:Content>
```

Web Page: AvailableBNEModels.aspx.aspx.cs [Code-behind]

```
using System;
using System.Collections.Generic;
using System.Data.Linq;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Domain.Entities;
using Domain;
using System.IO;
using AFDPresentation.Dlls;
using Parser;


namespace AFDPresentation
{
    public partial class AvailableBNEModels : System.Web.UI.Page
    {

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                BindGrid();
            }

        }

        private void BindGrid()
        {


            AFDOnlineDBDataContext dc = new AFDOnlineDBDataContext();
            var result = (from models in dc.Models select models).ToList();
            if (result.Count == 0)
            {
                setupPopUp();
                linkbuttonUploadNewModel.Visible = true;

            }
            else
            {

                repeaterAllBNEModels.DataSource = result;
                repeaterAllBNEModels.DataBind();
            }


        }

        protected void buttonInitilize_Click(object sender, EventArgs e)
        {
            //write the Input
            // Compose a string that consists of three lines.
            string path = @"C:\TempAFDTool\Model\net_x.txt,";

            // Write the string to a file.
            System.IO.StreamWriter file = new System.IO.StreamWriter("C:\\TempAFDTool\\AFD_BBN_Library_Init_input.txt");
            file.WriteLine(path);
            file.Close();


            // call the Wrapper Method
            UnsafeNativeMethods.AFD_BBN_Library_Init();

            // explore the Output file
            var parser = new fileParser("C:\\TempAFDTool\\AFD_BBN_Library_Init_output.txt");
            var outDto = parser.ParseInitOutFile();

            if (!outDto.result)
            {
              labelStatus.Text = outDto.ErrorMessage;
            }
            else
            {
              Server.Transfer("Test.aspx");
            }
        }


        protected void imageButtonHelpPopUp_Click(object sender, ImageClickEventArgs e)
        {
            setupPopUp();
        }

        private void setupPopUp() {
            popupWin.ActionType = EeekSoft.Web.PopupAction.MessageWindow;

            //Set popup and window texts
            popupWin.Title = "Informative Message";
            popupWin.Message = "</br></br></br><b>Oops! There are NO existing CTM Model on the Server, To Proceed futher, 
Click here </b>";
            popupWin.Text = " </br>  - There are no uploaded Model on the Server, you on Continue with the AFDTool by 
Uploading a New CTM Model </br></br></br> - For Uploding a New Model, click on the Upload New Model Button on the 
ToolBar </br>  ";

            //Change color style
            popupWin.ColorStyle = EeekSoft.Web.PopupColorStyle.Blue;

            //' Change timing
            popupWin.HideAfter = 20000;
            popupWin.ShowAfter = 500;

            //Show popup (after page is loaded)
            popupWin.Visible = true;
        }
        protected void linkbuttonUploadNewModel_Click(object sender, EventArgs e)
        {
            Server.Transfer("UploadModel.aspx");
        }
    }
}
```

References

- Calling an unmanaged dll from .NET http://blogs.msdn.com/b/jonathanswift/archive/2006/10/02/780637.aspx
- Flot Graphs http://people.iola.dk/olau/flot/examples/
- Data Tables (JQuery) http://datatables.net/
- JQuery Syntax http://www.w3schools.com/jquery/jquery_syntax.asp
- Bayesian network http://en.wikipedia.org/wiki/Bayesian_network , http://www.dsto.defence.gov.au/publications/2424/DSTO-TN-0403.pdf
- Windows Communication Foundation (WCF) http://msdn.microsoft.com/en-us/library/dd456779.aspx
- Linq to SQL http://www.simple-talk.com/dotnet/.net-framework/designing-a-data-access-layer-in-linq-to-sql/
- Deployment of web site http://www.codeproject.com/Articles/32210/Deployment-of-a-Website-on-IIS
- Asp.net; understanding and practicle implementation http://www.w3schools.com/aspnet/default.asp