Fall 2015

Appointment Scheduling System

Sanjay Jaival
Governors State University

Srinivasa Raju Kallepalli
Governors State University

Vaibhav Sharma Kumar
Governors State University

Himaja Vejju
Governors State University

Follow this and additional works at: http://opus.govst.edu/capstones

Table of Contents

1. Project Description:

1.1 Project Abstract:

According to the existing system, students schedule appointments either through email or going in person to the advisor's assistant and request for an appointment. This process is very tedious and time taking.

In our project we will build an online appointment scheduling web application for every student to schedule an appointment by them self whenever and wherever they are. This system helps to lessen the burden of waiting at the advisor's assistant's desk or to conserve time and use it in an efficient manner we will bring this appointment scheduling process online. All this process is available online through our web application "APPOINTMENT SCHEDULING SYSTEM". Any changes in the scheduled appointments such as cancellation of visiting hours will be notified through email.

Modules:

Administrator – In this module admin will take privilege to add advisors and students.

Advisor – In this module advisor can access through the student files and can give updates regarding appointment.

Student – Here students can login and select the slots.

FRONT –END:

Asp.net, C#.

BACK-END:

Database -: SQL server 2012.

1.2 Competitive Information:

Doodles, Day Pilot, Outlook are the major web based scheduling system highly used around the world. These scheduling systems are considered to be competitors of our Appointment Scheduling System.

1.3 Relationship to Other Applications/Projects:

Our project is developed using Google Calendar and Scheduling systems. We are using Google Calendar for scheduling the events and meetings.

1.4 Assumptions and Dependencies:

This system is supposed to be a web based application which can be accessed 24/7. Network connection should be available to use the application. System assumes that all the participants will be actively involved in responding to meeting requests. The initiator must decide about the importance of participants. MMS can only schedule a meeting, not the priority of participants. Priorities of meetings also has to be done by initiator, the system is not responsible for the importance level of meetings. System assumes the users are familiar with basic windows and web browser operations.

1.5 Future Enhancements:

Since this system is intended for many different types of organizations where a scheduling system is necessary to schedule appointment, it is very important for the code to be customizable due to the distinct needs of different organizations. For example, some organizations only need to schedule appointment with same time, where time is not an issue during each appointment, and some places need to schedule workers with different time frame for each project. Also, some organizations make their schedules once a month, and others more frequently. For the above reasons, the system needs to be schedule very well in close detail, and the system should be easy to reuse, and customize for future usage. Future customization will include features that will simplify time entry and reduce opportunity for time entry errors.

1.6 Definitions and Acronyms:

SRS – Software Requirement Specifications.

MMS – Meeting Scheduler System.

FR – Functional Requirement.

NFR –Nonfunctional requirement.

DR- Domain requirement.

SE- Software Engineering.

IC- Identity Card.

HTTP- Hypertext transfer Protocol.

HTML- Hypertext Markup Language.

CS- Computer Science.

2. Technical Descriptions:

The objective of the project is to develop a meeting scheduler system within allocated time, budget and specified quality. The project is prioritized due to high benefits to the organization. One of the important usages of this project is that it will automate the process of meeting scheduling and thus save the time and efforts of meeting organizer. More benefits will be further discussed ahead. The primary focus of our team is the reliability, usability, and quality. Satisfying requirements with precision is very important to us while the products intends to meet all the user's needs.

A) Microsoft .NET Framework

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.

To provide a code-execution environment that minimizes software deployment and versioning conflicts.

- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.

- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.

- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

For example, ASP.NET hosts the runtime to provide a scalable, server-side environment for managed code. ASP.NET works directly with the runtime to enable Web Forms applications and XML Web services, both of which are discussed later in this topic.

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the runtime enables you to embed managed components or Windows Forms controls in HTML documents. Hosting the runtime in this way makes managed mobile code (similar to Microsoft® ActiveX® controls) possible, but with significant improvements that only managed code can offer, such as semi-trusted execution and secure isolated file storage.

The following illustration shows the relationship of the common language runtime and the class library to your applications and to the overall system. The illustration also shows how managed code operates within a larger architecture.

B) SQL SERVER 2012

Microsoft SQL Server 2012 is comprehensive, integrated data management and analysis software that enables organizations to reliably manage mission-critical information and confidently run today's increasingly complex business applications. SQL Server 2012 allows companies to gain greater insight from their business information and achieve faster results for a competitive advantage.

2.1 Project/Application Architecture:

Fig: System Architecture (a)

Fig-1: Architecture

2.2 Project/Application Information flows

Fig: Use Case Diagram

2.3 Interactions with other Projects:

Our project is developed using Google Calendar and other Scheduling systems.

2.4 Interactions with other Applications:

Our project uses Event calendar for scheduling purpose. Finally our project uses Gmail for sending emails.

2.5 Capabilities:

- Scheduling the meeting in efficient way.
- Gathering the feedback from attendee.
- Cancelling the meeting.
- Changing the meeting schedule.
- Scheduling concurrent meetings in timely manner.
- Confirming the time of the meeting.

- Minimize users effort in coordinating and scheduling meetings

2.6 Risk Assessment and Management:

- Before risk management begins it is imperative that a foundation is established for providing structured project information, thus, the following project elements were completed and defined prior to developing this Risk Management Plan:
- Define work scope, schedule, resources, and cost elements
- Develop project WBS/WBS dictionary
- Develop master schedule and detailed schedules
- Estimate project cost and finalize budget
- Identify required and available resources
- Establish performance measurement metrics
- Define minimum and maximum baseline thresholds
  Schedule
  Resources
  Cost
- Baseline reporting requirements
  Format
  Frequency of distribution
  Distribution list
- Define Risk Management Roles and Responsibilities
- Project Manager chairs the risk assessment meetings
  Project team participates in risk assessment meetings and members serve as meeting recorder and timekeeper Key stakeholders participate in risk assessment meetings Project Sponsor may participate in risk assessment meetings.

3 Project Requirements:

3.1 Identification of Requirements:

<GSU-001-0.1 STUDENT-STUDENTID/000101>
Require to generate and store the student ID for the student
Implementation: Mandatory
<GSU-001-0.1 STUDENT-EMAIL/000102>
Student email must identify each student and is unique for every student
Implementation: Mandatory

<GSU-001-0.1 STUDENT-FIRSTNAME/000103>
Require to store the first name of the student
Implementation: Mandatory

\<GSU-001-0.1 STUDENT-LASTNAME/000104\>
>   Require to store the last name of the student
>   Implementation: Mandatory

\<GSU-001-0.1 STUDENT-PHONEONE/000105\>
>   Require to store the primary phone number of the student
>   Implementation: Mandatory

\<GSU-001-0.1 STUDENT-PHONETWO/000106\>
>   Require to store the secondary phone number of the student
>   Implementation: Mandatory

\<GSU-004-0.1 DATABASE-RELIABILITY/00401\>
>   About 100,000 customers will be visiting the web site per week. So, performance must be
maintained through 24/7
>   Implementation: Mandatory

\<GSU-005-0.1 DATABASE-MAINTENANCE/00501\>
>   Database maintenance issues must be fixed scheduled every Sunday 10PM to 6AM
central time. Daily DB maintenance and backup, can only be performed between 12PM to 2AM
central time
Implementation: Mandatory

\<GSU-006-0.1 DATABASE-FAILOVER/00602\>
To handle fail over case, the system must support at least 30 seconds worth of data in temporary
memory space. This is estimated as 5000 customer records and their transient state info. Duplex
system should be maintained.
Implementation: Mandatory

\<GSU-007-0.1 SCHEULE APPOINTMENT/000701\>

Advisor create the time slots when student need to meet her.
Implementation: Mandatory

\<GSU-007-0.1 SCHEULE APPOINTMENT-DATE/000702\>

When students need to meet the professor.
Implementation: Mandatory

\<GSU-007-0.1 SCHEULE APPOINTMENT-TIME/000703\>

Timing information will be updated basing on professors availability.
Implementation: Mandatory

\<GSU-007-0.1 SCHEULE APPOINTMENT-PLACE/000704\>

Place of appointment will be stored as it will be different for every professor.

Implementation: Mandatory

<GSU-007-0.1 SCHEULE APPOINTMENT-EMAIL/000705>

Students will be informed any updates regarding appointments via emails.
Implementation: Mandatory

<GSU-008-0.1 SLOT BOOKING/000801>

Using this students can select their own appointment slots.
Implementation: Mandatory

<GSU-008-0.1 SLOT BOOKING-AVAILABLE TIMES/000803>

Students can check which timing slots are available for appointment.
Implementation: Mandatory

<GSU-008-0.1 SLOT BOOKING- STUDENT ID/000804>

While booking the slots students must provide their Student ID
Implementation: Mandatory

<GSU-009-0.1 ADD STUDENT/000901>

Administrator will add the new students who just started their semester.
Implementation: Mandatory

<GSU-009-0.1 ADD STUDENT-STUDENTID/000902>

Student ID required to register for the application
Implementation: Mandatory

<GSU-009-0.1 ADD STUDENT-NAME/000903>

Student Name is required for the database storage.
Implementation: Mandatory

<GSU-009-0.1 ADD STUDENT-EMAIL/000904>

Student must provide valid college Email ID
Implementation: Mandatory

<GSU-009-0.1 ADD STUDENT-PHONE NUMBER/000905>

Student should provide valid phone number for communication purpose
Implementation: Mandatory

<GSU-010-0.1 ADD ADVISOR/001001>

Administrator can only register the advisors.
Implementation: Mandatory

<GSU-010-0.1 ADD ADVISOR-ID/001002>

To register advisor they should provide their College ID
Implementation: Mandatory

<GSU-010-0.1 ADD ADVISOR-NAME/001003>

Account will be created based on their name
Implementation: Mandatory

<GSU-010-0.1 ADD ADVISOR-EMAIL/001004>

When their account is created they can communicate using their emails.
Implementation: Mandatory

<GSU-010-0.1 ADD ADVISOR-PHONE NUMBER/001005>

After registration they will get password as a text message
Implementation: Mandatory

<GSU-010-0.1 ADD ADVISOR-PASSWOPRD/001006>

Administrator will create a password for Advisor.
Implementation: Mandatory

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P):

Appointment scheduling system is a web based application. The Client Side Design is developed using ASP.NET and Server Side Validation is done by using C# for storing student details, professor details and time slots. We are using SQL server 2012 .If students want to use this web application they need a web browser (Chrome, Firefox, Safari, Internet Explorer) of any versions.

If you want to deploy the web application we need windows server with IIS enabled and administrator should backup the database for every one month. It is necessary for the advisors & students to change their passwords for every 90 days.

If a student or a advisor wants to change their password they need to contact the administrator with their ID and password.

Those who are registered for the web application they can perform the following operations based on their role:

a) **Admin** :
   i) Add a advisor
   ii)Add student
   iii)Maintain Database
b) **Advisor** :
   i) Creating Appointment Slots
   ii) Verifying Student Appointment.
   iii) Communicating with the Students by email.
c) **Students** :
   i)Registration for the Available time slot
   ii) Selecting Slots for the Appointment.

3.3 Security and Fraud Prevention:

Our Project is a web based application whenever the user wants to login in to the Appointment Scheduling System it automatically records the IP address of the particular user login session and its stores the information in the database. Using this we can easily track the user details those who are trying to perform the fraud.

3.4 Release and Transition Plan:

Release Process

This SOW shall commence on Aug 31st, 2015 (the "Effective Date") and shall continue until 29th Nov, 2015. The final demonstration is to be given on Nov 30 2015. Below are the timelines and Deliverable details.
- Interim Project I (Phase 1.1) has been given on Nov 03, 2015
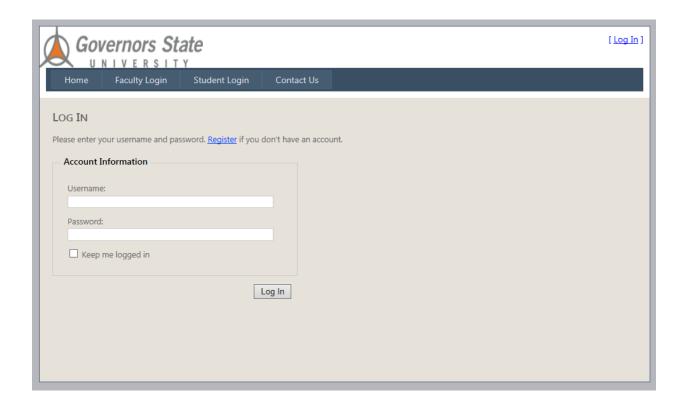- Final Project I (Phase 1.2) has been given on Nov 25, 2015

Transition Schedule

	Member1	Member2	Member3	Member4
Phase I				
Project Plan	X	X	x	X
System Requirement Specification		X	x	X
Prototype	X			
User Manual				
Presentation	X	X	x	X
Document Formatting				
Phase II				
Project Plan				
Vision Document	X			
Supplementary Document				
System Requirement Specification Refinement		X		
Process and Product Specification	X		x	X
User Manual				
Implementation				
Document Formatting				

Fig-3: Transition Diagram

4 Project Design Description:

The ASP.NET conventional facilities will be designed to provide the Security, Services and Utility Infrastructure needed to support the technical scope of the project and the mission of a high technology user facility.

The Appointment Scheduling System is a web-based meeting scheduler system to schedule various types of meetings. It would efficiently schedule meetings and determine the available resources such as location and free-time-slots suitable for all attendees which are necessary for the meeting to be initiated. The purpose of this system is to support the faculty in scheduling meetings by determining each attendee's free time slot, date and location. The MMS system will monitor meetings, plan meetings under constraints expressed by the participants, reschedule meetings based on constraints, support conflict resolutions, and manages all the interactions among participants. Since Appointment Scheduling is an online system, it can be easily accessed from web-browser with internet access, thus removing any constraints of time or place. The system also sends relevant notifications and information to respective users through emails. The system will have a user friendly interface which will make it easier for all kinds of participants.

Governors State
UNIVERSITY

Home   Faculty Login   Student Login   Contact Us

LOG IN

Please enter your username and password. Register if you don't have an account.

Account Information

Username:

Password:

☐ Keep me logged in

Log In

Governors State
UNIVERSITY

Home   Faculty Login   Student Login   Contact Us

CREATE A NEW ACCOUNT

Use the form below to create a new account.

Passwords are required to be a minimum of 6 characters in length.

Account Information

User Name:

E-mail:

Password:

Confirm Password:

Create User

Governors State
U N I V E R S I T Y

Home | Email | Schedule Meeting | Upload Documents

SEND EMAIL FROM YOUR ACCOUNT

Your Mail Id :

Your Mail Password :

To :

Subject :

Message :

Send

Governors State
U N I V E R S I T Y

Home | Email | Schedule Meeting | Upload Documents

Event Registration

Meeting Title

Description

Location

Email

Date   11/18/2015

<	November 2015					>
S	M	T	W	T	F	S
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Time   06 : 35 : 00 PM
            To
            07 : 35 : 00 PM

Submit

Governors State
U N I V E R S I T Y

[Log Out]

| Home | Email | Schedule Meeting | Upload Documents |

Multiple Files Upload

Browse...

Upload Files

GridView Example with Ec × GridView Example with Ec ×

← → C ⌂ | localhost:59697/GridView.aspx

Apps ☐ Dell Y! ★ Bookmarks   ☐ Other bookmarks

Row's Inserted

Term	Section Name and Title	Meeting Information	Faculty	Last Add Date	Refund Deadline	Last Drop Date	Available
Fall 2015	CPSC8400	F2324	xyx	10/20/2015 12:00:00 AM	11/12/2015 12:00:00 AM	12/12/2015 12:00:00 AM	1/20/1900 12:00:00 AM
Fall 2015	CPSC8400	F2324	xyx	10/20/2015 12:00:00 AM	11/12/2015 12:00:00 AM	12/12/2015 12:00:00 AM	1/20/1900 12:00:00 AM
Fall 2015	CPSC8400	F2324	xyx	10/20/2015 12:00:00 AM	11/12/2015 12:00:00 AM	12/12/2015 12:00:00 AM	1/20/1900 12:00:00 AM

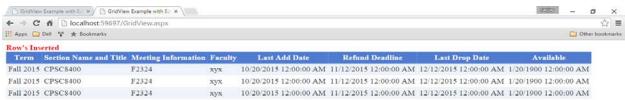Search the web and Windows

2:56 PM
11/4/2015

5. Project Internal/external Interface Impacts and Specification:

i) **Structure :**

Structure is an internal Factor that impacts operation of Appointment Scheduling System .The structure the number of advisors work for the Graduate Classes, the levels of hierarchy, the extent of advisor and department collaboration and the roles of advisor and admin.

ii) **External Communications**:

This is external Impact factor for the web based application system. The way the system interacts with the students impacts the workflow of the system.

6. Project Design Units Impacts:

The Techniques we will be using in our project will be focused on Model driven development, Test driven development, and object oriented development.

MDD gives architects the ability to define and communicate a solution while creating artifacts that become part of the overall solution. MDD is also comprised of the ability to visualize the domain, such as a business domain, and the generation of implementation artifacts. The Model-Driven Architecture (MDA) defines an approach to modeling that separates the specification of system functionality from the specification of its implementation on a specific technology platform. In short it defines a guideline for structuring specifications expressed as models.

Test Driven Design (TDD). With a TDD approach you create a test then write enough production code to fulfill that test. In other words, the tests form your detailed design model (as executable specifications), arguably making TDD a modeling approach.

Object Oriented Development (OOD) promises to reduce development time, reduce the time and resources required to maintain existing applications, increase code reuse, and provide a competitive advantage to organizations that use it.

6.1 Appointment Scheduler System:

The objective of our project is to develop an Appointment Scheduler System within allocated time, budget and specified quality. The project is prioritized due to high benefits to the organization. One of the major use of our project is that it will automate the process of meeting scheduling and thus save the time and effort of meeting organizer. More benefits will be further discussed ahead.

The primary focus of our team is the reliability, usability, and quality. Satisfying requirements and producing a perfect output of our product is mainly concentrated. Meeting user requirements is our top goal.

6.1.1 Functional Overview:

With this system, users can do the following major functions:

   I.   Set up meetings.
   II.   Re-plan meetings.
  III.   Cancel meetings
  IV.   Send emails to the users.

6.1.2 Impacts:

Impacts of the system are:

   I.   User should be able to access the system over the network.
   II.   Participants should be the professors or students of GSU.
  III.   Participants must have the GSU email for the authentication purpose.

6.1.3 Requirements:

<GSU-002-0.1 ADVISOR-LOGIN/000201>

If the advisor provides valid credentials, they will be redirected to our home page..
Implementation: Mandatory

< GSU -002-0.1 ADVISOR-PWD-REQUEST/000202>

 Prompts for Advisor ID. If a valid Advisor ID is provided then prompts security questions. If not prompts not a valid employee.

Implementation: Mandatory.

< GSU -002-0.1 ADVISOR-FGT-VALIDATE/000203>

If valid security answers then send password recovery link to concerned email Id

 Implementation: Mandatory

< GSU -002-0.1 ADVISOR-PWD-RESET/000202>

 Clicking link in the mail must redirect where the password can be reset

Implementation: Mandatory

6.2 Meeting Scheduler System:

I. A "meeting initiator" may cancel the meeting or reschedule the meeting at any time prior to the start of the meeting.

II. A meeting scheduler may automatically propose another meeting if current meeting is canceled by an important participant.

III. A "meeting initiator" shall confirm the meeting and the system shall change the "time slots" of accepting "meeting participants" from a temporary reservation to a scheduled meeting, once all "potential meeting participants" have responded to the "meeting proposal.

IV. A meeting scheduler will inform the "meeting initiator" that no "time slot" exists for all "potential meeting participants" and may optionally suggest an alternative "date range", "duration", and "location" which is available.

V. The system shall keep participants informed about meeting schedules and their changes.

VI. The meeting scheduler system must in general handle several meeting requests in parallel. Requirements: list of meeting schedules and their adopted room.

6.2.1 Functional Overview:

The Meeting scheduler system has three main members involved; the initiator, administrator and the attendees (participants). The steps for the event scenarios include the user input, processing for the meeting schedule and the notification sent to the various users. The user input consists of login with his/her email. The pre-condition to this function is "register" incase the user hasn't registered to the system. The initiator logs in to process the function for initiating, altering and canceling the meeting when required. Once the meeting has been finalized all the participants registered for the meetings are notified.

6.2.2 Impacts:

I. GSU students and employees database is needed for the list of participants.

II. Meeting room database is required for scheduling the meeting's location.

6.2.3 Requirements:

<GSU-003-0.1 DEPARTMENT-NAME/000302>

Every Department has its own Name

Implementation: Mandatory

< GSU -003-0.1 DEPARTMENT-EMAILID/000303>

Every Department has its Email ID for communication

Implementation: Mandatory

< GSU -003-0.1 DEPARTMENT-PHONE/000304>

Require to store department's phone number

Implementation: Mandatory

<GSU-008-0.1 DEPARTMENT-BUILDING/000305>

Department's block/building name or number must be stored

Implementation: Mandatory

<GSU-008-0.1 DEPARTMENT-HEAD/000311>

Every Department has a Head who is the employee

Implementation: Mandatory

7. Open Issues:

I.   Technical – Relating to a technological problem in the project.
II. Business process – Relating to the project's design.
III. Change management – Relating to college, students, or environmental changes.
IV. Resource – Relating to equipment, material, or people problems.
V.   Third party – Relating to issues with outside scheduling Systems.

8. Acknowledgements:

We kept all our efforts for the success of our project. However, it would not be possible to get this success without the kind support and help of our beloved Professor and Guide Dr. X. Tang and many other people. We would like to extend our heartfelt gratitude to all of them.

We are highly indebted for their guidance, constant supervision and also providing us necessary information regarding the project. Thank you for your support in helping us to completing the project.

9. References:

[1] Requirement Engineering – Advanced Requirement Engineering. CS/SE 6361 Section 001, Spring 2010. http://www.utdallas.edu/~chung/RE/syllabus.htm

[2] Project Description: Requirements Elicitation: Initial Understanding
http://www.utdallas.edu/~chung/RE/Project1.pdf

[3]
http://www.utdallas.edu/~chung/CS6354/CS6354_U07_source/Team_2/SoftwareProjectManagement_Plan_v1.1_Team2.doc

[4] http://re-project.org/Documents/Final_Project_Plan.doc

[5] http://www.utdallas.edu/~chung/RE/Presentations09F/Pioneer-Team/RE%20Deliverable/

[6] Roles and responsibilities www.csuchico.edu/cms/Readiness/roles.doc

[7] www.search.org/files/doc/sample%20project%20plan.doc - section 3.2, 3.4

[8] IEEE standards for SRS [IEEE-STD-830-1993] http://pro-10.design.officelive.com/aboutus.aspx

10. Appendices:

Features of the Common Language Runtime:

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally featuring rich.