ABSTRACT

This project implements a Web CMS for an online news service that heavily uses XML content. Managed content include text, images, audio and video, and can be managed by multiple, authorized users with Web connectivity at any location in real-time. All content is gathered in a repository from Web forms after being converted into XML. Finally, multiple XSL style-sheets for different sections of the Website publish the stored XML files online. The project is carried out in the Visual Studio.NET environment and uses ASP.NET technology to build the site and its native C# language to implement the front and back end functionalities.
# TABLE OF CONTENTS

- ABSTRACT ................................................................................................................................. II

- LIST OF FIGURES .......................................................................................................................... 5

1. BACKGROUND AND RATIONALE ............................................................................................. 6

  1.1 WHAT IS WEB CONTENT? ............................................................................................................ 6
    
    1.1.1 Web Content Types ................................................................................................................. 7
  
  1.2 CONTENT COMPONENTS ............................................................................................................ 7
  
  1.3 CONTENT MANAGEMENT SYSTEM ........................................................................................... 8
    
    1.3.1 Types of CMS .......................................................................................................................... 8
    
    1.3.2 CMS benefits .......................................................................................................................... 9
    
    1.3.3 The Need for a Web CMS ......................................................................................................... 10

2. CONTENT MANAGEMENT SYSTEM ............................................................................................. 11

  2.1 THE COLLECTION SYSTEM ....................................................................................................... 11
    
    2.1.2 The Authoring System ........................................................................................................... 11
    
    2.1.3 The Acquisition System ......................................................................................................... 12
    
    2.1.4 The Conversion System ......................................................................................................... 13
    
    2.1.5 The Aggregation System ....................................................................................................... 13
    
    2.1.6 Repository ............................................................................................................................ 14
  
  2.2 THE MANAGEMENT SYSTEM .................................................................................................... 14
    
    2.2.1 Editing repository content ..................................................................................................... 14
    
    2.2.2 Meta-content .......................................................................................................................... 15
    
    2.2.3 Content Lifecycle .................................................................................................................... 15
  
  2.3 THE PUBLISHING SYSTEM ......................................................................................................... 16
    
    2.3.1 Templating System .................................................................................................................. 16
    
    2.3.2 Personalization System .......................................................................................................... 17
    
    2.3.3 Deployment System .............................................................................................................. 17

3. SYSTEM DESIGN AND ARCHITECTURE ...................................................................................... 18

  3.1 THE NEWS WEBSITE DESIGN .................................................................................................. 18
3.2 COLLECTION SYSTEM .......................................................................................................................... 18
  3.2.1 Content Formats ................................................................................................................................. 19
  3.2.2 Content Acquisition Interface ........................................................................................................ 21
3.3 THE MANAGEMENT SYSTEM .............................................................................................................. 23
  3.3.1 Enforcing content format ................................................................................................................... 24
3.4 THE PUBLISHING SYSTEM ................................................................................................................... 25

4. TESTING AND EVALUATION ..................................................................................................................... 28
  4.1 WEBSITE AND LAYOUT ........................................................................................................................ 28
  4.2 ADMIN SECTION FOR ENTERING AND EDITING CONTENT ................................................................. 28
  4.3 XSD SCHEMAS FOR VALIDATING CONTENT ......................................................................................... 29
  4.4 XSL STYLE-SHEETS ............................................................................................................................ 30

5. TESTING RESULTS AND CONCLUSION ....................................................................................................... 31
  5.1 WEBSITE AND LAYOUT ........................................................................................................................ 31
  5.2 ADMIN SECTION FOR ENTERING AND EDITING CONTENT ................................................................. 32
  5.3 XSD SCHEMAS FOR VALIDATING CONTENT ......................................................................................... 37
  5.4 XSL STYLE-SHEETS ............................................................................................................................ 37
LIST OF FIGURES

Figure 3.1. Admin Interface .......................................................... 19
Figure 3.2. News Content in XML Format ................................. 20
Figure 3.3. Poll content ............................................................... 21
Figure 3.4. Weather content ......................................................... 21
Figure 3.5. Content entry interface ............................................. 22
Figure 3.6. Content addition ......................................................... 23
Figure 3.7. Business Content XSD ............................................... 25
Figure 3.8. Content sections ......................................................... 26
Figure 3.9. Headline News article .............................................. 27
Figure 5.1. Main site ................................................................. 31
Figure 5.2. FTP files directory .................................................. 32
Figure 5.3. Login demonstration ................................................. 33
Figure 5.4. Content Management section ............................... 33
Figure 5.5. Headline News Editing ........................................... 34
Figure 5.6. Editing changes displayed ....................................... 34
Figure 5.7. Webpage before deleting article .............................. 35
Figure 5.8. Deleting article from Admin section ......................... 35
Figure 5.9. *Classical Painting* article deleted in Admin ............ 36
Figure 5.10. Article removed from Webpage ............................ 36
Figure 5.11. Required field missing ............................................ 37
Figure 5.12. Dynamic XHTML page ........................................... 38
1. BACKGROUND AND RATIONALE

It is hard to imagine how society functioned properly and in a timely fashion in the previous era when the Internet was non-existent. Whereas applying for a job position in another part of the world through an online form and being selected for the position the same day is technically possible today, such a concept would have been ridiculed in the past, let alone the means. With the birth of the Internet, a whole new paradigm of human functioning was invented.

However, Web sites of the previous decade were quite simple in effort and functionality. They were mostly simple text formatted with HTML. Although most pages of the site included a few pictures and were capable of rudimentary personalization techniques, they were static in nature and their patrons could only view the information that the authors had originally placed online. Meaning, content selection was not possible and would require host intervention each time.

1.1 What is Web Content?

Although a matter of debate, Web content is generally agreed upon to include information, such as text and images on a Web site, and the applications or software running on a Web server that control the display of information [Boiko 2005]. It also includes the storage format of content and the rendering format of content that aids in presentation; however, they are often called metadata and are simply content about
content. All content has structure embedded that defines their characteristics within a certain context. It is these structures that allow us to distinguish two copies of a single data in different contexts.

1.1.1 Web Content Types

Most Web content fall under text, image, video, audio, and animation. Of these, only text is processed in ASCII whereas the other four are stored and processed in binary format. Other content not listed above are not common and generally include already structured content from a different application. Inserting an Excel spreadsheet (.xls format) in a Word document is one example, since the raw data in the spreadsheet that is already structured in Excel format is further wrapped in Word formatting of its own before presentation.

1.2 Content Components

Although text and images, for example, are both content, treating them as the same would not work due to their inherent semantic differences. Just as an image pixel cannot be converted into an ASCII character, converting an ASCII character into an image pixel, although possible, would result in gibberish [Fraser 2002]. The need for suitably compartmentalizing content into different categories is necessary and would allow us to develop different sets of properties, formats, and actions for the content, according to their classification. Content components are essentially single entities within a Webpage, the collection of which constitutes the content of that page. As an example, banners,
navigation bars, textboxes, pie charts, and visual delimiters on a typical Web page are all content components with different properties and presentation format that are combined together on a page for form content. This content could portray a single news item or story, or several, depending on the purpose intended.

1.3 Content Management System

A Content Management System (CMS) is a system that manages content components [Boiko 2005]. A CMS built specifically to manage online content is a Web Content Management System. It facilitates the creation, functionality, storage, and publishing aspects of Web content components or content as a whole. In other words, it oversees the full life cycle of content components and meta-content beginning with the repository where they are stored until their display on a Website.

1.3.1 Types of CMS

There are many types of CMSs, each with different purpose and functionality: however, their common purpose is to organize or create documents or other content collaboratively. Most CMSs are built in client/server architecture either across the corporate inter-network or the Internet. Web CMSs assist in automating the various aspects of Web publishing such as acquiring and approving content to be published, replacing obsolete content with fresh content on the Website, and generating dynamic HTML pages that are already formatted and ready for display, among many others.

"Transactional CMSs assist in managing e-commerce transactions for large, multinational
corporations. This includes integrating pricing structures and synchronizing e-commerce transactions across a distributed application.

Document management systems track and store media content of text documents, synchronize multiple document types to extract meaningful data that is either available for viewing in a well formatted document or as input to another program that requires all content to have a specific type or format.

1.3.2 CMS benefits

Corporate Web document standards such as layouts, fonts, and styles can be centrally controlled by the administrator and barring individual authors from using their own. This is so because all content components must go through a single workflow provided by the CMS to ensure uniform consistency.

Authors could manage their Web content from anywhere in the world provided they have internet access and allowed authoring tools. Password access could be setup to prevent intruders.

No client software installation is necessary. All content are entered through standard Web interfaces or another established method regardless of the processor or operating system type, provided that the system has a standard, Web-enabled browser.

A non-technical author without any knowledge of HTML or programming language could tailor content with ease.

Given the CMS client/server architecture, multiple users could be logged on to the CMS, with one editing the content, another editing the meta-content, and the third person viewing the content, all simultaneously.
1.3.3 The Need for a Web CMS

Older Web sites mostly belonged to the government and business corporations and were plain HTML files that spanned 15 to 20 pages at most, which listed their regular operations, offered services, and contact information. Such small Websites were easily created, edited, and disposed of manually, by an administrator. A formal workflow design that dictated this process was not required.

The same cannot be said of the Websites of today. Almost all major corporate Websites are information powerhouses that cater lavishly to their client's fancy by providing a wide range of goods and services. They deploy distributed databases that hold terabytes of information at several locations. Even a horde of Webmasters employed to manage such a behemoth system would fail simply due to lack of uniformity in standards among them. Also, a large number of authors from different locations simultaneously trying to modify the same content would jeopardize its accuracy. A carefully chosen CMS would be the perfect candidate for this situation and would save the company vast sums of money in the long term.
2. CONTENT MANAGEMENT SYSTEM


2.1 The Collection System

The Collection System turns raw information into content components, organizes and stored them in the repository [Boiko 2005]. The process is complicated and subdivided into a series of steps.

2.1.2 The Authoring System

The authoring system refers to the process of creating contents from scratch. The Collection System provides an authoring environment for authors to create content, which is most commonly a Web form or an authoring template.

Web forms extend full control over what is acceptable in terms of the type of content, their format and length. This is so because Web form server controls are preprogrammed and have properties associated with them that leave the author with little opportunity in changing the structure or format of the content. Images and other non-text content can be uploaded directly into the repository and the Web form has the functionality to format, organize, and link them with their associated text. Further, validation tools can be employed to ensure this process.
Authoring templates contain placeholders and sections separated by content type within an authoring environment such as MS Word or Notepad where the author can type text and metadata, and also include images or other media within the document. This gives the author much more freedom than Web forms yet guides him against breaches in structure or formatting. Built-in programming tolls such as Macros and VBA for Office can be used to create sections within Word that restrict the content type within a particular section. For instance, a Word document having four such sections would allow only formatted text in the first section, images in the second section, audio content in the third section, and video content in the fourth section. Each section is preformatted with structure geared toward the content type allowed. The text section may have placeholders for the title of the article, name of the author, date and time, etc. The image section would have a mechanism to include as many images as reasonable with placeholders for title and description, and likewise for the other two sections.

Building such a template based authoring environment is quite complex and resource intensive, but very useful. In many less-expensive systems, the authoring environment resides outside of the CMS and although it saves the CMS programmer much effort, it compromises the structure and formatting of the incoming content.

2.1.3 The Acquisition System

The acquiring system gathers content that was not originally created for the CMS [Fraser 2002]. They include related or similar content from another Web site, external databases or XML feeds. Fortunately, most HTML and relational database content today
are duplicated into XML schemas and made publicly available as XML feeds. Relational databases store information in hierarchical tree structure composed of levels of nodes, which is essentially the format of XML feeds as well.

2.1.4 The Conversion System

If the located content is not available in the format or structure required by the CMS, it must be converted to match the accepted standards of the CMS. The conversion process might involve stripping the content of unnecessary information such as headers, footers, and navigational data, changing the content's binary format to an acceptable format, or making explicit structure changes. For instance, extracting HTML pages or relational and object databases from another site involves content validation through XML schemas and formatting with style-sheets before content can he stored in the repository.

2.1.5 The Aggregation System

Aggregation is the process of bringing disparate information sources into a single overall structure [Boiko 2005]. Most content from external sources are not neatly packaged for use by the CMS; rather they are scattered and need to be rearranged into the desired format after conversion. Moreover, multiple content, each from numerous Websites, have to be aggregated or put together before they can he stored in the repository. Broadly, the aggregation process involves editorial processing, segmentation processing that breaks information into discrete chunks called content components, and
metadata processing, which uses metadata to assemble related data and thus form complete content.

2.1.6 Repository

Finally, content can be loaded into the repository in bulk upon completion of the steps mentioned before. The repository is simply a Web directory that stores binaries, executables, HTML pages, relational or object databases, or XML documents.

2.2 The Management System

The Management System exercises full control over the repository such as providing user-interfaces to edit content, storing meta-content that specify the types of content in the repository and their stage in the lifecycle, and determining content that are ready for publication or disposal. It manages the full lifecycle of content components, from their inception to removal. It also includes an administration section that controls access to content based on user permissions.

2.2.1 Editing repository content

Once content has been collected in the repository, collection system interfaces cannot be used to edit them due to security measures, since all the work performed and people involved in bringing content to the repository stage could he lost in no time. The administrator must approve the editing of repository content and grant access only to authorized users who are personally liable in case of damage to corporate data.
The editing process is complex in that the format of the repository content to be edited must be converted to match that of the editing interface provided, and once again, changed back to the original after the editing process is completed. For instance, to edit XML documents, they must be parsed first, and external style-sheets must be applied to extract them into appropriate controls on a Web form. After editing, once the form is submitted, a new set of style-sheets must be applied to convert form data back into an XML document.

### 2.2.2 Meta-content

The main reason for separating content and meta-content is that they have different workflows and their users or consumers are different [Nakano 2002]. The editorial staff is the primary user of the content whereas the Webmaster alone has the authority to control the look and feel of the individual content. Meta-content provides an elegant, user-friendly look to the content components.

It is necessary to distinguish the meta-content of individual content components from the meta-content that publish all of the content components together on a Website, which is explained in the next section. Whereas the former involves style-sheets that control the color, font, size and spacing of components, the latter controls the layout of these components on a Website relative to other components.

### 2.2.3 Content Lifecycle

This portion of the Management System controls the lifecycle of content
components ranging from storage in the repository, publishing on the Website and disposal after expiry. Only the site administrator reserves the authority to dispose of the content, in order to avoid mismanagement. Content expiration could be as straight forward as based on date and time of the publication or as complex as to require it own workflow. Such a workflow could involve several steps such as submission of content for disposal, review by the editor, suggestion of successor content by the editor, and final approval of the administrator or the managing authority. Once these required steps are completed, content disposal is simply the deletion of those XML documents, HTML pages or relational databases from the repository.

2.3 The Publishing System

The sole purpose of the Publishing System is to retrieve the content components from the repository, apply meta-content, and display them on the Website. The layout, background color of the site, indentation, and other similar formatting that require uniformity throughout the entire site are controlled by the Publishing System. The Publishing System can be further broken down into Templating System, Personalization System, and the Deployment System.

2.3.1 Templating System

Publishing templates are programs that build publications automatically [Boiko 2005]. They guide the creation of publication from repository content, and can be further subdivided into component templates and navigational template.
Component templates control the structure and formatting of the content on the Website. They have a static and a dynamic element; the static template publishes content that remains unchanged on every page of the site, whereas the dynamic template publishes dynamic content that require changes in layout relative to neighboring content. Component templates are generally built using style-sheets that dictate how a content component publishes onto a site. The style-sheet may specify the layout for each element of the component, in which case it overrides the meta-content settings from the Management System, and the layout of a collection of such components onto a Website, which generally forms an article or story. In an XML file, these elements would simply be the information enclosed within matching tags.

2.3.2 Personalization System

The Personalization System gathers and stores user profiles, mainly through cookies, in order to provide personalized content [Nakano 2002]. Such information is also gathered for statistics and simulations to obtain input for the betterment of services offered. Personalized content include tracking a user's interest in a specific product or service and displaying an improvised version or a cheaper alternative.

2.3.3 Deployment System

The Deployment System is only used in very large scale CMSs in multi-national corporations that house distributed content among far away locations. In this case, distributed content from each location must be aggregated and synchronized before publishing to the corporate Website.
3. SYSTEM DESIGN AND ARCHITECTURE

This project is designed to be a small-scale, integrated Web content management system for an online news service built in ASP.NET. The project components are the Website design, the Collection System, the Management System, and the Publication System, all of which function together seamlessly.

3.1 The News Website Design

The news Website is built in C# using ASP.NET and Cascading Style Sheets (CSS) technologies. It is essentially a set of dynamically generated HTML pages with ASP.NET server controls and CSS markups. External CSS are employed to maintain display uniformity across the entire site.

The main page is divided into nine sections, which are simply frame controls, each containing several news links. The nine sections are Headline News, World, Business, Regional, Local, Weather, Sports, Emergency Notices, and Polls. Each section has top level hyperlinks that are news categories or headlines whose content reside on other pages. It is a requirement that there be only one online poll at any given time.

3.2 Collection System

The Collection System (CS) acquires news and other content from the user interface and stores them in the repository as XML documents. The content structure is predetermined, and incoming content is expected to match one of the nine content formats
before they can be displayed on the website.

Figure 3.1 Admin Interface

As seen above, each content type has its own user interface. The user selects the type of content he/she wishes to enter and then proceeds to enter the data. Editing and entering content is only possible in the Admin section of the website and requires user authentication via a username and password.

3.2.1 Content Formats

The news content format has a News element as the root element with *Section*, *Title*, *Date*, *Time*, *Author*, *Image*, *Link*, *Media*, and *Body* elements nested within. The *Section* element of the news content determines if the article is a cover article or not. If the value of the `<Section>` tag is ‘Cover’, then it is a cover article. Other articles have a `<Section>` tag value of ‘Other’. Cover articles have the first link under each section along
with a brief summary and image. Other articles are listed as links to the right and are displayed in subsequent pages. The \texttt{<Link>} tag may contain web links and the \texttt{<Media>} tag may contain audio or video content. A sample Headline news content is shown below in Figure 3.2.

\begin{verbatim}
<HeadLineNews>
  <Section>Cover</Section>
  <Title>Obama’s Fallout Inevitable</Title>
  <Date>04/12/2008</Date>
  <Time>7:04 PM</Time>
  <Author>Mike Hogley</Author>
  <Img>http://localhost:1026/PortalImages/obama.jpg</Img>
  <Link>http://www.obama.com</Link>
  <Media>obama.wmv</Media>
  <Body>Obama’s future in the Presidential election…</Body>
</HeadLineNews>
\end{verbatim}

**Figure 3.2 News Content in XML Format**

The online poll content format has a \texttt{Poll} element as the root element with \texttt{Title}, \texttt{Date}, \texttt{Time}, \texttt{Question}, and several choice elements nested within. The poll content does not require a Section tag since its position within the main page always remains the same. A sample poll content is shown below in Figure 3.3.
<Poll>
  <Question>Bush or Clinton</Question>
  <Choice>Bush</Choice>
  <Choice>Clinton</Choice>
</Poll>

Figure 3.3 Poll content

The Weather content has a Weather element as root, Section, Temperature, Wind, Windchill, and Conditions elements nested within. A sample Weather content is shown below in Figure 3.4.

<Weather>
  <Section>Weather</Section>
  <Temperature>52 F</Temperature>
  <Wind>15 mph</Wind>
  <Windchill>45 F</Windchill>
  <Conditions>Partly cloudy</Conditions>
</Weather>

Figure 3.4 Weather content

Emergency Notices has similar XML content with Section and Body tags. It has its own XSD schema for content validation. Other content follow similar format.

3.2.2 Content Acquisition Interface

The interface for acquiring news and other content consists of a Web form with an ASP.NET server control for each of its elements, except for the Media Link element,
which requires a server object control for uploading image, video or audio data into the content repository. The author enters the news content metadata such as title, and the article itself in these controls, and submits the form for processing by the web application. The user interface for managing Business News content is shown below. *FCKEditor* is embedded within the page to provide the user with a variety of formatting tools. It is a GNU freeware that captures all formatting and passes it on to the ASP.NET application layer. The application layer extracts the formatting and stores it as metadata within the content, which is subsequently used by the XSL style-sheets during content display by converting the metadata into appropriate XHTML metadata. Once entered, the content is converted to XML format by the built in content generator.

![Figure 3.5 Content entry interface](image.png)
3.3 The Management System

In the Admin section, each content type is managed in two parts. The first part displays the list of all the articles of that content type. This is achieved with a separate .aspx file that parses the entire XML repository for all entries of a certain content type and lists them in order. For instance, Figure 3.6 lists all Business News content by parsing the Business.xml file in the XML repository and listing the titles and their images on its own page. This functionality is coded in the Business.aspx file and forms the first part of the process. Link is provided to the right for managing the displayed entries. Once the link to the right of a content entry is clicked, functionality is transferred to the BusinessEdit.aspx file that displays the article along with its metadata in ASP.NET user-controls. The article body is displayed in the embedded FCKEditor control. The users can now edit or delete the content body and the associate image/media and save the changes, which are written to the XML database. Likewise, when new articles are entered, content is automatically added to the respective XML file in XML format. It is important to note that the XML tags are automatically generated from the server controls. Figure 3.6 displays the screen after a new Business News content is added and saved.

![Figure 3.6 Content addition](image)

No validation tools are provided at this time to check for human errors in spelling.
and against false data. Likewise, the Weather, Poll and Emergency Notices content types have separate Web-form interfaces.

### 3.3.1 Enforcing content format

All content types have their own established structure and it must be ensured that incoming content follow the same structure if they are to be displayed on the site. This can be achieved through XML Schema Definition (XSD) files, which list all elements, attributes, entities, and their contexts. XSD schemas enforce constraints on the form or structure an XML document takes, but with much flexibility.

The following is an XSD schema for Business News content that lists all the elements that business news content should have, in that sequence. This XSD schema specifies that the first XML tag should be a `<Section>` tag, followed by a `<Title>` tag, and so on…

```xml
<xsd:element name="Business">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Section" type="xsd:string" minOccurs="0" />
      <xsd:element name="Title" type="xsd:string" minOccurs="0" />
      <xsd:element name="Date" type="xsd:string" minOccurs="0" />
      <xsd:element name="Time" type="xsd:string" minOccurs="0" />
      <xsd:element name="Author" type="xsd:string" minOccurs="0" />
      <xsd:element name="Img" type="xsd:string" minOccurs="0" />
      <xsd:element name="Link" type="xsd:string" minOccurs="0" />
      <xsd:element name="Media" type="xsd:string" minOccurs="0" />
      <xsd:element name="Body" type="xsd:string" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```
It should be noted that the XSD schema shown in Figure 3.7 states nothing about the attributes or the length of the content. Similarly, other content types have their own XSD schemas that validate the structure of the content before display.

### 3.4 The Publishing System

The publishing system has two functions: Loading the main page with links to the content under appropriate sections as stated in the Section element of the content, and displaying the content itself uniformly when these links are chosen. Each XML file to be published has its own XSL formatting style-sheet that it is linked to, as a declaration in the header of the XML file. The style-sheets essentially convert raw XML data into XHTML for presentation. They specify the position, indentation, text spacing and all other aspects of the physical layout for each content type. Further, CSS style-sheets specify the font and color attributes, among many others, of the content. Each content type has its own XSL style-sheet. The publishing system parses the XML repository for XML files, retrieves them from the repository, and publishes them as hyperlinks on the main site, in appropriate sections based on the section element.

There are three parts to the Publishing System controlled by three different style-sheets. The first style-sheet parses the XML repository and collects all the articles that
are to be displayed on the main site. It selects all content that have a value of ‘Cover’ or ‘Other’ in the Section tags. The ‘Cover’ articles are displayed to the left with the associated image and a brief summary. The ‘Other’ articles are listed to its right with a hyperlink only. Both, the Cover and Other articles are listed in their belonging sections of the site, for instance, all business news content would be listed under the Business section. The following figure illustrates this.

![Content sections](image)

**Figure 3.8 Content sections**

At the bottom of the section, a *more >>* hyperlink is displayed, which when clicked, displays all articles in that section on a subsequent page within the same window, and is controlled by the second style-sheet. Figure 3.9 below illustrates this.
Internet search giant Google Inc. unveiled a new feature Tuesday for its popular mapping programs that shines a spotlight on the movement of refugees around the world. The maps will aid humanitarian operations as well as help inform the public about the millions who have fled their homes because of violence or hardship, according to the office of the U.N. High Commissioner for Refugees, which is working with Google on the project. “All of the things that we do for refugees in the refugee camps around the world will become more visible,” U.N. Deputy High Commissioner for Refugees L. Craig Johnstone said at the launch in Geneva. Users can download Google Earth software to see satellite images of refugee hot spots such as Darfur, Iraq and Colombia. Information provided by the U.N. refugee agency explains where the refugees have come from and what problems they face.

Google Maps
by Software Services
Google, Inc.

Figure 3.9 Headline News article

This style-sheet lists all articles in sequence and further provides a complete story hyperlink at the end of each article, which is controlled by the third style-sheet. This style-sheet displays each article in its entirety on a subsequent page, within the same window, along with the associated images, links and media. News content with images or media is positioned so that the text flow inline with the image or video, which are always placed at the top-left hand corner of the page. An advertisement banner may be displayed beneath the image.
4. TESTING AND EVALUATION

In this section, different evaluation criteria for the project will be analyzed and documented. They will form the KPIs (Key Performance Indicators) that determine a successful implementation of the project. The KPIs are successfully demonstrated in the following section.

4.1 Website and Layout

Criteria 1:

The website should be subdivided into nine different sections: Headline News, World, Business, Regional, Local, Weather, Sports, Emergency Notices, and Polls. The top banner of the website should display links to the above sections as well.

Criteria 2:

No content should be stored as HTML pages but as XML pages only, with .xml extension. All pages will be generated dynamically by converting XML content into XHTML pages on the fly via XSL style-sheets.

4.2 Admin Section for Entering and Editing Content

Criteria 1:

A separate interface should be provided as Admin section that is restricted to the general public via password authentication. This interface would be used for adding new content and editing existing content and only content editors should carry the username
and password to access this section.

Criteria 2:

A separate user interface should be provided for each content type. Functionality should be provided to manage existing content (Edit/Delete) and to enter new content including images and audio/video media. All content should be listed by their title and associated image. Changes should be reflected on the website immediately.

Criteria 3:

After existing content is edited or new content is entered, and the Save button is clicked, all content should be saved to the respective XML file. Likewise, when existing content is deleted, it should be removed from the XML file along with its links and media content.

4.3 XSD Schemas for validating content

Criteria 1:

Before content is displayed from the XML repository, they must be checked against their respective XSD schemas for validation. If there is an exact match in the format, the content should be displayed. Else, an application error should be thrown in a user-friendly format stating that the link is unavailable at the moment and the user should be navigated back to the Homepage.
4.4  XSL Style-sheets

Criteria 1:

Under each section, the cover article should be displayed to the left with an associated image and a brief summary. Beneath the cover article, a complete story > link should be displayed that opens the full cover article on a new page in the same window. To the right, links to other articles should be displayed. A more >> link should be displayed at the very bottom that takes the user to a subsequent page with all the articles within that section. In addition to the above formatting, the exact positioning of text, images and media should be controlled by the style-sheets.

Criteria 2:

When links to individual articles are clicked, each article should be displayed along with all associate image and media content in its own page. All content should be displayed in the same window and a new window will never be used except in cases where a link to another website is provided within the content or as external links. A separate style-sheet should be used for Home, Sections, and individual articles.
5. TESTING RESULTS AND CONCLUSION

The Website, Online News Portal, is hosted at http://www.saprfid.com and can be accessed by most modern browsers. The top banner lists the different sections such as Headlines, World, Regional etc. The core website is also subdivided into nine sections. Each section lists the title of the first article – the Cover article – right beneath the section heading, and the first few lines of the article follow, with a link to the rest of the article. Links to other articles are displayed to the right, under the same section.

5.1 Website and Layout

Criteria 1:

![Figure 5.1 Main site](image)

As seen in the image above, the top banner lists the links to all ten sections followed by a logo image. To the right, links to cover articles from each news section are
displayed. Clicking any of these links takes the user directly to that individual article, processed by a separate style-sheet at the article level.

Criteria 2:

![Figure 5.2 FTP files directory](image)

No HTML pages are stored on the server. All webpages are dynamically generated using XSL style-sheets from XML content after content validation using XSD schemas. Figure above displays a snapshot of the XML content in the DataFiles directory on the FTP server of the Web Host.

### 5.2 Admin Section for Entering and Editing Content

Criteria 1:

Managing site content is restricted by user authentication by means of a username
and password that are possessed only by content editors and site administrators. General users are free to peruse news articles and other content but are not allowed to modify content. As seen below, when an invalid username or password was entered, the user was denied entry to the Admin section.

Figure 5.3 Login demonstration

Criteria 2:

Figure 5.4 Content Management section
Figure 5.5 demonstrates editing a Headline News article. The title of the 3rd article was edited to reflect the change and a new Google logo was also added to this article. All changes were instantaneous and were displayed on the site immediately, shown in Figures below.

**Figure 5.5 Headline News Editing**

**Classical painting**

By Minnie Driver  
Updated: 04/14/2008 10:26 PM

THIS ARTICLE WAS EDITED AT 6:12 PM ON APR 27

Popular artist Quincy Adams is displaying his works at the New York Museum this Saturday on a special debut with proceeds going to the local charities. Patrons are also encouraged to buy tickets to the evening dinner and social gathering.

**Figure 5.6 Editing changes displayed**
Criteria 3:

The following demonstration shows a sequence of steps taken to successfully delete an article from the Admin section. The last image in the sequence shows the Headline News section that is rightfully missing the link to the deleted article.

**Figure 5.7 Webpage before deleting article**

**Figure 5.8 Deleting article from Admin section**
Figure 5.9 *Classical Painting* article deleted in Admin

**HEADLINE NEWS**

<table>
<thead>
<tr>
<th>Title</th>
<th>Image</th>
<th>View</th>
<th>Edit/Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nano Bomb Invented</td>
<td></td>
<td>View</td>
<td>Manage</td>
</tr>
<tr>
<td>Obama’s Bitter Fallin</td>
<td></td>
<td>View</td>
<td>Manage</td>
</tr>
<tr>
<td>Google Earth, U.N. Pair Up EDITED APR27 4:21 PM</td>
<td></td>
<td>View</td>
<td>Manage</td>
</tr>
<tr>
<td>Blast Kills 9 at Iranian Mosque</td>
<td></td>
<td>View</td>
<td>Manage</td>
</tr>
<tr>
<td>People Sicken from Cereal</td>
<td></td>
<td>View</td>
<td>Manage</td>
</tr>
</tbody>
</table>

You are viewing page 1 of 1.

Figure 5.10 Article removed from Webpage
5.3 XSD Schemas for validating content

Criteria 1:

An article in the Sports section was edited with a purposeful omission of the Author. When attempting to save the article, an error message was thrown alerting the user to enter the Author’s name, which demonstrates the success of the XSD schemas.

Figure 5.11 Required field missing

5.4 XSL Style-sheets

Criteria 1:

Hyperlinks to articles under each section were displayed correctly using the appropriate style-sheets. The \textit{more >>} and \textit{complete story >} links were also displayed and
functioning properly for navigating to all articles in that section and to individual articles, respectively.

Criteria 2:

Individual articles in all sections were generated dynamically and published as expected. The style-sheets successfully placed all images and media in appropriate placeholders within each article.

In summary, all testing criterias were demonstrated successfully. The content repository was already loaded with valid XML content. The XSL style-sheets used for displaying hyperlinks to articles, weather and other content use pre-defined templates to position them on the page, under appropriate sections based on the Section element of the content. The style-sheets are versatile enough to accommodate different content lengths to produce HTML documents as per the expectations.

The project idea was borne from the need of a local firm specializing in corporate Website building and management. Since many of their clients are in the same industry
and require similar Websites, a Web content management system would enable the company to maintain on an on-going basis high quality Web sites with little effort.

The designed solution was built using the ASP.NET paradigm in the Visual Studio 2003 (.NET version) environment. The front and back end functionalities such as the XSL style-sheets for publication and XSD schemas for validation were programmed in C#.NET.

Possible improvements could include an automated interface for gathering online content from trusted sources that are obtained and published on a periodic basis without any manual intervention.


APPENDIX A - PROGRAMMING CODE