

Report

SA1033A: Enterprise Internet Solutions

Nicolas Lanquetin
0604918@abertay.ac.uk

27th April 2007



UNIVERSITY
of
ABERTAY DUNDEE

University of Abertay Dundee
School of Computing & Creative Technologies

Contents

Requirements of the WFDF	1
1 Suitability of ASP.NET	3
1.1 Basic Differences	3
1.2 Price & Ease of Maintenance	4
1.3 Benefits for the User	4
2 Mobile Systems	5
2.1 Content for Mobile Devices	5
2.2 ASP.NET for Deployment of Mobile Applications	5
3 Content Management and the Publication of Information	7
3.1 Possible Content Management Systems	7
3.1.1 Own CMS Implementation	7
3.1.2 Adobe Contribute	8
3.1.3 TYPO3 CMS	8
3.2 Generation of Excel Sheets and Alternatives	9
3.3 Conclusions	9
4 Security Implications	10
4.1 Attacks, Prevention & Reaction	10
4.2 Roles and Rights	11
Bibliography	13

Requirements of the WFDF

The World Flying Disc Federation (henceforth WFDF) has asked to be advised regarding the following requirements:

1. The public area of the application will provide information about the tournament, the timetable, results, etc. The existing Championship application runs under ASP Classic but, because they want a complete rewrite, have heard that they should take the opportunity to move to a solution based on ASP.NET. They are interested to learn about the basic differences, whether a .NET solution will make their application any cheaper to build and maintain, and also whether the adoption of ASP.NET would bring any benefits to the end user.
2. The WFDF is aware of the increasing trend towards people consuming information via their mobile phones and other devices. They have asked you to suggest some possible content that would be suitable for delivery on mobile devices and to comment on the suitability of ASP.NET for the deployment of such an application.
3. All information relating to the championships will be stored in a central data store. A content management system will be used to generate content, update it and prepare it for distribution in various formats. They would like you to make recommendations about the type of content management system that will be most suitable for their needs and about how they should implement the system. In particular, news releases will be issued in PDF format. They would like results (which are stored in the content management system's data store) to be available to the media as Microsoft Excel spreadsheets, but are not sure if this is possible. If it is not, they would like advice on how they can issue results in a format that the media can use to manipulate data – they are aware that most journalists don't want to have to (or have the expertise to) use import/export techniques.
4. The WFDF is acutely aware of the need to preserve the integrity of their data, and are aware that their application is likely to be the subject of malicious attacks from people who do not share their love of the sport and think it 'geeky'. You are therefore asked to report on security measures that you would apply. You will not have access to network security settings, so authentication would have to be done within your application. They are also concerned that people with access to the content management system should only be allowed to perform certain tasks. An

official who is allowed to enter the players for a match, for example, should not be able to enter the scores for that match, and they don't want him/her to be able to enter data for matches other than those for which they have authorisation – for example, the administrator of the Masters tournament should only be able to work with information about matches in that tournament. They appreciate that there has to be some form of demarcation or 'ring fencing' in operation, but they are not sure how that should be done.

1 Suitability of ASP.NET

According to requirement 1 (p. 1), the client is already using classic ASP for their current system and is considering to migrate to ASP.NET. However the migration of a complete application is difficult and very time-consuming, as ASP.NET is a completely new development tool (Peterson, 2002, p. 9) (Harris and Macdonald, 2002, p. 6). Therefore a complete rewrite as proposed by the client (c.f. requirement 1 on p. 1) is necessary anyway.

As there are many aspects to consider for a comparison of ASP and ASP.NET, this report will focus on the client's requirements:

1.1 Basic Differences

The move to ASP.NET is justified, as classic ASP has various limitations. Harris and Macdonald (2002, p. 2 et seq) highlight some of the most important disadvantages of classic ASP, which are solved in ASP.NET: The language used in classic ASP is VBScript. The latter is less functional, based on a primitive design¹ and slower². ASP.NET, on the other hand, is based on the Common Language Runtime (CLR) and supports many *modern* programming languages. Furthermore, ASP.NET separates controller³ and view⁴. This makes it easier to separate tasks in a team, and also ease the maintenance of the code. In ASP, however, controller and view are both combined into a single file. Furthermore, the state management in classic ASP is not appropriate for web farms⁵, a problem solved in ASP.NET. The performance also increases with ASP.NET, as the framework supports numerous caching mechanisms (Harris and Macdonald, 2002, p. 4 et seq). In addition, the debugging and testing is made easier in ASP.NET, and the security management and crash tolerance has been improved as well (Harris and Macdonald, 2002, p. 5). Finally the process of providing and consuming web services has been drastically simplified in ASP.NET. This is an important feature for the client, as it is very likely that web services will be used for accommodation information.

¹VBScript is procedural, rather than object-oriented

²Like all script languages, VBScript is interpreted rather than being compiled

³the business logic

⁴interface elements

⁵A web farm consists of more than one server running the application.

1.2 Price & Ease of Maintenance

The development of a new application is less expensive, when implemented with ASP.NET. Firstly, no initial framework development is needed, since the framework is the ASP.NET technology itself. In addition, the development speed is drastically increased through the framework capabilities, including the powerful server controls⁶; the object-oriented and event-driven architecture; and the separation of configuration, model⁷, view and controller. The architecture allows developers to write less code for the same results and ease the maintenance of the application. Moreover, developers using other programming languages can easily participate in the development, since ASP.NET is not bound to VB only: Over 20 languages are supported (Peterson, 2002, p. 7) and new one, such as C#⁸ or J#, are introduced. Therefore developers don't need to get acquainted with a completely new language and can start developing right away. To summarise, the important shortening of development time and the ease of maintenance make the application much cheaper.

1.3 Benefits for the User

Most important aspects for the user includes access time, design consistency and compatibility. Classic ASP is only used for server-side processing. Consequently, all client-side interaction must be implemented from scratch. ASP.NET, however, generates code which can be processed on client-side, lessening the server round-trips and giving a faster and more interactive feeling for the user. Additionally, the use of templates and web-controls guaranties a consistent layout. Finally, the interface presented to the user, automatically adapt the content to the client's browser, and therefore maximise compatibility.

⁶Additionally, own user controls can be defined (Peterson, 2002, p. 26).

⁷the stored data

⁸C# is very similar to Java. Microsoft intended Java developers to rapidly adopt that language in order to migrate to ASP.NET.

2 Mobile Systems

Mobile systems differs from PC systems in many points: the supported markup language (HTML, WML, cHTML¹), the browser used, the number of display lines (or the screen size accordingly) and cookie and Javascript support ([ASP.net, 2007](#); [W3Schools, 2007](#)). All these factors have to be taken into account when creating an application and its content for mobile systems.

According to requirement 2 (p. 1), the client asked to be advised on possible content suitable for mobile devices, and if ASP.NET is a good choice for the deployment of mobile applications.

2.1 Content for Mobile Devices

The content for the application will mainly consist of events timetables, events results, and third party content, such as accommodation information. All these contents are possible and easily representable on a mobile device. Nevertheless, because of the limitations of mobile devices mentioned above, the client must be aware that information must be kept to a minimum. This is especially the case for devices supporting only WML and having a small screen size. Some older cell phones for instance can only display 4 lines of text. In addition, the output types are restricted to WML for older devices, and simple HTML for PDAs, Blackberries and other recent devices. It is not advised to use other media types, such as PDFs, Excel sheets or larger pictures (e.g. for the accommodation); they are not suited for displaying on mobile devices.

2.2 ASP.NET for Deployment of Mobile Applications

Regarding the implementation using ASP.NET, an extension to the .NET Framework can be used. The latter is called the Microsoft Mobile Internet Toolkit (MMIT) or simply .NET Mobile. Similar to using server controls, the content must be represented using predefined *mobile controls* shipped with the MMIT ([MSDN, 2007](#)) ([W3Schools, 2007](#), Mobile Reference). The huge advantage of ASP.NET, is that only one single code is required, even though many outputs are possible. The code must be written using the

¹compact HTML (cHTML) for Japanese i-mode phones

mobile controls mentioned above. As for the output, it is then automatically generated (in HTML, WML or cHTML) by the framework, depending on the device (or the browser accordingly) requesting the page ([W3Schools, 2007](#), Mobile Example). This makes the development very fast and easy, since the interface is produced dynamically for every device. Additionally, developers do not have to worry about compatibility issues, because they are taken care of by the framework. Last but not least, the manageable amount of mobile controls makes the code easy to produce and maintain.

The MMIT extension exists since .NET 1.1² and is widely and successfully used. The client can be sure that ASP.NET is a fast and easy choice to implement their mobile interface, especially if the intention is to use ASP.NET for the web application anyway.

²At the time of writing, .NET 3.0 is already release.

3 Content Management and the Publication of Information

According to requirement 3 (p. 1), the client needs advice on which type of Content Management System (henceforth CMS) to use in accordance with their needs, and how that CMS should be implemented. The CMS should allow generation and distribution of content in various formats, including PDF format for the news, and if possible Excel sheets for events results. Furthermore, requirement 4 (p. 1) states that role and right settings should be possible. The latter will however be discussed in further detail in chapter 4 (p. 10).

Possible solutions are the development of a CMS tailored to the specific needs of the client (section 3.1.1), or the use of already existing CMS, such as the commercial solution ‘Contribute’ from Adobe (section 3.1.2, p. 8) or the open-source solution ‘TYPO3 CMS’ (section 3.1.3, p. 8). The following section will address what a CMS would require if developed from scratch, and also which already existing CMS (commercial and free) would be the most appropriate for the client’s needs.

3.1 Possible Content Management Systems

3.1.1 Own CMS Implementation

An own CMS implementation is appropriate when there is a limited amount of users, task to perform, and output formats. This is the case for the WFDF, as the application is mainly used for an annual event and requires only a few tasks. Moreover, the output formats consist only of HTML, PDF, WML and Excel (or similar). It can be assumed that the client wants a CMS running on ASP.NET (cf. requirement 1, p. 1; chapter 1, p. 3). As mentioned in the previous chapters, ASP.NET is very powerful, both for web applications and mobile applications. Compared to other possible technologies, ASP.NET ensures a relatively easy and fast implementation of the client’s requirements.

However, it is important to launch the application in time, as the World Championship starts in July. Therefore, acquiring a third party CMS can be a safer choice: Instead of being developed from scratch, the application only needs to be adapted to the organisation’s needs, which is cheaper and drastically shorten the lead time.

3.1.2 Adobe Contribute

Adobe Contribute is a relatively cheap¹ mid-range CMS (Adobe, 2007). Installed with the Adobe Contribute Publishing Server and an extension, it partially fits the client's needs. Firstly it complies with the requirement for roles and rights, as it has a sophisticated user management system. Additionally, provided that a third party extension is installed, the content data can also be retrieved from the client's central data store (instead of Adobe Dreamweaver's editable regions). Regarding the publishing in different media formats, Contribute includes a free copy of FlashPaper 2. The latter is a program for converting documents to PDF format, which the client needs for the news publishing (Skalbeck, 2004, Six Selling Points for Contribute). Unfortunately Contribute depends on Dreamweaver and must be installed on every system. Furthermore it is not possible to generate Excel files from the stored data. Section 3.2 (p. 9), however, covers how this problem could be solved.

3.1.3 TYPO3 CMS

TYPO3 CMS is a solution, which is free² and covers all client requirements (TYPO3, 2007). TYPO3 CMS is a very powerful and successful open-source CMS, implemented in PHP. It comes with many core features which would satisfy most users. It is however possible to install various extensions for the CMS in order to adapt the system to the specific needs. One of the latter is a PDF extension (Klinger, 2004), which allows output in PDF format and can be used to generate the news (see requirement 3, p. 1). Furthermore TYPO3 CMS also generates WML and WAP, an interesting feature with regard to requirement 2 (p. 1) of the client. Additionally, it supports an undo history where all changes made from the beginning can be undone. The content is versioned, so that all changes can be traced and/or reverted, similar to a WIKI system. Regarding requirement 4 (p. 1), the CMS also supports group and user privileges, enabling the administrator to grant granular rights. It is also possible to get content updates approved before publishing. As for the Excel sheet generation, TYPO3 CMS does not support the creation of a real³ Excel file. It is however possible to directly generate CSV⁴ data of parts of the data store (TYPO3, 2007, Task Center). Given the fact that CSV files are by default associated with Microsoft Excel, the journalists will have no problem opening the CSV files. With regard to usability, the editing is also made very easy, as TYPO3 CMS provides a Rich Text Editor (RTE), allowing WYSIWYG⁵ content editing. The user interface is very similar to Microsoft Word

¹USD 238 per seat (including the Adobe Contribute Publishing Server)

²licensed under GNU/GPL

³The reason why there are no support for Excel file generation, might come from the fact that Excel (till Office 2003) is a proprietary format and therefore not supported in open-source projects.

⁴Comma Separated Values

⁵What You See Is What You Get

and intuitive to use. Therefore, only minimal training is required for the editors. Finally, TYPO3 might be too complex⁶ for the relatively small tasks required by the client. It is however important to consider future changes. Any new requirement might already have been implemented as an extension that the client can easily install in the CMS, making TYPO3 CMS also an interesting choice in the long run.

3.2 Generation of Excel Sheets and Alternatives

Most CMS offer an interface for plug-ins or extensions. If the CMS does not offer any support to create Excel files from stored data, it is still possible to create an own extension for this purpose. In any case, it is advised to provide RSS feeds for the results in addition to the Excel files, knowing that many news websites can easily include them. Furthermore it could also be possible to provide a web service for the results. However, since this is a more complicate and time-consuming step, there should be a guaranty that other important websites will consume the web service.

3.3 Conclusions

An own CMS implementation can be considered, since the client does not have many requirements and a limited number of users and output formats. The application would fit the client's requirement perfectly. However, beside of the high costs, the lead time might be too long, and the client risks to launch the application behind time. Using an existing CMS, in turn, guaranties a much faster deployment and a lower price. Adobe's Contribute CMS is a good solution, if the editors or content managers are used to work with Microsoft Office products and Adobe Dreamweaver. On the other hand, it needs to be tweaked to support all user requirements and is not an ideal solutions regarding the publishing of results. TYPO3 CMS is the best choice as it fulfill all requirements from the start and is also more interesting in the long run.

⁶[CMSMatrix \(2007\)](#) shows that TYPO3 CMS has much more functionality and features than any other competing CMS.

4 Security Implications

The likeliness of an attack on the application is very high according to requirement 4 (p. 1). To prevent attacks, it is important to know which one are possible on a web application, and how the risk of such an attack can be minimised. There exist classic attacks and security principles which must be taken into consideration ([Kriha, 2007a](#)).

4.1 Attacks, Prevention & Reaction

An important weak spot used by most attacks is the user input (e.g. over a web form): One possible attack is the SQL injection, consisting in submitting malicious SQL code, which will be executed on the server side. This can be prevented by processing user input as strings and correctly validating input. When processing the user input, the order of security check and validation is very important as well: First the authorisation of the user must be verified, and only then should the input be validated and processed. By respecting the right order, data integrity can be assured, and dangerous attacks (e.g. directory traversal) can be prevented. Furthermore it must be prohibited to let the user submit HTML code directly, as it can be used for cross-site scripting (XSS). It is necessary to either filter HTML tags which could be used to inject arbitrary code, or to provide a special markup language which can be transformed to the correct format by the application.

Special attention must also be spent on state holding and session handling. It should be avoided to do the state holding on the client side (e.g. by passing variables in hidden fields), as they can be modified by the user. A similar problem applies to the use of session tokens¹ (or session IDs), as they could be used for a session take-over. In order to prevent such an attack by a man-in-the-middle, the use of SSL is crucial. Furthermore, the session must time out after inactivity of the user: Most of the applications transmit the session token in the URL, when cookies are disabled. These URL are sometimes cached by search engines such as Google. Without the timeout, a malicious user could follow the link with the session token in it and act with an authorisation level belonging to another user.

Another important security principle is to avoid ‘security through obscurity’. The latter is used when the security relies on the secrecy of design and implementation. If the attacker has knowledge about a system, but still cannot compromise it, only then the system is truly secure.

¹strings which identifies a user’s session

Beside trying to prevent attacks, it must also be thought of emergency measures in case an attack comes through. A fail save stance can be used in this case, preventing the use of the system, even for legal operations. This way, the data integrity cannot be compromised. Furthermore it is important to have physical replications and backups of the application and its content, in order to roll back the changes to a safe state.

The client asked to be advised about security measures on the application layer. However, it must be stressed that – according to the attackers profile – it is also likely that attacks will occur on the physical layer, e.g. with a DOS² attack on the server itself. Securing the application alone is consequently not enough: The security should rely on several line of defense, according to the principle of ‘defense in depth’. The application could for instance run behind a proxy server and be protected by a host based firewall (Kriha, 2007b).

Most of the principle mentioned above must be considerate for the development of an own CMS. When acquiring a commercial or open-source CMS, the chances are high that the security is better than in the own developed application. However, hackers might more easily find a vulnerability to exploit, as they know the system.

4.2 Roles and Rights

The maybe most important security principle is the principle of least privilege (need-to-know, need-to-do): Not more rights should be granted to a user, than needed to fulfill a task (Kriha, 2007c). This principle can be applied not only to users, but also to processes. Hence, the process running the application (e.g. an Apache web server) should have its own system user account with restricted rights on the operating system. In a worst-case scenario, only data can be lost or compromised, for which this user had access to. The same principle should also be used when creating roles and rights for users on the application layer. The best approach for creating a new user, is to satisfy the principle of ‘default is deny’: By default, all actions are forbidden for the user. Rights are only granted when required for specifics task. This way the user can never do more than he should be able to.

The client highlighted the importance of granular rights for the CMS users (see requirement 4, p. 1). The best solution would be to create an administrator user, responsible for a set of matches. The administrator has the right to work only on the matches he has been assigned to, and he can further give rights to users that should work on the matches. For every match, the administrator can assign users with the right to update match information and/or match results. Users sharing the same rights, can further be classified into

²Denial Of Service

groups sharing common rights. This makes the maintenance easier, as privilege changes can be made to an entire group in only one operation.

The reviewed CMS ‘Contribute’ and ‘TYPO3 CMS’ (see chapter 3.1.2 and 3.1.3, p. 8 et seq) both supports group and user management and the management of privileges is granular enough to separate the tasks of updating match information and match results separately.

Finally, it is important to be consequent about the abuse of privileges and forbid further use of the account. This intervention is not only a disciplinary action, but also a necessary step to guaranty data integrity. As all activities can be logged and are stored using a versioning system, it should be easy to trace the changes and find out who is responsible. To avoid having a situation where an account could have been compromised by a man in the middle, it must be ensured that authentication data is handled with care, e.g. not cached or send in plain text.

Bibliography

- Adobe. 2007. “Adobe Contribute: Complete feature list.”. Available at:
<http://www.adobe.com/products/contribute/features/>.
- ASP.net. 2007. “The Official Microsoft ASP.NET 2.0 Site: Microsoft Mobile Internet Toolkit.”. Available at: <http://www.asp.net/mobile/intro.aspx>.
- BeanSoftware. 2007. “How to make ASP.NET Applications that support mobile devices.”. Available at:
<http://www.beansoftware.com/ASP.NET-Tutorials/WML-Mobile-PDA.aspx>.
- CMSMatrix. 2007. “The Content Management Comparison Tool.”. Available at:
<http://www.cmsmatrix.org/matrix/cms-matrix>.
- Harris, Steve and Rob Macdonald. 2002. *Moving to ASP.NET: Web Development with VB.NET*. Berkeley, CA 94710: Apress.
- Kieley, Jim. 2001. “Migrating to ASP.NET: Key Considerations.”. Available at:
<http://msdn.microsoft.com/en-us/library/aa479019.aspx>.
- Klinger, Ursula. 2004. “HTML to PDF converter.”. Available at:
http://typo3.org/documentation/document-library/extension-manuals/pt_html2pdf/0.1.0/view/.
- Kriha, Walter. 2007a. Attacks on Web Applications. In *Advanced Internet Security*. University of Applied Science Stuttgart, Germany. Available at:
<http://www.kriha.de/krihaorg/docs/lectures/security/webattacks/webattacks.pdf>.
- Kriha, Walter. 2007b. Firewall architectures and types. In *Advanced Internet Security*. University of Applied Science Stuttgart, Germany. Available at:
<http://www.kriha.de/krihaorg/docs/lectures/security/firewallarc/firewallarc.pdf>.
- Kriha, Walter. 2007c. Security as a System. In *Advanced Internet Security*. University of Applied Science Stuttgart, Germany. Available at:
<http://www.kriha.de/krihaorg/docs/lectures/security/secintro/secintro.pdf>.
- Kriha, Walter. 2007d. Web Application Security. In *Advanced Internet Security*. University of Applied Science Stuttgart, Germany. Available at:
<http://www.kriha.de/krihaorg/docs/lectures/security/webapps/webapps.pdf>.

- MSDN. 2007. "Mobile Controls." Available at:
<http://msdn.microsoft.com/en-us/asp.net/aa336586.aspx>.
- Peterson, John. 2002. Migrating To .NET. In *ASP.NET Developer Conference & Expo – Spring 2002*. Available at: <http://www.asp101.com/articles/john/migration/>.
- Skalbeck, Roger V. 2004. "Macromedia's Contribute 3.0 – Sophisticated Web Editing Made Simple." Available at: <http://www.llrx.com/features/contribute.htm>.
- TYPO3. 2007. "TYPO3 CMS: Feature list." Available at:
http://typo3.com/Feature_list.1243.0.html.
- W3Schools. 2007. ".NET Mobile." Available at:
<http://www.w3schools.com/dotnetmobile/>.