

# Using J2EE Web Services for the Implementation of an Application on Demand Portal

Ch. Bouras	A. Gkamas	D. Primpas	K. Stamos
Research Academic Computer Technology Institute, & Computer Engineering and Informatics Dept., Univ. of Patras, bouras@cti.gr	Research Academic Computer Technology Institute, & Computer Engineering and Informatics Dept., Univ. of Patras, gkamas@cti.gr	Research Academic Computer Technology Institute, & Computer Engineering and Informatics Dept., Univ. of Patras, primpas@cti.gr	Research Academic Computer Technology Institute, & Computer Engineering and Informatics Dept., Univ. of Patras, stamos@cti.gr

## Abstract

*This paper describes the design of the ASP-NG portal, which is a portal for providing the user with the necessary interface in order to access an Application on Demand (AoD) service. The ASP-NG portal is responsible for the interaction with the user of the Application on Demand service. The implementation of the ASP-NG portal is going to be based on the Web Services of the Java 2, Enterprise Edition (J2EE) platform. The ASP-NG portal will offer to its users the capability to select and customize the language of the user interface in order to present information in their preferred language. Moreover the ASP-NG portal will offer to the portal administrator the capability to customise the look and feel of the ASP-NG portal.*

**Keywords:** J2EE, AoD, Web Services, e-business, e-marketplace

## 1. Introduction

This paper describes the design of the ASP-NG portal of the ASP-NG<sup>1</sup> project. The strategic goal of the international consortium undertaking the ASP-NG project is to research, develop and implement a new middleware for Application Service Providers (ASPs), which will integrate all the required functionality for running an ASP, including new methods such as streaming application, to deliver productivity tools and rich content, secure keying of IP and Digital Rights Management (DRM) with capabilities of running on different platforms and operating systems. The ASP-NG project is aimed at providing ASPs, small, medium, or large enterprises (SMEs and Telecoms), with a complete set of features required to run an ASP service.

Using the Application on Demand service the user can rent and use an application even if he has not purchased the complete application package. Instead, the user agrees with the ASP company to rent the application and each time the user only needs to download only the necessary modules for a specific session. The user is not able to use the application outside of the mutually agreed frame with his ASP, but he experiences the application as if he actually used the complete application package and installed it in his computer. The whole procedure is handled by the AoD system and is completely transparent to the end user.

The main role of the ASP-NG portal is to provide to the user the necessary interface in order to use the Application on Demand (AoD) service. The ASP-NG portal is responsible for the interaction with the end user. Through the ASP-NG portal the user will be able to browse the available applications, rent and use

---

<sup>1</sup> ASP-NG (ASP-New Game - IST-2001-35354) is a project founded by the Information Society Technologies (IST) Programme of the European Commission.

them. In addition, the ASP-NG portal offers functionality such as searching for specific applications, checking already rented applications, creating “subusers”, that is, users with limited capabilities that are under the direct supervision of the parent user, credit purchase and renewal, viewing transaction history, chatting with other users of the ASP-NG system, asking questions at the ASP-NG portal’s forum, viewing already answered questions and more. For this reason it implements some functionality itself and some functionality is implemented in the other modules of the ASP-NG system, such as the Backoffice module and the CRM module.

Apart from the end user (“client”) the ASP-NG portal architecture defines a number of administrative roles. The administrators are responsible for maintaining and managing parts of the AoD system. It is assumed that the administrators are going to be persons from the ASP company that provides the AoD service. More particularly, the following administrators will have access through the ASP-NG portal:

- Portal Administrators: These administrators will be responsible for configuring the ASP-NG portal.
- Content Administrators: These administrators will be responsible for application and content management (upload application metadata describing an uploaded application, create application packages, remove application from list of available applications).
- User Administrators: These administrators will be responsible for users management (create users, edit users, remove users).
- System Administrators: These administrators will be responsible for managing the rest of the administrator users (Portal Administrators, Content Administrators, User Administrators) of the ASP-NG portal.

In addition, the ASP-NG portal will offer to its users the capability to select and customize the language of the user interface in order to present information in their preferred language. Initially, the ASP-NG portal will support only a small set of languages but the portal administrator will have the capability to add support for more languages. Moreover, the ASP-NG portal will offer to the portal administrator the capability to customise the look and feel of the ASP-NG portal.

## **2. Usage Scenario of ASP-NG Portal**

The ASP-NG portal is a generic web site designed for Applications on Demand (AoD) implementation. The ASP-NG portal’s front-end interface includes a comprehensive set of features designed to facilitate rental uptake and to enhance user experiences with an Application on Demand service. The user will access the AoD service through the ASP-NG portal. The user will have the capability to browse the ASP-NG portal and collect information about the available applications in the AoD server. The ASP-NG portal will offer to the user different ways of browsing (for example browse by category or browse all applications) and searching (for example searching by keyword, searching by category etc).

In order to rent and use an application the user must be registered in the ASP-NG portal. A registered user has to submit his login and password information in order to be authenticated. After passing the authentication control and if the user has enough credits in his virtual account, he can rent and use applications. The ASP-NG system will offer to the user many capabilities for renting an application (rent one application, rent a bundle of applications, etc). The system will also offer to the user the capability to buy credits in different ways, like paying by credit card, paying through a bank account, paying directly or paying using a micropayment mechanism (the first implementation will only simulate a billing system). In addition, after the user has passed the authentication control, he will have the capability to update his profile, which contains all the user-defined details of the user’s account. Through the profile the user will have the capability to access and update information relative to his account such as his personal information, his transaction history, and his system preferences. In addition, the user will have the capability to manage (create, edit, remove) sub-accounts for subusers with limited access to the applications rented by the parent user. The level of access for the subusers will be determined by the parent user. This way, the parent user will be able to create for example accounts for his children by restricting access to applications that the parent finds unsuitable for their age. The subaccounts only have access to the applications through the credits of their parent account (cannot rent their own applications).

Moreover, the ASP-NG portal will offer several methods for communication among its users. For example the ASP-NG portal will offer chat rooms (general chat rooms or chat rooms with specific subjects) and forums (general forum or forums with specific subjects). When a problem occurs or if the user has any questions regarding the operation of the system, the ASP-NG portal will guide the user to the Helpdesk/CRM module (Customers Relationships Management) through which the end-user will have the capability to access help pages (for example guidelines, FAQ) or will be able to contact Helpdesk personnel.

### 3. ASP-NG Portal Architecture

Figure 1 presents the general architecture of ASP-NG system mainly from the ASP-NG portal point of view (This means that some modules which do not interact directly with the ASP-NG portal may not be displayed in this figure). As this figure shows, the ASP-NG portal interacts mainly with the Backoffice module and the users access the ASP-NG system services through the ASP-NG portal. The ASP-NG portal is responsible for the interaction with the user. For this reason it implements some functionality itself and some functionality is implemented on the other modules of the ASP-NG system. In addition the ASP-NG portal accesses the system database through the application server of the Backoffice module, but it has the capability to directly access the portal database. This database is specific to the ASP-NG portal module and keeps information that is related to the display languages and the different presentation skins of the ASP-NG portal. It resides within the Backoffice module, together with the system database, although it can reside in a different module, if it is desired in a specific case. The structure of the ASP-NG portal database is described in detail in section 4.1.

The Backoffice module consists of the application server, the mail server and the database management system (DBMS). The application server provides services to other modules (the ASP-NG portal is using these services) and access to the system database. Other databases, which are module specific, are directly accessed using standard database interface (JDBC). The system database stores information related to the users, their credits, their accounts, the applications, which application has been rented by which user, etc., while the portal database stores different versions (different languages) of the ASP-NG portal, customizable interface, graphics, etc. The CRM module is responsible for handling customers' relationships. The ASP-NG portal will "drive" the end-user to the CRM through hyperlinks for reporting problems and incidences and get connected with the CRM.

The AoD server is responsible for providing the core Application on Demand technology.

The ASP-NG portal will interact with the Backoffice module in two ways:

1. *Interaction with the Backoffice module application server:* In this case the application server will provide service in API style to the ASP-NG portal. This interface will be based on direct communication with the use of Enterprise JavaBeans (EJB) technology and more particularly with the use of Java RMI, which is used by the EJB technology for distributed communication. The EJB technology offers a number of alternative methods for implementing the business logic of a web application. For this reason, the developer can choose between Entity Enterprise JavaBeans, Session Enterprise JavaBeans and Message Enterprise JavaBeans. The Backoffice module will be implemented using Entity Enterprise JavaBeans for managing data persistence in the system database. It will also use Session Enterprise JavaBeans in order to implement business logic and provide access to the Entity Beans. Therefore, the ASP-NG portal module will not have direct access to the Entity Enterprise JavaBeans, but it will rather access the Session Beans, which construct a layer covering the Entity Beans. This technique offers a number of benefits and is called the Session Façade [4].
2. *Interaction with the Backoffice module database:* In this case the ASP-NG portal will collect information directly from the Backoffice module database (and more specifically from the portal database).

There is not any direct interaction between the ASP-NG portal and the CRM module except that the ASP-NG portal will provide links to the CRM module. Through these links the ASP-NG portal will provide to the CRM module the user ID and the user preferred language.

The diagnostic tools will collect information about the status of the ASP-NG portal with the use of an SNMP agent and the SNMP protocol, which will be installed in the web server of the ASP-NG portal.

The interface between the ASP-NG portal and AoD module will be based on the use of Javascript or VB script functionality. In addition the web browser must be compatible with ActiveX/Netscape Plug in for interactive with the AoD client and the web server must be compatible with JavaBeans in order to access the AoD system. More particularly the interface between the ASP-NG portal and the AoD module will be implemented with the following components:

- AoD control: The AoD control is a web component that can run on the user's computer, and can be activated from the web. It comes in three forms: an ActiveX component for use with Internet Explorer, a Java Applet for use with Netscape 4.x or a Netscape Plug-in for use with Netscape 6.2 and above. It exports two functions. The first retrieves some information about the user's computer and is used for example to get the installed components versions on the user's computer (such as DirectX) and compare it to the required version by an application. The second function exported from the AoD control runs the selected application on the user's computer.
- AoD JavaBean: The AoD module uses a JavaBean that exports functions for retrieving directions for executing and buffering an application on the user's computer. Those directions can be used as parameters for the AoD control function that runs the selected application.
- Web and scripting Interfaces: The web and scripting interfaces will provide important AoD functionalities to the ASP-NG portal and are going to be embedded into it. There is going to be a function that will compare the components detected on the user's computer to the application's requirements in the database, and the following pages:
  - Shortcut page: Redirects an application launch request from a start menu shortcut.
  - Scheduling page: Retrieves scheduling directions to the Player on the user machine.
  - Error Log page: Posts error events from the Player on the user machine to the client error log database table.

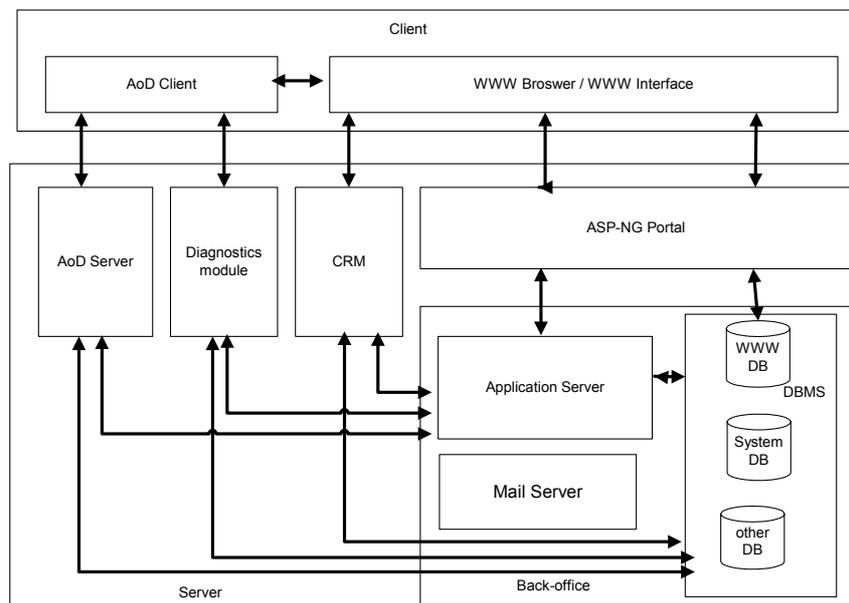


Figure 1 General Architecture of ASP-NG System

#### 4. ASP-NG Portal Design Aspects

The implementation of the ASP-NG portal is going to be based on the Web Services of the Java 2, Enterprise Edition (J2EE) platform ([7]). More specifically, the ASP-NG system will make use of the JavaServer Pages (JSP), Servlet and Enterprise JavaBeans (EJB) technologies.

The J2EE technology provides a component-based approach to the design, development, assembly and deployment of enterprise applications, in order to reduce costs and enable faster enterprise application design and development. Its multitier distributed application model is not tied to the products and Application Programming Interfaces (API) of any one vendor. J2EE applications are made up of components. A J2EE component is a self-contained functional software unit that is assembled into a J2EE application with its related classes and files and that communicates with other components. The web services of J2EE specification defines client components, Servlet and JavaServer pages components that run on the server, and Enterprise JavaBeans components that run inside the EJB container.

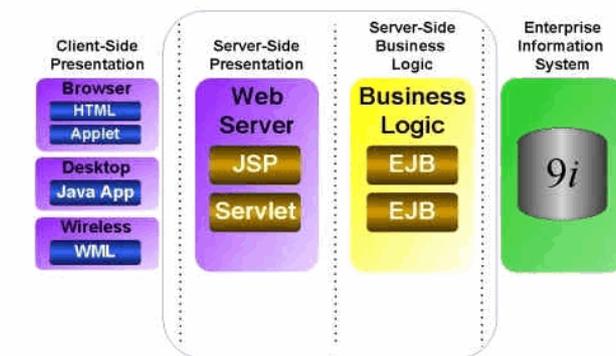


Figure 2 J2EE Web Services Architecture<sup>2</sup>

Figure 2 shows the J2EE Web Services Architecture. In the ASP-NG system the server-side presentation module will be implemented as part of the ASP-NG portal and the server-side business module will be implemented as part of the application server of the Backoffice module. The architecture of the ASP-NG portal is going to be based on JavaServer pages that will mainly contain the presentation data for the ASP-NG portal. In order to separate business logic from presentation logic, the JavaServer pages are going to use functionality, which will be implemented inside Enterprise JavaBeans, hosted by the Backoffice module. Using this approach we can also benefit from the transaction, security, performance and stability features offered by the EJB container of the Application Server. The JSP pages will not directly access the main ASP-NG database. Instead, they are going to use Enterprise JavaBeans methods that will enable the JSP pages to access the data they need. The ASP-NG portal is also going to use the internationalisation features offered by the Java technology in order to implement the multilingual functionality of the ASP-NG portal.

Customization features of the ASP-NG portal are going to be partly implemented using the Cascading Style Sheet mechanism. The Cascading Style Sheet mechanism allows authors and readers to attach style such as fonts, colours and spacing to HTML documents, using a human readable language. The ASP-NG portal is going to store a different stylesheet for each different skin that the portal administrator installs. The contents of the selected (by the portal administrator) stylesheet will be requested from the user through an intuitive web form. From then on, each time this user requests any page from the ASP-NG

<sup>2</sup> Source of figure: Oracle Technology Network (<http://otn.oracle.com>)

portal, the system will provide the requested page with a link to stylesheet included in the HTML <link> tag.

#### 4.1 The ASP-NG Portal Database

The ASP-NG portal database will support the operation of the localization of the customized look and feel of the ASP-NG portal. In order to support that service the ASP-NG portal database contains 8 tables. Figure 3 shows the ASP-NG portal tables and their relationships.

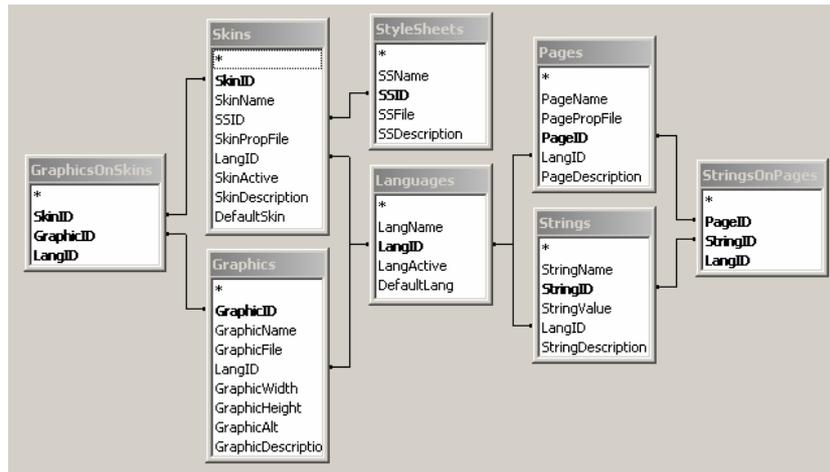


Figure 3 ASP-NG portal Database Schema

- *Language table (LangName, LangID, LangActive, DefaultLang)*: Lists all languages that have been inserted so far. The LangActive is a boolean attribute that determines whether the corresponding language will be displayed to the user (it is FALSE when for example a language has been inserted but not all necessary string translations have been made, in order to avoid presenting incomplete language support to the user). The DefaultLang identifies the default language of the ASP-NG portal.
- *Strings table (StringName, StringID, StringValue, LangID, StringDescription)*: Lists all Strings anywhere in the ASP-NG portal in any language. The LangID is a foreign key that shows in what language the corresponding string is written. The StringName attribute is the name that the string has in the .properties file. The StringDescription attribute gives an English description for the string, which will help the portal administrator during the translation of the strings.
- *Pages table (PageName, PagePropFile, PageID, LangID, pageDescription)*: Lists all pages of the ASP-NG portal in all the available languages. The LangID is a foreign key that shows in what language the corresponding page is written. The PagePropFile contains the name of the .properties file for this page. The PageDescription attribute gives an English description for the page, which will help the portal administrator to identify the page in the ASP-NG portal.
- *StringsOnPages table (PageID, StringID, LangID)*: This table associates the pages with the strings they contain, so PageID is a foreign key on the Pages table, and StringID on the Strings table. The LangID attribute is also foreign key on the Languages table and it must agree with the LangID of the corresponding String record and with the LangID of the corresponding Page record.
- *Stylesheets table (SSName, SSID, SSFile, SSDescription)*: List all inserted Stylesheets to the ASP-NG portal. SSName is the name that is given to the stylesheet, while SSFile is the name of the corresponding stylesheet file. The SSDescription attribute gives an English description for the stylesheet for the portal administrator convenience.
- *Skins table (SkinID, SkinName, SSID, SkinPropFile, LangID, SkinActive, SkinDescription, DefaultSkin)*: Lists all available skins. SSID is a foreign key on the StyleSheets table that shows the

stylesheet associated with the corresponding skin. SkinPropFile is the skin properties file, while SkinActive is a Boolean attribute similar to the LangActive attribute of the Languages table. If a skin has been inserted in the Database but not all its components (stylesheet, graphics) are ready yet, it is marked with the SkinActive attribute FALSE. Each skin is associated with a language (there are different versions of each skins for all the available language), and LangID is the foreign key that realizes this relationship. The SkinDescription attribute gives an English description for the skin for the portal administrator convenience. The DefaultSkin identifies the default language of the ASP-NG portal.

- *Graphics table (GraphicID, GraphicName, GraphicFile, LangID, GraphicWidth, GraphicHeight, GraphicAlt, GraphicDescription):* Lists all available graphics, with different graphics for each language. The LangID is a foreign key that associates a Graphic record with its language. The GraphicFile attribute contains the name of the graphics file, while GraphicAlt contains the text that will be used as alternative (the “Alt” attribute in <IMG> tag) when a browser cannot display images. GraphicWidth and GraphicHeight hold the dimensions of the graphic. The GraphicDescription attribute gives an English description for the Graphic for the portal administrator convenience.
- *GraphicsOnSkins table (SkinID, GraphicID, LangID):* This table associates the skins with the graphics they contain, so SkinID is a foreign key on the Skins table, and GraphicID on the Graphics table. The LangID attribute is also foreign key on the Languages table and it must agree with the LangID of the corresponding Graphic record and with the LangID of the corresponding Skin record.

When the portal administrator wants to insert a new language to the ASP-NG portal, the necessary information is inserted in the Languages table. The portal administrator can then select the page to translate the strings contained in that specific page for the new language. He is then presented with a form where he can translate all strings contained in the selected page. If some string has already been translated, the corresponding edit box is already completed. With the Database tables completed (through the WWW forms that the portal administrator can view), the .properties files can be generated. A similar approach is applied in order to add new skins and create the necessary .properties files.

For the correct operation of the ASP-NG portal, during the installation of the ASP-NG system, the portal database will contain (at least) all the necessary information for the operation in the default language (English) and the default skin. After the installation, the portal administrator will have the capability to add new languages and new skins. Because the default language and the default skin will be used as guidelines for adding new languages and new skins, it is the responsibility of the portal administrator to check that the new default skin and default language are properly installed, when he changes the default language and the default skin.

## **4.2 Authentication and Security Issues**

The ASP-NG portal will offer a built-in authentication system for registering and validating registered users. The authentication of a user will be based on a login/password combination over secure HTTPS/SSL ([13]) connections and the users’ passwords will be stored encrypted in the ASP-NG Backoffice database. With this approach even the system administrator will not have the capability to read a user password (of course the system administrator will have the capability to reset the user password). We have decided to base the authentication mechanism on the common approach of login/password combination and not to use more advanced mechanisms (for example user authentication with the use of certificates) for the following reasons:

- The proposed authentication system (login/password combination over secure HTTPS/SSL connections) provides a good security level, which satisfies the requirements of an e-commerce solution like the ASP-NG service.
- Most of the end-users are familiar with the use of login and passwords.
- A more advanced authentication system, which needs specific technical knowledge from the user, adds overhead to the authentication mechanism and may discourage a user without the necessary technical knowledge.

In addition, the ASP-NG portal will provide the capability of secure communication in the means of secure HTTP/SSL connections. Every time the user provides personal information and passwords (for example during registration or authentication) or during renting transactions (for example rent an application) all the communication will be done over secure HTTP/SSL connections. We select the approach to provide only the most sensitive parts of ASP-NG portal over secure HTTPS/SSL connections and not the whole ASP-NG portal, in order to increase the performance of the ASP-NG portal, due to the fact that the HTTPS/SSL connection includes an overhead to the overall performance of the ASP-NG portal, since it has to encrypt the whole content of all pages protected by HTTPS/SSL.

### **4.3 ASP-NG Portal localization Features**

The ASP-NG portal will offer to its users the capability to select and customize the language of the user interface in order to present information in their preferred language. Initially the ASP-NG portal will support only a small set of languages (during the developing phase these languages are going to be English and Greek) but the portal administrator will have the capability to add support for more languages at any time, through a very simple web-based interface. The implementation of the ASP-NG portal multilingual user interface will be based on the internationalisation features of the J2EE platform. The J2EE platform provides a rich set of APIs for developing internationalised WWW portals like the locales (class `java.util.locale`) and resource bundles (class `java.util.ResourceBundle`).

On the J2EE platform, a locale is simply an identifier for a particular combination of language and region (for example language English and region USA). Java locales act as requests for certain behaviour from another object. This object tries to fulfil the requested locale but if this is not possible, it then selects the best available locale (for example USA English instead of UK English) or if there is not any matching locale then the object uses the default locale. The process of selecting the best matching locale for a specific locale request is handled by the J2EE core classes and there is no need for special programming. The identification of users' preferences regarding localization is based on ISO 639 for languages identification and ISO 3166 for country identification.

Resource Bundles contain locale-specific objects, which are in most of the cases locale-specific strings. When a program needs a locale-specific resource, for example a string, the program can load it from the resource bundle that is appropriate for the current user's locale. With this approach, the programmer can write code that is largely independent of the user's locale isolating most, if not all, of the locale-specific information in resource bundles. A Resource bundle obtains locale specific resources from properly formatted text files (with the extension ".properties") or properly defined java classes.

More specifically, for the implementation of the ASP-NG portal multilingual user interface we will use locales for identifying the user language preference either indirectly by the user browser setting or directly by asking him for his language preferences. The only information we are interested in for each locale is the locale's language (and not the country or the specific variation of a language, since we differentiate by language selection). The information for each different language of the ASP-NG portal will be stored in the portal database and every time an interface item (for example a string) changes or every time the portal administrator decides, the appropriate resources bundle files (.properties files) will be created based on the portal database information. With the above approach - building the resources bundle files on demand and not extracting the information of the portal database during the operation of the ASP-NG portal - we improve the performance of the ASP-NG portal. In order to generate a web page and serve it to the user's browser, the server does not have to make any additional database access. It only reads a local .properties file, which contains the strings of the specific page, translated in the suitable (preferred) language. In addition, the fact that the localization information is not changed very often leads us to the above-described approach.

#### **4.4 ASP-NG Customised Look and Feel**

The ASP-NG portal will offer to the portal administrator the capability to customise the look and feel of the ASP-NG portal. With the term “customise” we mean the capability that the portal administrator will have to perform the following operations:

- Change (create, edit, remove) the stylesheet (.css file), which controls the look and feel (for example fonts and colour of each element) of the ASP-NG portal.
- Change (create, replace, remove) the graphic files of the ASP-NG portal with others with the same dimension.
- Change the strings of the ASP-NG portal (this capability will be provided through the localization features of the ASP-NG portal).

The realisation of the above features of the ASP-NG portal will be based on the concept of Skins [1]. Skins are a common approach to provide customised look and feel not only to WWW portals but also to other kinds of applications. With the term “Skin” we refer to a collection of interfaces objects (for example graphics, font sizes, font colours, etc), which can be used to change the user graphic interface. In the case of the ASP-NG portal the skins will consist of the portal stylesheet and the portal graphics. For the implementation of the customised look and feel feature of the ASP-NG portal the concept of the Skin Server will be used. The Skin Server is a program (in the ASP-NG portal it is going to be a Servlet program), which is responsible for loading the skins information and providing the correct object for each request of a skin element based on the ASP-NG portal configuration. The SkinServer provides its services to the other JSP pages of the ASP-NG portal. The skin information will be stored in the portal database and based on that information the portal administrator will build special formatted files (.properties files) with the skin information. We choose this approach – similar to the one implemented for the localization features – for better performance as we have already explained in the previous paragraph.

#### **5. ASP-NG Portal Implementation Technologies**

The ASP-NG portal is responsible for the interaction between the system and its clients. The main service of the ASP-NG portal is the access to the Application on Demand (AoD) service that the ASP-NG system offers. In addition to that service, the ASP-NG portal offers a number of supportive services, which increase the effectiveness, and the user-friendliness of the ASP-NG system. The most important supportive service is the usage of the user profiles. Each user profile contains information regarding the end user, his/her authorities, statistical information, billing information etc. In addition the ASP-NG portal offers customized look and feel capabilities and internationalisation.

The security and authentication features of the ASP-NG portal use state of the art technologies and protocols in order to ensure the secure and proper operation of the system. These technologies and protocols include secure protocols like the HTTPS (Secure HTTP) SSL (Secure Socket Layer) and public/private key technologies. The ASP-NG portal is implemented with the use of JSP and Servlets technologies, and EJBs for implementing business logic with regard to localization and customization features.

The Backoffice is implemented using J2EE technologies and more particularly using Enterprise Java Beans (EJB) technologies. We use the EJB version 2.0 standard and the following types of EJBs defined in EJB version 2.0 specifications:

- Entity EJBs - mainly BMP (Bean Managed Persistence) Entity EJBs: For modelling the ASP-NG system database in the Java Environment.
- Session EJBs (both stateless and statefull): For providing the Backoffice API calls to the Backoffice users (JSP pages of ASP-NG portal). The session EJBs use the entity EJBs, which represent the System Database in the Java Environment.

The Backoffice API EJBs are used by the ASP-NG portal using the following concept: The Backoffice API provides the business logic to the ASP-NG portal and the ASP-NG portal provides the presentation layer to the end user.

The Backoffice API EJBs are deployed to the application server of the Backoffice and the communication between the entity EJBs and the session EJBs is based on Local Interfaces (running in the same Java Virtual Machine – for performance reasons). The communication between the session EJBs and the ASP-NG portal is based either on Local Interfaces (both Backoffice EJBs and ASP-NG portal running in the same Java Virtual Machine / Container and the same physical server in order to achieve better performance) or on Remote Interfaces (the Backoffice EJBs and ASP-NG portal running on different physical servers in order to achieve better scalability).

## **6. Layout and ASP-NG Portal Browsing**

Figure 4 shows the general layout of the ASP-NG portal and more particularly the layout of the logged user. The logged user page consists of the following:

- Advertisement (Top): Displays an advertisement banner. The contents of the banner are determined by the portal administrator, who can for example use this space for commercial or promotional purposes.
- User Login or User profile area (Upper left): There is one link about the user's subusers, which leads to a page that displays all subusers this user has created. Another link leads to the user's transaction history. The transaction history page displays all purchases and rentals this user has made. The next link leads to this page, which lists the applications that the user has rented. The following link leads to a page where the user can see and edit his personal information and preferences, and the last link leads to the page where the user can view and edit his ranks/reviews for applications he has tried and wants other users to know his opinion on them.
- Browse area (Middle left): Here the user can browse the applications offered by the ASP-NG system. He can either select a specific application category, or view all individual applications, all application bundles, or all products offered by the system.
- Search area (Lower left): the user enters the keywords in the text box and chooses the sorting criteria (product rank, product category). There is also a link to the advanced search page. The advanced search page offers many more possibilities for the user to enhance his search and achieve more focused and relevant results.
- Main area: There is a list of the applications and application bundles this user has rented and a link for ranking them, and a line showing the user's remaining credits and offering the possibility of buying more. The user's bandwidth has also been measured and is reported at the lower part of the page. The user has also the possibility to perform the bandwidth measurement again. The user can also select to view his most recently used applications.
- Collaboration area (Upper right): Links to the chat, forum pages and CRM functionality.
- Top Titles area (Middle right): The names of the top 10 applications are shown in this area, each one with a link to more applications details.
- New Titles area (Lower right): Here the names of the latest applications added to the ASP-NG portal are displayed, each one with a link to more application information and details.
- Featured application (Bottom): This area allows the promotion of one application, presenting some details and perhaps graphics. If the portal administrator chooses not to offer some featured product, this page area is removed.

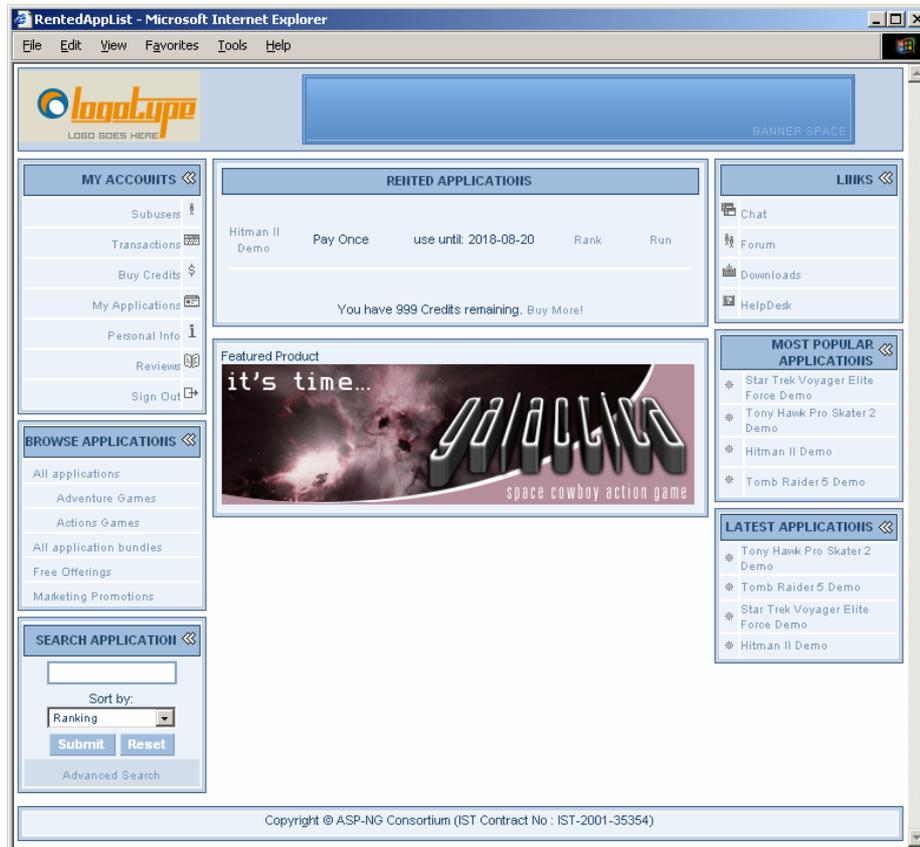


Figure 4 Logged user home page

## 7. Conclusion – Future Work

In this paper we presented the design of the ASP-NG portal, which constitutes a web interface in order to use the Application on Demand (AoD) service. The design of the ASP-NG portal is based on current state of the art J2EE technologies and offers enhanced capabilities like multilingual interface and customised look and feel. The design of the ASP-NG portal offers enhanced performance and scalability based on J2EE technologies, separates the presentation and the business logic and allowing these two entities to operate independently.

Our future work includes the implementation of the ASP-NG portal based on the design presented in this paper and the integration of the ASP-NG portal with the other modules of the ASP-NG system in order to provide the complete AoD solution that the ASP-NG project envisions.

## 8. Acknowledgements

We want to thank the European Commission (EC) for supporting this work in the context of ASP-NG IST project (Contract number: IST-2001-35354). In addition, we want to thank our partners in the ASP-NG project for their valuable cooperation. More information regarding ASP-NG project can be found in <http://www.asp-ng.eu.org>.

## References

- [1] Advanced JavaServer Pages by David Geary, by Publisher: Prentice Hall PTR; ISBN: 0130307041; 1st edition (May 29, 2001)

- [2] Core Servlets and JavaServer Pages (JSP) by Marty Hall, Publisher: Prentice Hall PTR; ISBN: 0130893404; 1 edition (May 26, 2000)
- [3] “Designing Enterprise Applications with the J2EETM Platform”, Second Edition, [http://java.sun.com/blueprints/guidelines/designing\\_enterprise\\_applications\\_2e/index.html](http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/index.html)
- [4] EJB Design Patterns, Advanced Patterns, Processes and Idioms, Floyd Marinescu Wiley Computer Publishing (ISBN 0-471-20831-0)
- [5] “Internationalize JSP-based Websites”, by Govind Seshadri, <http://www.javaworld.com/javaworld/jw-03-2000/jw-03-ssj-jsp.html>
- [6] Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions by Budi Kurniawan, Publisher: New Riders Publishing; ISBN: 073571195X; 1st edition (April 12, 2002)
- [7] Java TM 2 Platform, Enterprise Edition (J2EE), <http://java.sun.com/j2ee/>
- [8] Jsp: Javaser Pages (Developer's Guide) by Barry A. Burd, Publisher: Hungry Minds, Inc; ISBN: 0764535358; Bk&Cd-Rom edition (April 2001)
- [9] JSR 168 Portlet Specification <http://www.jcp.org/en/jsr/detail?id=168>
- [10] Mastering Enterprise JavaBeans (2nd Edition) by Ed Roman, Scott W. Ambler, Tyler Jewell, Floyd Marinescu, Publisher: John Wiley & Sons; ISBN: 0471417114; 2nd Bk&acc edition (December 14, 2001)
- [11] “Personalize your Website with skins”, by Brian R.J. Heumann <http://www.javaworld.com/javaworld/jw-06-2001/jw-0629-skins.html>
- [12] “The J2EE Tutorial”, [http://java.sun.com/j2ee/tutorial/1\\_3-fcs/index.html](http://java.sun.com/j2ee/tutorial/1_3-fcs/index.html)
- [13] “The SSL Protocol” <http://wp.netscape.com/eng/ssl3/draft302.txt>
- [14] Web Development with Java Server Pages by Duane K. Fields, Mark A. Kolb, Shawn Bayern, Publisher: Manning Publications; ISBN: 193011012X; 2nd edition (September 15, 2001)
- [15] Web Services for Remote Portlets TC [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsrp](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp)