UNIVERSITY OF GOTHENBURG

Content management via mobile devices

Accessing and modifying data and content in SharePoint

Rikard Malm

Master of Science Thesis

Report No. 2013:095
ISSN: 1651-4769

University of Gothenburg
Department of Applied Information Technology
Gothenburg, Sweden, June 2013

Abstract

This thesis considers content management within SharePoint via a mobile device. There are several means of communication between SharePoint and a non-mobile system. However, the possible means of communication between a mobile device and SharePoint is more limited. Moreover, some methods for communication will only be available for specific types of mobile devices. However, it is of interest to examine a more general method that is not limited to a specific type of mobile device.

The work was carried out as a laboratory experiment, where a Portal was constructed in SharePoint. In order to communicate with the portal, a mobile application triggered procedures on the server that accessed and modified data within the portal on behalf of the client. This approach was subsequently compared to accessing a portal through the web browser of the mobile device. This comparison was carried out according to Functionality and Efficiency of ISO/IEC 9126, and shown that this method increases the Efficiency while maintaining the Functionality.

Acknowledgements

This project was carried out at Sigma AB & Delta (Dpec AB). I would like to especially thank the supervisors, Tobias Linde at Sigma AB and Faramarz Agahi at Chalmers. I also thank Sanna Anjou and Daniel Tengvall for providing a link between me and Sigma AB.

I thank Antony Norman and Marion Blatt at Chalmers Open Communication Studio (CHOCS) for useful discussion.

List of figures

List of tables

Table of Contents

List of abbreviations

API	– Application programming interface
SDK	– Software development kit
IDE	– Integrated development environment
URL	– Uniform resource locator
CSS	– Cascading style sheets
XML	– Extensible markup language
STS	– Secure token service
CMS	– Content management system
RPC	– Remote procedure call
IIS	– Internet information services
WCF	– Windows communication foundation
WAS	– Windows process activation service
SMB	– Server message block
CIFS	– Common internet file system

1 Introduction

This project will address how to develop mobile applications that may manipulate data and content within SharePoint. This section will introduce some fundamental concepts for this problem such as SharePoint and mobile applications. Furthermore, it will describe the existing solutions, and give some examples of what this approach can provide that is out of reach for the existing solution.

1.1 Background

The possible approaches for communication between a mobile device and a resource may be categorised based upon the logical structure of the mobile device that is used to access the resource. That is, while it is possible to access a resource that is exposed to the web through a web browser, it is also possible to use applications on the mobile device to communicate with the resource.

A mobile application is a logical sequence of code that may be executed on a mobile device. There are several types of mobile devices, each with its own capabilities. These mobile devices may be pigeonholed based on the OS of the mobile device. The most significant OS' for smart phones are Android, iOs and Windows Phone [38]. A distinction can also be done based on the type of application. More specifically, dividing the applications into two categories, native application and hybrid application. In this context, a native application is an application specific to the OS of the mobile device and a hybrid application is a web site wrapped in a native application.

A Hybrid application is a slightly more complex concept than a native application. More specifically, a hybrid application is a web site combined with a native application. Basically, the web site is encapsulated into a native application. By wrapping the site into a native shell it forms a site that is capable of accessing the native functionality of the device. This is achieved through the web browser control that the application contains. The web browser control will manage the site. That is, it will display the site when the application is launched. Furthermore, when any resource is requested, the web browser control will examine it. If the resource corresponds to a location the web browser control will navigate to that location. However, the requested resource may also correspond to an internal resource on the device. In that case the web browser will invoke the native functionality. The internal resources of the mobile device are referred to as platform specific functionality. The application that encapsulate the web site and allows the hybrid application to access platform specific functionality will have to be specific for the targeted platform. That is, for an Android based system the web site would have to be encapsulated into an Android application and so on. In contrast, the site that is encapsulated by the application could be common for the different devices. Both hybrid and native mobile applications are developed similarly.

Mobile applications are developed using integrated development environments (IDE). The IDE may simulate the effect of the code on the mobile device by emulating a mobile device and simulate the effects of the code on that emulator. The emulators will acts as the mobile device they are emulating within a reasonable limit. That is, if the operation to be performed upon the emulated device is based upon hardware that would be available to the mobile, but is not available to the device emulating it, such as camera or GPS the result would differ from that of an actual device. However, the access to additional hardware is not the only difference between developing for a mobile and a stationary device. In particular, it is important to note that, development for mobile devices tends to be more restricted than for stationary devices due to the fact that there are less available technologies. Where the same technologies are available, they may be in restricted capacity. For instance, development of native applications for devices with the Android OS is based on Java. However, the version of Java used is specific for Android and not the same as the version used for stationary devices. Java consists of several editions, another edition is Java ME which is intended for mobile devices. It is a reduced version of Java intended for mobile devices. Since the development for smart phones is restricted to a single programming language which is specific to the particular device, the communication between the mobile device and the SharePoint server is restricted to components available within that programming language.

SharePoint, a content management system (CMS), is a computer program that supports management of web content and data. When the user submits information specific to the syntax required by the CMS, additional information may be generated by the CMS. Thus, when performing some tasks the developer is only required to perform a part of the task while the CMS generates the rest. In other cases it is sufficient to provide the CMS with the parameters of the operation. This functionality is often accessed through a web interface, typically via a web browser on a workstation. However, it is not limited to access from a workstation. Mobile devices may access the CMS through a web browser as well. In the case that a mobile device accesses the CMS through a web browser the result will be significantly different as discussed further in section 1.2. However, that approach relies on communication through a web browser, there is great potential from accessing SharePoint directly from applications running on a mobile device. This would allow web developers to be able to create and update web sites in SharePoint using other devices than workstations. Furthermore, users of the site managed by SharePoint could potentially benefit from accessing the site through a mobile application rather than through a web browser. When the site is accessed through a web browser, the web browser will simply display the content of the site. However, when a mobile application is used to access the site, the application may access the SharePoint data and content as well as functionality native to the mobile device. More specifically, an application can access the hardware of the mobile device. The functionality accessed on the mobile device would be dependent on what is available and this may vary from one device to another. However, general functionality common to smart phones could be assumed. This would include access to memory which would allow the application to store content of the SharePoint server locally on the device. This data could then be used to create an offline version of the site. Another example would be, if the location of the user would be relevant, the application could use the GPS to determine the location and integrate that with the information from the SharePoint server. However, it should be noted that some of this functionality can be obtained through html also, such as, for instance, geolocation that can be used to obtain the users geographical position by associating a location with an IP address.

1.2 Existing solutions

Within SharePoint there can be several versions of each page, and the version which is returned by the SharePoint server for display on the accessing device is determined by the type of accessing device that is making the request [1]. For example, if the accessing device is detected to be a smart phone, then a version of the page that is suitable for display on a mobile device will be returned. This version of the page is denoted by "?Mobile=1" in the URL. Display of the version of the page intended for display on a device like a laptop or workstation with a larger screen can be forced by replacing this with "?Mobile=0".

The pages intended for mobile devices are called Mobile pages and are automatically generated by SharePoint. This does not mean that all the content of the pages for stationary devices will be available for mobile devices. Some site elements such as lists will be available to the mobile pages while other site content while other content may need to be adapted to mobile device [2].

The Mobile pages often simply replicate the content and functionality of the version intended for stationary device [3]. However, since it is still a web page being utilized to access the content of the site, the possibilities to adapt the page specifically to the device that is accessing it is limited. Web sites can adapt the size of the elements within the site to fit the screen resolution of the device accessing it. Applications on the other hand may access platform specific functionality.

1.3 Problem description

This thesis is aimed at determining if SharePoint is a suitable CMS when a mobile device is used to access the system. For a system consisting of a server and a client, determining if the system would be suitable in this context could consider how this system would affect the server or how the client will be affected. For both the server and client, there are several possible aspects that could be considered in order to determine if SharePoint can be used advantageously. This work focuses on how the client will be affected by the system in order to answer the research question:

How is the mobile device affected when it is used to access a site developed in SharePoint via an application. However, there exist several types of mobile devices, and in some cases there are methods that are only available for a subset of the mobile devices. That is, there are inherent problems for the communication. In particular, there are several possible OS that the mobile devices may be based upon. These OSs implies restrictions upon the means of communication, in particular when it is done programmatically. Therefore, this work will evaluate communication between SharePoint and a mobile device in a manner that is not specific to a particular kind of mobile device, in order to provide a meaningful investigation.

SharePoint contains several application programming interfaces (APIs) for programmatic access from another machine then the server. These APIs are called Client Object Model. The reason there are several APIs are not in order to provide distinct options in terms of functionality, but rather to allow it to be used in different programmatic contexts. More specifically, there are versions of the API for JavaScript, Silverlight and .NET managed applications. They function by allowing the client to manipulate the content and information of the SharePoint site by constructing queries. These queries will contain instructions for the server, such as selecting some data element based on criteria included in the query or requesting some operation to be performed.

Due to the limitations for smart phones, where they are limited to a single programming language for native applications, the only possible approach for directly communicating with the SharePoint server through SharePoint’s API would be through a hybrid application. Since a hybrid application consists of a web site wrapped in a native application, this website could utilize JavaScript to access and modify individual site elements in SharePoint. That is, specifically, the web site would use the JavaScript API to communicate with the server. The API would then handle the communication with the server is by constructing a query. This query will contain the actions to be performed on the SharePoint server. None of the actions in the query will be performed before the query is transmitted to the server. When the query is executed the commands are bundled together into a XML request and sent to the server. The server will perform the requested operations and reply to the client. With this approach, the SharePoint server is accessed by JavaScript code that will be global over the hybrid applications. That is, the web site, containing the JavaScript code, that is part of the hybrid application is not specific to a certain kind of mobile device. Consequently, the SharePoint functionality provided to the hybrid applications would be identical regardless of the type of mobile device. However, this approach relies on cross-site scripting. That is, when a script on a site performs operation on another site. Cross-site scripting is prohibited for SharePoint’s API. As a result of this, there is no way for a mobile application to directly access SharePoint programmatically.

Another possible method is to have procedures on the server that performs some operations on the data and content, then allowing another machine to request the execution of these procedures. In this case, the Server object model may be used. The Server object model is an API just like the Client Object Model with the distinction that the code is executed on the server. This means that the information is already available and may be manipulated directly rather than using queries. However, if the data is modified by procedures that reside locally on the server, the client will still need to call these procedures remotely.

The communication between the mobile device and the SharePoint server can in some cases be dependent on functionality specific to a certain kind of mobile device. The Client Object Model may be used within the application framework Silverlight. Development of native applications for Windows phone is based on Silverlight, thus, a native Windows phone application may access the SharePoint server directly. However, this approach would not be viable for devices based on Android or iOS. Therefore, a more general solution that is not specific to a small number of devices would be better. The code for accessing the native functionality of the device and the communication with the server would still be specific to the particular device since different programming languages are used on different devices. However, the methodology should be general enough that it does not impose restrictions on what kind of mobile devices are capable of being used. Regardless of the method used to communicate and manipulate data, the CMS has to verify that client has sufficient privileges within the system for the attempted operation.

Accessing the information and data on the SharePoint server will be dependent on the configuration of the server. In particular, how users and processes attempting to access the server are authenticated. The authentication process for a mobile device is identical to that of a stationary device. However, since the primary intended use of the CMS is through a web browser, the CMS works well in this situation. That is, the authentication process will be managed by the browser, other than supplying credentials, the user will not notice much of the authentication process. Furthermore, manipulation of site content and data is supported by the web interface viewed in the browser. However, when a mobile application is used, the application has to authenticate and access the data and content programmatically. The possible approaches when developing are not identical for mobile and stationary devices.

1.4 Limitations

This thesis will focus on sites developed within SharePoint 2010. The applications considered on the mobile device will be limited to native and hybrid applications. The mobile devices considered will be limited to smart phones based on Android, Windows phone and iOS. The means for communication between the SharePoint server and mobile device will be general. That is, how the communication is carried out should not be specific to a certain kind of mobile device.

2 Theoretical framework

The starting point for this project was existing research in the field. This section will present the content and result of related work. Furthermore, general theory of how SharePoint's functionality is provided will be considered followed by how this functionality can be provided to a mobile device. Moreover, a model for software evaluation is presented.

2.1 Non-mobile software solution

Many multi-user systems distinguish between their users by the use of user credentials. This serves to determine the permissions for specific users within the system. Since SharePoint is designed as a multi-user system, the resources being accessed by the client requires authentication. There are several alternate methods of authentication.

SharePoint supports Claims-based and "Classic mode" authentication. This determines how the client is authenticated to the SharePoint server. Within these methods of authentication there are several mechanisms that perform the actual authentication. Classic authentication uses Windows authentication [17]. For Windows authentication, the mechanisms that perform the authentication are limited to Kerberos and NT LAN Manager (NTLM). In both cases, the credentials provided to the system are obtained by default from the computer that performs the authentication rather than by user input. These credentials will be the same credentials that were used to log into the computer that will be forwarded to the CMS.

Each authentication method is of interest for different reasons. However, for further versions of SharePoint (after SharePoint 2010), classic authentication is deprecated [37]. Consequently, Claims authentication is of interest to consider. However, NTLM authentication on the SharePoint server implies restrictions upon the system accessing it. In particular, Windows phone does not support NTLM authentication. While Claims authentication is of interest since it seems to be what SharePoint is moving towards, NTLM authentication is of interest because of the restrictions it imposes. However, while the process of authentication prefixes any access to resources, the manner in which the resources are accessed are not specific to the authentication scheme. That is, more specifically, the same entity can provides access to the SharePoint data regardless of authentication scheme. Therefore, both Claims and NTLM will be considered.

However, the standard approach to interact with SharePoint is through a web browser. This means that, in order to access SharePoint's functionality, it has to be exposed in a manner that makes it manageable by the web browser. This includes that the resources of the system has to be exposed for access through a web browser, but also that the web browser has to implement the authentication scheme in order to obtain sufficient privileges within the system. In contrast, if the resources are accessed programmatically, the system containing the resources then has to expose the resources for programmatic calls.

Accessing and modifying data

When accessing a resource on another machine, the resource has to be exposed in some manner. That is, the machine containing the resource has to listen for incoming requests. By hosting the resource within a server, the server can listen for requests to the resource. However, the syntax of the requests is not arbitrary. The exact syntax of the request depends on how the resource is hosted within the server. This section will consider how the resources are exposed and consequently how the resources can be accessed.

Internet information services (IIS), the application server hosting the SharePoint service, may also expose other applications and programs. IIS is constructed in a modular manner with a core containing basic functionality and a number of modules [5]. The modules extend the functionality of the server in some way. This means that, depending on the tasks to be performed an arbitrary number of modules can be attached. Each module have a specific functionality, for instance protocol listeners that receive manage protocol specific requests.

Several applications may be contained within the IIS, therefore it is necessary to determine to what application the request were directed. The Windows process activation service (WAS) is the part of the IIS that manages worker processes. The listeners will capture the requests and forward them to the WAS [5]. URLs are used by the listeners to map incoming requests with specific applications [6]. When an incoming message requests a resource on the server, the WAS may then start a process in response [7]. The URLs consists of two parts, a base prefix and a relative path [7]. The base is shared for all applications within a site. The relative path will determine the specific application within the site.

WAS clusters applications into logical groups called application pools [7]. This logical grouping is based upon some characteristic the applications have in common, and may contain applications from several different sites. Each application pool have its own worker process. This provides isolation between the application pools. Thus, an application that fails will not affect other applications outside of the application pool even if the worker thread fails. The application pools will not only share worker process but also memory space. The application pools will also impose restrictions. In particular, the version of the .NET framework the application is based upon must be consistent for all applications within an application pool. That is, if the application pool contains an application based upon the .NET framework, only applications based upon the same version of the .NET framework may share the application pool. However, the versions 2.0, 3.0 and 3.5 are compatible and do not require separate application pools.

It is possible to host applications based on WCF inside IIS [7]. WCF is a framework for developing service based applications, and is being used as Microsoft's unified programming model for the development of service-oriented applications on Windows [8, 9]. This means that, applications developed with WCF may expose services to other machines within a computer network without imposing restrictions upon the platform of the requesting machine. In WCF, the services provided by the application are based upon messages being sent between endpoints, where an endpoint refers to an entity that sends or receives messages [8]. The messages may trigger the application to perform some operation and transmit the result back to the requesting machine. The endpoints determine the parameters of the messages being sent.

There is a distinction made that categorises an entity as a client or server [8]. The applications that initiate the communication is called client while the endpoint that awaits messages is referred to as a server. However, it is possible for an endpoint to be both a server and a client. For instance, peer-to-peer networks may have endpoints that function both as server and client. The client that is connecting to the server will generate an endpoint that is compatible with the servers endpoint [8]. To ensure that incoming messages follows the expected format, the end points exposes metadata. This meta data is information about the syntax of messages being sent from and to the endpoint. The component that determines the syntax of the messages sent between the client and server is called a binding [10]. The binding will determine the transport protocol used for communication and the syntax of the messages. The binding may also specify what security mechanism should be used to secure the communication.

The services provided by the application are bundled together by a service contract which defines properties of the application [10]. The service contract is typically a interface which the actual service will implement. This will allow the service contract to impose constraints upon the service such as encryption without specifying how it should be implemented [11]. The service contract will contain operation contracts. The operation contracts are then methods in the interface. Thus, the operation contracts will defines operations by specifying the methods of the class that implements the interface. In this manner, the operation contracts will determine the methods of the service with their parameters and return types.

The constraints imposed upon the service by the contracts must be supported within the syntax of the message as determined by the binding. If not all constraints are fulfilled the service will not be operational. The contracts are typically expressed in XML format and exposed to allow developers to construct application that may interact with the service [11]. This exposed metadata is, however, only a description of the provided service.

When a WCF web service is deployed within SharePoint, the endpoints of the service will be automatically configured [12]. However, the type of the service still needs to be selected. The endpoints supported for a WCF web service within SharePoint limits the type of possible services to SOAP, REST, ADO.Net data service. However, only REST and SOAP are considered within this work since these approaches can be used within the constraints of the method. That is, SOAP and REST is an available approach for any mobile device that can connect to the internet and not just smart phones.

REST

REST is an architectural style for developing distributed applications which is supported within WCF [13, 14]. An architectural style in the context of software engineering is a set of constraints or features that can be used guide the development of the software system. REST has significant parity with the Internet in that resources are distinguished by a url. The resources can then be accessed via standard HTTP requests. A service that implements the REST architectural style is referred to as a RESTful service.

The possible operations upon a RESTful service through HTTP requests are shown in *Table 1 - REST operations* [25]. However, the service has to implement the infrastructure in order to manage the requests. Thus, for example, in order for a HTTP PUT request to be able to update data, a method has to exist within the service that listens for HTTP PUT requests and the process the data of the request.

HTTP verb	**Data operation**
GET	Retrieve data
POST	Submit data
PUT	Update data
DELETE	Delete data
MERGE	Update data

Table 1 - REST operations

SOAP

SOAP uses xml to structure information [15]. Within the SOAP structure the information is divided into xml tags. These tags determine the parameters of the message. A SOAP message consists of several sections. The SOAP envelope section is the root element of a SOAP message. The SOAP envelope identifies the message as a SOAP message and determines the encoding of the message. The envelope section may contain a SOAP header section. The header contains application specific information that determines how the recipient should process the message. The envelope must also contain a SOAP body section. The body contains the actual information being sent.

The system structure and how external systems can access data and information is summarised in figure 1. IIS hosts SharePoint, which in turn contains web services. The syntax for communicating with these web services is determined by WCF. This work will focus on REST and SOAP as means of communication.

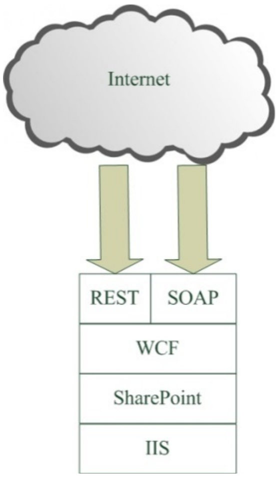


Figure 1 - system

Claims

Claims keeps track of the systems users with tokens [18]. A token contains a number of claims. Each of these claims contains some information about the user such as name or privileges within the system. The information contained within a token can be customised depending on the specific needs of the system. Regardless of what information is required of the system, each piece of information is contained within an individual claims, within the token. The claims may be expressed by, for instance, a numerical value or character string.

STS are used to create the tokens [18]. The process of acquiring a token is as follows

- The client provides some form of identifying information and specifies the service it wishes to access to the STS.
- The STS validates the information provided by the user.
- The STS looks up the information contained for the user with respect to the service specified is step 1.
- The information acquired in the previous step is used to create a token. The token is then signed and returned to the client.

Once the client has acquired the token it will contact an application that provides some service [18]. The application will perform verification on the user's token to authenticate the user and determine privileges within the system. The verification process includes validating the tokens signature. Validating signatures is done by decrypting the signature with the sender's Public key. This will allow the application to determine which STS issued the token. If the STS is trusted by the application, the claims within the token are considered correct by the application. The application may then use the claims within the token to determine the users privileges within the system. This process is depicted in figure 2.

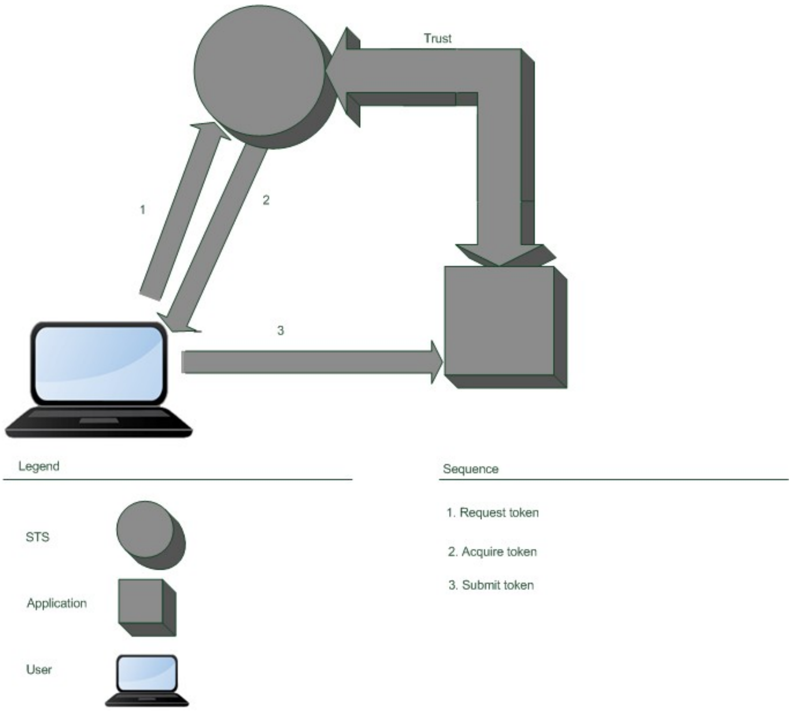


Figure 2 –Claims authentication

However, there may be several STS. If the STS that issued the token is not trusted by the application, the user may still be granted privileges within the system [18]. If there exist a STS that trusts the service that issued the token and is trusted by the system, it may recreate the token. This new token may be used to access the application.

When using Claims, the applications that provide some service to the users do not have to authenticate the users [18]. The authentication is done by the STS. The applications only have to validate the token provided by the user.

In SharePoint, Claims-based authentication is built on Microsoft Windows Identity Foundation [19]. Which is a framework that provides a set of APIs. These APIs can be used to process tokens created by a STS within a web application or web service that uses the .NET framework. This process typically involves decrypting the security token, validating the signature, and dividing the token into a set of claims. These claims can then be used in the application.

Within SharePoint, there are several methods of claims based authentication [19]. They are:

- Windows Claims - Standard windows authentication is used. The resulting windows identity is translated into a claims identity.
- Forms-Based authentication - ASP.NET membership provider is used to authenticate the client. The resulting identity object is translated into a claims identity object.
- SAML-Claims - External STS provides are used to authenticate the client. The user provides the SAML token to the SharePoint server that uses it to create a claims identity object for the user.

Regardless if Classic or Claims mode authentication is used to authenticate the users [20], SharePoint services use Claims for internal communication. However, it is in some cases possible to convert claims into Windows credentials. This can be used for outgoing communication.

NT LAN Manager

Rather than transmitting the users password and username through the network, a challenge/response is used by NTLM [32]. The challenge is a mathematical computation based on the password. The client will supply the challenge with some identifying information such as username and password. The result is sent to the server. The server will then perform the challenge with stored value for that user's password. If the value resulting from that computation is identical to the value provided by the client access is granted.

If the client is not part of the same domain as the server, NTLM authentication will be used [33]. The challenge/response procedure of NTLM authentication is defined by Microsoft [34] as follows:

1. (Interactive authentication only) A user accesses a client computer and provides a domain name, user name, and password. The client computes a cryptographic hash of the password and discards the actual password.
2. The client sends the user name to the server (in plaintext).
3. The server generates a 16-byte random number, called a challenge or nonce, and sends it to the client.
4. The client encrypts this challenge with the hash of the user's password and returns the result to the server. This is called the response.
5. The server sends the following three items to the domain controller:
   - User name
   - Challenge sent to the client
   - Response received from the client
6. The domain controller uses the user name to retrieve the hash of the user's password from the Security Account Manager database. It uses this password hash to encrypt the challenge.
7. The domain controller compares the encrypted challenge it computed (in step 6) to the response computed by the client (in step 4). If they are identical, authentication is successful.

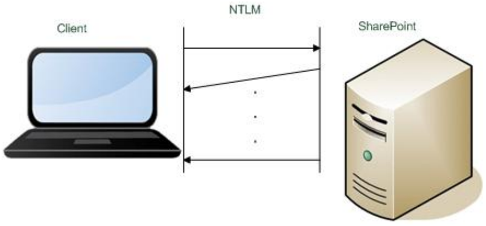


Figure 3 – NTLM authentication

NTLM uses the Server message block (SMB) which is an application layer protocol for network file sharing. There are several versions of the SMB protocol [35, 36]. These versions are known as dialects. The common internet file system (CIFS) protocol is such a dialect of the SMB protocol.

When the site is accessed through a web browser, it will suffice for the user to provide user name and password since most web browsers have implemented windows authentication. However, when the site is accessed through a mobile application, the challenge/response scheme has to be implemented within the application.

2.2 Mobile software solution

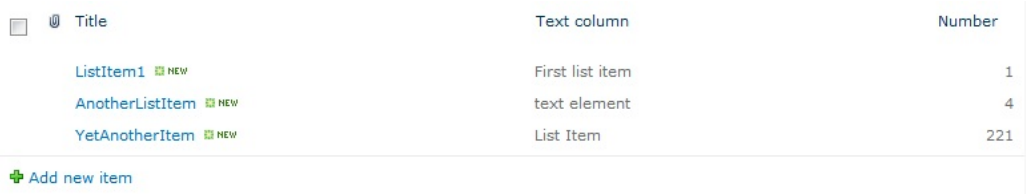
Since SharePoint is a multiuser system there have to be a distinction between which user is attempting to call which resource. In particular, not all users will have the same privileges within the system. Therefore, not all users should have access to all resources. When SharePoint’s resources are accessed through the web services, the authentication scheme that determines the privileges of users within the system are inherited from SharePoint.

Another aspect to consider is how the resources of the system are accessed. That is, when the CMS is accessed through a web browser, the authentication scheme will have to be implemented by the browser, otherwise it would not be possible to access the system. The end user will not notice much of the authentication scheme other than a request for credentials. Any message parsing between the user and system in order to fulfil the authentication scheme will then be automatically carried out by the web browser in the background. In contrast, when the system is accessed by an application, that application would then have to implement the authentication scheme of the CMS. However, the development for mobile devices is more restricted than for stationary devices. In this case, the restrictions on mobile devices may inhibit the implementation of the authentication scheme. This means that, in those cases, the mobile application will not be able to communicate with an entity that implements such an authentication scheme. More specifically, for this case, Windows phone does not support NTLM authentication. Hence, it will not be able to communicate directly with the web services that exposes SharePoint’s functionality. Claims, on the other hand, is based upon tokens, and the use of Claims as authentication scheme will not impose restrictions upon the mobile device accessing it.

The information from the portal needs to be exposed in a manner that makes it accessible from a mobile device. It should be mentioned, that the manner in which the mobile device will access a resource will be of importance. More specifically, in our case, the resource will be accessed from an application on the mobile device. This is significant, because, the authentication protocol will have to be implemented by the web browser in order to access the resource of the site when it is accessed through a web browser. Consequently, the application accessing the site needs to implement the authentication protocol in order to access any resource. Once the privileges within the system have been determined, the user may access and modify data accordingly. This procedure will be similar to the authentication, in that, when it is done from a browser, any information that the system needs from the user will be provided by the browser.

Accessing and modifying data

How the content is represented within SharePoint is significant for determining how it can be accessed. In particular, within the context of SharePoint, lists are collections of information that can be edited and modified by users of the portal. These collections resemble a relational database, where items within the list consist of tuples. The tuples are ordered sequences of information. Within a tuple, the type of a value will be consistent for every attribute of a specific index. More specifically, the type of the value at a particular index will be the same for all tuples within the list. Figure 4 shows an example SharePoint list.



□	Title	Text column	Number
	ListItem1 NEW	First list item	1
	AnotherListItem NEW	text element	4
	YetAnotherItem NEW	List Item	221

Figure 4 – SharePoint list

The application on the mobile device will not be able to access and modify the data, such as lists, on the SharePoint server directly. However, it is still possible to modify and access the data programmatically from the server. Thus, it is possible to construct methods on the server that access and manipulate the information. These methods can then be exposed to calls from other devices. That is, another device will initiate the method and provide the parameters of the method. The server will then execute the method with the provided parameters and reply with the resulting value of the operation. The approach of triggering a procedure remotely is known as remote procedure call (RPC). In order to use RPC, the privileges of the procedures will be executed with needs to be determined.

Depending on the desired operation to be performed, a web service capable of performing the operation through RPC might be available. A number of web services exist within the default implementation of SharePoint. These web services are clustered together in the folder _vti_bin, and function in standard RPC manner. That is, provided the input, the web service will carry out some operation and reply with the result. Consequently, the application on the mobile device will only need to trigger the methods on the server. The input from the mobile application to the web service must be provided in the format and manner expected by the web service. What information the web service expects is, of course, dependent on the operation to be performed by the web service. However, the manner in which the information is provided to the web service will depend on how the web service is constructed.

The default web services within SharePoint are based upon WCF. These web services may be used to expose information or to perform operations on the SharePoint site via RPC. As aforementioned, for a site developed within SharePoint they will be available at the folder _vti_bin within the site. The URL that the web service is accessed at consist of, the address for the site where the operation is to be performed at, and also the web service to be invoked. Thus, for a SharePoint site named mySite the services would be available at the url http://mySite/_vti_bin/Service where Service is the name of the specific service being invoked. However, a portal may consist of several sites. Therefore, it is important to note the context of the web service's execution. That is, when a web service is called through the concatenation of the site and the web service, the web service will be executed at that site. Consequently, the resources available will be limited to the resources available for that site.

While a lot of functionality may be provided by these web services, not every possible operation upon the SharePoint portal is covered. As a result, it may be necessary to construct additional web services in order to expose more of the functionality of the portal. The basic requirement is for the service is to programmatically, through procedures, access SharePoint's data. These procedures then have to be exposed for remote calls. The process of programmatically manipulating data on SharePoint is possible through SharePoint's server object model. The server object model is SharePoint's API for server side manipulation of the SharePoint data. Therefore, the procedure utilising this method to access and manipulate SharePoint's data has to be located on the same machine that hosts the SharePoint site.

It is not sufficient to have procedures that accesses SharePoint's data, the procedures have to be accessible through RPC. In order to expose the procedures for remote calls and allowing the mobile devices to communicate with them, the procedures implement WCF. Consequently, the bindings of the service must be specified, as required by the architecture. The information specified by the binding includes the address of the endpoint, a service contract and a binding configuration. The Service contract declares the available procedures within the service. The service contract is programmatically constructed as an interface. The operation contracts within the service contract are implemented as the methods exposed by the service. This interface is then implemented by the actual service.

The endpoints are automatically configured by SharePoint's service factory. The service factory is an entity that is specific to the type of WCF service. Consequently, there are three possible service factories corresponding to the three types of WCF service types, SOAP, REST and ADO.NET. Programmatically selecting the factory is sufficient to determine the type of the service.

However, if a service that exposes the requested content already exists, developing another would be redundant. There are by default a number of web services built-in to SharePoint. To access and manipulate information contained within a list, the web service ListData.svc can be used. ListData.svc is a restful service. Therefore, the RPC calls are initiated by HTTP verbs at a specific URLs. For instance, ListData.svc would be available at http://mySite/_vti_bin/ListData.svc where mySite is the site the operations will be performed upon.

Communication with the service ListData.svc in order to invoke operations is done according to *Table1*. That is, HTTP GET requests may be used to obtain information regarding lists on the site. HTTP Post may submit information and so on. The same syntax that was used to retrieve data may be used to submit, if the http verb used is post instead of get. The http post request will trigger the server to create the resource using the data contained in the request. In particular, the operations of interest are HTTP GET that allows the access of information and HTTP POST that allow information to be submitted to the server. However, the resources, that is, in this case the lists are accessed by extending the URL. Specifically, performing a HTTP GET request at the base address of the web service ListData.svc will return an xml document containing the SharePoint lists within the site. By appending the base address with the name of a specific list, the web service would access that list. These lists are exposed as RSS feeds. Individual elements of a list may be accessed, also as a RSS feed, by attaching an index in brackets to the URL of the http request.

Furthermore, it is possible to limit the information being accessed with the URL of the request. In particular, the information accessed may be filtered so that only elements matching a certain criteria will be returned. This is done by appending the URL of the resource with a filter. Beyond accessing lists, ListData.svc also exposes other elements of a SharePoint portal that are internally represented as a list. A significant amount of SharePoint components are internally represented as lists. This includes announcements and Calendars. In addition, the SharePoint element document library which is the structure that contains documents within a SharePoint portal is internally represented as a list. However, accessing an index in the document library will access that list entry rather than the file. To access the file, it is sufficient to create a file stream from the files location on the site.

However, while providing access to list data, there are operations upon that data that is not supported by the ListData.svc service. For instance, discussions which are the SharePoint component that provides a forum for the users of the portal are internally represented as a list. Due to a bug, confirmed by the Microsoft SharePoint support team, discussion lists may not be accessed this way [26].

Since the mobile application does not perform any operations upon the data, it is sufficient for the mobile device to trigger RPC via messages to the server and display the result. Since the RPC calls are triggered by HTTP requests, it is sufficient to construct the HTTP requests and display the response to the user.

Claims

Before any resources can be accessed, a token must be obtained. This token will represent the users privileges within the system and will be obtained from the STS. Fortunately, the default implementation of SharePoint contains a STS. The STS is called Authentication.asmx, and is based upon SOAP. Navigation to it within a web browser will reveal the syntax of token request and response based on SOAP version as shown in figure 5 and figure 6.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /_vti_bin/Authentication.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://schemas.microsoft.com/sharepoint/soap/Login"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Login xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <username>string</username>
      <password>string</password>
    </Login>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <LoginResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/">
```

```
      <LoginResult>
        <CookieName>string</CookieName>
        <ErrorCode>NoError or NotInFormsAuthenticationMode or
PasswordNotMatch</ErrorCode>
        <TimeoutSeconds>int</TimeoutSeconds>
      </LoginResult>
    </LoginResponse>
  </soap:Body>
</soap:Envelope>
```

Figure 5 – SOAP 1.1 authentication

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /_vti_bin/Authentication.asmx HTTP/1.1
Host: www.zevenseas.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <Login xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <username>string</username>
      <password>string</password>
    </Login>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <LoginResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <LoginResult>
        <CookieName>string</CookieName>
        <ErrorCode>NoError or NotInFormsAuthenticationMode or
PasswordNotMatch</ErrorCode>
        <TimeoutSeconds>int</TimeoutSeconds>
      </LoginResult>
    </LoginResponse>
  </soap12:Body>
</soap12:Envelope>
```

Figure 6 – SOAP authentication

Following the protocol for communication with the STS, the caller will submit username and password within the body of the SOAP envelope. Upon a successful request, the STS will return a token in form of a cookie. The cookie may then be attached to a subsequent RPC request. However, if the request is not successful, an error will be returned instead.

When the authentication process is completed, the aforementioned web services may be used via RPC. The web services are structured by the type of operation they perform. As the web services are hosted within WCF they are not limited to SOAP based services. The type of the web service will depend on the operations it performs.

NT LAN Manager

While direct communication between the mobile device and the web service exposing the information is excluded as an approach, it is still possible for the mobile device to communicate with the web service. The web services were used as intermediaries since the mobile devices could not programmatically access or modify the information on the SharePoint server. Similarly, continuing with the same approach, yet another intermediary can be added.

This other intermediary was on a stationary device such as a server, thus the restrictions for mobile devices would not apply. In computer networks, an entity that acts as an intermediary and requests resources on a server on behalf of the client is called a proxy. Any request to the server will be seen by the server as originating from the proxy. A reverse proxy acts as a proxy on behalf of the server. For requests to the server through the proxy, the response will be seen by the client as if it originated from the proxy rather than the server. By this notation, the entity that accesses the information and data is a proxy and the entity that performs the authentication on behalf of the mobile application is a reverse proxy. The purpose of the proxy is to access and modify content on the CMS while the purpose of the reverse proxy is to implement the authentication scheme of the CMS.

The reverse proxy, performing the NTLM authentication on behalf of the mobile device, was implemented as a web service in Java. In order to provide the authentication, the Java library JCIFS (http://jcifs.samba.org/), which is a implementation of the CIFS/SMB networking protocol. This allowed the reverse proxy to authenticate towards the proxy on behalf of the client.

System overview

The process of accessing and manipulating data is prefixed by the process of authentication. For Claims, the mobile device initiates the communication by sending a SOAP based message to SharePoint’s STS. This message contains information verifying the identity of the client. Assuming the information is correct, the server respond with a token in the form of a cookie.

The cookie obtained from the STS is attached to the message to a service. Since the operations available depend on the service, the service contacted will be specific to the requested service. The service will serve as a proxy and perform operations upon the information of the portal on behalf of the client. These operations include accessing and submitting information.

Upon completion of the requested operation, the service responds to the client. This response could contain the requested information, or an acknowledgement that the requested operation was performed successfully. This process of accessing the resources on SharePoint with claims authentication is shown in f*igure 7.*

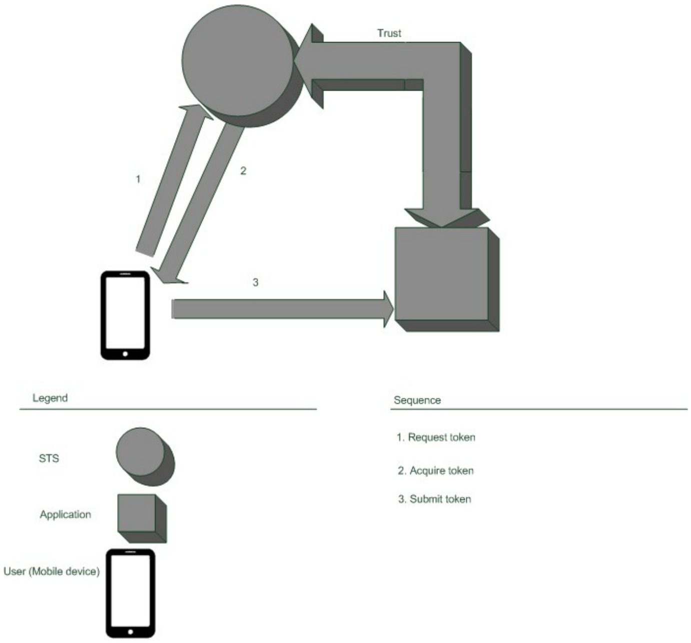


Figure 7 - Claims

In contrast, the authentication for NTLM is done by a reverse proxy on behalf of the client. Thus all communication between the mobile device and the web service exposing SharePoint's functionality must pass through the reverse proxy. The mobile device will trigger the reverse proxy to access or manipulate data upon the SharePoint server by executing procedures on the proxy. The resulting value of the procedures are transmitted to the reverse proxy, which in turn returns the value to the mobile device as depicted in f*igure 8*.

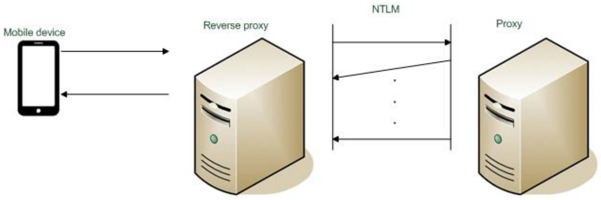


Figure 8 - NTLM

2.3 Software evaluation

When comparing software systems, there are several distinct aspects of the system that can be used for the comparison. This section presents a model for categorising the aspects of the software system that are used for the comparison. More specifically, the model used is ISO/IEC 9126. ISO/IEC 9126 is a general software evaluation model [24]. The objective of the standard is to meet user needs. This is done by evaluating the quality of the software. Where quality is defined as follows: "Quality: the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs."

The model proposed by the standard ISO/IEC 9126 consists of several parts [24]. The first part is a quality model that categorizes software quality into a structured set of characteristics, which in turn contains sub-characteristics as shown in *figure 9*. The hierarchical structure of the model constructs are illustrated in *figure 10.* The further parts of the standard are concerned with metrics. In particular, in order to complement the aforementioned model. These metrics are as follows:

- Internal quality: Static properties of the code, such as, for instance, path length.
- External quality: Dynamic properties of the code, such as, for instance, response time.
- Quality in use: The extent to which the software meets the needs of the user in the working environment.

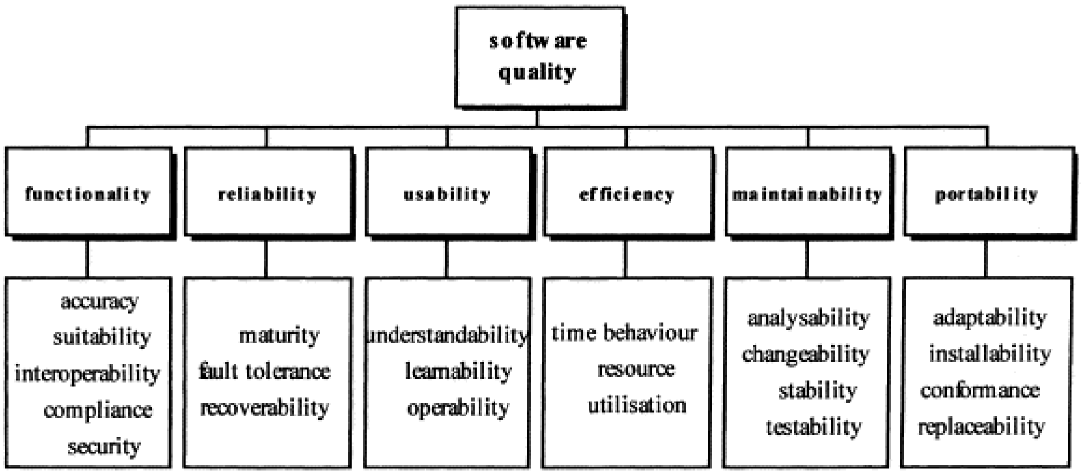


Figure 9 - Software quality characteristics [24]

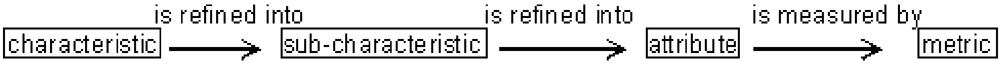
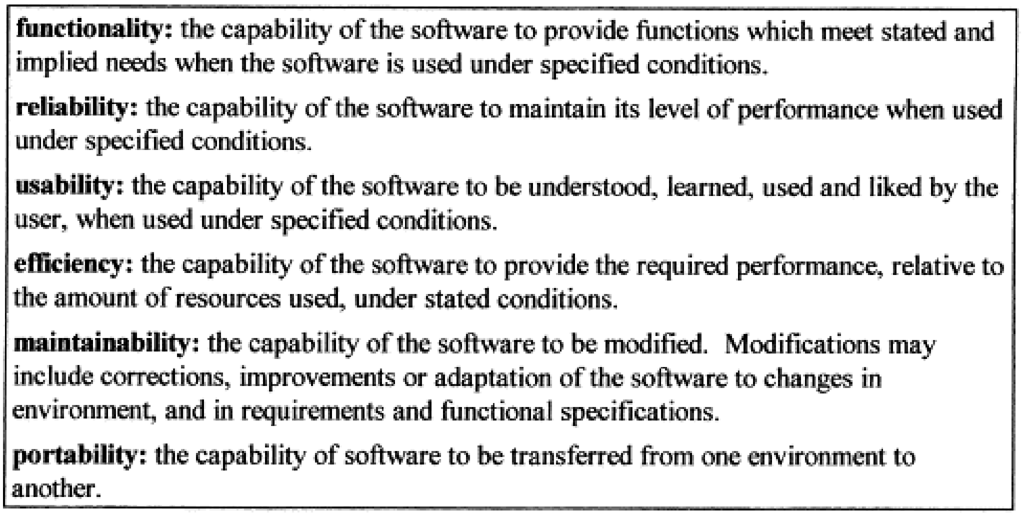


Figure 10 - ISO9126 constructs hierarchy [28]

The quality model in *figure 9* is applicable to internal and external quality [27]. The model for Quality in use, on the other hand, contains four characteristics: effectiveness, productivity, safety and satisfaction. However, unlike the quality model used by internal and external quality, these characteristics are not refined further into sub characteristics. The characteristics of the software quality model are defined by the model [24]. That is, for every system the same set of characteristics apply. The characteristics are defined in terms of “the capability of the software” as shown in *figure 11.* Each of these characteristics contains a set of sub-characteristics. A clarification of the meanings of the sub-characteristics is shown in *Table 1.* These sub-characteristics are also defined independently of the system that the model is applied to.



functionality: the capability of the software to provide functions which meet stated and implied needs when the software is used under specified conditions.

reliability: the capability of the software to maintain its level of performance when used under specified conditions.

usability: the capability of the software to be understood, learned, used and liked by the user, when used under specified conditions.

efficiency: the capability of the software to provide the required performance, relative to the amount of resources used, under stated conditions.

maintainability: the capability of the software to be modified. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications.

portability: the capability of software to be transferred from one environment to another.

Figure 11 - ISO/IEC FCD 9126 definitions [24]

Characteristic	Sub-characteristic	Explanation
Functionality	Suitability Accurateness Interoperability Security	Can the software perform the tasks required? Is the result as expected? Can the system interact with another system? Does the software prevent unauthorized access?

Reliability	Maturity Fault tolerance Recoverability	Have most of the faults in the software been eliminated over time? Is the software capable of handling errors? Can the software resume working and restore lost data after failure?
Usability	Understandability Learnability Operability Attractiveness	Does the user comprehend how to use the system easily? Can the User learn to use the system easily? Can the user use the system without much effort? Does the interface look good?
Efficiency	Time behaviour Resource utilisation	How quickly does the system respond? Does the system utilise resources efficiently?
Maintainability	Analysability Changeability Stability Testability	Can faults be easily diagnosed? Can the software easily be modified? Can the software continue working if changes are made? Can the software be tested easily?
Portability	Adaptability Installability Conformance Replaceability	Can the software be moved to other environments? Can the software be installed easily? Does the software comply with portability standards? Can the software easily replace other software?
All characteristics	Compliance	Does the software comply with laws or regulations?

Table 2 - ISO 9126 characteristics and sub characteristics

The method for determining the quality is dependent on the type of quality to be determined [24]. Moreover, within an aspect of quality there may be several distinct methods for obtaining measurements [29]. Typically, internal quality is determined by code inspection [24]. However, specification documents review and checking models are also applicable methods [29]. This is done in order to provide means to evaluate the software at early stages of the project, before the product becomes executable. Therefore, measuring the internal quality is done as part of the verification during development [28]. Moreover, the primary objective of the internal metrics is to ensure the external quality [24]. External metrics are derived from the behaviour of the software system in its operational environment [24]. That is, external quality is measured from the execution of the software. Typically, external quality is determined by testing, operating and observing the executable software system. The objective of external metrics is to provide a target for validation [28]. Quality in use is the quality perceived by the end user executing the software system. However, a system may have several types of users. Such as, for instance, the person responsible for the maintenance of the system, or the person responsible for migrating the software system between environments.

It is the overall objective of the quality evaluation is to ensure that product will have the desired effect in its operational environment. Furthermore, the quality aspects influence each other. Therefore, it is sufficient to examine a single aspect of quality in order to estimate the effect of the system in its operational environment. Consequently, this work will focus on external quality as a means of determining the overall quality of the system. Furthermore, there are several possible attributes that could be used to evaluate a system. However, the factors that influence the quality of a system are dependent on the context where the system is used. Therefore, the quantity attributes are not predefined by the standard. The selection of attributes is thus dependent on the context of the system and a non-trivial activity [28]. The generic format of the standard has been the cause of some criticism. In particular, the standard lacks a rationale for deciding which criteria relate to a particular factor. Consequently, the selection of characteristics and sub-characteristics may seem arbitrary [30].

In this case where the objective is to evaluate different methods rather than to compare implementations, it is very important that the measurements are not related to the specific implementation. An example of an implementation specific measurement would be the

Attractiveness sub characteristic of Usability. That is, “Does the interface look good? “, which is related to the design of the system. Consequently, this work will focus on the factors that affect the system without being related to a specific implementation. More specifically, the characteristics considered will be limited to Functionality and Efficiency, since both of these characteristics relate to the capability of the software in a manner that is general and not related to a specific implementation. Within these characteristics Suitability, Accurateness, Time behaviour and Resource utilisation were considered.

3 Method

The considered methods for this thesis were case study and laboratory experiment. On the one hand, a case study would allow the examination of a system within the context of its intended use. On the other hand, a laboratory experiment would provide a greater freedom in terms of ability to manipulate the aspects of the research environment. Due to the advantages associated with the laboratory experiment approach, it was hence selected as method for this thesis.

However, it should be noted that the laboratory experiment approach to the problem also has significant disadvantages. More specifically, since, within this method, the SharePoint site is constructed. This means that the components of the site will be selected. Therefore, it should be noted that, it is possible that the selected subset of components will omit some functionality that would have been of interest to examine. This method entails the construction of a SharePoint portal as a starting point. The content and functionality of the portal was obtained from pingpong. That is, the created portal in SharePoint replicated a select set of the content and functionality of pingpong. This provided a means of comparison for the developed system. More specifically, since both systems provides the same functionality and content, how this functionality and content is provided can be compared. In particular, the aspects considered for this work were Functionality and Efficiency from the ISO/IEC 9126.

3.1 The case: A university course portal – pingpong

Knowledge management systems (KMS), is a subcategory of CMS, focuses on organisational objectives by the sharing of knowledge. Barber, G.B., Haque, N. and Gardner B [4] proposed a hierarchical approach to determine how a knowledge management system affects various aspects of an organisation in the work "'OnePoint' - Combining OneNote and SharePoint to facilitate knowledge transfer". The model is divided into four tiers, which are as follows:

- External world - Regulatory & key documentation, Historic record of key events, Validated documentation.
- Corporate - Accessible company-wide, summary information.
- Team - Project team centric, idea evaluation, Transient information.
- Individual - Personal, idea capture, Unstructured.

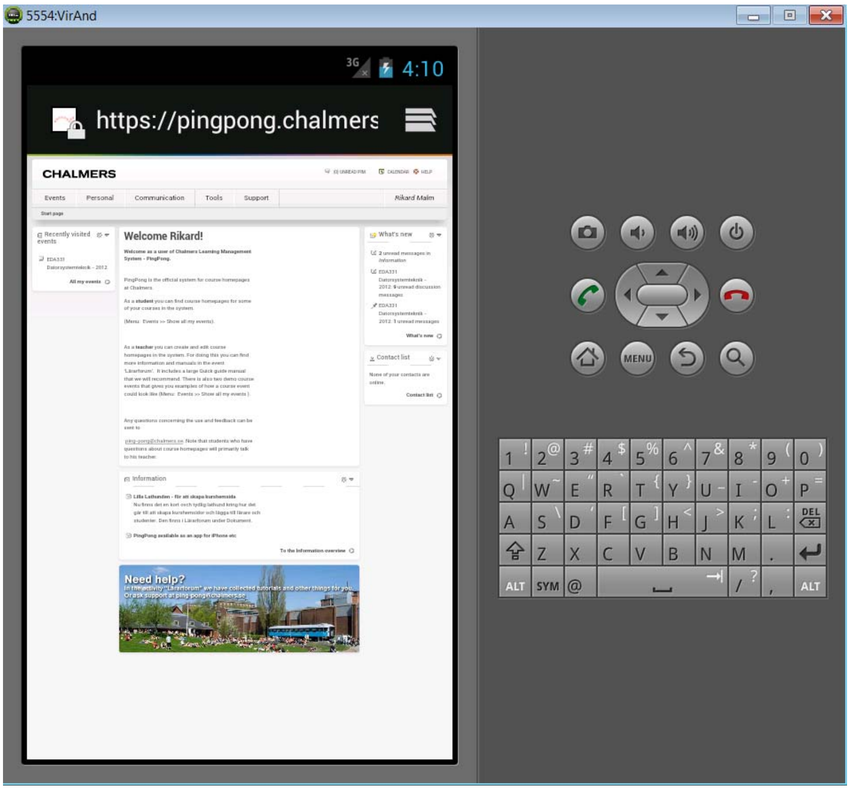
In the context of our portal, the team tier would represent courses which also contain groups of individuals in the same manner as a team. The corporate tier would represent the institution that was hosting the courses.

No common standard or definition seems to exist for the term portal. Furthermore, the context in which the portal is used seems to influence the definition. [21]. However, the concept of a portal has attracted a great deal of attention. As a result of the attention it attracted, several attempts were made to define it. The earliest, to us known, definition is from the paper, enterprise information portal - industry overview [22], where it is defined as a application that provides a single interface to the information of the enterprise. Since the context of the portal considered in this work is to expose information from an enterprise, the definition can thus be considered a definition for an enterprise portal.

However, since the work presented in this paper is concerned with the broader aspects of communication and content management, no distinction of portals will be made based upon the context in which it is used. The context in which the portal is used will only affect the type of information exposed, and not affect the manner in which information may be accessed and modified.

Technological advances have made more functionality possible from the portals, as reflected by subsequent definitions of the term portal. These technological advances have also enabled portals to perform more complex operations, such as, for instance, application integration capabilities [23]. These capabilities of the portal enable front-end integration of information, which is an important part of a KMS [21]. Consequently, portals have become the dominant front end for KMS.

Chalmers University of technology uses a course portal that contains means of conveying course related information to students, and in some cases allowing the students to submit information to the portal. The user submitted information can be in the form of files submitted to the system, but also in the form of comments and questions within forums. However, while the portal allows student submitted information, the arguably more important functionality of the portal is that it allows user obtain information. This information could be, for instance, lecture slides or past exams. Furthermore, it allows the presentation of news, and also simply text content on the portal's sites. The start page of pingpong when accessed through a mobile device is shown in *figure 12.*



EDA331 Datorsystemteknik - 2012

Welcome Rikard!

Welcome as a user of Chalmers Learning Management System - PingPong.

PingPong is the official system for course homepages at Chalmers.

As a **student** you can find course homepages for some of your courses in the system.

(Menu: Events >> Show all my events).

As a **teacher** you can create and edit course homepages in the system. For doing this you can find more information and manuals in the event 'Lärarforum'. It includes a large Quick guide manual that we will recommend. There is also two demo course events that gives you examples of how a course event could look like (Menu: Events >> Show all my events).

Any questions concerning the use and feedback can be sent to

ping-pong@chalmers.se. Note that students who have questions about course homepages will primarily talk to his teacher.

2 unread messages in Information

EDA331 Datorsystemteknik - 2012: 9 unread discussion messages

EDA331 Datorsystemteknik - 2012: 1 unread messages

None of your contacts are online.

Lilla Lathunden - för att skapa kurshemsida
Nu finns det en kort och tydlig lathund kring hur det går till att skapa kurshemsidor och lägga till lärare och studenter. Den finns i Lärarforum under Dokument.

PingPong available as an app for iPhone etc

Need help?
In the activity "Lärarforum" we have collected tutorials and other things for you. Or ask support at ping-pong@chalmers.se

Figure 12 - pinpong

There may be several means of performing an operation. In particular, distinct systems may use distinct components and site elements in order to achieve certain functionality. However, in order to provide a meaningful comparison, both systems has to provide the same content and functionality. Therefore, a subset of the functionality provided by pingpong was replicated within SharePoint in order to consider the components that can be used within SharePoint in order to achieve certain functionality. However, there may in some cases exist several alternate means of conveying the information. In order to provide consistency and reduce ambiguity, this section considers how the functionality was provided in a specific case. More specifically, it provides a description of how SharePoint's functionality was used to implement a course portal.

News and calendar

The front page of the portal displays upcoming activities and news regarding the portal. This is done visually through a calendar containing the events. However, not every event or announcement will relate to a specific day. Therefore, an announcement list is used as an alternate means of displaying the news. An announcement list is a substructure of the SharePoint list structure which contains short bits of information. The entries in an announcement list, unlike other list entries, contain an expiration date. When an expiration date of an entry is reached, the entry will automatically be deleted.

Forum

In order to facilitate communication within groups, forums are provided within the portal. These forums are specific to the part of the portal they are contained within. That is, the forum on the front page may contain general discussion, while the forums on specific course pages would contain information regarding that specific course.

Course site

Information regarding specific courses are clustered together into course sites, as subsites of the front page. The course site allows course specific information to be published and presented in a structured manner. This information may consist of, for example, a course schedule, course discussion, published files, and other relevant information.

Documents

The portal also contains a number of documents. Primarily, pdf files containing lecture slides and previous exams. These documents are part of the course site. However, the possible files that can be submitted to the portal are not restricted to pdf files.

Tasks

Activities such as courses are often composited of a number of tasks. These tasks may be, for example, laborations, reading course material and examination. In order to track the collection of tasks, SharePoint provides a task list, which is a type of list, containing the progress of the task and the prerequisites.

3.2 Data Collection

In order to evaluate the results of the constructed system, it was compared to the course portal used by Chalmers University of technology (pingpong.chalmers.se). The quality of the systems was be evaluated according to ISO/IEC 9126-1. In particular, the quality model was be used to evaluate the systems.

Since ISO/IEC 9126-1 is a general model for measuring the quality of a software system it has to be adapted to the system if is measuring the quality of, both in terms of the attributes and the characteristics used to evaluate the system. The attributes will depend on the usage of the system. In particular, the attributes have to represent the purpose of the system in a meaningful way. The characteristics used also have to represent the intended purpose of the system. Consequently, when determining the quality of a system according to ISO/IEC 9126-1, not every characteristic have to be a part of the investigation.

While ISO/IEC 9126 addresses several aspects of quality, the remainder of this paper will only address external quality characteristics. This is a consequence of the limited access to pingpong. That is, in particular, no access to the source code of pingpong prevented the measurement of internal quality. Moreover, the measurement of quality in use would require a set of users. Therefore, both internal quality and quality in use was omitted.

The ISO/IEC 9126-1 standard needs to be adapted to the context of the software system it is used to evaluate. This work focuses on information management within a portal through a mobile device rather than evaluating distinct systems. Therefore, any implementation specific measurements such as design choices or layout are considered irrelevant for this work. Consequently, the focus is on the operations performed through the portal in terms of possible functionality, and how the usage of this functionality affects the users of the portal. There are several aspects to the usage of the functionality. These aspects have to be evaluated individually through the standard in terms of characteristics and sub-characteristics. In particular, this work will consider the suitability and accurateness sub-characteristics of functionality and the sub-characteristics Time behaviour and Resource Utilisation of Efficiency.

Furthermore, in practice, some influence among the attributes of the quality model is possible. However, in this work the attributes was be considered to be independent.

The sub-characteristic interoperability evaluating if the system can interact with other systems are not explicitly evaluated. However, the manner in which the system was constructed makes it easy for any other system to communicate with it. Any system capable of constructing and receiving HTTP requests may access the systems resources.

For some measurements, a distinction has to be made based upon where in the system the measurements are performed. Specifically, what part of the system the measurements are carried out within. A distinction could also be done based upon the role of the user within the system, in the case that the system contains specific groups of users who use the system differently. For instance, in our case of a course portal, the users may be students consuming the information, or course responsible supplying the information.

The functionality sub-characteristics are highly interconnected, and consider therefore the same set of functionality. Specifically, the functionality considered was a subset of pingpongs functionality. That is, possibility to access files and documents, access temporal information through calendars, and also access to course announcements. While the same set of operations was considered for the sub-characteristics of functionality, different aspects of the operations were be considered. That is, on the one hand, the Suitability sub-characteristic considers if the system can perform the task required with respect to a specific operation. The accurateness sub-characteristic, on the other hand, consider if the result of the performed operation is as expected.

The efficiency characteristics may in many cases consider the entities of the system individually. The time required to perform a task may measure the time required by the server, client or even the total time required. Furthermore, the resources required may be resources on the server or client, in some cases the resources may even be shared between the client and server. In particular, the bandwidth is a shared resource. However, the resources used by the client are arguably more significant than the resources used by the client since it might be simpler to improve performance by scaling the server rather than upgrading the mobile device for every user of the system.

The Time behaviour sub-characteristic is the time required to perform a task. This is highly dependent on the amount of resources necessary. Thus, this sub-characteristic is strongly connected to the resource utilisation sub-characteristic. Furthermore, measuring the time required to perform a task may also be affected by the implementation. In particular, the transportation time of the message between the client and server depends on the path of the messages. In order to compare methods for content management rather than implementations of systems, the latency of the systems was disregarded from the measurement of time behaviour by measuring the amount of data transmitted. The amount of data transmitted in order to perform an operation is not affected by the location of the server in relation to the client. Consequently, measuring the data transmitted by an operation will be an indirect measurement of the time required to perform that operation. The operations Time behaviour was considered for were restricted to authentication and file transfer.

The Resource utilisation sub-characteristic was measured in terms of the resources used by the system. Specifically, the resources used by the mobile device in terms of memory and CPU usage were measured. Determining the amount of resources needed to perform an operation gives an indication of what is required of the device to perform an operation. If an operation can be performed with less resources, that frees up resources to be used by other processes. Furthermore, if an operation can be performed with fewer resources consumed, it will enable devices with fewer resources available to perform that operation. While the system was constructed to not impose any restrictions upon the type of device using the system, the possible combinations of every possible action with every possible mobile device would result in a large set of permutations. Therefore, set of mobile devices that the measurements were performed upon was restricted. Specifically, the measurements were carried out with an Android based mobile device. Furthermore, the operations considered for this sub-characteristic were also restricted to authentication, file transfer, and accessing a specific announcement.

3.3 Measurement tools

In order to measure the sub-characteristics for efficiency, tools were used to monitor the resource consumption of the system. Specifically, the traffic transmitted was recorded as well as the CPU and memory usage of the device accessing the system.

For the measurement of the systems resource consumption, a device based upon the Android OS was used. More specifically, a Samsung galaxy GT-I6300 with specifications according to *Table 3.* Furthermore, in order to obtain measurements of resource consumption, an application was used to record the CPU and memory usage of the process.

Android version	4.1.2
RAM memory	1GB
CPU	1.4GHz quad core
Chipset	Exynos 4212 quad

Table 3 - Android specifications

The measurement of Resource utilisation were recorded with the Android application Diagnosis (https://play.google.com/store/apps/details?id=eu.thedarken.diagnosis). Diagnosis were configured to record the CPU usage and memory consumption. The recording process of recording was carried out as follows: the recording was started, the event to be measured was triggered, and the recording stopped after the event was completed.

The measurement of Time behaviour was based upon the traffic being sent. Therefore, the traffic had to be captured and monitored. This was done through a proxy server and a packet analyser tool. The proxy server and packet analyser tool captured and recorded the traffic. More specifically, the proxy server used was called “Fiddler 2”, and the packet analyser tool was “Wireshark. This approach provided the means to inspect the traffic. In particular, the information with relevance in this research was the amount of information transmitted, and the execution time of the request on the server. The packets being transmitted contains a header and a body, both are inspected and measured. Furthermore, for each request, Fiddler 2 captured for HTTP traffic, information regarding events related to the request. More specifically, events recorded for the server, by Fiddler 2, are: ServerGotRequest, ServerBeginResponse, and ServerDoneResponse. The time between the events ServerGotRequest and ServerDoneResponse were used as a measurement of the execution time of the resource on the server.

The IDE used for development of Android applications, Eclipse, contains functionality to simulate the effect of the code on an Android device. This is done through emulating the mobile device and simulating the effects of the code. However, it also contains the default applications of an Android device such as the web browser. Therefore, this type of virtual Android device was used on a machine that was recording the traffic. Thus, the traffic of the Android device could be recorded and monitored.

Accessing the portal through a web browser is, in this work, considered as a distinct alternative to the access through an application. However, there are several distinct browsers that can be used in order to access the portal. The measurements carried out as part of the Time behaviour only considered the default web browser, contained within the default implementation of the virtual device. In contrast, the measurements related to Resource utilisation carried out with an Android device, with specifications according to *Table 3,* considered both the accessing of the resource through the default browser of the device, but also through Chrome web browser. The default web browser of the Android device were simply referred to as “browser” by the system. Therefore, hereinafter, browser will refer to the embedded web browser of the Android device.

3.4 Reliability of measurements

Measurements always contain some amount of measurement error [31]. However, on the one hand, the measurements for the functionality sub-characteristics are binary. That is, either the functionality can be provided correctly or it cannot be provided. On the other hand, the measurements for the efficiency sub-characteristics are more complex.

In order to improve upon the reliability of the measurements, and provide an as accurate as possible representation of the effects of the requests upon the system, the measurements carried out towards pingpong related to the state of the server were carried out at times where the workload of the server were expected to be at minimum. That is, during the weekend, but more importantly all the measurements related to the state of the server were carried out during the same day. To further improve upon the reliability of the measurements, all measurements related to Efficiency characteristics were carried out repeatedly, and the results documented. Since the measurements for the characteristics of Functionality were less complex, they were not repeated.

The measurements of resource consumption are pessimistic for both the application and web browser. In particular, the process of recording the Resource usage required the application set in foreground to be the application performing the measurements in order to start the recording. After the recording of resource consumption was initiated, the foreground application could be set to the component performing the request. This process caused the current site to be reloaded when a web browser were used to access the resource. That is, the site containing the operation the measurement was to be performed upon were reloaded as part of the measurement. Furthermore, the measurements related to resource consumption for the application included the initiation of the application into the measurements.

Finally, it should be noted that, measurements carried out where some form of tool were used to calculate the measurement, the reliability of the measurement is dependent on the reliability of the tool used to obtain it.

4 Results

This section will present the results of the conducted empirical investigation. Since the ISO/IEC 9126 standard was used to categorise the measurements, the measurements are presented according to the standard's characteristics.

4.1 Functionality

This section will address if the functionality can be provided by the portal, both in term Accurateness and Suitability. The measurements for these sub-characteristics are binary. That is, either the functionality can be provided correctly or it cannot be provided.

There may be several means of performing an operation. In particular, distinct systems may use distinct components and site elements in order to achieve certain functionality. This results in the different visual appearance of the portal, and possibly also a distinction in how the operations are performed by the server. However, how the functionality can be provided is not considered, it is sufficient that it can be provided.

Functionality	**SharePoint**		**Pingpong**	
	Accurateness	Suitability	Accurateness	Suitability
Course information	*1*	*1*	*1*	*1*
Download files	*1*	*1*	*1*	*1*
Calendar	*1*	*1*	*1*	*1*
Navigation	*1*	*1*	*0*	*0*

Table 4 - functionality

***Matrix legend**:*

0. The result of the action upon the portal with respect to the specific sub-characteristic was not satisfactory.

1. The result of the action upon the portal with respect to the specific sub-characteristic was satisfactory.

The results of the measured functionality are presented in *Table 4.* However, some values need some clarification. In particular, the measurements related to pingpong's navigation. The navigation on pingpong was partly based upon a menu containing hyperlinks. The default state of the menu was collapsed so the hyperlinks were not accessible. However, upon hovering over the menu, it expanded and the hyperlinks within it became accessible. Unfortunately, when accessed through a mobile device, touching the screen is registered as a click. Consequently, the mobile device was unable to produce a mouseover event to trigger the expansion of the menu in order to access the hyperlinks within it.

The course information was in the form of shorter text blocks containing course announcements. These announcements could adequately be accessed by the mobile device. However, information could also be contained within documents on the site. Therefore, the possibilities to download files from the portal to the mobile device were examined, and both the portals were found to provide a means of file transfer.

Some for the functionality consists of a graphical representation of an object as well as entries of information. An example of such functionality is the calendar. A calendar consists of a graphical representation of calendar as well as entries in the calendar. The entries are events recorded in the calendar. For SharePoint, when the application on the mobile device accesses the functionality, the information accessed is, in the case of the calendar, the events. That is, information about how the information is graphically displayed within the portal is not obtained. This is similar for other functionality in that, it is the information contained within the site elements, such as events in the calendar, that will be accessed is. That is, while some functionality in the portal consists of a graphical element, the information acquired is limited to the entries. More specifically, the entries will be obtained in the form of a XML document .Consequently, in order to obtain the same functionality in the application on the mobile device, the application must process the obtained information and display it to the user in some manner. In contrast, for pingpong, when some functionality is accessed, both the graphical element and the entries are obtained.

4.2 Efficiency

The results for the sub-characteristics of efficiency will be considered individually.

Time behaviour

The values presented within this section were obtained from performing the operation upon the portal. In some cases, performing an operation triggered subsequent operations. For instance, through the web browser, the process of authentication also triggered a redirect. However, there is neither a way to authenticate without redirection to the start page, nor would it be very useful as there are no resources accessible from the login page. Consequently, the sequence of processes executed in order to perform an operation was attributed to the operation. Therefore, the values presented here are the sum of the processes the operation consists of.

The results presented in this section were obtained from the information transmitted between the mobile device and server. Specifically, the measurements were obtained from the packages sent. Each package contains a head, body and also, for the requests recorded with Fiddler 2, information about the stages of the request. In particular, the stages of the request contain the times where the server obtained the message and when the server finished processing the request. The events ServerGotRequest and ServerDoneResponse were used to calculate the execution time of the request on the server. However, it should be noted that the execution time of a request is dependent on the state of the server, in particular, the workload of the server. The values for server execution time were only obtained from the measurements of pingpong.

Unlike the execution time of the request, the amount of information transmitted in the messages is not related to the state of the server. However, a few bits will be related to the specific implementation, such as the location of the resource being accessed. Furthermore, the password and username provided to the server were not identical for pingpong and the SharePoint system. Consequently, the information transmitted for a specific request may vary slightly. However, it was the judgement of the researcher that while the packages sent contains a few bits that are implementation specific, the general size of the messages can still be inferred.

Operation	**Bytes sent**	**Bytes received**	**Server execution time**
Download file (52 432B) request 1	head:739 ;body:0	head:481 ;body:52 432	0.066s
Sum for request 1	739	52 913	0.066s
Download file (52 432B) request 2	head:739 ;body:0	head:481 ;body:52 432	0.153s
Sum for request 2	739	52 913	0.153s
Download file (52 432B) request 3	head:739 ;body:0	head:481 ;body:52 432	0.109s
Sum for request 3	739	52 913	0.109s

Table 5 - pingpong downalod file - Time behaviour

The resulting measurements of downloading a file from pingpong are shown in *Table 5. The requested information, the file, is exactly the size of the body of the response, while the headers are the method calls and response. The method calls and response are a necessary* *consequence of obtaining the desired* information. In computer science, any excess resource consumption when performing an operation is referred to as overhead. More specifically, when some information is requested from the portal, some additional information may need to be transmitted in order to obtain the requested information. In this case, the overhead corresponds to the headers.

Bytes sent	**Bytes received**	**Content-Type**	**Time 1**	**Time 2**
headers:219; body:0	headers:107; body:0	Tunnel	-	-
headers:1 550; body:273	headers:4 055; body:11 199	text/html	0.088s	0.091s
headers:328; body:0	headers:160; body:1 245	text/html	0.018s	0.019
headers:219; body:0	headers:107; body:0	Tunnel	-	-
headers:1 084; body:10 770	headers:681; body:231	text/html	0.046	0.043s
headers:1 174; body:0	headers:603; body:0	application/octet-stream	0.040s	0.053s
headers:1 242; body:0	headers:465; body:20	text/html	0.027s	0.037s
headers:1 186; body:0	headers:399; body:5 032	text/html	0.053s	0.086s
headers:641; body:0	headers:444; body:54 813	application/JavaScript	0.076s	0.075s
headers:219; body:0	headers:107; body:0	Tunnel	-	-
headers:219; body:0	headers:107; body:0	Tunnel	-	-
headers:656; body:0	headers:426; body:832	text/css	0.011s	0.010s
headers:219; body:0	headers:107; body:0	Tunnel	-	-
headers:642; body:0	headers:395; body:1 955	text/css	0.021s	0.021s
headers:219; body:0	headers:107; body:0	Tunnel	-	-
headers:644; body:0	headers:437; body:20	application/JavaScript	0.010s	0.010s
headers:219; body:0	headers:107; body:0	Tunnel	-	-
headers:619; body:0	headers:440; body:379	application/JavaScript	0.010s	0.010s
headers:629; body:0	headers:407; body:1 975	application/JavaScript	0.011s	0.011s
headers:660; body:0	headers:431; body:14 521	text/css	0.035s	0.033s
headers:644; body:0	headers:380; body:630	image/png	0.009s	0.009s
headers:644; body:0	headers:381; body:1 293	image/png	0.010s	0.011s
headers:640; body:0	headers:381; body:1 246	image/gif	0.011s	0.011s
headers:646; body:0	headers:380; body:428	image/png	0.010s	0.010s

headers:738; body:0	headers:398; body:288	text/html	0.700s	0.821
headers:219; body:0	headers:107; body:0	Tunnel	-	-
headers:643; body:0	headers:381; body:1 377	image/gif	0.011s	0.010s
headers:642; body:0	headers:381; body:1 049	image/png	0.010s	0.010s
headers:639; body:0	headers:381; body:1 146	image/gif	0.010s	0.010s
headers:647; body:0	headers:381; body:1 193	image/gif	0.011s	0.010s
headers:643; body:0	headers:381; body:1 204	image/gif	0.013s	0.012s
headers:641; body:0	headers:381; body:1 193	image/gif	0.010s	0.010s
headers:643; body:0	headers:381; body:1 296	image/png	0.010s	0.010s
headers:650; body:0	headers:381; body:1 258	image/gif	0.010s	0.010s
headers:634; body:0	headers:419; body:67 793	image/jpeg	0.091s	0.093s
headers:647; body:0	headers:381; body:1 276	image/gif	0.011s	0.011s
headers:639; body:0	headers:380; body:2 497	image/gif	0.011s	0.011s
headers:642; body:0	headers:381; body:1 249	image/gif	0.011s	0.010s
headers:647; body:0	headers:381; body:1 156	image/gif	0.010s	0.010s
headers:660; body:0	headers:427; body:829	text/css	0.011s	0.010s

Table 6 – pingpong authentication messages data and server execution time

The messages for the authentication request to pingpong, shown in *Table 6*, were transmitted in parallel. Consequently, the order in which the messages were received varied between requests. However, significant portion of the bandwidth and server execution time are consumed by the overhead in the form of images, css and html.

The resulting measurements of authentication and file transfer between the mobile device and SharePoint and presented in *Table 7.* The headers are the method calls and method response, while the body is the parameters of the operation. For instance, for the process of claims authentication, the method call and location of the STS are contained within the headers, while the authenticating information such as username and password are contained within the body. Similarly as for pingpong, the body of the file transfer request corresponds exactly to the size of the requested file, while the headers are the overhead of the operation. Furthermore, it should be noted that no server execution times were recorded for the SharePoint portal.

Operation	Byte sent	Byte received
Authentication (Claims)	headers:132; body:437	head:112 ;body:378
Download file (52 432B)	headers:592;body:0	head:721;body: 52 432

Table 7 - SharePoint Time behaviour

While the requests and responses contain some information that is specific both to the requested operation and the implementation, examining the transmitted data can still provide an approximation of the information transmitted between the client and server. Each request contains a lot of information, both in the headers and body. The header contains meta information about the message. Therefore, the headers are of interest to examine. *Table 8* displays the headers of the SharePoint authentication and part of the authentication for pingpong. The first row of the table is the request, and the second row is the reply corresponding to that request.

SharePoint	pingpong
POST /sites/KursPortal/_vti_bin/authentication.asmx HTTP/1.1 User-Agent: ksoap2-android/2.6.0+ SOAPAction: http://schemas.microsoft.com/sharepoint/soap/Login Content-Type: text/xml;charset=utf-8 Connection: close Accept-Encoding: gzip Cache-Control: no-cache Pragma: no-cache Host: 10.11.127.10:200 Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2 Content-Length: 437	GET https://pingpong.chalmers.se/startPage.do HTTP/1.1 Host: pingpong.chalmers.se Connection: keep-alive Cache-Control: max-age=0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31 Referer: https://idp.chalmers.se/adfs/ls/?SAMLRequest=fZFLb4MwEIT%2FCvI9GENDghWQaHJopLRFgfbQS2WDGywZm3pNH%2F%2B%2BJPSVS847OzP77QpYp3qaD67Ve%2FE6CHDeR6c00NMgRYPV1DCQQDXrBFBX0zK%2F3dHQD2hvjTO1UcjLAYR10ui10TB0wpbCvslaPOx3KWqd64Fi3Et96I0%2B%2BHXL1KgBHwQuW8m5UcK1PoDBR%2B8QF%2FdlhbzNWEZqdrT9M5FNf7bPmhfACjDytpsUPXORRDyJkyXnccjDRbRckKQWwVUUxQlpolEGMIitBse0S1EYkGgWzGckrEhMSUij%2BRPyiu%2FDrqVuxtKXKfBJBPSmqorZVP1xbHeqPQpQtjqypKdg%2B4%2FuZVv2gxRlFwHCL8AV%2Fpczhfb0bjTebgqjZP3p5UqZ97UVzIkUEYSzaeX8%2F9kX&Rel

	ayState=ss%3Amem%3A6ed1bc8e58e25c13bb1e69ba985c7a25 Accept-Encoding: gzip,deflate,sdch Accept-Language: en-US,en;q=0.8 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3 Cookie: __utma=222584388.718323499.1364307004.1364384342.1364992772.3; __utmz=222584388.1364992772.3.3.utmcsr=google|utmccn=(organic)|utmcmd=organic|utmctr=(not%20provided); _shibsession_64656661756c7468747470733a2f2f70696e67706f6e672e6368616c6d6572732e73652f73686962626f6c657468=_0135f69fd517acc6aff90409456f226a; JSESSIONID=A0744BFC9951E32703098B9EE3A3A5A0; PPLoggedIn=18516_chalmers
HTTP/1.1 200 OK Cache-Control: private, max-age=0 Content-Type: text/xml; charset=utf-8 Content-Encoding: gzip Vary: Accept-Encoding Server: Microsoft-IIS/7.5 SPRequestGuid: 27fd832a-e08e-42e6-bc3b-ed21b8002404 X-SharePointHealthScore: 0 X-AspNet-Version: 2.0.50727 Set-Cookie:	HTTP/1.1 200 OK Date: Sun, 12 May 2013 16:14:14 GMT Server: Apache/2.2.22 (FreeBSD) DAV/2 mod_jk/1.2.32 mod_ssl/2.2.22 OpenSSL/0.9.8q X-UA-Compatible: IE=edge X-PP-Userid: 18516_chalmers Pragma: no-cache Cache-Control: no-cache Vary: Accept-Encoding Content-Encoding: gzip

FedAuth=77u/PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTgiPz48U1A+MCMuZnxhc3BuZXRzcWxtZW1iZXJzaGlwcHJvdmlkZXJ8cmlraSwwIy5mfGFzcG5ldHNxbG1lbWJlcnNoaXBwcm92aWRlcnxyaWtpLDEzMDEyMDM0NDcwMDU0ODEyNSxUcnVlLFNlb0p5bnc2ekVTVzRLaHUzQ0xpbFl5dUtscGVWN09Gc2pvQm00YkRPTW4xMCsxL2hOc0JmVm5BcC90YU5DR0RrL3VWTkx2TklQQitzV0pHdXhqUEhKQUwzUkFtbkRWUlR6UituaUFiUlU1anl6blZsbnJlbDJ0ME9zcGhyOThlU2hzb3lrcFk4L3VkK0hlWTh4YWIrb3FlRDAyOEFYMUt2VmdhM2t6UGN2eElCLzZhMXFVRXpXQmJVNWUrSDUxTDY0em05eCs3L3Rhd3hIaGZTMFVIUjdNQXBOQWR4Sk9uME1mSFM3VzRuT0tiVy9jdVJacldDVEZiZmp0YjVhRVFYWjhwcy9SNVFBYTFkSHhMWkpqelZPSmUyNk91a2NSZ25oaS9oejIxN3V6RWlqY015SUIwS0p6V1ZaMS9hRy81SVFsMFZuSTF4bHFJWFVlaENaTjF1QT09LGh0dHA6Ly8xMC4xMS4xMjcuMTA6MjAwL192dGlfYmluL2F1dGhlbnRpY2F0aW9uLmFzbXg8L1NQP g==; expires=Fri, 03-May-2013 06:01:10 GMT; path=/; HttpOnly X-Powered-By: ASP.NET MicrosoftSharePointTeamServices: 14.0.0.4762 Date: Thu, 02 May 2013 20:01:10 GMT Connection: close Content-Length: 378	Content-Length: 5032 Keep-Alive: timeout=2, max=2 Connection: Keep-Alive Content-Type: text/html;charset=UTF-8

Table 8 – example request /response headers

Resource utilisation

The resources considered in this section are limited to CPU and memory usage of the mobile device. The bandwidth is a resource of the system and could be included here. However, since it were already considered in the previous section as a means to determine the time required for and operation disregarding the latency, it will not be considered again in this section. The operations these measurements were carried out for are as follows:

- Authentication.
- File transfer.
- Accessing course announcements.

While authentication and file transfer were also measured for Time behaviour, the measurements were performed separately. That is, the operation was carried out to record Time behaviour, than the operation was carried out anew to record Resource utilisation.

The measurements related to the authentication process for pingpong are shown in *figure 13.* For each operation, the graphs depict the recorded measurements, with each measurement as a point in the graph. As shown, the amount of CPU used by the system during the authentication process is fairly consistent for Chrome, while Browser had a higher CPU consumption. However, the memory consumption was consistent between chrome and browser. Nevertheless, the measurements for SharePoint's authentication, as shown in *figure 14,* were even more consistent and showed even less divergence than pingpong. The memory consumption showed less than a percent deviation between the highest and lowest measured value. Similarly, the values for CPU usage also showed only a minor divergence between the highest and lowest recorded value.

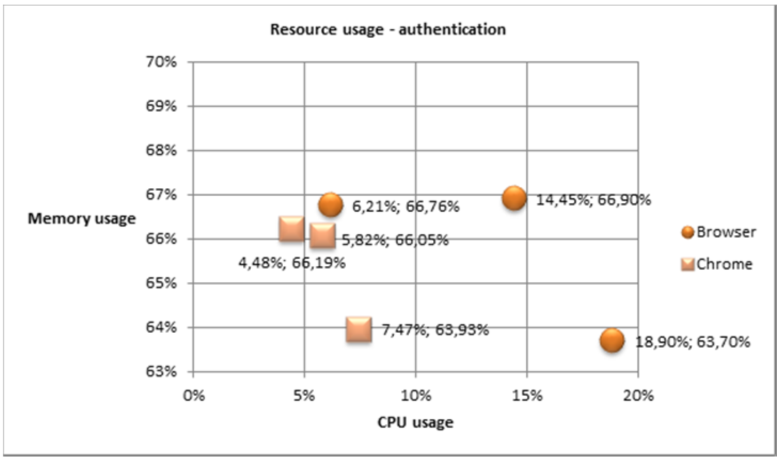


Figure 13 - pingpong Resource utilisation - authentication

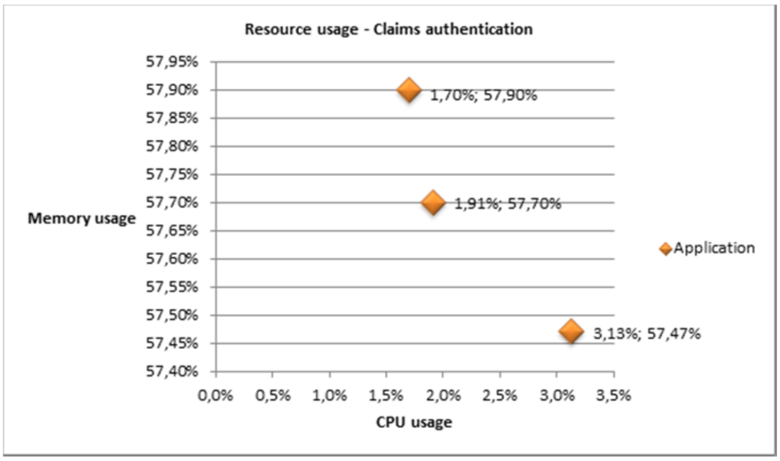


Figure 14 - SharePoint native application Resource utilisation - Claims authentication

The measurements related to the file transfer process for pingpong are shown in *figure 15. T*he memory consumption for Chrome is consistent while the memory consumption for Browser shows a few percent divergence. The CPU usage, on the other hand, exhibited a more significant divergence. In particular, the difference between the greatest and lowest recorded value, for Browser, was slightly more than ten percent. In contrast, the CPU usage of the mobile application, as shown in *figure 16,* only displayed a minor divergence between the highest and lowest recorded values. That is, slightly above two percent, as opposed to the ten percent of Browser. Furthermore, the recorded measurements for memory consumption only deviated by a few percent, and thus displayed a greater degree of consistency than the CPU consumption. However, it is important to note that the results from the file transfer measurements of pingpong and SharePoint used the exact same file.

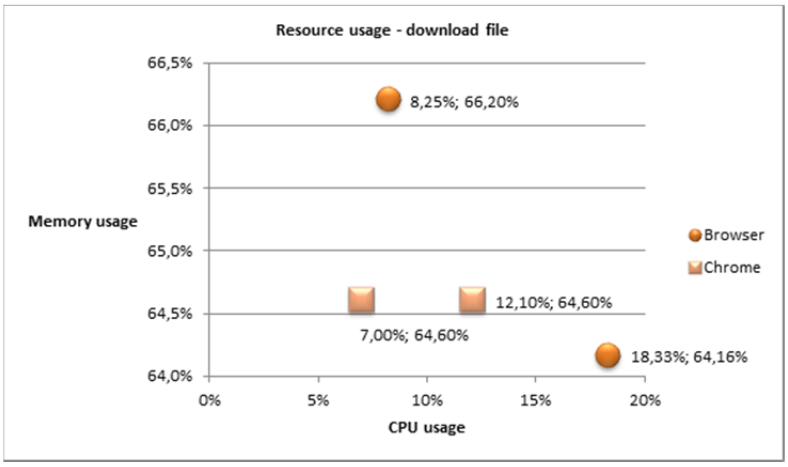


Figure 15 - pingpong Resource utilisation - download file

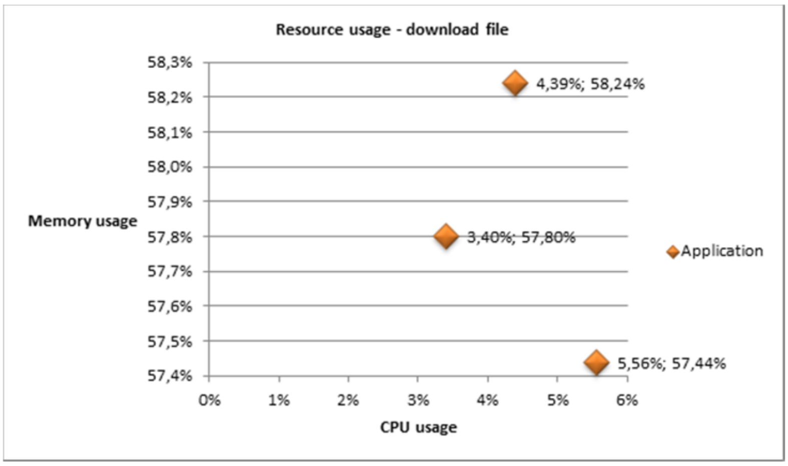


Figure 16 - SharePoint native application Resource utilisation - download file

The measurements related to the course announcements displayed the least consistency of all the examined operations. The measurements for pingpong (*figure 18*) exhibited a larger divergence for memory consumption when accessed through a web browser. However, Chrome had less divergence than Browser. Furthermore, the recorded values for CPU consumption displayed significant divergence. In particular, if the type of web browser accessing the resource is disregarded, In this scenario, the least CPU consumed (4,3%), compared to the greatest amount of CPU consumed (20,7%). While this divergence in CPU consumption is significant, the higher consumption of CPU resulted in less memory consumption for Browser. However, this relationship between CPU and memory consumption was less clear for Chrome. Similarly as for access through a web browser, the measurements for access through an application (*figure 17*) displayed significant divergence between the highest and lowest recorded values for both CPU and memory. Furthermore, when the application consumed a higher amount of memory, the CPU consumption was reduced.

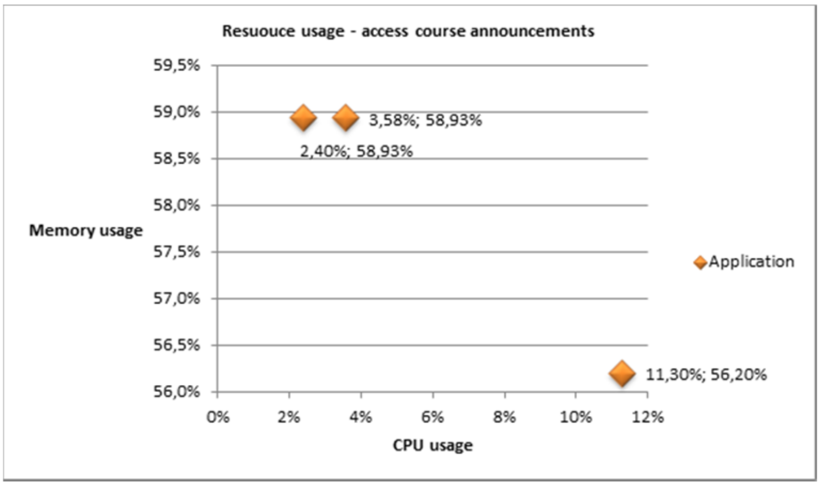


Figure 17 - SharePoint native application Resource utilisation - access course announcements

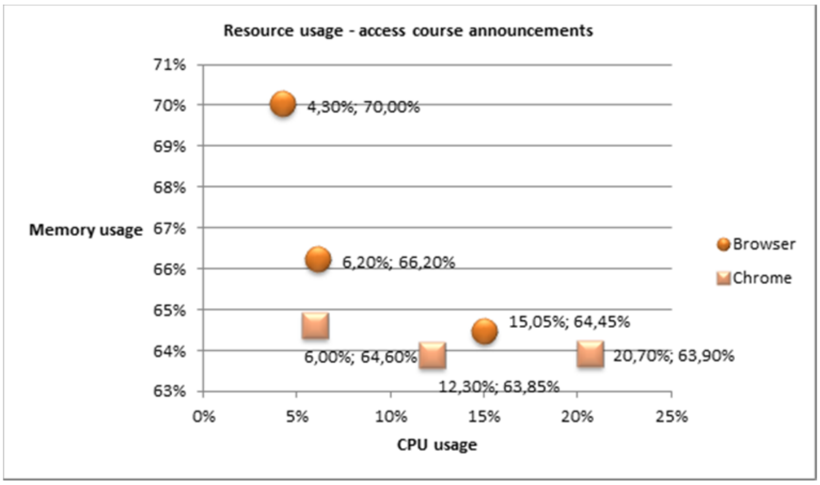


Figure 18 - pingpong Resource utilisation - access cource announcements

There are several measurements for each operation. In order to provide an overview of these results, and an estimation of the amount of resources an operation can be expected to consume, the average resource consumption for the operations were calculated. However, the average does not account for skewed values. Therefore, the median was also calculated. This summary is presented in *Table 9* and *Table 10.*

	pingpong	**SharePoint**
Authentication	Average:65,59% Median:66,12%	Average:57,69% Median:57,7%
File transfer	Average:64,89% Median:64,6%	Average:57,83% Median:57,8%
Access course announcements	Average:65,5% Median:64,53%	Average:58,02% Median:58,93%

Table 9 - memory consumption summary

	pingpong	**SharePoint**
Authentication	Average:9,52% Median:6,84%	Average:2,25% Median:1,91%
File transfer	Average:11,42% Median:10,18%	Average:4,45% Median:4,39%
Access course announcements	Average:10,38% Median:8,12%	Average:5,76% Median:3,58%

Table 10 - CPU consumption summary

The resource consumption is higher for pingpong both in terms of CPU and memory consumption. However, in particular, the CPU consumption, which is reduced to a fraction of the CPU consumed for pingpong when the same operation is carried out through the mobile application.

5 Discussion

The purpose of the work presented in this thesis was to investigate to what extent SharePoint is a suitable system for developing a portal, if the portal is accessed through a mobile device. That is, if the functionality and content can be provided with efficiency to a mobile device in a manner that does not exclude some types of mobile devices. For the functionality, the factors of consideration was if the required task could be performed (Suitability) and if the performed operation produced the correct result (Accurateness). While there are several components of the system, the factors considered for Efficiency were limited to aspects of Efficiency on the mobile device. More specifically, the time required by the mobile device to perform a select operation (Time Behaviour) and the resources consumed by the mobile device in the process of accessing and manipulating data and content on the portal (Resource utilisation).This section will discuss the outcome of the investigation in order to answer the research question.

Since the factors investigated are Functionality and Efficiency, these characteristics can be used to refine the research question. That is, how is the functionality of the system affected when an application on the mobile device is used to access a resource and is it an effective system. The measurements for pingpong where the Functionality and Efficiency were considered are used to provide context to the measurements related to SharePoint. Furthermore, Functionality and Efficiency can both be considered in terms of the sub-characteristics.

How the resources are exposed, through WCF, are intended to not impose restriction on the type of device that can access the resource. The approaches considered in this paper are very similar, both SOAP and REST communicate with HTTP request, the difference is that SOAP specifics the requested operation within the body of the message while REST determines the operation to be performed based on the URL of the request. However, while this approach does not exclude any mobile device from accessing and manipulating data on SharePoint, the process of authentication might exclude types of mobile devices from direct access. Claims authentication is performed through SOAP messages and is therefore available for the mobile devices. In contrast NTLM required an intermediary to provide access to some types of mobile devices.

By using proxies to access and manipulate the content on behalf of the client, the clients only have to produce a request for the information. Consequently, when the system scales in terms of users, the resources consumed by the mobile device will be consistent. In contrast, the strain on the server will increase with the number of users of the system. That is, the resources consumed by the server when increase with the number of users of the system increases. However, how the amount for resources required by the server in order to process the requests will increase might be of interest but is outside the scope of this paper. More specifically, if the amount of system users double, and the amount of requests double, how much resources will be consumed by the server in order to process this increase in requests, since it is not certain that the resources required to process the requests will increase. In contrast, the bandwidth required by the system will scale linearly since the same messages will be transmitted regardless of the number of clients within the system. Consequently, this approach will minimize the resource consumed by the mobile device regardless of the number of users within the system. However, the resources consumed by the rest of the system are not considered within this paper.

5.1 Analysis of Efficiency

By not measuring the time directly but rather deriving the transmission time from the amount of data transmitted, the measurements are independent of the implementation. That is, regardless of where and how the system is implemented the same amount of data needs to be transmitted in order to perform an operation. If the time had been measured directly instead, then factors such as latency would have affected the measurements. The measurements of resource consumption were carried out with a specific device. Consequently, if another device had been used, the resulting measurements might have been different. However, the manner in which the mobile device communicates with the system is the same for both pingpong and SharePoint. That is, for both systems resources are acquired by HTTP requests. Therefore, while another mobile device might have produced another measurement for Resource utilisation, the relationship between the values for pingpong and SharePoint would likely be consistent. That is, another mobile device might have a different architecture and consequently consume a different amount of resources on the system. However, if a method for communication consumes more resources when both methods use the same approach, it is likely that it will consume more resources on another system as well.

Reduced resourced consumption of the mobile device in terms of CPU and memory usage

When the resources of the portal are accessed through a web browser, a lot of functionality is included in the browser. This functionality consumes resources of the system even if it is not currently in use. That is, even if not all the functionality of the web browser is currently being used, the code containing the functionality still has to be loaded into memory. However, when an application is used to access the portal's resources, the functionality of the application can be customized to the bare essentials. Consequently, memory consumption was consistently lower when the resources were accessed by an application on a mobile device rather than through a web browser. Similarly, CPU consumption also showed a similar pattern, however, not to the same extent as for memory consumption. That is, for memory consumption, the highest recorded value for a mobile application was lower than the lowest recorded value for the web browser for every investigated operation. Similarly, for CPU consumption, for the operation of accessing course announcements, a single recorded value was greater than some of the recorded values when the same operation was carried out through the web browser. However, for the other investigated operations, the CPU consumption showed the same drastic reduction as the memory consumption where the highest recorded value for the application was lower than the lowest recorded value for the web browser. Despite that, both the average and median consumption of CPU and memory were for every investigated operation lower when the operation was performed by the mobile application.

Reduced overhead

When performing some operation or accessing some resource in the CMS, additional data beyond the requested data may need to be transferred. There are several distinct purposes for this additional information. In some cases, this information provides access control to the resources. In other cases, the additional information determines the resource to be accessed or the operation to be performed. However, in some cases this overhead is resources that the user did not request. In particular, the overhead of the authentication of pingpong in comparison to pingpong's file transfer. The authentication had significant overhead both in terms of transmitted data and server processing time for the request. This overhead was largely in the form of a redirect triggered by the operation. Consequently, a new web page was transmitted. This web page contained structural information regarding how the content should be displayed in the form of html and css files, content of the web page such as images, and also functionality in the form of JavaScript. If the mobile application also wanted to perform this side effect, the images would have to be transferred. However, the structural information, how to display the information, is a part of the application. While containing some of the information within the application reduces the information transmitted, it increases the information that needs to be stored. Similarly, the images could also be contained within the application. The client needs to process the information regardless of where it originates. However, if the content is stored on the client, the server does not need to process it, and it does not need to be transmitted.

The approach of using a mobile application can reduce the overhead since less information needs to be transmitted. This will result in less information that needs to be transmitted and less resources needed to process requests. However, this is at the cost of storage on the mobile device. While the approach of accessing the portal through a web browser offers the possibility to cache information, how the application structures the information is part of the applications code and is thus more stable.

However, in the cases where the requested operation does not have any side effects, the requests are very similar for both systems. The request, for both systems, consisted of a HTTP verb specifying the resource to be accessed. This request had overhead in the form of access controlling information, for security. In the cases where the user submitted information to the server, the overhead also included information to help interpret the user submitted information.

In summary, while, the request contained very little overhead for both system, the responses contained significantly more overhead from pingpong then from the mobile application alternative. That is, because, the response contains meta information, which specifies how the information should be displayed to the user. When an application is used, the structural information, how the received information should be displayed is part of the application.

Reducing the overhead further is a complex matter. The overhead caused by access control is necessary and should not be omitted for performance issues. Likewise, information specifying the resource to be accessed and how the transmitted information should be interpreted is difficult to reduce. However, the actual content can be compressed. While this would result in less information being transmitted, the process of compression for the server, and the process of decompression for the client would both cause new overhead.

5.2 Analysis of Functionality

While the measurements related to functionality are not related to a specific implementation for SharePoint, the manner in which the investigation is carried out imposes restrictions. That is, since the investigation was carried out as a comparison, the system that the comparison is carried out against must provide the functionality to be examined. Consequently, aspects of functionality that are not present at pingpong cannot be compared. Such as operations not performed by pingpong and the Interoperability sub-characteristic of functionality. However, regardless of the system compared against, the functionality has to be present in both systems for a meaningful comparison and pibngpong contains sufficient functionality that it is not

Maintained functionality

The default manner in which a user interacts with a CMS is through a web browser. Therefore, the CMS will expose its content and functionality for access through a web browser. However, it is possible to obtain the content and functionality through an application on a mobile device. While it can be provided, it is also important to consider how the functionality is provided. When the portal is accessed through a web browser, the information accessed includes how the information should be displayed. When the application is used to access the information, the information regarding how to display the information to the user is omitted. Consequently, the application must programmatically specify how the information should be displayed to the user. However, regardless of where the directives specifying how the information should be displayed are obtained, they still have to be processed by the mobile device. Consequently, when the information is accessed by an application on the mobile device, the application must know the format of the information in order to display it to the user in a meaning full way. More specifically, the information is obtained as text in XML format, if it should be displayed in some other manner to the user it must be processed. This means that the mobile application must know the format of the data it will receive.

When new content is added to the portal, the alternative solutions will behave differently. On the one hand, pingpong accessed through a web browser will display the new content. On the other hand, the mobile application accessing the SharePoint site will not display the new content before code is added to the application that manages this new content. This management process could be in the form of requesting the information, and then processing the reply and displaying it to the user.

In summary, while the same functionality can be provided in both cases, providing it for the application on the mobile device requires a bit more effort. Furthermore, as discussed in the introduction, an application on the mobile device can access functionality specific to the mobile device that is out of reach for the web browser. Consequently, since the functionality that the CMS provides through a web browser can be obtained through an application, the alternative of using an application provides greater opportunities in terms of functionality. That is, while both alternatives can provide the functionality of the CMS, an application can integrate the functionality of the CMS with the functionality of the mobile device.

6 Conclusion

This paper explores content management within SharePoint from an application on a mobile device in an approach that is not limited to a specific type of mobile device, or type of application. That is, this approach is viable from both hybrid and native applications and not restrictive to a certain type of mobile device. In order to develop a more nuanced perspective of the approach, a portal was implemented within SharePoint. This portal was accessed from an application on the mobile device, and compared to an existing portal while accessed through a web browser. This comparison of the systems was carried out according to the standard ISO/IEC 9126.

The default means of accessing a CMS, such as SharePoint, is through a web browser. Therefore, the infrastructure necessary in order to access and manipulate the content from a web browser is provided by the CMS. In contrast, if the content is accessed in a manner that was not intended, the infrastructure has to be developed. Due to the constraints on development for mobile devices, for SharePoint, the information within the portal could not be accessed directly from a mobile application. However, while the application on the mobile device was unable to access and manipulate the data directly, it could still be accessed from an entity that does not share the restrictions of the mobile devices. Therefore, a proxy was used to access resources within the portal on behalf of the mobile application. Since SharePoint contains such a proxy, it did not have to be developed. However, in the case that the portal used NTLM authentication which Windows phones do not support, an additional entity was implemented. The sole purpose of this additional entity was to allow the performing of authentication of the requests on behalf of the application on the mobile device.

A limitation that should be acknowledged is that, while there are several components in both systems, the measurements were taken only from the mobile device in both cases. Furthermore, no access to pingpong other than executing requests and observing the results were available. This limited the possible measurements. In addition, the measurements were carried out manually, which reduced the number of measurements and the number of operations investigated.

The same functionality was provided to the mobile device from both systems. However, the functionality of the systems is not limited to the functionality that can be accessed from the server. In particular, the alternative of using an application on the mobile device can beyond providing the functionality of the CMS also integrate that functionality with that of the mobile device. That is, the mobile application can integrate the functionality provided by the hardware of the mobile device with the functionality accessed on the server. However, this can in some cases, depending on the specific functionality, also be achieved through html.

The time required in order to perform some operations was reduced for the system using SharePoint through an application on a mobile device. This was partly because some operations had side effects for pingpong but more significantly because of the responses when the client requested some operations. That is, while, the request contained very little overhead for both systems, the responses contained significantly more overhead from pingpong then from the mobile application alternative. More specifically, the response contained meta information, which specifies how the information should be displayed to the user. When an application is used, the structural information, how the received information should be displayed is part of the application.

For every investigated operation, the mobile device consumed fewer resources when the operations were performed on SharePoint through an application on the mobile device. The cause of this is not determined in this work. Hypothetically, it could be caused by less data required from the server and thus be related to Time behaviour. Another possibility is that the web browser contains a lot of built in functionality that will take resources of the system. While the cause of the reduced resource consumption is not clear, the implications of it are. Since fewer resources are being used to perform the same operation, the mobile device may either provide the user with more functionality for the same resource consumption or provide the same functionality more effectively.

The advantages for using SharePoint over pingpong can largely be attributed to whether the system is accessed through an application or through a web browser. This approach requires the system to expose the content of the portal in a manner which is accessible for the mobile application. SharePoint contains a number of web services that exposes the content of the CMS. This limits the effort required to develop this system in SharePoint, and, therefore, suggests that SharePoint can be used advantageously in this case.

Since the measurements were intended to compare approaches to content management rather than specific implementations, the possible measurements were severely limited. However, there are possible other factors that could be used to examine content management from a mobile device which would provide a meaningful extension to this research. More specifically, this work focused on how the client is affected by the system, examining how the server is affected by this approach would provide a meaningful extension to this work.

References

[1] Microsoft (2010) Overview of Mobile Pages and the redirection system http://msdn.microsoft.com/en-us/library/ms462572(v=office.14).aspx (13-02-01)

[2] Microsoft Office Team (2010) Configuring Sharepoint 2010 for mobile device access http://blogs.technet.com/b/office2010/archive/2010/03/09/configure-sharepoint-server-2010-for-mobile-device-access.aspx (13-02-04)

[3] Microsoft (2010) Mobile Pages http://msdn.microsoft.com/en-us/library/ms479269(v=office.14).aspx (13-02-04)

[4] Barber, G.B., Haque, N., Gardner B. (2009) 'OnePoint' - Combining OneNote and SharePoint to facilitate knowledge transfer. Drug Discovery Today, vol 14, nr 17/18, 845-850

[5] Microsoft Planning Your IIS Architecture http://www.iis.net/learn/get-started/planning-your-iis-architecture (12-03-27)

[6] Baier, D., Weyer, C., Maine, S. (2007) IIS 7.0:Extend Your WCF Service Beyond HTTP With WAS http://msdn.microsoft.com/en-us/magazine/cc163357.aspx (13-02-07)

[7] Microsoft (2012) Hosting in Windows Process Activation Service http://msdn.microsoft.com/en-us/library/ms734677.aspx (13-02-08)

[8] Microsoft What Is Windows Communication Foundation http://msdn.microsoft.com/en-us/library/ms731082.aspx (13-02-08)

[9] Microsoft Windows Communication Foundation http://msdn.microsoft.com/en-us/library/vstudio/ms735119(v=vs.90).aspx (13-03-01)

[10] Microsoft (2012) Fundamental Windows Communication Foundation Concepts http://msdn.microsoft.com/en-us/library/ms731079.aspx (13-02-07)

[11] Microsoft (2013) Designing and Implementing Services http://msdn.microsoft.com/en-us/library/vstudio/ms729746(v=vs.90).aspx (13-02-05)

[12] Microsoft (2010) WCF Services in SharePoint Foundation 2010 http://msdn.microsoft.com/en-us/library/ff521586.aspx (13-02-11)

[13] Microsoft Use the REST Endpoint for Web Resources http://msdn.microsoft.com/en-us/library/gg334279.aspx (13-02-11)

[14] Microsoft An Introduction to RESTful services with WCF http://msdn.microsoft.com/en-us/magazine/dd315413.aspx (13-02-11)

[15] w3schools SOAP Tutorial http://www.w3schools.com/soap/default.asp (13-02-11)

[16] Singh, R.R., (2012) A Beginner's Tutorial for Understanding ADO.NET http://www.codeproject.com/Articles/361579/A-Beginners-Tutorial-for-Understanding-ADO-NET (13-02-11)

[17] Microsoft (2012) Plan authentication methods (SharePoint Server 2010) http://technet.microsoft.com/en-us/library/cc262350(v=office.14).aspx (13-02-15)

[18] Chappell, D. et. al (2011) Claims-based identity for Windows

[19] Brown, k., Mani, S. (2009) Windows Identity Foundation (WIF) Whitepaper for Developers. Microsoft

[20] Crawford, D. et al (2011) Implementing Claims-Based Authentication with SharePoint server 2010. Microsoft

[21] Urbach, N. et al (2010) An empirical investigation of employee portal success. Journal of Strategic Information Systems, vol 19, nr 3, 184-206

[22] Shilakes, C.C., Tylman J., (1998) Enterprise Information Portal - Industry Overview. Merrill Lynch, Inc., New York City

[23] Daniel, Elizabeth and Ward, John (2005). Enterprise portals: addressing the organisational and individual perspectives of information systems. In: Bartmann, D.; Rajola, F.; Kallinikos, J.; Avison, D.; Winter, R.; Ein-Dor, P.; Becker, J.; Bodendorf, F. and Weinhardt, C. eds. Proceedings of the Thirteenth European Conference on Information Systems. Regensburg, Germany: ECIS Standing Committee.

[24] Bevan, N., (1999) Quality in use: Meeting user needs for quality. The Journal of Systems and Software, vol. 49, nr 1, 89-96

[25] Microsoft Using the REST Interface http://msdn.microsoft.com/en-us/library/ff798339.aspx (13-02-15)

[26] Estrem, J., (2011) ListData.svc returns HTTP 500 error on Discussion list. Bug? http://community.office365.com/en-us/forums/154/t/16404.aspx (13-02-17)

[27] International organization for Standardization (2003) “ISO Standard 9126: Software Enginering – Product Quality, parts1, 2 and 3”, International organization for Standardization, Geneve. Cited in Botella, P. et al (2004) "ISO/IEC 9126 in practice: what do we need to know?", In Software Measurement European Forum 2004, 297-306

[28] Losavio, F. et al, (2003) Quality Characteristics for Software Architecture. Journal of Object Technology, vol. 2, nr 2, 133-150

[29] Zeiss, B. et al (2007) Applying the ISO 9126 Quality Model to Test Specifications - Exemplified for TTCN-3 Test Specifications

[30] Kitchenham, B., Pfleedger, S.L., (1996) Software quality: the elusive target. IEEE software, vol. 13, nr 1:January, 12-21

[31] Carmines, E.G., Zeller, R.A., (1979) Reliability and Validity Assessment. Sage University Paper Series on Quantitative Applications in Social Sciences, Sage, Newbury Park

[32] Microsoft (2008) How Windows NT Challenge/Response works http://support.microsoft.com/kb/174775 (13-02-17)

[33] Microsoft (2011) Windows Authentication Documentation for Windows 7, Windows Vista, Windows Server 2008 R2 and Windows Server 2008 http://technet.microsoft.com/en-us/library/cc755284(v=ws.10).aspx (13-02-18)

[34] Microsoft (2013) Microsoft NTLM http://msdn.microsoft.com/en-us/library/windows/desktop/aa378749(v=vs.85).aspx(13-02-18)

[35] Microsoft (2013) NTLM Over Server Message Block (SMB) http://msdn.microsoft.com/en-us/library/cc669093(v=prot.20).aspx (13-02-18)

[36] Microsoft (2013) Microsoft SMB Protocol Overview (Windows) http://msdn.microsoft.com/en-us/library/windows/desktop/aa365233(v=vs.85).aspx

[37] Microsoft (2012) Authentication, authorization and security in SharePoint 2013 http://msdn.microsoft.com/en-us/library/ms457529.aspx (13-02-19)

[38] IDC (2013) IDC Press Release http://www.idc.com/getdoc.jsp?containerId=prUS24108913 (13-06-19)