Messaging for Office Doors through Ubiquitous Systems

David Schneider
Department of Computer Science
University of Cape Town
dave.schneider@gmail.com

Neil Stuppel
Department of Computer Science
University of Cape Town
nstuppel@gmail.com

ABSTRACT

The goal of the MODUS project is the creation of a communication system allowing visitors the ability to leave messages for or book appointments with staff members. This implementation exists in a ubiquitous environment that creates problematic conditions for both usability and security aspects of computing.

The MODUS project aims to ensure that the system is as secure against attack as possible. This involved developing security features according to fundamental security principles: Authentication & Authorization, Integrity and Confidentiality. During the course of development, issues relating to usability and security in a ubiquitous computing environment were identified and explored.

Throughout the design and implementation phases of the project, solutions to the usability and security issues were identified, tested, and implemented. Ultimately, the outcome of the MODUS project is a fully-functional system that achieves the goals set out at the beginning of the project.

1. INTRODUCTION

Recent improvements and increased popularity in mobile technology have created an environment where users are familiar with such technology and willing to use such technology in their everyday lives.

The objective of the MODUS project is the establishment of a ubiquitous network of Personal Digital Assistants mounted on office doors at the location of this research. Conventionally, staff members allow visitors to leave messages on post-it notes or white-boards. This solution is not always effective as visitors are able to remove the post-it notes and alter them. This solution does not provide a means for 2-way communication required for prompt and reliable interaction with staff members. Visitors are able to communicate with the staff members although the staff members are not able to communicate with the visitors.

Projects like HERMES have investigated the creation and maintenance of an office door messaging system in a ubiquitous environment. This project provides new ideas in implementing usable systems. This research however does not investigate the application of security for ubiquitous messaging systems.

This paper presents research in developing a multi-functional messaging system in a ubiquitous environment. The paper details its design, implementation and evaluation. The project attempts to extend the research based on the previous work and attempt to provide an investigation of the security application that is required in implementing a messaging system in a ubiquitous environment.

To ensure that the system is secure, the approach taken has been to implement the system according to the fundamental security principals: Authentication, Authorization, Confidentiality, Integrity and Availability.

The MODUS project aims to

- Create an efficient and convenient system which can assist in communication between staff members and visitors
- Provide a secure communication s environment to ensure all interaction between visitors and staff members remain confidential.

The MODUS system is divided into separate sub-systems. These include:

- The MODUS client provides a front end to the visitors to post messages and schedule appointments.
- The MODUS web service that provides the secure communication between client and back end server.
- The MODUS web administrator to provide each staff member a means to view messages and create appointment time slots for visitors.

2. RELATED WORK

This chapter serves to provide the reader with an overview of prior work carried out in the field of ubiquitous computing as well as existing research into the topic of “pervasive security”[3]. Included is a discussion of identified points of concern which could prove detrimental to the integrity of a ubiquitous system.

1.1 Usability

The HERMES[1] system implemented at Lancaster University is similar in concept to the MODUS system. It is displayed in figure 1. Developed by Cheverst et al, it allows visitors to leave messages on office doors in the Computing Department at Lancaster. The HERMES project focused on the usability of such a ubiquitous system, whereas the MODUS project aims to expand on their work by exploring the security ramifications of a ubiquitous messaging system. This decision was made as the MODUS project team felt that research performed in a fresh context would be more useful than carrying out research similar to that already accomplished in the UK.

Figure 1: HERMES System at Lancaster University

In the book **Designing the User Interface**, Ben Shneiderman identified 8 principles that should be adhered to when designing interfaces for desktop applications. [9]
[10] discusses the limitations of mobile device interfaces using the 8 principles as specified by Ben Shneiderman and moves to adapt these principles for the development of mobile devices.
It suggests that four of the original guidelines can be directly used when developing interfaces for mobile devices

- Enable frequent users use of shortcuts
- Offer informative feedback
- Design dialogs to yield closure
- Support internal locus of control

The remaining 4 guidelines require modification to adapt to mobile devices due to resource limitations including memory space, limited screen size and slow response times.

The paper moves on to propose new guidelines to developing interfaces for mobile devices

- Design for multiple and dynamic contexts
- Design for small devices
- Design for limited and split attention
- Design for speed and recovery
- Design for "top-down" interaction
- Allow for personalization
- Design for enjoyment

1.2 Security

1.2.1 Authentication & Authorization

"Both features are needed, in order to restrict any malicious user from entering the network or to prove one's own identity."

Most research into the field of pervasive security focuses on the nomadic character of most ubiquitous computing systems. Haque and Ahamed provide the following definition of such a network: "Thousands of pervasive devices are able to arbitrarily join and leave a network, creating a nomadic environment known as a pervasive ad hoc network." [3] This description does not quite fit the system under development by the MODUS project team. The office door system is better described as a pervasive managed network as devices cannot leave and join the network without express permission of the network administrator. The Vigil[2] system developed by Kagal et al is founded on the concept of trust based security. Vigil affords trusted users the authority to delegate their access rights to other personnel. However, this is irrelevant in the case of the MODUS system as the low number of devices on the network allows for management through an access control list.

Certificates are a fairly high-maintenance form of authentication as they must be issued by a verified Certificate Authority. However, as devices rarely leave or join the network in question, the initial overhead is justified. In a position paper on security in ubiquitous computing, Creese et al discuss the notion of certificates verifying certain attributes of a node[6]. The paper makes the point that names are fairly irrelevant in ubiquitous networks as they do not dictate much information regarding the type of device or which services it may attempt to access. The idea of verifying attributes is extremely useful to the development of the office door system. Each PDA has several attributes that are strong candidates for authentication purposes. As communication takes place over the 802.11 wireless network, the MAC address of the PDA can be used as an identifier. Note that the wireless access point can easily be configured to filter unknown MAC addresses as suggested by Bhagyavati et al[5], however this can be defeated with MAC spoofing[4]. However, MAC filtering remains a viable candidate as there is still a degree of complexity involved in spoofing a MAC address.

1.2.2 Integrity

"Integrity indicates information has not been altered or falsified by an unauthorized user." [3]

Ensuring the integrity of messages stored on the device assures users that the messages communicated to them originate from the stated source and have not been tampered with in any way. The importance of maintaining integrity in the office door system is obvious when considering the academic consequences of a professor receiving fraudulent information regarding a student.

Integrity can be compromised by man in the middle attacks[4]. In these instances, attackers place a rogue access point in the vicinity of the network which act like legitimate AP's. The nodes may then attempt to connect to the rogue AP, supplying it with vital authentication information and the contents of messages intended for the central server. Attackers are then able to modify message contents (provided they have broken the encryption in use) before passing the message on to a legitimate access point. As Zahur and Yang point out, these attacks are possible as there is no mutual authentication. By this, they mean that access points are not required to authenticate themselves to clients. As such, clients transmit sensitive information to unknown recipients.

1.2.3 Confidentiality

"Confidentiality ensures information is not exposed to any unauthorized user." [3]

This goal is effectively accomplished by encrypting data transmissions. This ensures that transmissions are not intelligible if intercepted by a third party. Also, third parties cannot transmit messages under the guise of legitimate users as they should not be in possession of the proper encryption key.

Bhagyavati et al discuss the issues inherent in Wireless LAN (WLAN) security[5]. They state that Wired Equivalent Privacy (WEP) encryption built into the 802.11 wireless protocol is weak and does not provide effective security against a determined attacker. WEP works by having a single key shared among every device and access point on the wireless network. However, due to the nature of the encryption, it is easily broken. If this occurs, attackers are free to intercept all messages left at any office door terminal.

1.2.4 Availability

As discussed by Haque and Ahamed, we must consider the limitations of the devices on the network [3]. PDAs are limited in battery life, memory capacity and processing power. While battery limitations can be overcome by supplying AC power to the device, the other two limitations can cripple availability and network security. Most forms of encryption are processor and memory intensive and cannot be practically implemented on such limited devices.

1.3 Summary

Research carried out on the subject of pervasive security is clearly geared towards solving the issues of authentication and authorization. This is most likely due to the fact that these issues are the most prominent due to the nomadic nature of a truly ubiquitous network. As such, literature does not tend to focus on other issues such as attacks on the integrity on the network. It is hoped that the findings emerging from this project will go some way towards filling this niche. Ubiquitous computing is clearly a growing field of interest due to the increased integration of computers into everyday life. However, if a serious security flaw is found in such systems, the outcome could shake consumer confidence in pervasive computing systems and hurt the advancement of this promising technology.

3. DESIGN AND IMPLEMENTATION

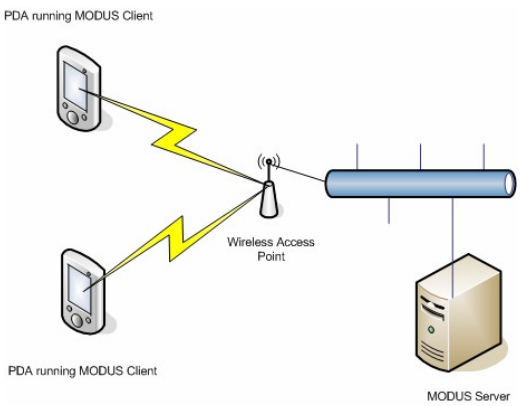

Figure 2: MODUS Network Diagram

The system consists of a series of HP iPaq Personal Digital Assistants connected via 802.11 Wifi to a central server as illustrated by figure 2. The server exposes a set of web services for communication with the deployed mobile devices. The server runs a database service for the storage of all information pertaining to the operation of the MODUS network. This includes fairly static information such as user login details as well as dynamic information such as messages left for device owners. The third technological grouping is that of remote clients. Device owners are able to login to the MODUS web administration system remotely and view messages and appointments that have been created on their devices. In this section, each aspect of the system is described in greater detail.

3.1 Client

The client is implemented using Visual C#.NET and the .NET Compact Framework. Custom user controls have been created to handle functionality not already implemented in the Compact Framework. These controls are

- DrawingControl: implements a panel on which the user can draw using the PDA stylus; and
- LoginPanel: implements a grid of 9 coloured buttons and raises an event when either four buttons have been pressed or the “Anonymous User” button is pressed.

The limitations of the .NET Compact Framework have caused problems with the implementation of some initial designs. The LoginPanel was initially conceived as a wheel divided into segments of different colours. Pressing a circular button at the centre of the larger circle would denote an anonymous user. However, this design required several functions missing from the Compact Framework. As such, the wheel approach was abandoned and the plainer squares approach was conceived.

The PDA display has been re-oriented using a third party piece of software titled Nyditot Virtual Display[8]. NVD replaces the resident display drivers on the PDA to enable a wide range of display settings not natively supported by the device. The only feature currently in use by the MODUS project is the ability to re-orient the screen to a right-handed landscape orientation. It was decided that the standard portrait orientation was not conducive to implementation of the agreed-upon design which called for an omnipresent panel on the left-hand side of the display.

Figure 3: Client home screen

The home screen shown in figure 3 is the initial point of contact for users of the MODUS system. It provides information about the staff member and is the starting point of any action the visitor may wish to take.

Owner Details
The owner details are dynamically retrieved from the database every time the MODUS application is loaded and the device brought online. The details are refreshed every 60 seconds.

Function Buttons
Leave a Message and **Make an Appointment** respectively initiate the processes of leaving a message for the staff member or making an appointment to see him. However, before these functions are accessible, the user is asked to either identify himself by entering a pass code or express a desire to remain anonymous.

Bluetooth Proximity Sensor
The user-developed Bluetooth Proximity Sensor control monitors the surrounding area for a given Bluetooth-enabled device. This device would likely be the staff member's mobile phone. If the device is found, the control will state that the staff member is in his office. The purpose of this feature is to allow owners to leave their office doors closed without mistakenly given the impression that they are not in.

Display Notice
The display notice is a combination of a pre-defined image stored on the server and custom text input. Like the owner details, the display notice is stored in a table in the database and is retrieved at every start-up. Every 30 seconds, the device checks for an updated display notice and retrieves it if one is found.

3.1.1 Messaging

Figure 4: Messaging screen

Once the user has opted to leave a message and has cleared the authentication phase, he is presented with the messaging panel (figure 4). The primary functionality of this screen is provided by an instance of the DrawingControl user control. This allows users to leave handwritten notes as opposed to tapping keys on an onscreen keyboard. Other features of the panel include the *confidential message* checkbox and the post button.

Messages with confidential content can be entered in a high-security mode which obscures the message from view as it is written. This is accomplished by fading the writing to a light shade of grey two seconds after it has been written. The post button converts the message bitmap into a XML-encoded string which is then transmitted to the web service for storage in the database.

3.1.2 Calendar

Figure 5: Calendar screen

The visitor is presented with a panel that displays a calendar and is able to select a date on which they wish to have a meeting. A listing of the available time slots for that day is illustrated to the visitor where they are able to select the time slot that suits them best.

The visitor is then required to enter a student number / name using a custom keypad control that has been developed for the MODUS project. As with the messaging component the visitor is able to write a message to be attached with the appointment without using a keypad. Serialization of the appointment information is performed and sent to the MODUS server and stored in the backend database.

3.2 Web Administrator

Each office occupant is able to control their MODUS client PDA using the Web Administrator. This is web based and decided upon as it will allow for anywhere/anytime access for each occupant.

Maintaining uniformity of technology compatibility the Web Administrator has been developed in ASP .NET. This provides a dynamic environment to create all functionality required to control the MODUS client.

The MODUS Calendar Web Service is designed to allow:

- Viewing or deletion messages sent by visitors;
- Viewing or deletion appointments scheduled by visitors;
- Creation of open time slots that visitors can use to schedule appointments; and
- Management of staff member information displayed on the client

The administrator is split into three sections:

- Messaging
- Calendar
- Administration.

3.2.1 *Messaging*

The messaging component allows staff members to view messages left on the client application. It is shown in figure 6.

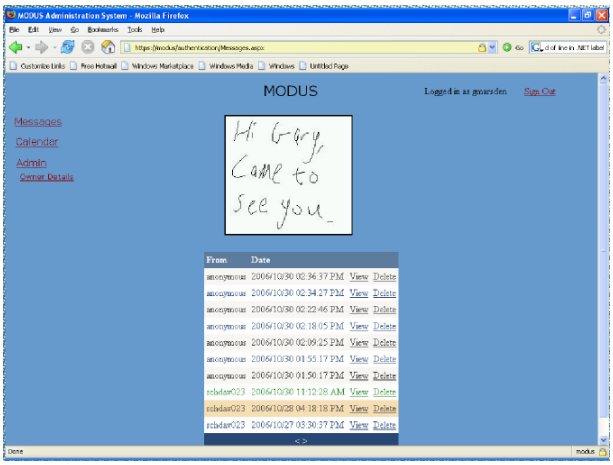

Figure 6: Web Administrator – Messaging Component

3.2.2 *Calendar*

The calendar component allows staff members to view created appointments through the client application and allows for the creation of time slots during which they will be available to see the visitor. These time slots are then published on the client application for visitors to book.

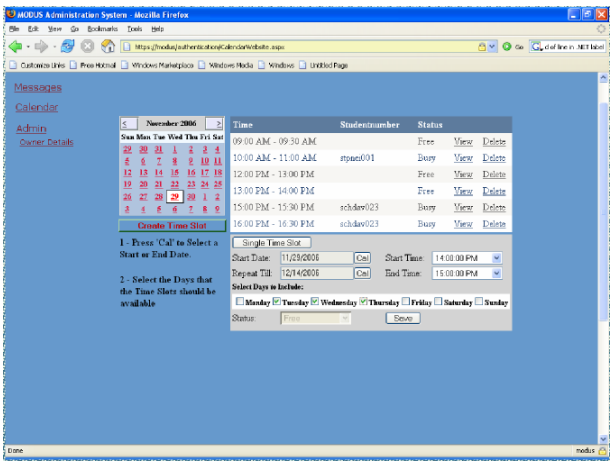

Figure 7: Web Administrator – Calendar Component

3.2.3 *Administration*

The administration component allows staff members to:

- Change owner details displayed on the client application;
- Turn e-mail and Bluetooth functionality on or off; and
- Create a temporary message to be displayed on the client application that indicates the owner's location

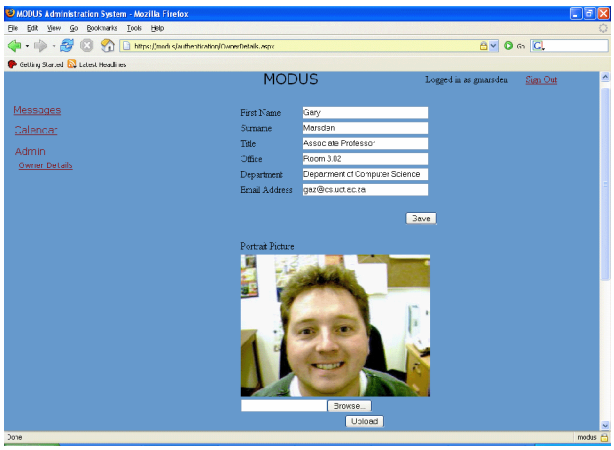

Figure 8: Web Administrator – Owner Details Interface

The web administration website contains RSS feeds of all the latest messages and appointments. The RSS feed has been implemented based on the RSS 2.0 standards and displays the latest 10 messages and appointments that were created using the MODUS client.

4. SECURITY

4.1 Encryption/Integrity

Encryption should ensure that no communication between client and server can be visible by any 3rd party monitoring transmissions. All communication must be undecipherable and provide enough security to ensure that the cost of breaking the encryption is greater than the value gained from the encrypted information.

The proposed security design for the MODUS project included the use of either Web Services Security through the use of Web Services Enhancement (WSE) provided in the .NET framework or the use of SSL. Due to the limitation of the compact framework, all encryption has been provided over SSL.

SSL is implemented to provide secure server authentication, encryption and integrity. SSL provides Encryption and Integrity for the underlying connection used for communication between client and server.

The implementation of SSL is provided by IIS. This is the web server of choice and provides a standard implementation of SSL that uses certificates to negotiate the secure connection.

4.2 Server Authentication

To ensure clients are able to authenticate the MODUS server to eliminate any rogue servers from masquerading as the MODUS server, certificates have been used. This implementation is coupled with SSL that provides an intergraded server authentication. Certificates are authenticated using a global trusted Certificate Authority.

The steps are as follows:

- Create a Certificate Authority Private/Public key pair.

- Create Server Certificate Private/Public key pair signed by the Certificate Authority private key.

The certificates contain a 1024 bit key that is used to negotiate an SSL connection.

Figure 9 illustrates the protocol used to authenticate and setup the initial SSL connection.

- The client requests a secure connection
- The server sends its certificate to the client whom checks to see if the Certificate Authority that signed the certificate is trusted.

If the certificate authority is trusted the client accepts the certificate and agrees to create the secure connection.

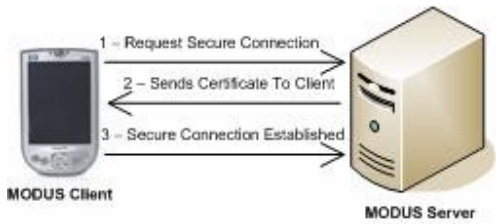

Figure 9: Server Authentication

The result is a 128bit encrypted session. This is the size of the encryption key used to encrypt the transmitted information. This effectively creates a key with 2^128 possible combinations required to guess the key to decrypt the information.

4.3 Device Authentication

To ensure that only authorized clients can connect to the MODUS server, device authentication has been implemented.

Authenticating devices using certificates is not possible due to the limitations of the compact framework and the open environment in a ubiquitous environment. The use of Client certificates is not possible. MAC address filtering has been implemented instead. Performing access control by MAC address requires all devices accessing the web service to be authenticated by MAC address.

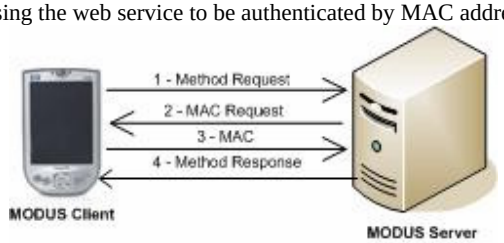

Figure 10: MAC Address Authentication

Figure 10 illustrates the sequence of request/response communications to determine the MAC address of the connecting device.

The response is matched against a list of registered MAC addresses in the database. If no match is found the connection is closed and the client is denied access. Due to the independence of each web service call, each method executes a MAC request and a MAC lookup.

4.4 User Authentication

Users identify themselves to the client application through the login panel which appears as soon as the user presses one of the function buttons. The login panel is shown in figure 11.

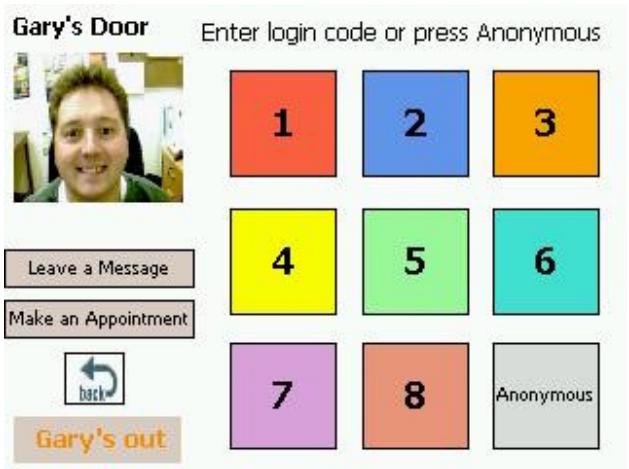

Figure 11: Login panel

The user either enters a four-digit login code or presses the *Anonymous* button. Registered users are identified personally and have access to the high-security confidential mode when leaving messages. If the device detects three consecutive failed logins, the system will lock itself down for 30 seconds to deter brute-force attackers.

5. TESTING

To ensure system security, each security principal has been tested. Various scenarios were tested to ensure all security is applied to the system in all situations.

The breakdown for security testing is as follows:

- Confidentiality/Integrity
- Server Authentication
- Device Authentication
- Resource Availability / Load Testing

5.1 Confidentiality/Integrity

Confidentiality/Integrity was ensured by implementing SSL security. SSL has been used to encrypt the link between the mobile devices and the server. In order to test this encryption, a piece of software called Ethereal was used. Ethereal is a packet sniffer and can be used to capture all network traffic to and from a node. Encryption was visually confirmed by capturing and analysing traffic between the client application and the server.

5.2 Server Authentication

A number of tests have been performed to ensure server authentication is valid and each client does not communicate with any rogue servers. Two scenarios have been put in place to see how each client reacts.

1. *Client does not contain the Certificate Authority certificate as a trusted Authority.*
2. *A Rogue server attempts to use a different certificate signed by the same Certificate Authority*

Results

1. When the server sends the certificate to the client. The client attempts to verify it using the Certificate Authority that signed it. If the CA is not trusted by the client it does not authenticate the certificate. The tests confirmed for clients that did not contain the trusted Certificate Authority would not allow any communication between the services when a method call was made. The client closed the connection to the server and waits for user response.

2. A situation can occur where a rogue server is able to get a certificate signed by the certificate authority that is used to authenticate the MODUS server. If this happens, the MODUS clients are required to recognise that this certificate is not valid. The standard approach of doing this is to allocate each certificate for a server with a common name. This common name is based on the server host address. The client matches all certificates with the server's host address. If this does not match the common name the client is attempting to connect to, the client can detect that a rogue server is masquerading as the MODUS server.

   To confirm this, a separate certificate was created using a different common name: "marvin.cs.uct.ac.za". Marvin represented the test rogue server. The server attempted to send its certificate to the client that is connecting to the MODUS server. The test resulted in the client application closing the connection and recognising it as an unauthenticated server.

5.3 Device Authentication

Client authentication assures the MODUS server that it is communicating with a legitimate MODUS client and allows the server to ascertain which device it is communicating with.

The client authentication scenarios that will be tested include:

1. *Client PDA attempts to access the web service but is not registered.*
2. *Client PDA attempts to access the web service as a registered device*
3. *Client is registered but attempts to access a different owner's information.*
4. *Two MODUS Clients attempt to retrieve information.*

Results

1. The connection between the client PDA and MODUS Server is terminated by the server when each method is called by a non-authenticated user.

2. The Server executes all methods correctly and returns all information.

3. To manipulate each client to send different owner details, it has been hard coded to set a new calendar item instance to match a different owner. The server places the updated appointment under the device lookup name and not the name associated with the calendar item instance from the client.

4. Using multiple devices, each client retrieves the correct information, not accessing any other owner's information.

5.4 Load Testing

Load testing was carried out to observe and extrapolate server performance under increasing strain. Load was created by adding more devices to the network and continuously performing operations on the client applications. Server performance was monitored using perfmon – the Windows XP performance monitor.

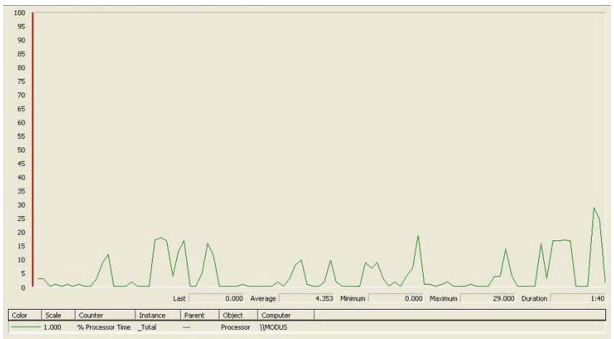

Figure 12: CPU load from single device posting messages

Load profiles such as the one shown in figure 12 were created to show server CPU load under varying conditions. Testing showed that using the current server, the system scales fairly poorly. It can be extrapolated that in the worst case, five devices on the network will cause the server CPU to operate at 100% capacity. This situation could be improved greatly by running the web server and database on dedicated server architecture. The current AMD Athlon 1.1Ghz CPU is clearly too underpowered to adequately serve a network of this type.

5.5 Usability Testing

Usability testing was performed in order to test the client interface design and the flow of the application in performing functions. Both members of the project team designed scenarios relevant to their individual sections with each scenario designed to test a specific execution path of the application. The first round of testing took place in the third floor corridor of the Department of Computer Science. Passing students and staff members were asked to complete a given task for the messaging component followed by a task for the calendar component.

Feedback from respondents did not indicate any critical flaws in the design of the client application interface. During the course of testing, positive comments were made suggesting that students believed the system to be a good idea. One lecturer suggested that users would not take to writing on a touch screen; however, this is an assertion that would have to be tested with a full-scale field test. This was unfortunately not possible due to time and logistical restrictions. Ultimately, feedback mainly called for increased guidance within the interface. This was implemented by increasing the number of instructional labels as well as clarifying the wording of existing labels.
Following the first round of usability testing, several changes were made to the client interface in order to implement

suggestions from respondents and eliminate points of hesitation observed during testing. The new iteration was then tested on a second group of respondents.

The presence of instructions clearly improved the usability of the interface. All respondents reacted positively to the interface and several stated that it was easy to use. Minor changes were made to the interface in response to comments received during this round of testing. Users were confused over the usage of the confidential message functionality. It is hoped that these changes will serve to both clarify the function of and draw user attention to the *Confidential Message* checkbox. However, time constraints prevented a third round of testing.

6. CONCLUSION

The main goal of the project is to create a multi-functional messaging system that will provide communication between staff members and students in a secure, ubiquitous environment.

The development of the MODUS project has revealed many issues surrounding the development of a ubiquitous network for communications purposes. Attempts to secure the network against intrusion have resulted in most of the security issues inherent in the distributed wireless networks being resolved using conventional security principals.

The limitations of the .NET Compact Framework caused problems with the development of the system functionality and security features. However, the project team was able to overcome these issues using alternative approaches.

From the results gained in security and usability testing it has been found that implementing a multi-functional messaging system in a ubiquitous environment that is both secure and usable is possible.

7. FUTURE WORK

A significant amount of work can be undertaken in order to improve the state of the MODUS project. Some of these extensions are the following:

7.1 Field Test

First and foremost, the system requires a full field test in order to properly evaluate the practical value of the system. By deploying at least one device to an office door, the researcher would be able to establish the uptake rate among students and staff and perhaps formulate new methods for encouraging system use within the department.

7.2 Optimisations

Certain optimizations can be made at various points in the code to reduce the server load and improve client response times. As an example, portrait images can only be displayed at a certain size on the client. Therefore, uploading a larger image is of no value. It would be beneficial to improve the image upload code to dynamically resize the image before it is stored in the database and transmitted to the device.

7.3 Message Compression

Messages left through the client application are saved and transmitted as uncompressed Bitmaps as the .NET Compact Framework does not support any other image format. These messages are prime candidates for compression as repeating strings comprised of common tokens are prominent. Some basic compression was initially implemented and tested during development. Results were extremely favourable with message sizes being reduced by up to four times with basic compression routines. However, compression was deemed to be outside the scope of this project and was abandoned for fear that it would lead to complications requiring already-limited time and effort to resolve.

7.4 Operating System

The current PDA being used is the HP IPAQ 4150. This PDA is an old model and operates on Pocket PC 2003. The new version of Windows Mobile 5.0 allows for better functionality and more sophisticated functions. Windows Mobile 5.0 allows users to change Bluetooth stacks from a Broadcom Stack to Windows Stack and allows for seamless integration using C# in the .NET environment.

7.5 Scalability

From testing it can be seen that there are scalability problems. To establish the reasons for this, it would be better to migrate from the current server to a more powerful system with more resources available to run the actual software that is providing the client with the service such as SQL Server and IIS.

7.6 Calendar Design

During implementation, it was suggested by the project supervisor that it could possibly be useful having each staff member display a full timetable and allow each student to create an appointment on any free slots not already booked in the time table.
Future work can be based on implementing a different way to allow users to sign up for an appointment.

7.7 Security

Other security techniques can be implemented, possibly with new versions of the compact framework WSE will be available and can provide a better solution.

Using IP addresses to allow for authentication can be useful. IP v6 eliminates the possibility of spoofing an address. This implementation can be a more secure solution if implemented.

7.8 Communication

From load testing it was shown that the MODUS server was not scalable with the MODUS clients. From previous research it was noted that using SOAP messages increases overhead. SOAP messages was initially used for security reasons and later realised was not possible. It was felt by the development team that it was too late to change the technology being used as the development process was not allocated enough time to change technologies.

Changing the communication to using a REST architecture or possibly a direct socket connection between client and server may provide a more scalable system in the future.

8. REFERENCES

[1] Cheverst, K. and Fitton, D. "Experiences Managing and Maintaining a Collection of Interactive Office Door Displays." *Proceedings of 1st European Symposium on Ambient Intelligence (EUSAI'03)* , Eindhoven, Netherlands.

[2] Kagal, L. , Undercoffer, J., Perich, F., Joshi, A., and Finin, T. "A Security Architecture Based on Trust Management for Pervasive Computing Systems." *Proceedings of Grace Hopper Celebration of Women in Computing 2002.*

[3] Haque, M. and Ahamed, I. Security in Pervasive Computing: Current Status and Open Issues. *International Journal of Network Security, Vol 3, No 3, November 2006, pp 203-214.*

[4] Zahur, Y and Yang, T. "Wireless LAN Security and Laboratory Designs." *Journal of Computing Sciences in Colleges, Volume 19 Issue 3. Consortium for Computing Sciences in Colleges.*

[5] Bhagyavati, DeJoie A., and Summers, W. "Wireless Security Techniques: An Overview." *Proceedings of the 1st annual conference on Information security curriculum development. ACM Press.*

[6] Zakiuddin, S. Creese, B. Roscoe, and M. Goldsmith, "Authentication in Pervasive Computing, Position Paper." *presented at PAMPAS '02 - Workshop on Requirements for Mobile Privacy & Security Royal Holloway, University of London, 2002.*

[7] Weiser, M. "Some Computer Science Issues in Ubiquitous Computing." *Communications of the ACM, Volume 36, Issue 7 (July 1993). Special issue on computer augmented environments: back to the real world. pp 75 – 84. 1993.*

[8] Nyditot. Virtual Display software. http://www.nyditot.com/NVDPage.asp [Accessed 05 November 2006]

[9] Shneiderman, Ben. "Designing the User Interface". 1997. Addison-Wesley Publishing Company.

[10] Gong, J., Tarasewich P. Guidelines for Handheld Device Interface Design. Proceedings of the DSI 2004 Annual Meeting.