# A WEB APPLICATION PROTOTYPE TO PROVIDE DRIVERS WITH AN INSTANT ACCESS TO VEHICLE KEEPERS ADDRESS AND INSURANCE DETAILS

by

Victor O Barnieh

MSc Distributed Computing Systems

THE UNIVERSITY OF GREENWICH

March 2004

Approved by Dr. Chaoying Ma
Supervisor

**ABSTRACT**

DRIVER VEHICLE INSURANCE CHECK

This dissertation is the development and implementation of a prototype web based application which is designed to help drivers involved in accidents to confirm the name, address and insurance details of the other party at the road side using their WAP enabled phone or on a personal computer. At present the Highway Code, rule 257 states that "If you are involved in an accident which causes damage or injury to any other person, vehicle, animal or property, you must stop give your own and the vehicle owner's name and address, and the registration number of the vehicle, to anyone having reasonable grounds for requiring them. If you do not give your name and address at the time of the accident, report the accident to the police as soon as reasonably practicable, and in any case within 24 hours."

*Law RTA 1988 sect 170*

The law also requires all motorists to be insured, unfortunately, however, a staggering 1.6 million are without motor insurance. Chances of being involved in an accident with an uninsured driver are one in 20. This is the motivation behind this project.

Drivers or motorist are required to register and login to be able use the service. Once logged in, they can verify the details of any vehicle i.e. the keepers name, address and insurance status.

This service is not limited to motorist only, law enforcements agencies such as the police can also use it to check the details of a vehicle, but this time with more enhanced results such as whether the vehicle has passed its annual road worthiness test (MOT) and also if the driver is competent to be in charge of the vehicle (if they have a full driving licence) or not. The system can also help the police find out if the driver is certified to be in charge of a particular type of vehicle and also if they are banned from driving or not.

All these services can be accessed by any wireless application protocol (WAP) enabled device or computers equipped with a web browser.

Table of contents

# ACKNOWLEDGMENTS

I wish to thank Dr. Chaoying Ma my project supervisor, who guided me throughout this project and Dr. Kevin McManus who introduced me to the open source programs. Special thanks to Anita who kept the kids busy whilst I seriously buried in the project work.

*C h a p t e r  1*

# INTRODUCTION

## 1.1 Aim

The aim of this dissertation is to design and implement a prototype web application whose concept hasn't been implemented as yet, although the Government has been deliberating over it for some time now. With the knowledge acquired from Software Tools and Techniques, Software Engineering and Web Application Technology, I am with the hope of delivering a well structured and robust application adhering to a good software design that will be easy to upgrade, administer and user friendly.

Getting involved in a road traffic accident is a terrible experience, but when your insurer informs you that the documents presented to you by the other party does not exist or invalid, you could be waiting for a very long time to reach a settlement. I thought about this and said surely there must be a way to verify or confirm this sort of information, in the very least, the Driver Vehicle Licensing Agency (DVLA) holds immense data about all drivers and vehicles i.e. the addresses of all vehicles registered in the UK and their keepers. Surely in these modern times people can not be bothered with writing letters to DVLA requesting for the addresses of these culprits, although this service rendered is not free, it is available, if you are prepared to pay the requested £2.00 in the form of a cheque or postal order. Once again the cheque book is fast becoming obsolete, as the electronic version in the form of debit cards are taking precedence. With all these in mind, this project topic was developed and with the approval of my supervisor, the design and implementation began.

## 1.2 Objectives

The objective of this project is to design and implement a web application that will help motorist verify and confirm the address and insurance status of the third party in an accident. It also gives the law enforcement officers such as the police, the ability to check at the roadside, the address of the vehicle, the road worthiness (MOT) and insurance. It also empowers the police to gain access to the drivers' details whether they have the appropriate licence to be in charge of a particular vehicle or better still if they are licensed to drive.

At present motorist are insured under their names and the insurance companies rely on the honesty of their clients such as if they have driving licence and how long they have held it for etc, this application provides them with an interface whereby drivers are insured using their driver numbers. This thus eliminate the need of fraudulent details been submitted to the insurers a classical example is a banned driver trying to purchase a motor insurance policy.

This project is also aimed at MOT centres so that they can update their daily register by completing a form that gets uploaded to the server's database.

*Chapter 2*

# LITERATURE SURVEY

Although I did not get much success in acquiring the sample database needed for this project from the Driver and Vehicle and Licensing Agency (DVLA) and the motor insurance companies, I did however managed to gather the information required from other sources.

## 2.1 Driving Licence

I obtained a D1 form which is the application form for a driving licence from the post office. This form gathers all the necessary information or data that the DVLA holds on drivers. This information formed the basis for the driver and licence table in the database.

## 2.2 VEHICLE KEEPER

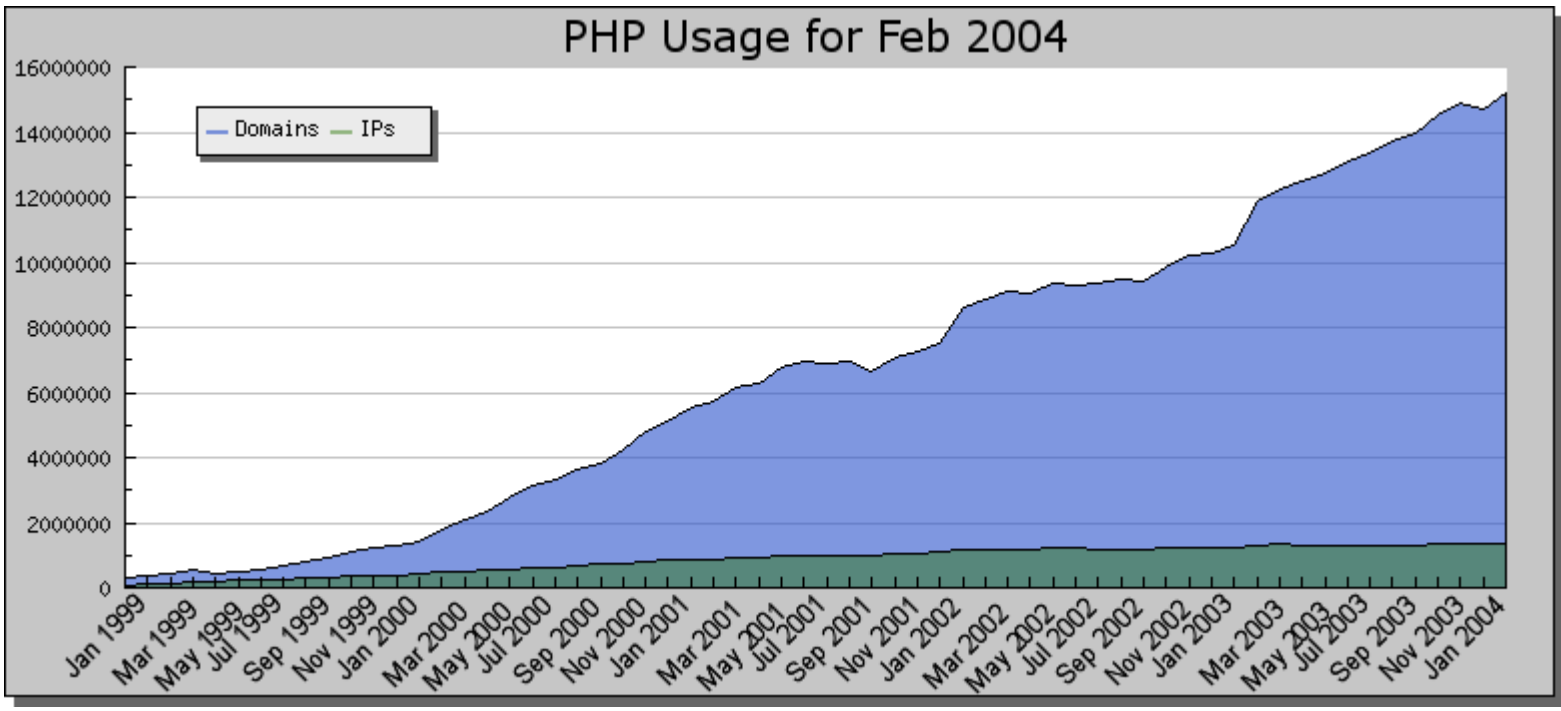
The Vehicle Registration Document Application or the V62 form outlines the details required to register a vehicle. The fields in this form were used to mimic the vehicle table of the database.

## 2.3 INSURANCE

Collected details form various motor insurance web sites i.e. application forms and also details of the proposal by the Association of British Motor Insurers in combating the increase in uninsured drivers.

**2.4 PHP**

PHP originally stood for Personal Home Page and was created by an independent IT contractor in 1994 called Rasmus Lerdorf. This was simply a set of Perl scripts he wrote to track visitors to his web site and to log information about them. As its usefulness and capabilities grew, he soon began to receive enquiries about these scripts from others so he rewrote them as a scripting engine that included a form interpreter. The revised package was released in 1996 as PHP-FI to reflect its new features. Developers around the world began contributing ideas and by 1997 over 50,000 web sites were using PHP for a variety of dynamic functions. Two of these developers, Zeev Suraski and Andi Gutmans, were primarily responsible for creating an Application Programming Interface (API) that became the PHP parser which was released in June 1998 as PHP3. From PHP4 to the current updated version 5, the Zend engine was incorporated so that its scripts can be used with virtually any web server and operating system in other words PHP is portable. At present PHP can be found on 15,205,474 Domains with 1,330,021 IP addresses. Source the official PHP web site http://www.php.net.




This chart from Netcraft (http://www.php.net/usage.php) shows the immense growth over last the5 years.

The stated goals of PHP is to equip web developers the ability to create dynamically generated pages quickly and effortless.

**2.5 MySQL**

MySQL is the worlds most poplar open source SQL database, and is developed and provided free of charge by MySQL AB. In 1979, Michael Widenius (also known as Monty) developed an in house database tool called UNIREG for managing databases. UNIREG is a tty interface builder that uses low-level connection to an ISAM storage with indexing. Ever since, UNIREG has been rewritten in several different languages and extended to handle big databases. The Swedish company TcX began developing a web based application using UNIREG to support its existence. Unfortunately UNIREG had too much over head to be successful in dynamic web pages so alternatives had to be sought. SQL and mSQL were considered, but the latter was a cheap DBMS that gave away its source code with database licences. At the time mSQL had very limited features, most important to TcX it did not support indexing and as such its performance was poorer than UNIREG. Monty the only employee of TcX contacted David Hughes, the author of mSQL, with a proposition if he was interested in connecting mSQL to UNIREG's B+ISAM handler to provide indexing to mSQL. Hughes was well on way in releasing the next version mSQL 2 that catered for indexing. Monty decided to go ahead and create a database server that suited his needs. There was one thing for sure, he wasn't going reinvent the wheel and he built on UNIREG and made use of mSQL 2's third party utilities by writing an API into his system which was initially very similar to Hughes', but was completely original.

By 1995, Monty had built a database system that supported his needs and my MySQL 3.11 was developed. David Axmark at Detron HB, a business partner persuaded Monty to release this server on the Internet and follow and follow a business model pioneered by Aladdin's L. Peter Deutsch. This business model enabled the developers at TcX to work on projects of their own and release the results as free software. Money is generated through support they provide, a very flexible copyright that makes MySQL free.

**2.6 WAP**

Wireless Application Protocol (WAP) is a result of continuous work to define an industry wide standard for developing applications over wireless communication networks. The WAP Forum was originally founded by Ericsson, Motorola, Nokia, and Unwired Planet. WAP, which displays Web information on small wireless devices, debuted to great elaboration in the late 1990s. At the time, most handsets had black-and-white displays and relied on dial-up connections for data transfer, making Internet-based services slow, expensive and difficult to use. Not surprising it was declared dead since its conception and on arrival by critics, but the technology is now experiencing resurgence, largely driven by WAP sites offering customized phone ring tones and logos. Because it works in already existing networks, WAP needs little modification by way of Web content and is easily available. Already, there are numerous companies providing e-commerce services through WAP around the world, and with the huge mobile telephone subscriber base, the potential for m-commerce is tremendous.

In the past half decade, the Internet has revolutionized the practice and procedure of trade, arising in the new world of e-commerce. Now people can buy or sell goods and services practically 24 hours a day, 7 days a week, if they have access to the Web. Vendors have been able to tap into markets that were impossible to reach due to remote geographic location and this technology has fuelled the new economy. The mobile telephone is the obvious choice for m-commerce. It is however, estimated that there will be 1 billion mobile telephones worldwide by the end of this year, according to IDC, with more than half them been internet-enabled. The most popular Internet-enabling technology being adopted as a whole by handset manufacturers and service providers is obviously the WAP. Nokia, Ericsson, Motorola, and the Japanese mobile operator NTT DoCoMo as well as the giants in the banking, retail and travel industries are developing their mobile e-sites, including Amazon and Schwab. And these sites are all using WAP, which works with all major wireless networks. The wireless networks can be built into any operating system, including Windows CE, Palm OS, Epoc, or JavaOS to run on many personal digital assistants (PDA).

The number of WAP pages viewed in the United Kingdom for November hit 947 million, up from October's 897 million, and significantly ahead of industry expectations. Source the *Mobile Data Association* (MDA). It had been projected that usage in the United Kingdom would hit 8 billion WAP page impressions for last year (2003), but with November's figures, the total already stands at 8.2 billion. Source the *U.K. GSM (Global System for Mobile Communications) operators O2, Orange, T-Mobile and Vodafone Group*. The wide availability now of color screens, better user interfaces and faster data connections is encouraging mobile users to log on, even if only to download ring tones, according to the MDA. Polyphonic ring tones based on pop songs are the main driver of WAP traffic. The rise in traffic is good news for mobile operators who earn revenue from download fees, and it also means opportunities for content providers and developers. WAP however still remains a far cry from the popularity of the most successful mobile Internet platform, NTT DoCoMo, which reached 40 million subscribers in October. It is also far less popular than the main carrier of data traffic on U.K. mobile network, Short Messaging Services (SMS) or text messaging. Person-to-person text messages reached an average of 59 million per day in November.

Web applications are traditionally designed based on the assumption that visitors will have a desktop computer with a large screen and a mouse. Although handheld devices are more limited than its desktop counterpart in several important ways, smaller screens, unable to display decent pictures, only a few lines of text, and often monochrome instead of colour, very limited input capabilities and entering data takes extra time, less processing power and memory to work with, their wireless network connections have less bandwidth, and they are much slower than those of computers hard-wired to fast LANs.

These limitations will hinder WAP as the choice for future technology as some analysts say. Let us not forget, however, that DoCoMo, the Japanese mobile operator, signed up 10 million subscribers to its I-mode service in less than one year.

*CHAPTER 3*

# SECURITY

Privacy and the protection of personal data on the Internet is a huge and complex problem to discuss and deal with. "Growing concerns over the abuse of personal information via the World Wide Web can be addressed at political, social, and technical levels." [Ryan Lee].

In seeking to restore control over private information to the individual, one must look to the field of online access control. Access to computers has always been a part of computer history since users were billed for the computing resources they used and had to be identified through a username and password for record keeping. The method of usernames and passwords was kept as computers connected to each other in networks for remotely accessing a machine and for broader Internet applications such as FTP. In its earliest days, the Internet dealt primarily with communication and sharing data. But as computer networking evolved, the number of users exploded, particularly due to the popularity of the World Wide Web, people involved with Web development began to recognize the need to provide finer grained access control over personally sensitive data. Early implementations of the Common Gateway Interface (CGI) and HTTP Authentication provided methods for differentiating between users, again with usernames and passwords. While these allowed individual servers to implement their own data protection schemes by somehow restricting access to certain users or groups of users, there was no way to universally express their criteria such that different software implementations of servers could enforce data protection. Even as the Web has evolved, no standard has emerged in declaring who has access to what.

### 3.1 DATA PROTECTION

It will be incomplete to discuss privacy and data protection issues without talking about the Act. Although it is about 20 years old it is widely misunderstood and misinterpreted by many, The Data Protection Act 1984 was introduced because of concern of concern about individuals' right and personal privacy in the light of the then rapidly developing computer technology. There was also a council of Europe Convention on Data Protection which required data to move freely between European countries, as Britain wanted to comply with the European conventions, so it introduced a similar act. Listed here are the principles:

1. Personal data shall be processed fairly and lawfully and, in particular, shall not be processed unless-
    a. at least one of the conditions in Schedule 2 is met, and
    b. in the case of sensitive personal data, at least one of the conditions in Schedule 3 is also met.
2. Personal data shall be obtained only for one or more specified and lawful purposes, and shall not be further processed in any manner incompatible with that purpose or those purposes.
3. Personal data shall be adequate, relevant and not excessive in relation to the purpose or purposes for which they are processed.
4. Personal data shall be accurate and, where necessary, kept up to date.
5. Personal data processed for any purpose or purposes shall not be kept for longer than is necessary for that purpose or those purposes.
6. Personal data shall be processed in accordance with the rights of data subjects under this Act.
7. Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data.
8. Personal data shall not be transferred to a country or territory outside the European Economic Area unless that country or territory ensures an adequate level of protection for the rights and freedoms of data subjects in relation to the processing of personal data. *[Source http://www.hmso.gov.uk/acts/acts1998/80029--l.htm#sch1ptI ]*

Although all the 8 principles concerns this dissertation, the third and eighth are of particular interest. With reference to the third, as the web application is designed and targeted at drivers, it is a requirement that all users of the site should posses a valid UK driving licence so that any misuse of information or data acquired from the site can be traced back to the user. The user name which users are required to use to register and login was designed in a way that it will only accept the full driver number which is unique to each driver and issued as part of the driving licence by the Department of Vehicle and Licence Agency, which is kept by the agency. The unique warrant numbers of the law enforcement officers, which again are issued to the police by their respective forces are used as the user names to enforce the principle. An audit trail is kept and maintained by the application of all transactions that users conduct at the site. The eighth principle was implemented in a way so that users of the application cannot have access to any form personal information or data without requiring the user to login or register. The hyperlinks that leads to pages where users can enter a search criteria to access data, checks to confirm that indeed the user has either previously registered or logged in, if a violation is detected, the users is redirected to the login page where they can login and begin a session that can be traced.

*CHAPTER 4*

# TOOL AND TECHNIQUES

## 4.1 Web Application

A Web application is a dynamic extension of a Web server and there are two types.

Presentation oriented. A presentation-oriented Web application generates dynamic Web pages containing various types of mark-up language example HTML and WML in response to requests.

Service oriented. A service-oriented Web application implements the endpoint of a Web service. Service-oriented Web applications are often invoked by presentation-oriented applications.

### 4.1.1 WHY WAP

The most recent trends in wireless technology are finally freeing us from the constraints that wires and desktop computing have placed on our access to information and services. Now we can execute transactions and communicate anytime and anywhere, through mobile phones and Personal Digital Assistant (PDA). As emerging technologies are making mobile computing easy and accessible, many aspects of the wireless Internet are constantly changing. Several different protocols are attempting to standardize the means through which server-based software applications communicate with mobile phones and PDA's to carry out administrative and service-related tasks.

The two main protocols competing for this large market are WAP and I-MODE.

WAP, as already discussed was created by a consortium of Phone.com, Ericsson, Motorola and Nokia. It is therefore no surprise that Phone.com's WAP-based micro-browser is some kind of a standard on most WAP-enabled devices.

The most basic difference is obviously the different graphic capabilities. WAP’s support for graphics is very limited compared to I-MODE, although I-MODE only supports simple graphics. I-MODE's packed switched data network is more suited for transferring data than WAP's circuit switched network. These new services are mainly for small devices with very small screens. However it is assumed that extensive graphic capabilities is not necessary, since most of what people might want to do over their wireless phone would be checking their email, some stock-quotes, weather, or traffic reports and the like.

Another major difference is that I-MODE like broadband, is always connected to the internet and does not require dial-up. Since users are not charged for the time they spend online, (after all, they always are), it's also cheaper than the dial-up version, WAP, this reason again increases the advantage I-MODE has over WAP.

On the other hand, the difference in Mark-up Languages utilised by the two competing technologies thus I-MODE, uses cHTML, a subset of HTML, while WAP uses WML, a subset of XML. cHTML, while certainly easier to develop in from a web designers point, it has its limits. The downside of WML, on the other hand, is similarly obvious, currently a WAP-Gateway is required to translate between HTML and WML for almost every data transfer. Since WML is derived from XML, it is much more extensible and it is assumed that XML will in some respect replace HTML in the future, since it allows for more dynamic content and various different applications. If this trend proves to be true, it might be a hint that a WML based service will, in the future be of more advantage than an HTML based one. So while WAP may currently require more complicated technology, limited functions and more expensive to use, it might in the future, enable the user to achieve more with this technology. Also WAP is far more popular in the UK than I-MODE.

### 4.2 Choice of languages

#### 4.2.1 Client-Side

Basic HTML pages, which were once the norm of creating web pages, are now difficult to find, the basic skills are now obsolete and a new direction is needed. It seems the only place where plain HTML is still in use, is in a basic brochure-style web site for a furniture Shop.

As scripting languages go, there's really very little to choose from between Netscape's JavaScript and Microsoft's VBScript and Microsoft has called their implementation, Jscript. As such, the 2 common client side scripting languages are JavaScript and VBScript or Jscript that are used in the development of modern web pages, to make web pages more interactive and the most reason for their deployment in my opinion is for validation. The scripts on the pages are processed by the individual client-side web browser that requests these web pages. This results in reduced traffic to and fro the web server, freeing up bandwidth and server resources.

It should be noted, however, that, Client-Side scripts can only run on browsers that support the specific scripting language that was used, meaning that you would need Netscape Navigator to achieve the full potential or functionality of JavaScript and Microsoft Internet Explorer for VB Scripts.

I feel that it is important to mention that my comparisons between JavaScript and VB Script are meant to be relevant only to Client-Side Scripting purposes. Both languages can be used for Server-Side scripting as well, however that is not the topic of this discussion and is a hardhearted debate as the old and prolonged battle between Netscape Navigator and Internet Explorer battle to dominate in this niche.

Some programmers and developers propose that VB Script is a better choice for the development of Client-Side Internet Applications, and say how quick it is to get an application up and running with fully functioning code, taking into

considerations factors such as support, learning curve, etc. I am well aware that VB Script is so easily attainable in that, the syntax is based on that of Visual Basic and in fact, “VB Script is identical, syntactically and grammatically, to Visual Basic and Visual Basic for Applications” (Thurott). There are literally millions of Visual Basic Developers who can instantly become web developers without much of a learning curve.

JavaScript, on the other hand, is not truly based on any other language. Despite the use of the phrase “Java” in the name, it has absolutely nothing to do with it at all. In fact, the original was LiveScript, but the name was changed at the last minute by the developers at Netscape, “so that Netscape could feed off the success of Java, like some fish hanging off the belly of a shark” (Thurott).
VB Script was created specifically for use on the internet, and was designed to be as close as possible to Visual Basic, that is probably the most popular ever created (Hatfield, P. 21) and it often has a strong resemblance to another language known as English. Another advantage of VBScript is its ease of integration with client-side controls, and a vast range of which are available. Quite literally, within minutes, you can create a form containing all the text boxes, drop-down lists, and options that you could wish for, link them all together with VBScript, and an application that looks, feels, and operates like a Windows application is created. After all VB Script is by Microsoft.

Although Netscape's JavaScript suffered in its original incarnation from a whole host of memory and other problems just as any other new software or language, most of these have now been addressed. It now has support for more browsers by a huge margin. In other words, this means that scripts embedded in an HTML page using JavaScript; will have the vast majority of Web users or audience been able to take advantage of the full functionality it provides.
I have no doubt at all about the stability, reliability, popularity and most of all the superiority of JavaScript over the Microsoft’s VBScripts. If Microsoft appreciates the power and strength of the open source JavaScript to adopt it on their main index page of their website, which evidence is produced below, then those die hard Microsoft fans out there should rethink, however, if one disputes this fact, well the code snippet below says it all, it was obtained from www.microsoft.com (click view source from browser).

```
  <script language="JavaScript">
<!--
            var userAgent = navigator.userAgent;
            var MSIEIndex = userAgent.indexOf("MSIE");
            if (userAgent.indexOf("Win")  != -1 &&
                        userAgent.indexOf("MSIE") != -1 &&
                        userAgent.substring((MSIEIndex + 5),(MSIEIndex + 8)) >= 5.5)
                        window.location.replace("/homepage/ms.htm");
//-->
</script>
```

Source www.microsoft.com

As the authors of VBScript use JavaScript for the Client-Side scripting, there is no need to discuss any further the advantages of JavaScript over VBScript. This is the reason that JavaScript is the first choice in Client-Side scripting language and that is why I chose it. As a reminder most Microsoft web pages use JavaScript.

Taken with the integration, support and clout afforded by their wide range of products, it is very compelling for some business managers to specify Microsoft products, rather than risk the hyped incompatibilities that tend to creep in down the line going with JavaScript.

**WML**

WML is a mark-up language that is based on XML (eXtensible Mark-up Language). The official WML specification is developed and maintained by the WAP Forum, an industry-wide consortium founded by Nokia, Phone.com, Motorola, and Ericsson. This specification defines the syntax, variables, and elements used in a valid WML file. A valid WML document must correspond to this DTD or it cannot be processed. As at present, WML is the only language for WAP.

**Server Side**

One of the most popular methods used to create today's modern dynamic web pages is Server-Side Scripting languages. These dynamic pages are constructed in such a way that all server processes take place before the page is delivered to the user. This means that you only need the most basic internet browsing software to view the most complex and dynamic pages on the web today

Server-Side Scripting has made it possible to "create platform-independent, easily deployable applications" (Rahmel). The thought of creating a program or application that will run anywhere in the world has obvious advantages over Client-Side Scripting. Scripting Engines seek to resolve the problem of having no functioning application code on the client. Because the script code is embedded in the HTML, it is downloaded every time the page is accessed (Rahmel). There several Server-Side Scripting languages available to choose from, and as this project goes, I have looked at Macromedias ColdFusion, Microsoft's ASP.NET, Java Server Pages and PHP. From my little experience I have noticed that although Java based application are very stable and reliable, they are often very slow and as this web application will also be serving the already slow WAP clients I had to rule JSP out. I seriously considered ASP.net and PHP.

ASP .NET which is the predecessor of ASP is Microsoft's flagship Sever-Side Scripting language. ASP .NET not just only runs in the Windows environment, it requires that all relevant service packs, security updates and Internet Information Services (IIS) are installed. Although it might run on Personal Web Server (PWS), IIS should be considered if a serious web application is to be developed. Having all this in place, the 130MB copy can be downloaded from Microsoft's web site.

On the hand the 3MB PHP can be downloaded free of charge from php.net. Unlike ASP.NET, PHP will run on virtually any platform including Windows. Although PHP is not restricted to IIS like ASP.NET, it can also run on the Apache Server. Some may argue that PHP is not or fully Object Oriented as ASP.NET, but that topic is beyond the scope of this discussion. I found PHP to be very simple to learn as I only recently begun to read about it for the purpose of this dissertation. I was however impressed to read that, PHP is specially optimized and tuned to MySQL (the database of my choice) for seamless web integration and in fact it has a host of specific functions geared for MySQL . These reasons among others justified my choice as PHP, and also not forgetting the myth about open source versus proprietary.

**Database Choice**

As many dynamic web applications use database as main the form of back-end data stores, so does this prototype. With many databases software available, it becomes a daunting task choosing the one to use, however, the two most popular that web developers use are MySQL and SQL Server. Like most modern relational databases, they have many features in common like judging from their names they use SQL to retrieve data because both claim support for ANSI-SQL and both support primary keys and key indices which can be used to speed up queries and for constraining input. They both provide some form of XML support.

There were a few factors I had to consider in choosing between the two, it initially appeared to be a bit confusing but as I read on it became clearer that the choice had to be platform independent or should be able to run on a UNIX server which is more stable and secure than the Windows platform, as the web application will be hosted on Unix. "*SQL Server 2000 only works on Windows-based platforms, including Windows 9x,Windows NT, Windows 2000 and Windows CE. In comparison with SQL Server 2000, MySQL version 4.1 supports all known platforms, including Windows-based platforms, AIX-based systems, HP-UX systems, Linux Intel, Sun Solaris and so on"[ Alexander Chigrik 2003]*.

*"In terms of performance, MySQL is the leader, mostly due to its default table format, MyISAM. MyISAM databases are very compact on disk and place little demand on CPU cycles and memory." [Sanders Kaufman, Jr. 2003]*. MySQL runs on Windows without problems but also performs better on UNIX. Much of the very busy Yahoo! Finance portal which bears very closely to my application uses MySQL as a back-end database purely because of performance. Although MySQL version 3.23 which is installed on our UNIX server doesn't fully support foreign keys, making it less of a desire to SQL Server, the version 4 onwards does. If the application is adequately coded this short fall is eliminated. Aside this setback, I felt the advantages far out weighed the disadvantages.

It is also very important to note that, I did not choose MySQL over Microsoft SQL Server solely based on the long outstanding feud over my core principles thus open-source versus proprietary, but I did. The learning curve for me as a beginner in database was very easy and in terms of free material, tutorials and support, the web proved very supportive and helpful. With a very minimal systems specification, MySQL can be installed and up and running in very little time. With no financial burden and licence constraints, it could be downloaded free of charge to serve and support a small to a very large corporate database. The obvious choice for me was MySQL.

A good thing about visual tools such as Macromedia's DreamWeaver, Adobe's GoLive and Microsoft's FrontPage is its ability to generate code. These program-generated codes often eliminate the need of debugging, cross browser compatible and also faster than hand coding. Having said this, these programs are very expensive and the codes they generate are quite difficult to understand. However, using Edit Plus (a basic text based program) gave me the ability to write simple codes that are easy to understand and fully functional.

*CHAPTER 5*

# EXISTING AND APPLICATION

## 5.1 Effect on the driver

At present the Driver Vehicle Licence Agency (DVLA) holds an immense amount of data on all motorists that they have issued driving licence to, in effect all drivers. They also hold data about all vehicles that is the vehicle specification, colour, modification etc and most importantly, the keepers name and address.

At present the Highway Code, rule 257 states that "If you are involved in an accident which causes damage or injury to any other person, vehicle, animal or property, you must stop give your own and the vehicle owner's name and address, and the registration number of the vehicle, to anyone having reasonable grounds for requiring them. If you do not give your name and address at the time of the accident, report the accident to the police as soon as reasonably practicable, and in any case within 24 hours

A letter has to be written to the DVLA requesting for a motorist address given the registration number of the vehicle concerned, and the requested details i.e. the name and address will be sent, and there is a fee for this service. The response according to the DVLA is sent within 5 working days, but in reality it takes weeks.

## 5.2 Effect on the police

The traffic police spend much of their time stopping motorist during their random stop check, when stopped your current driving licence is checked to ensure that you have the right licence for the vehicle type that you are in possession of and indeed if you're really allowed on the road. The current MOT and motor insurance certificates are also checked. At present it is not a requirement to carry these documents however failure to produce them at the roadside will result in a producer form HORT1 (Home Office Traffic Reporting), been issued by the police. The producer should then be presented at any designated police station of choice with the stated documents (which was failed to be produced at the roadside) within seven days of its issue. The officer then sends another form HORT2 to the designated police station. When the driver turns up at police station, his documents are checked and a third form HORT3 is filled out by the receiving officer to corroborate that, the said documents have been

submitted and sent back to the first constable. This system is not only cumbersome, resource intensive and subject to abuse it is also estimated it has an operational cost of £15m a year and it's of course paper based.

### 5.3 Effect on Insurance Companies

To purchase an insurance policy, the insurance company takes your details among many other things ask for your name, date of birth, address and whether or not you have a full driving licence and indeed how long you have held it for, they also ask about your driving history and will want know if you have any penalty points or convictions on your driving licence. They seldom, if at all demand to see your driving licence before issuing the insurance cover, probably this is to enable them sell many illegitimate covers. This system relies solely on the honesty of the potential client.

### 5.1.1 PROPOSED APPLICATION

The proposed application will make use of the existing database held by the DVLA and the police, the Insurance companies will be required to keep information about their policies and certificates electronically and to transmit it to a centrally managed database and to allow motorists and other law enforcement officers to access the information electronically. It should however be observed that, motorist will have access to very basic information such vehicle keepers name and address, insurer name, policy number and expiry date so as to facilitate with their claim. There will be the need to create a nationwide database for the MOT test centres so that current paper based MOT registers could be upgraded and stored in this database. This eliminates the need for MOT inspectors to visit test centres for the purpose of inspecting these registers. As it is a requirement for all vehicles over a certain age, generally 3 years, to have valid a MOT certificate, renewed annually, extracts from the MOT database will be integrated into the application to facilitate not only the police during their checks but also to provide an important source when renewing or buying a road fund tax.  Please note: a valid MOT certificate is a requirement when purchasing the road tax.

### 5.1.2 ADVANTAGES OF THE APPLICATION

This is a web based application that can be accessed by any Wireless Application Protocol (WAP) enabled handheld device or computer connected to the Internet. It will provide motorists with a source to confirm the details of other motorist when involved in an accident. As all current mobile phones are WAP enabled, this can be done at the accident scene. When the details are

confirmed, it is practical to assume that the correct details are sent to the insurer and this facilitates the process of the claim. The police and other law enforcement agencies have a tool that will enable them to check and confirm the details of a vehicle before the driver is even stopped. Documents such as the vehicle log book, MOT test and Insurance certificates will not be required to be produced at the road side as these details will all be available to the officer. The need to fill out the HORT1 form (producer) by police and for the motorist to produce these documents at the police station will be obsolete, so will the HORT2 and 3 forms. This will not only save the police the estimated £15 million year, but will also release valuable police time and save the motorist the trip to the police station. The insurance companies will benefit from the application in that, the way they insure their clients will change to insuring drivers, as a policy will be sold using your driver number as opposed to them currently relying on the honesty of the potential client. Insurance fraud will be greatly reduced e.g. at present a new driver can tell the insurer he has been driving for probably 5 years thereby reducing the insurance premium but this application will prevent this, as the driver number issued by the DVLA and kept in their database will reveal the exact details. The issues of motorist using false insurance and MOT certificates in acquiring road fund taxes all will be a thing of the past, because when purchasing a road tax, the officer or clerk will have access to the vehicle details the tax is intended for. This will probably change the way we purchase road tax, as there is every possibility it can be online as the entire prerequisite (insurance, MOT and vehicle details) will be available. Hopefully this might encourage the 1.25 million motorists driving without valid insurance to purchase the appropriate policies and save the collective motor insurance industry the estimated £400 million a year and also £15 -£30 a year off the premium of legitimate motorist

*CHAPTER 6*

# DESIGN OF THE APPLICATION

Although PHP is not or not fully Object Oriented, an analysis of the application helped to deal with the complexity inherent in the real world by focusing on the essential and interesting features of the application.

Below were the diagrams used to depict the application model during the design.

- **Use case diagram**

This was used to represent the functional requirements of the application, both from the client and broker standpoints without worrying about how those requirements will be implemented

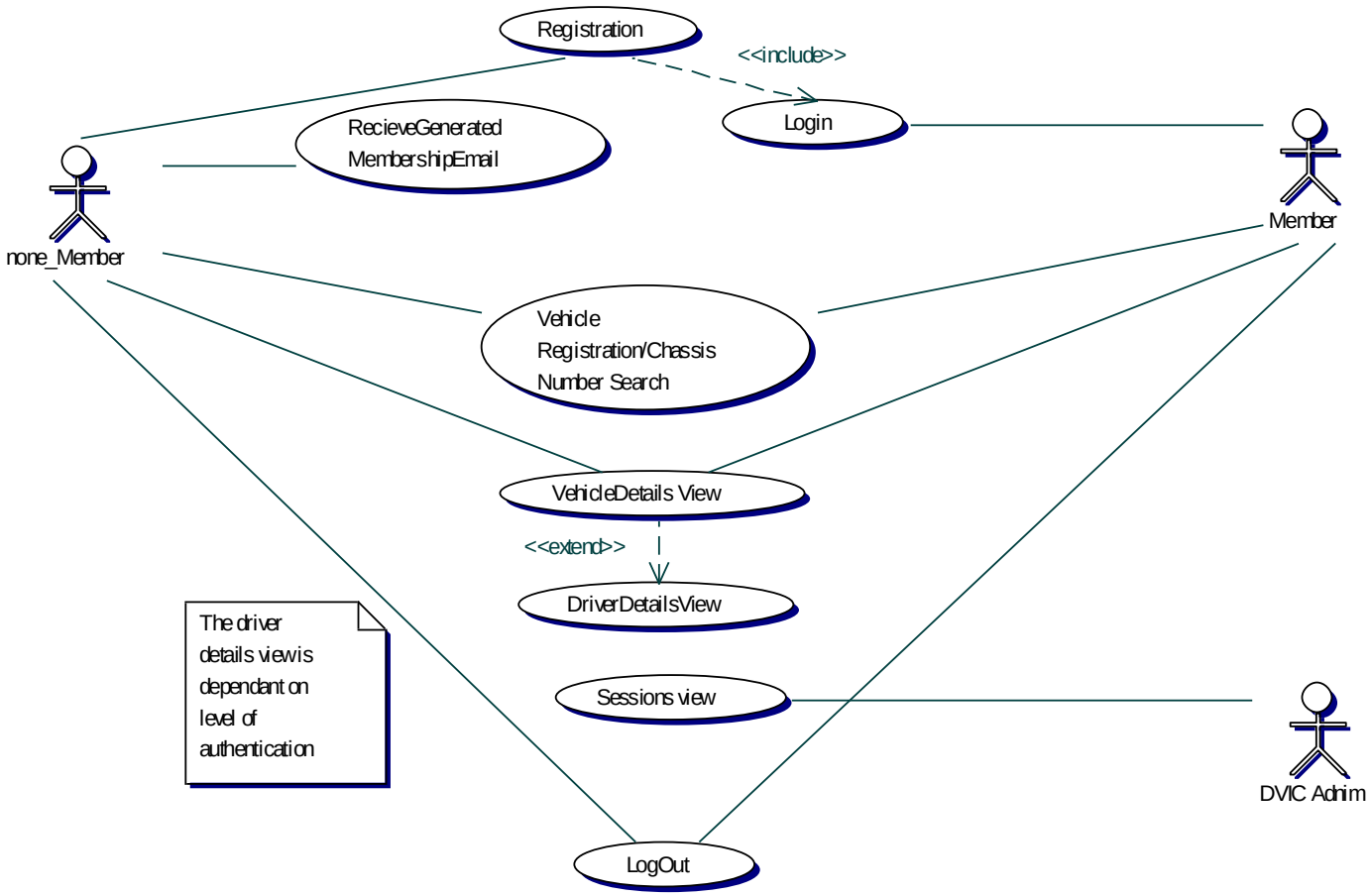
- **Class Diagram**

  This diagram was used to show the static structure of data and the operations that acts on the data.

- **Sequence Diagram**

  Sequence diagrams were used to represent the dynamic models of interactions between the class objects

## 6.1 USECASE DIAGRAM



| USE CASE NAME | REGISTRATION |
|---|---|
| Brief Description | This use case describes how a new member register with the application |
| Actor | This can be a police officer or a driver, and can accessing the application by a WAP device or PC. |
| Flow of events | 1.The application requests the user to enter their first and surnames, email address, username, password and to confirm the password<br>2. The user completes the registration form and submit it<br>3. The application then validates the registration<br>4. An email is generated and sent to the user<br>5. The new member is logged on to the system<br>6. The user table in the database is updated with the new user details<br>7. The application generates a unique session ID and stores this ID together with current date and time and the username in the session table of the database |
| Post-condition | User enters correct registration details<br><br>User is welcomed with a personalised message using their original first name stored in the driver or police table |

| USE CASE NAME | LOGIN |
|---|---|
| Actor | Member a driver or police officer |
| Brief Description | This use case describes how a user logs into the application |
| Flow of events | 1. The application requests the user to enter username and password.<br>2. The user enters the username and password<br>3. The username and password is validated by the application<br>4. the user is logged on to the system<br><br>5. The application generates a unique session ID and stores this ID together with current date and time and the username in the session table of the database |
| Post-condition | User enters correct user name and password |

| | |
|---|---|
| Use case name | Receive Generated Membership email |
| Actor | New member |
| Brief description | This use case describes how an email is generated by the application and sent to the user at successfully completing the registration process |
| Flow of events | The application sends an email to the new member using the email address provided by the user on the registration form |
| Post-condition | User receives an email confirm registration confirm user name and password. |

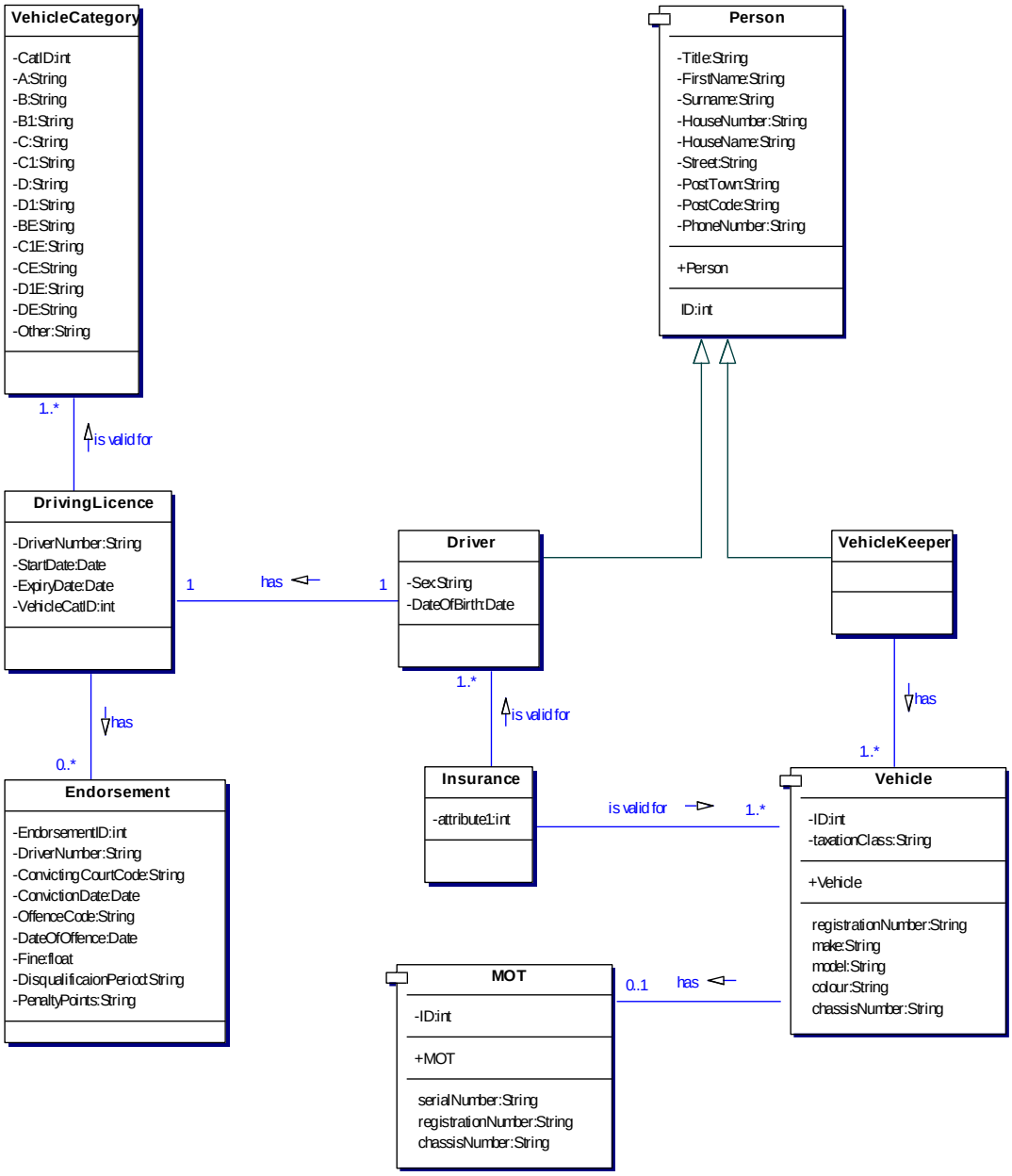
| USE CASE NAME | VEHICLE REGISTRATION OR CHASSIS NUMBER SEARCH |
|---|---|
| Brief description | This use case describes the core function of the application. It provides the user text field to either enter a vehicle registration or chassis number to begin the search |
| Actor | Member, who can either be a driver or police officer using a WAP enabled device or a PC to access the application |
| Flow of events | 1. The application displays a text field requesting the user to enter a vehicle registration or chassis number.<br>2. The user enters the vehicle registration or chassis number that they wish check and press the search button<br>3. The system searches the vehicle, vehicleKeeper and insurance tables in the database for the requested details, if the user is a police officer, the MOT table is included in the search. |

| | |
|---|---|
| Use Case Name | Vehicle details view |
| Actor | Member, who can either be a driver or police officer using a WAP enabled device or a PC to access the application |
| Brief | The application displays the first and surname, address, insurer name, insurance policy number and the expiry date to the user. If the user is police officer the MOT certificate number and expiry date is also added. |
| Flow of events | 1.The application displays the users first name obtained from either driver or police table and the registration or chassis number requested for which details is been enquired.<br><br>2. A list of the first and surname and address of the vehicle is retrieved from the vehicle and vehicleKeeper table, the insurer name, policy number and expiry date and time are also retrieved from the insurance table in the database and displayed, if the user happens to be a police officer, the MOT certificate number and expiry date is retrieved from the MOT table and displayed.<br><br>3. The logout button is also displayed<br><br>4. The session table is updated with vehicle registration or chassis number and date and time stamped. |

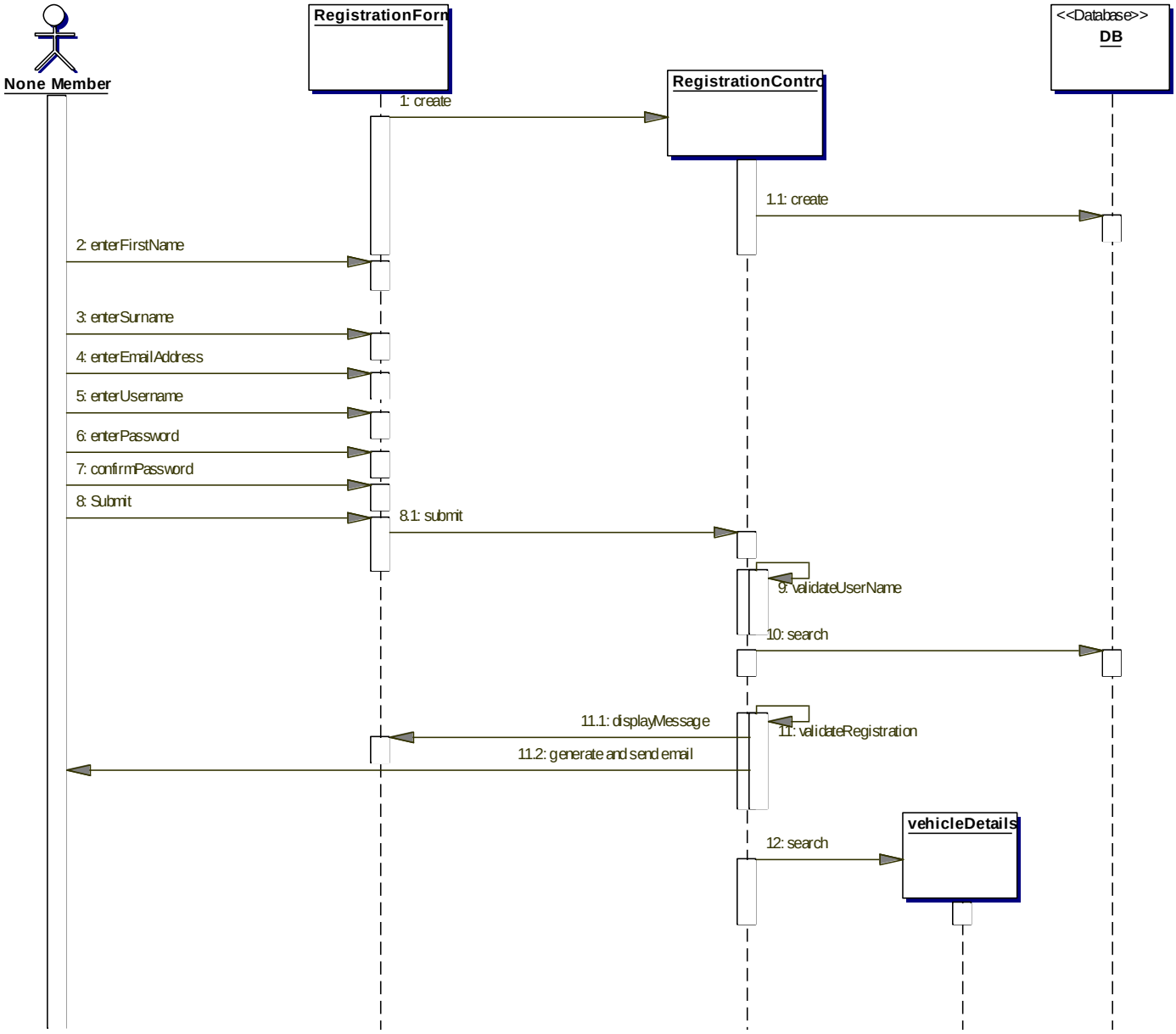
| Use Case Name | Sessions view |
|---|---|
| Actor | DVIC administrative staff |
| Brief Description | As a form of security, this use case provides a view of the user activities e.g. what user viewed what details |
| Flow of events | 1.The application prompts the actor to enter the vehicle registration or chassis number or the username of a user whose usage to view<br><br>2. Usage is displayed |

| Use Case Name | Logout |
|---|---|
| Actor | Driver or police officer |
| Brief description | This use case log out the user and terminates the session |
| Flow of events | 1.The application presents the logout button<br><br>2. The user invokes the logout button which runs the logout script.<br><br>3. The application ends the sessions.<br><br>4. A message confirming logout is displayed. |

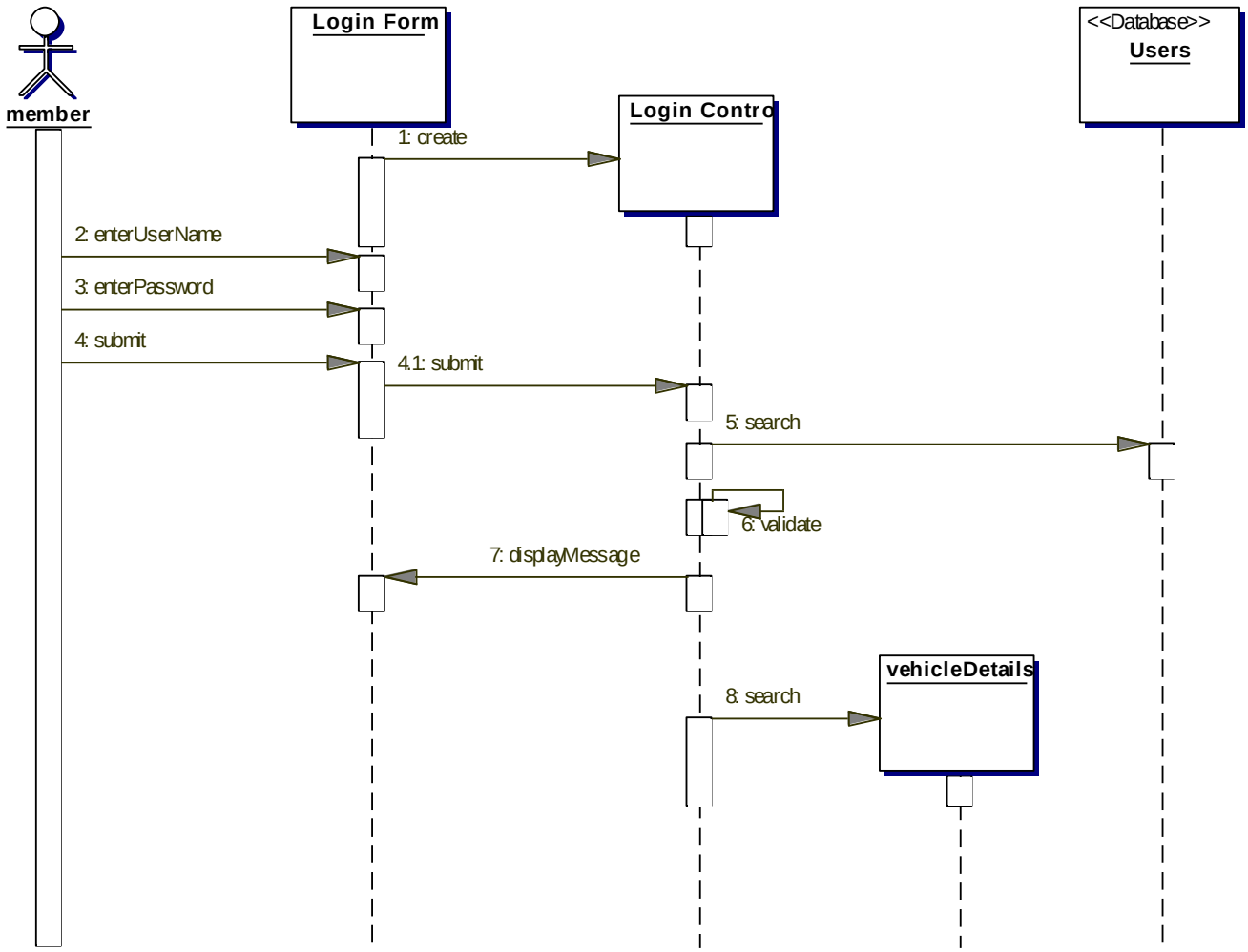
## 6.2 Class Diagram



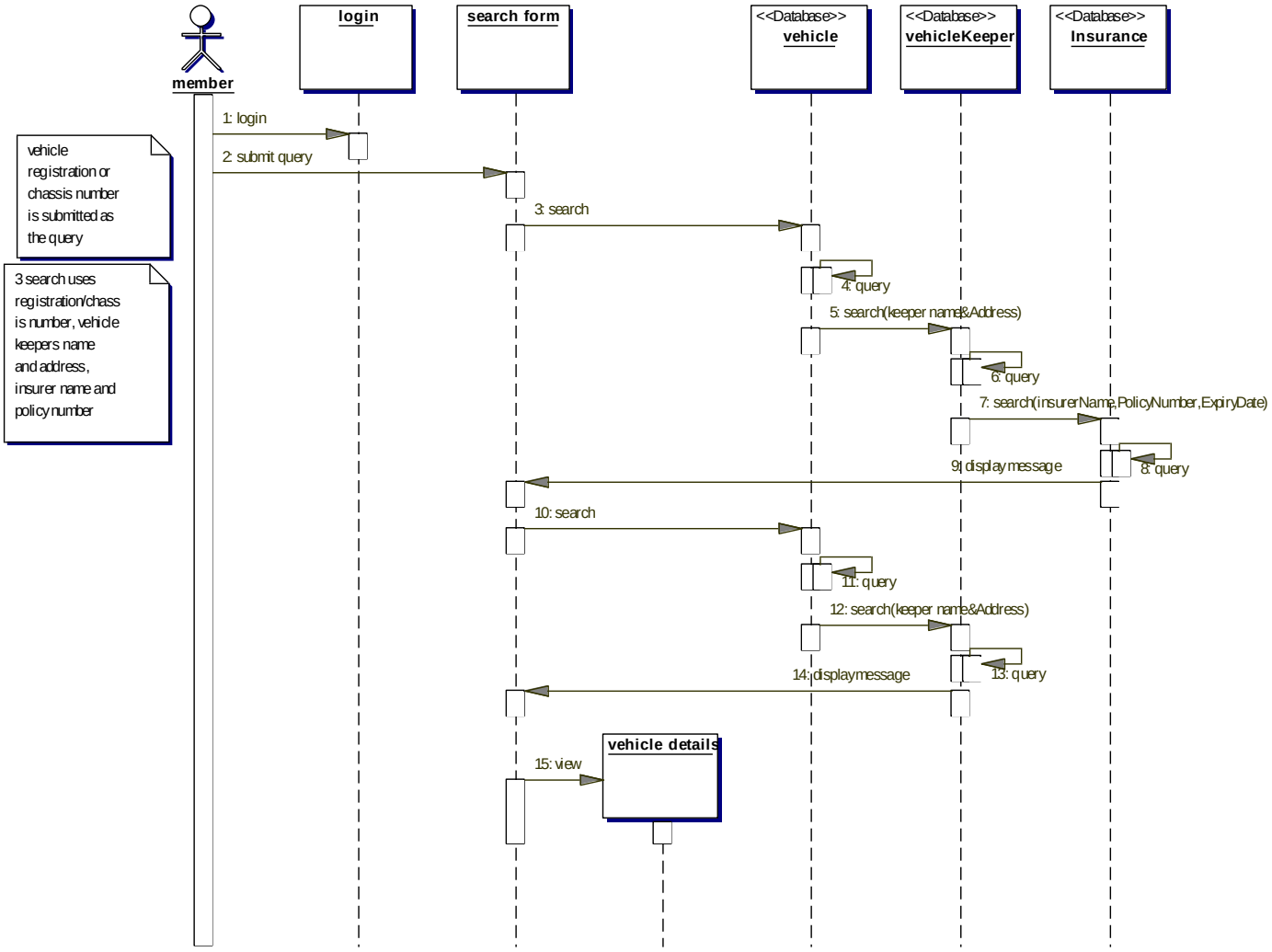
## 6.3 REGISTRATION SEQUENCE DIAGRAM



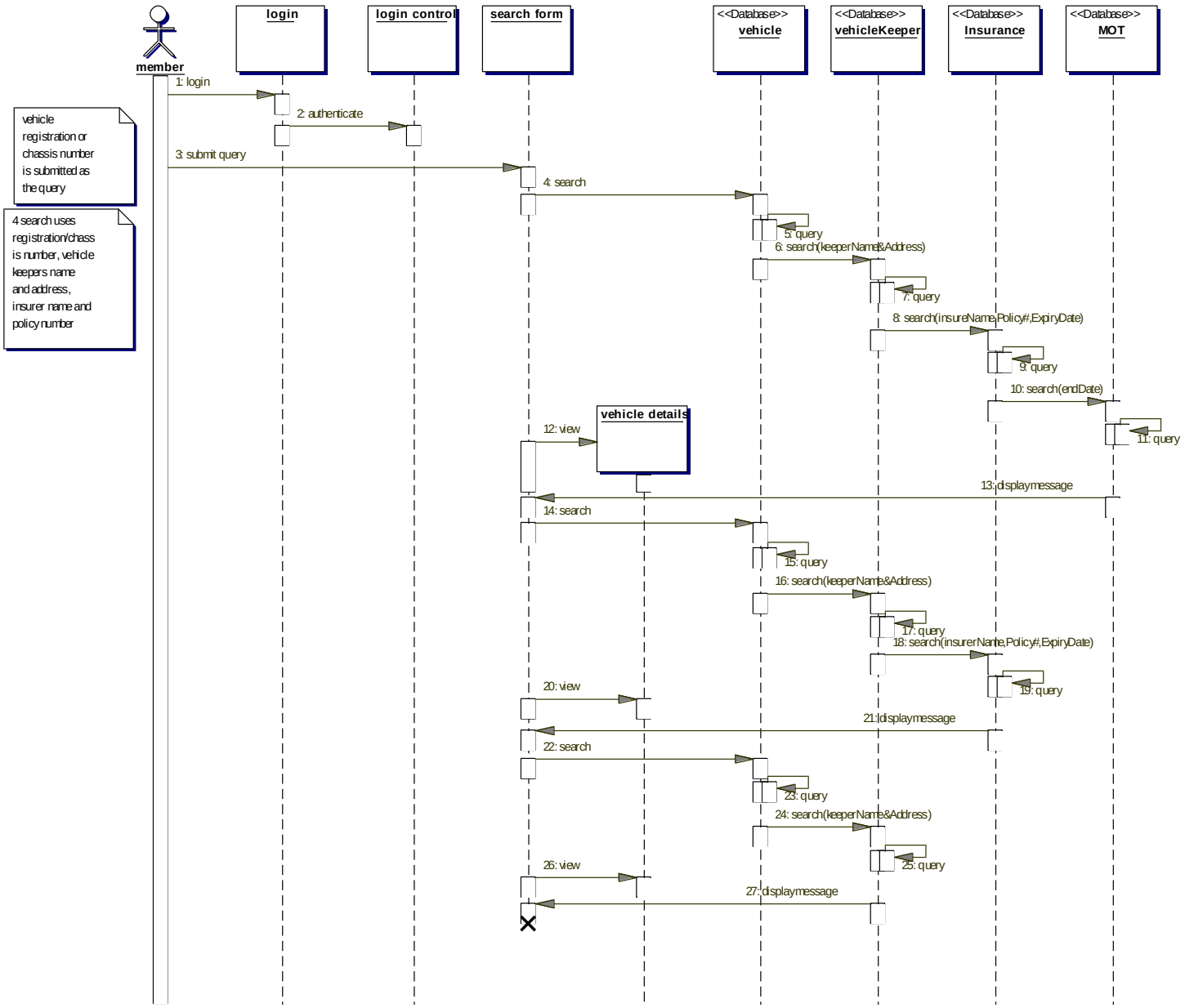
### 6.3.1 LOGIN SEQUENCE DIAGRAM



## 6.3.2 MEMBER SEARCH VEHICLE DETAILS



## 6.3.3 POLICE SEARCH VEHICLE DETAILS SEQUENCE DIAGRAM



*CHAPTER 7*

# IMPLEMENTATION

## 7.1 ABOUT THE SYSTEM

When the URL http://stuweb.cms.gre.ac.uk/~ov102 is typed into a web browser, the index.htm is opened displaying a welcome page and a brief introduction of what the site is about. A tool bar with 6 buttons (login, what we check, the facts, sample report and about us) at top of the page is provided for navigating the site. The URL of the WAP version is http://stuweb.cms.gre.ac.uk/~ov102/index.wml, and can be viewed using a WAP enabled browser. This also presents the user with a similar welcome page and brief description about the site, a next soft button is provided so that when clicked the user can proceed to the options page where they select to login into the application, register or view the what we check, the facts, sample report and the about us pages.

**weCheck.htm and weCheck.wml**

The "we check" button when clicked takes the user to "weCheck.htm", this is a basic html page that tells the user the sort of results to expect and it also describes our search criteria. Its WML version is can be invoked when the "what we check" hyperlink is followed.

**facts.htm and facts.wml**

The facts button leads to "facts.htm", this also provides a brief information about the current issues regarding uninsured drivers. It also talks about driving issues and effects of breaking the law. The WML version displays the message when its hyperlink is selected from the options page.

**sample.htm and sample.wml**

The sample.htm and sample.wml displays a specimen report of the results search in the various versions.

**aboutUs.htm and aboutUs.wml**

The about us button and about us hyperlink leads to the contact details of the site in the two versions.

**Login.htm and login.wml**

The main functionality of the application really begins when the login button or hyperlink is invoked, here the user is given option to either login or register. If a user has already registered, they can proceed and login using their username and password. If they haven't already registered and try to login, access is denied and a message describing the error is displayed and then automatically redirected to the registration page (after six seconds only in HTML version). Upon successful login, the user is handed the appropriate level of access depending on their level of access. The JavaScript or WML ensures that both username and textbox fields are not empty. This form is then sent for authentication (search.php or p_search.php for the HTML version or w_search.php or p_w_search.php for the WML version).

**registration.htm and registration.wml**

On the registration form a series of simple questions are asked and most importantly, the username and email address. As the registration form is designed for both motorists/drivers and law enforcement officers (the police), a series of validation are required. The username of drivers are their driver number which is the 16 digit unique alphanumeric found on every driving licence (this is issued by the DVLA at the time the licence is first issued) and for the police they use their 6 or 4 digit warrant numbers. This so, to ensure security, and as this site is purposely designed to support drivers, only drivers with driving licence can use the site and of course the police. During the registration process, the username submitted is checked with the entries from the driver table in the database to ensure that it is a valid driver number, if a match is found and every thing else is fine, the user is registered and an email is generated and sent to him confirming the registration and information about their account. If the user has already registered and tries to re-register, an error message is generated and the "regerror.htm" or "regerror.wml" pages are invoked explaining the error, the user is then redirected to the login page. If the username supplied is not valid or not found in the driver or police tables in the database, again an error message is generated and displayed and the user is redirected back to the registration page. As there are currently two main username formats, if a user tries to register, the application checks for the length of the username supplied and is redirected to the appropriate page. With the police registration, the username (warrant number) is validated from the police table, once again if a match is found and is not a re-registration attempt, it is authenticated and a welcome message incorporating the first name that is stored in the driver or police table in the database displayed and an email is generated and sent to the new member. The registration form is validated thoroughly to ascertain that it

is filled out correctly. The first and surname fields are checked that they contain at least an alphabet, the email field is validated for a valid email address and the password and confirm password fields are checked that they contain the same values. Theses fields on the form are validated by JavaScript. This form is also authenticated by Rsearch.php. The password is encrypted using a built in MySQL function, the user table is then updated with the new user information. After a successful registration and login, a session is generated and , and for the purpose of security, the session ID generated together with the username and the current date and time are inserted into the session table in the database.

The registration.wml behaves very similar to the HTML version expect that it does not use JavaScript for validation but instead WMLS script.

**Rsearch.php and w_Rsearch.php**

This handles and authenticates the registration form, it checks for the user type by way of username and redirects to the appropriate page. As a form of double security, it checks that the user has filled out the registration form and it contains at least the username and password, if assuming the user just typed "Rsearch.php" in the address bar of the browser, the user will be redirected to the registration page. This is where decisions such as valid and invalid usernames are picked up, if a username is not valid, the user is sent to regerror.htm which displays an error message telling the user that, the username is not valid and that after 6 seconds will be redirected to the registration form. If the registration form has no errors, however, it generates a personalised welcome message, addressing the user by the first name basis as recorded in the driver table and also composes and sends a personalised email using a similar technique and confirming to the user that, an email has been sent and that they check their inbox. A search box is then presented prompting the user to enter the vehicle registration or chassis nuumber that they wish to check, once again JavaScript ensures that the form is field out properly. Once this form is filled out and submitted, "data.php" takes over.

The WAP version has a similar functionality except that WML does the validation i.e. ensures that the text field has a valid value.

**search.php and w_search.php**

This file works in a similar way as Rsearch.php except that it authenticates the login (login.htm or login.wml). If the login form is not filled out prior to accessing this page, the user is redirected to the login page. When a user attempts to log into the application, this file loops through the users table to be find a match for the username and the password, if no match is found, the

user is sent to the loginerror.htm or loginerror.wml where an error message is displayed informing the user to check their username and password and retry, (assuming the username was correct and wrong password was entered or wrong username and password, the user will still get the same error message as this is a form of security, denying illegitimate users to the opportunity to guessing passwords) the user is then automatically redirected to the login page. If the login details are correct, a welcome message similar to that of Rsearch.php is displayed, but this time no email is sent. Depending on the level of access (username format) the appropriate page is displayed. A search text box prompting the user to enter the vehicle registration they wish to check is also provided. A new session then starts and the username, date and time is inserted into the session table, this form is then passed on to either data.php or p_data.php to search and display the details of the vehicle.

The WAP version has the same functionality except that the validation is done by wml and the form is posted to w_data.php or w_p_data.php respectively.

**data.php and p_data.php**

These two files work in a very similar manner, in that, they both receive similar requests from Rsearch.php, search.php and p_Rsearch.php,p_search.php respectively. They both search the vehicle table using the vehicle registration or chassis number as a search string. If the search is successful and all the criteria are met, i.e. a valid vehicle registration or chassis number and insurance, data.php will display the results in a table format listing the vehicle keeper's name, address, and insurance details. The p_data.php displays in addition, the MOT status of the vehicle. The session table is now updated with the vehicle registration or chassis number and the current date and time. If all of the above mentioned details are not found, the searcherror.htm file is invoked. The user has the option to engage the logout button which will invoke logout.php or just close the browser to terminate the session.

The WAP version w_data and w_p_data works similarly and they in turn receive requests from w_search, w_p_search or w_Rsearch and w_p_Rsearch respectively.

**searcherror.htm and searcherror.wml**

This form informs the user that no insurance detail of the requested vehicle was found, and that if the user wants to search for the name and address only of the vehicle. A vehicle registration or chassis number search box is displayed, when a registration or chassis number is typed in, the submit button posts the contents to address_searcherror.php for processing. If a match is found,

the details are displayed in a table format and the session table is updated with registration or chassis number and the date and time. If no results are found, the bigerror.htm or bigerror.wml is invoked, and this displays an error message and the user is redirected to index.htm and index.wml respectively.

**bigerror.htm and bigerror.wml**

This displays an error message saying that no details were found on the vehicle and the user is redirected to the respective index welcome pages.

**Db.php**

This file defines the PHP functions that deals and maintains the session.

*CHAPTER 8*

# TESTING

The application was tested using the two main testing methods i.e. the functional and structural testing schemes. For the HTML (PC based) version, the two main types of web browsers, Microsoft's Internet Explorer and Netscape Navigator were used to test for the functionality and presentation and phone.com's Openwave emulator, Deckit WML previewer by PyWeb.com, Ericsson's R380 emulator, WinWap from winwap.org and finally Nokia's Nokia Mobile Browser were used for WAP version.

Although the application worked to its full functionality as expected on both Internet Explorer and Netscape Navigator, there were minor differences in the presentation, the navigation bar was slightly displaced when viewed in Internet Explorer, but perfect in Netscape. Having said this it was clearly visible and acceptable. In fact most of my tester failed to recognize it. Another trivial problem was that when the login and registration forms were loaded, Internet Explorer was able to interpret the JavaScript onload function and set focus on the first text field but Navigator failed to do this.

The WAP testing was a bit more of a challenge, although most of them displayed the pure wml content as expected, there were slight variations when viewing wml pages embedded in PHP code. But overall they were all acceptable. Wbmp images did not display in all the emulators so they had to be dropped.

## 8.1 Functional or Black Box Testing

This type of testing is derived from the program specifications and the tests are devised to explore the various inputs and corresponding outputs. The text fields in the registration form were tested thoroughly to ascertain that they had valid entries before been passed onto server. The first name, surname, username, email, and both password text fields were initially tested for null entries and if they returned true, the JavaScript displayed an alert box describing why the error has occurred and sets focus on the offending text field and in this particular case a message saying that the field was left empty and has to be filled, giving the user the opportunity to correct the error. It's fair to assume that first and surnames usually begins with an alphabet, so this was tested for as well, so that if those fields began with a space, special character or a number, JavaScript displayed an alert box complaining about the format of the entry and again the focus is set on the offending text field for re-entry. The email text field was probably the biggest challenge in the validation process as this was tested to conform to a valid email address format. I did check that the field started with an alphanumeric character, no spaces, followed by the @ symbol and at least an alphanumeric

followed by a dot (.) and then followed by an alphabet. It was also tested to ensure that it did not end with a dot (.). Also the presence of just a single @ character. If any of these rules were broken or did not follow the sequence as stated above, again, JavaScript displayed an alert box explaining the nature of the error. The password and confirm password fields were also tested to ensure that they had the same the value, else an alert box is displayed describing the error and the focus is set on text field with the error. The fields in the vehicle registration and chassis number search forms were also tested to ensure that they did not have zero length values. The WAP version of the application was also tested with all the above discussed and violations were caught by the WMLS Scripts.

**8.2 Structural Testing**

Structural or White Box tests are derived from the knowledge of the program or application's structure and implementation. This prior knowledge allows the testing of the various components and paths that the program is executing and takes.

As this web application is designed for driver, it is expected that all visitors and users of the site or application, with the exception of law enforcement officers (police) should posses a valid driving licence. The user name required to complete the registration process on the registration form were tested to ensure that, the registration process could only be completed successfully only when a valid driver or warrant number is entered into the user name field. The driver and warrant numbers (user name) are in different formats i.e. 15 and 4 or 6 characters respectively. Serving the appropriate or correct pages is very paramount in enforcing security and thus the Data Protection Act as to prevent standard users being able to access certain details which they are not authorised to. As the application uses the length and format of the user name to distinguish user authentication levels, a thorough test was conducted with this in mind, but this time if a violation is detected, users are redirected to the respective HTML and WML pages which informs them of the violation and after five to six seconds are redirected again back to the login page. The Java Scripts on these error pages i.e. loginerror.htm, regerror.htm, regrror1.htm and bigerror.htm were tested on both Netscape and Internet Explorer to ensure that users were automatically redirected to the appropriate pages after the five seconds period for redirection. Upon successful registration, new members are sent an email to confirm their membership details, this was also tested that, it worked for both the HTML and WAP versions. As security and privacy is a big issue in this type of application, the Session table that store all user activities and transactions was also rigorously tested to ensure it stored and maintained the audit trail.

*CHAPTER 9*

# CONCLUSION

A long awaited web based system that can address the needs and empower law abiding drivers to be able to check and confirm details of illegitimate drivers at roadside when a motor accident occurs. At present it's only the police that have the resource and access to this sort of information, and even them, they have to radio to their station to get this sort of information relayed to them at the roadside. It has been a personal challenge to design and implement a system that no one has designed over all these years and yet a very powerful, easy and useful tool. It is amazing that in this day and age the law requires us to just accept anything that third party provides without checking it, taking into consideration that an estimated 1.25 million drivers are driving without insurance.

The motivation for choosing this topic was to help curb this problem, although the system doesn't deter or discourage those drivers without the proper documentation, it does help set the minds of the legitimate ones at rest when there is the need to check for these details.

The police on the other hand will have the advantage of checking and receiving driver details such as their driving licence, MOT and motor insurance status instantly by the roadside eliminating the need to issue producers and to save themselves from the unnecessary paper work (completion of HORT 1 and 2 forms) thus increasing productivity, and also saving drivers the need to produce their documents at the police station. As I am a driver myself, it gives me a sense of satisfaction and indeed achievement.

The technologies used for the application were all relatively new to me and had to learn it for the purpose of the project. With the exception of the JavaScript, HTML and some WML which was taught in the Web Application Technology lectures, the rest of the languages and scripts were all new. As a personal challenge, I wanted the entire project and application to be built using just pure open-source software, and no Microsoft products and indeed I managed to achieve just that, and in fact this report was written in word perfect. The client-side of this web application was written in HTML and WML whilst the validation was in JavaScript. PHP was used as the Server-side scripting language whilst MySQL served as the database. The web application is hosted on an Apache Web Server which runs on the University's Unix Server.

Initially, I planned to do a quick prototype using Microsoft Access as the database as I had no prior experience or better still never used MySQL, but as I went through the tutorial provided with database, I was amazed how quick and easy it was to learn

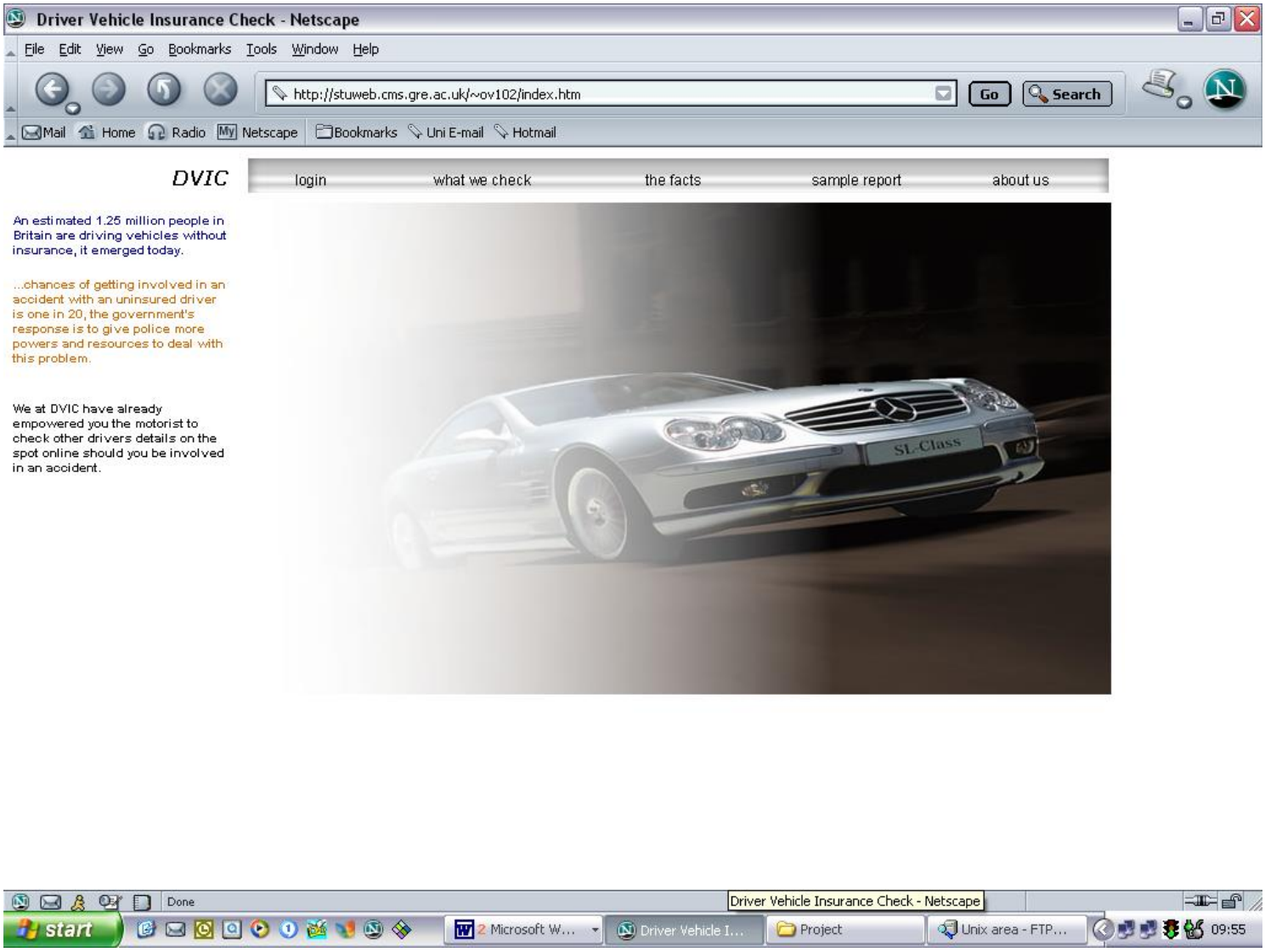
MySQL, and so the whole idea of prototyping in Access had to be abandoned. PHP was equally quite straight forward to learn and for a project like this it was the perfect choice. I have had a previous encounter with Microsoft's ASP, but PHP's ability to run from virtually any web server was a motivation to learn it.

Limitations

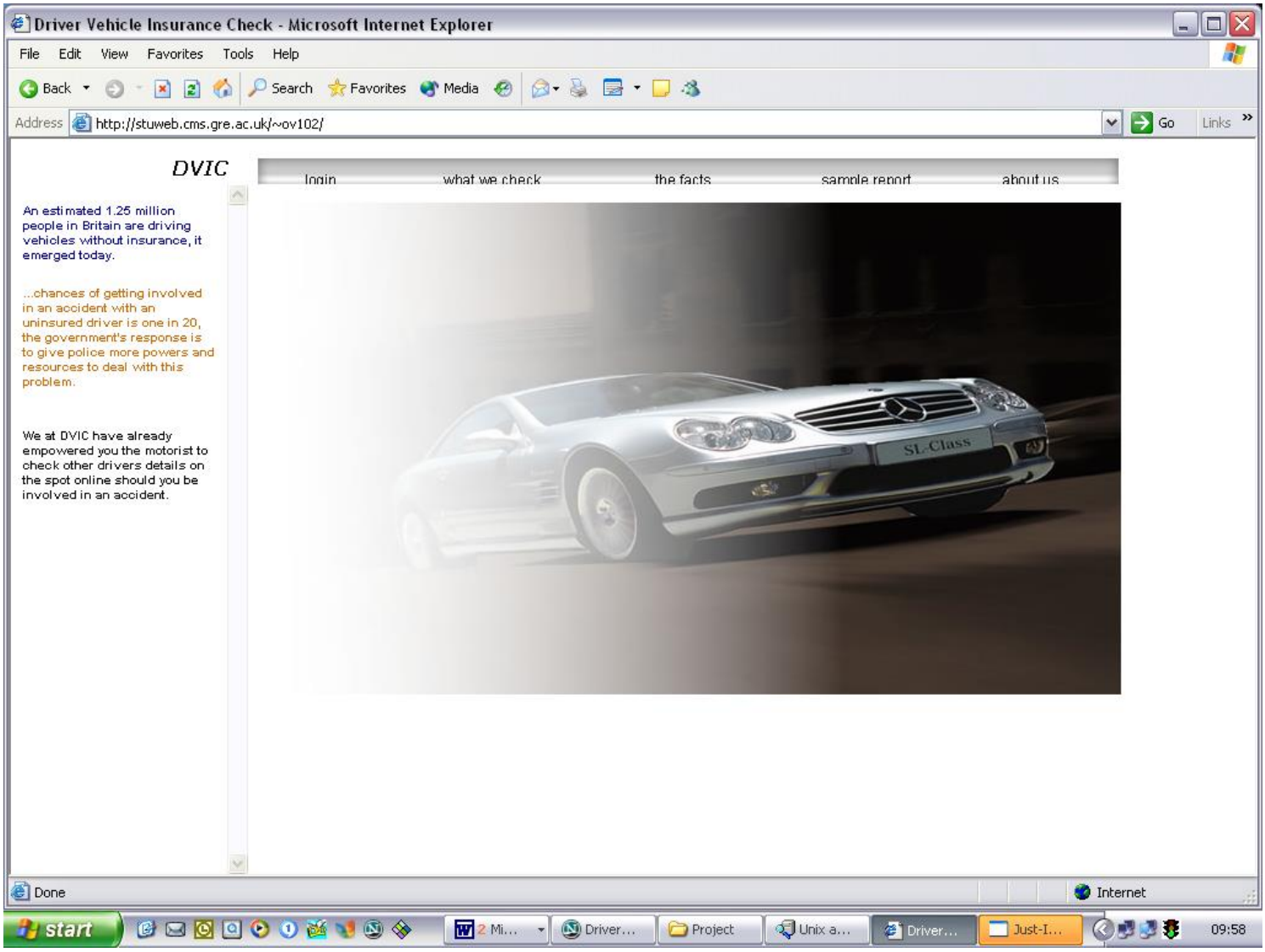
It would have been appropriate if the application could charge or handle some form of payment. Like many of web based application example www.192.com, www.theAA.com, and a few others that provide similar services, they charge for the services they provide.

Appendix A

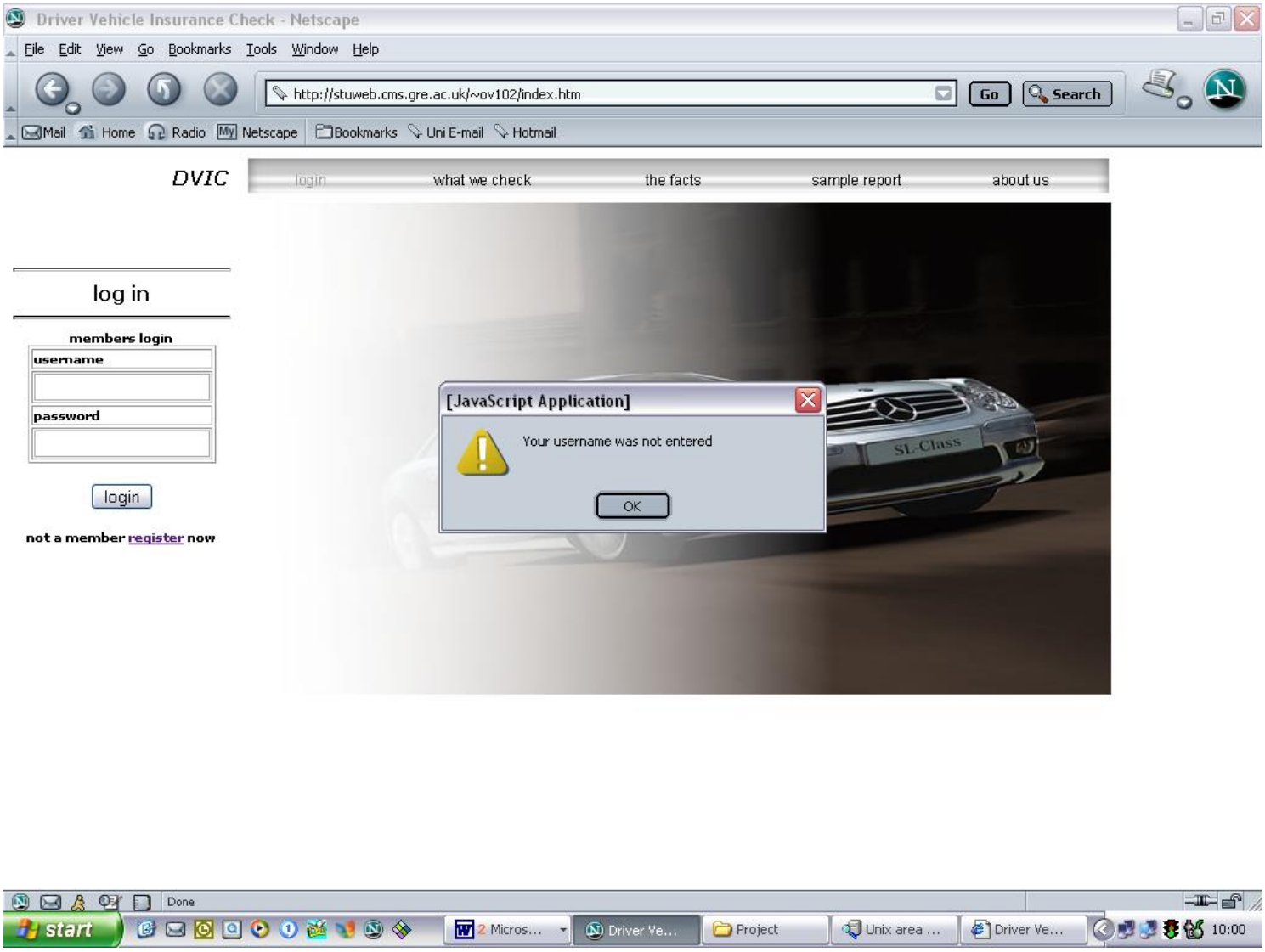
Screen shots



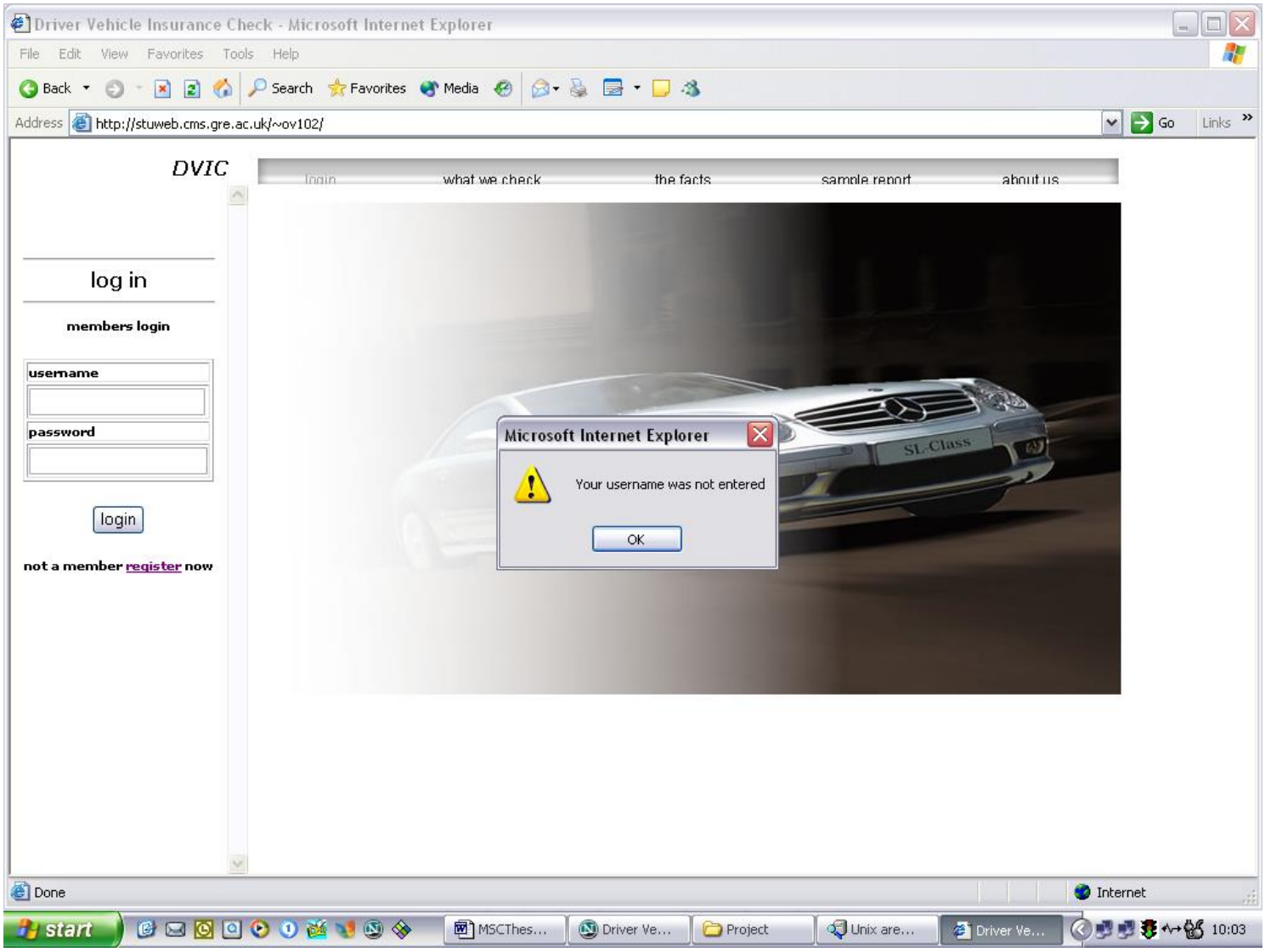

The welcome screen as displayed in Netscape



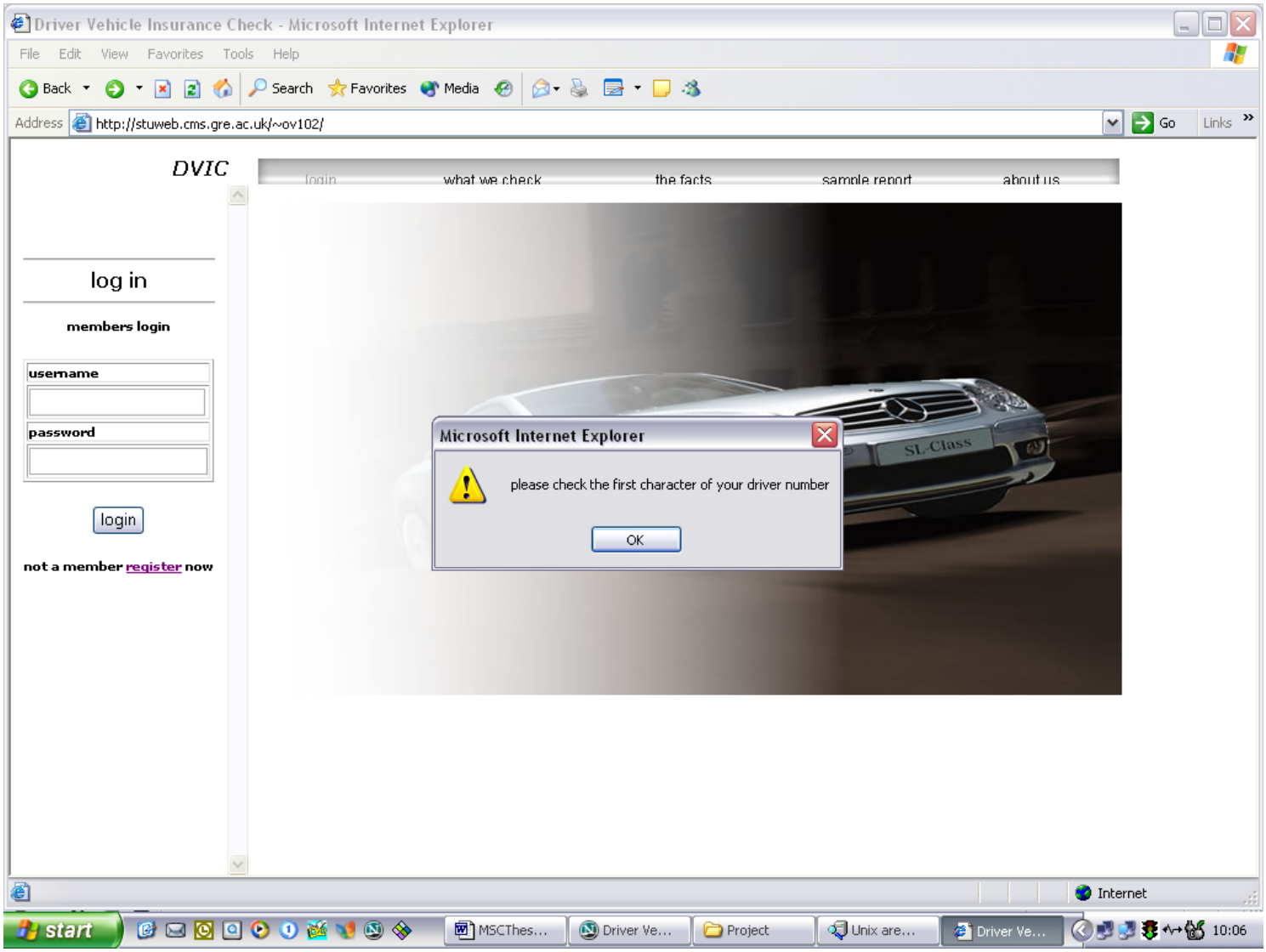

the welcome screen as displayed in Internet Explorer



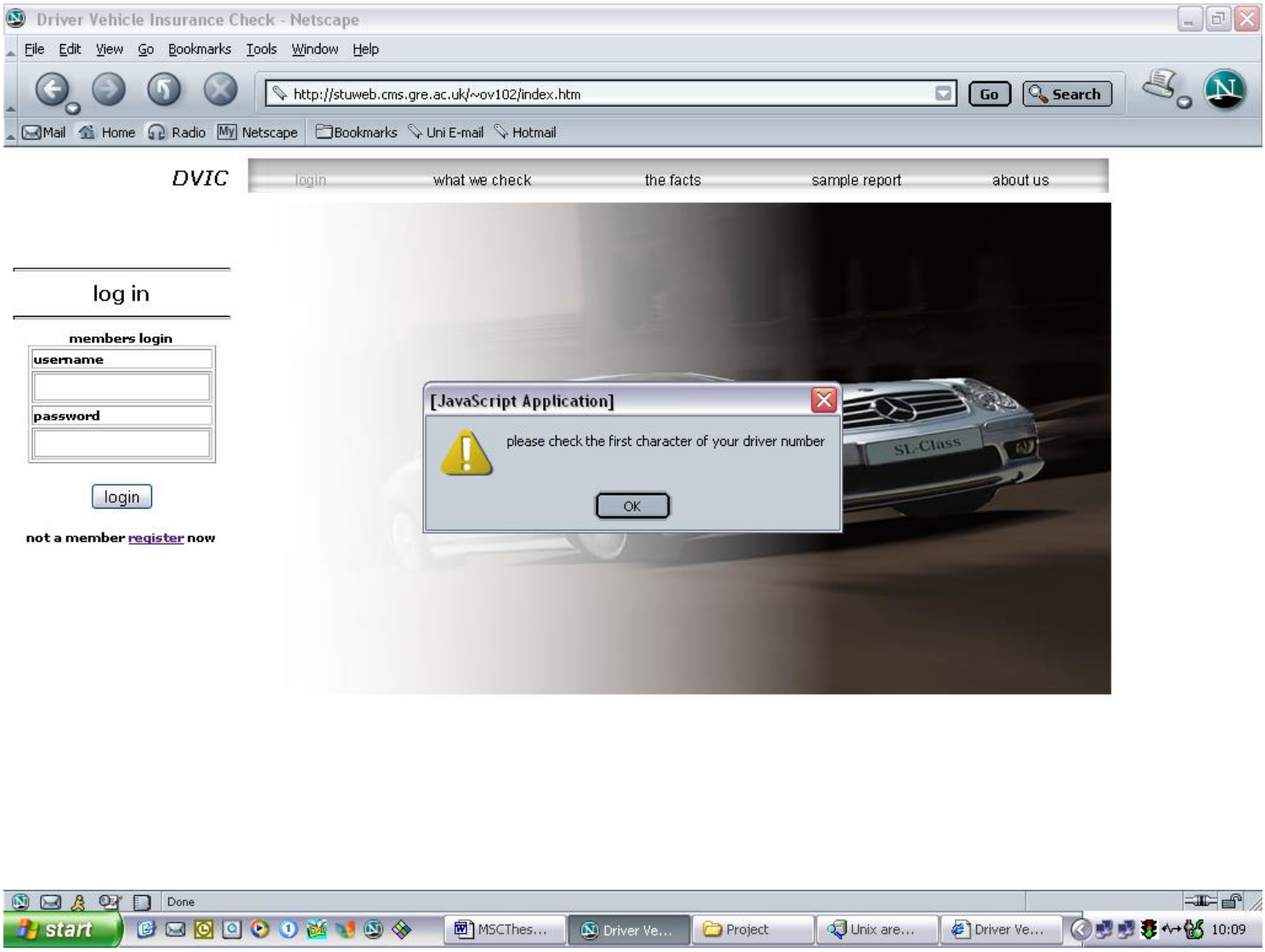

the login window Netscape displaying an error and prompting the users



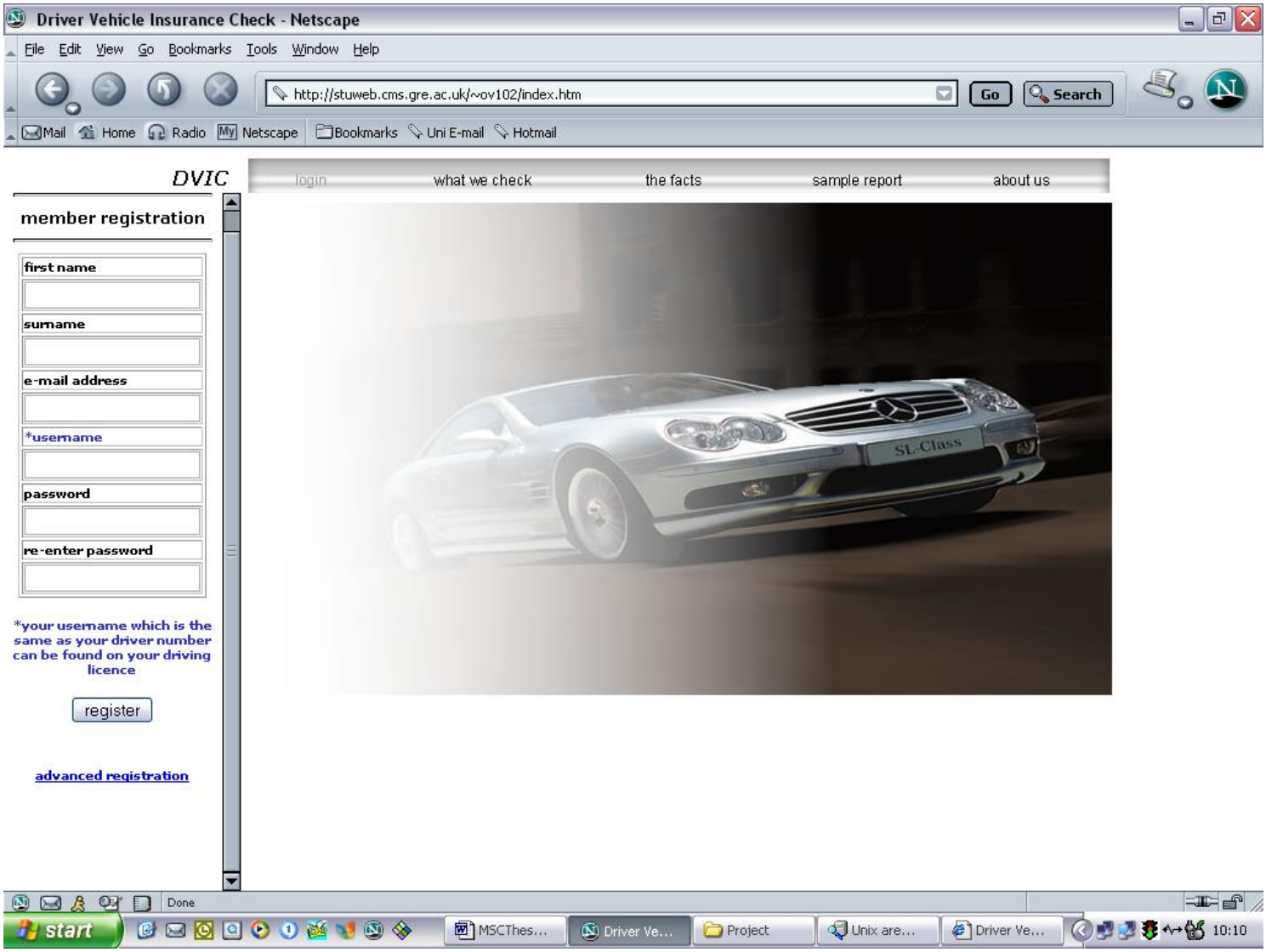

the login window in Explorer displaying an error and prompting the users



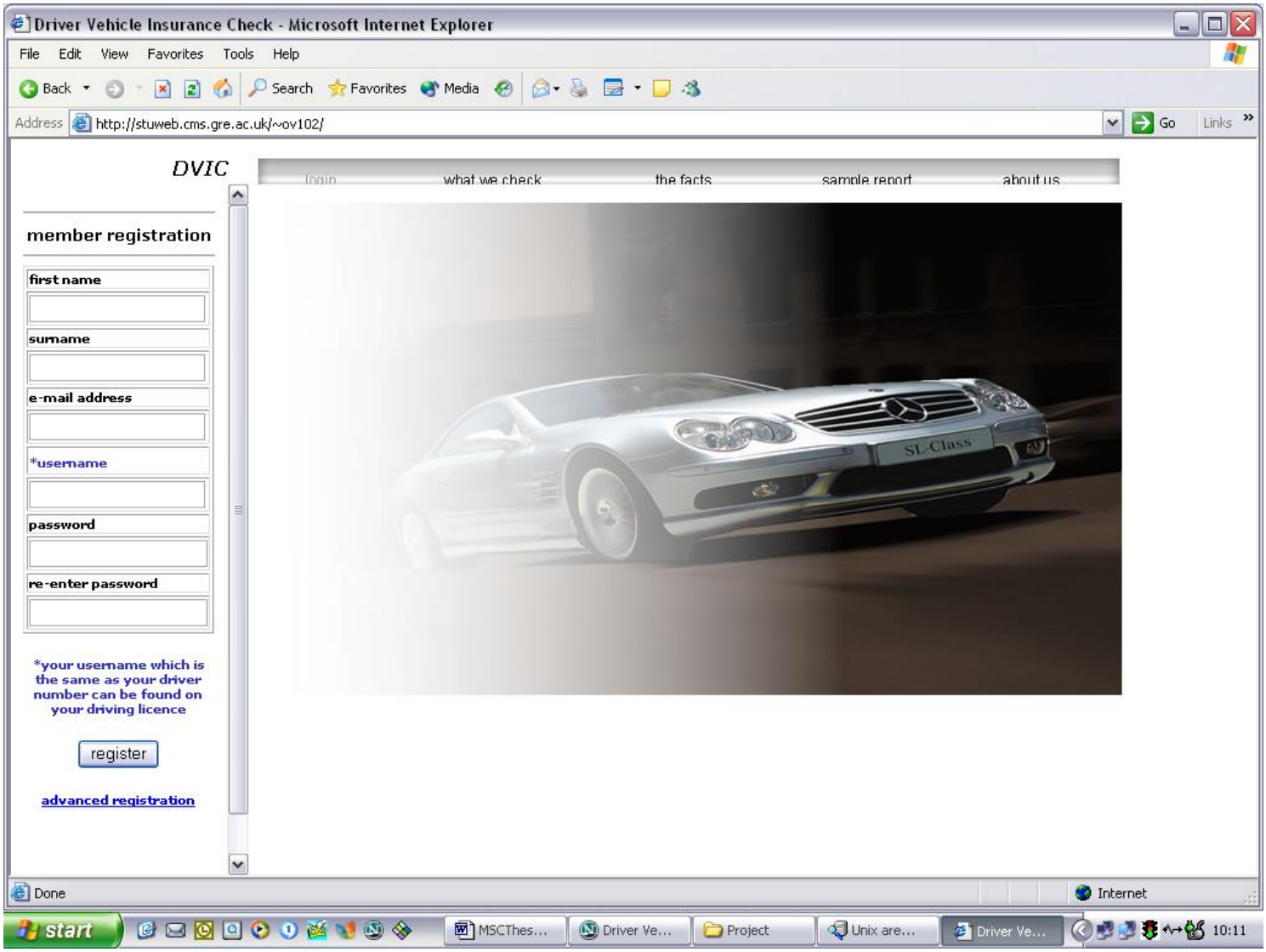

the login window in Explorer displaying an error message and prompting the user, as the first character is a space



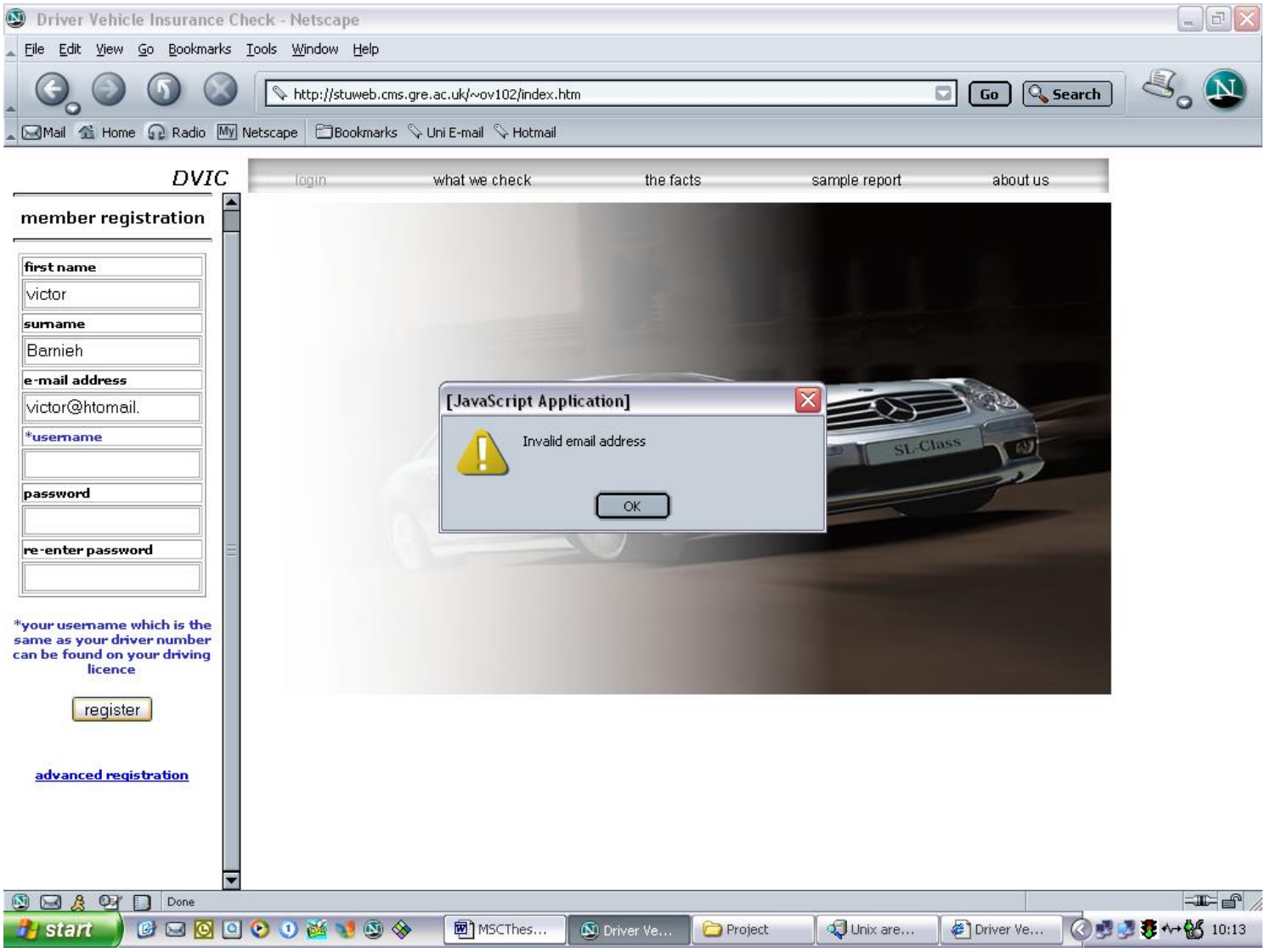

the login window in Netscape displaying an error message and prompting the user, as the first character is a space



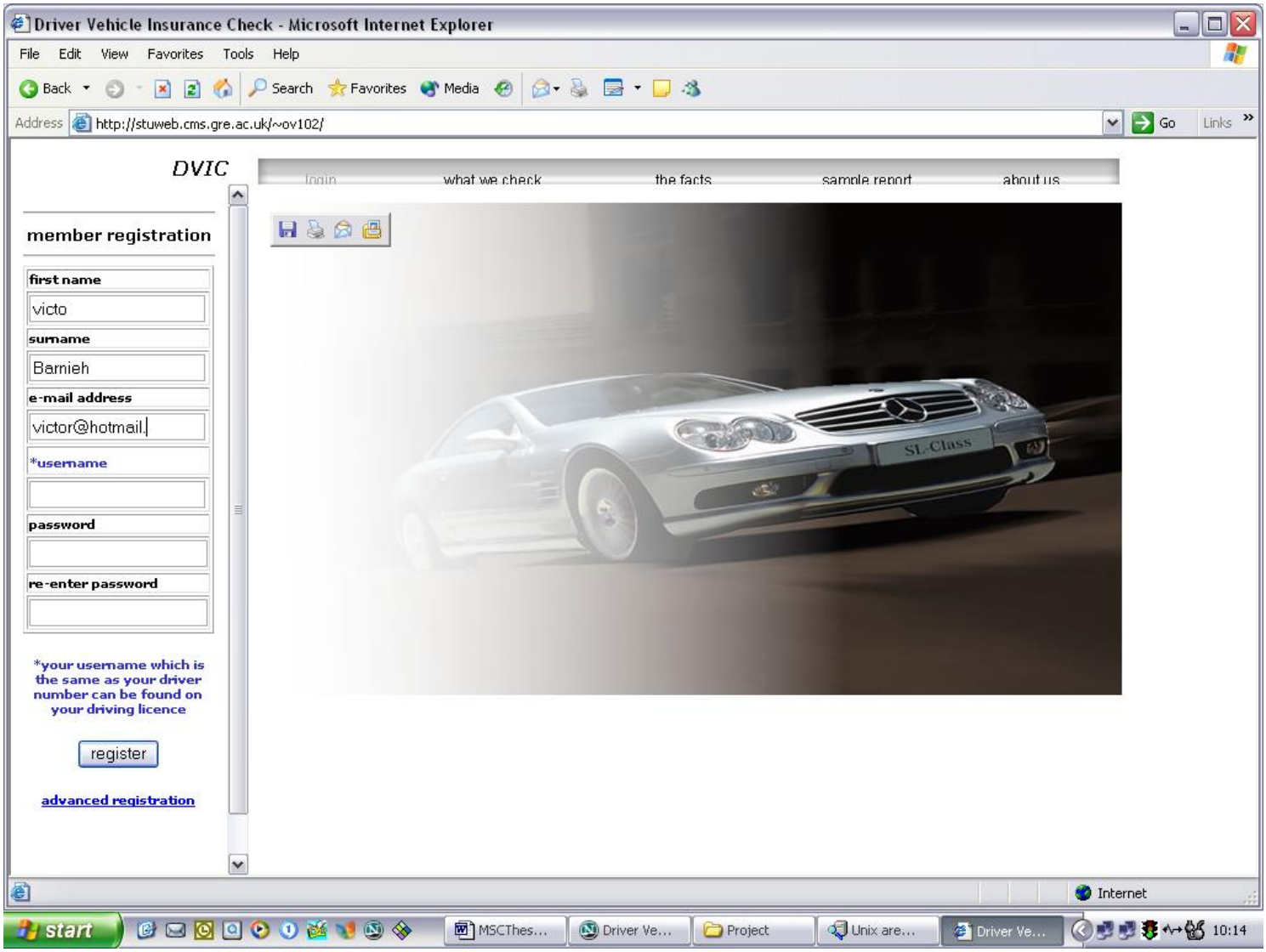

The registration page as displayed in Netscape



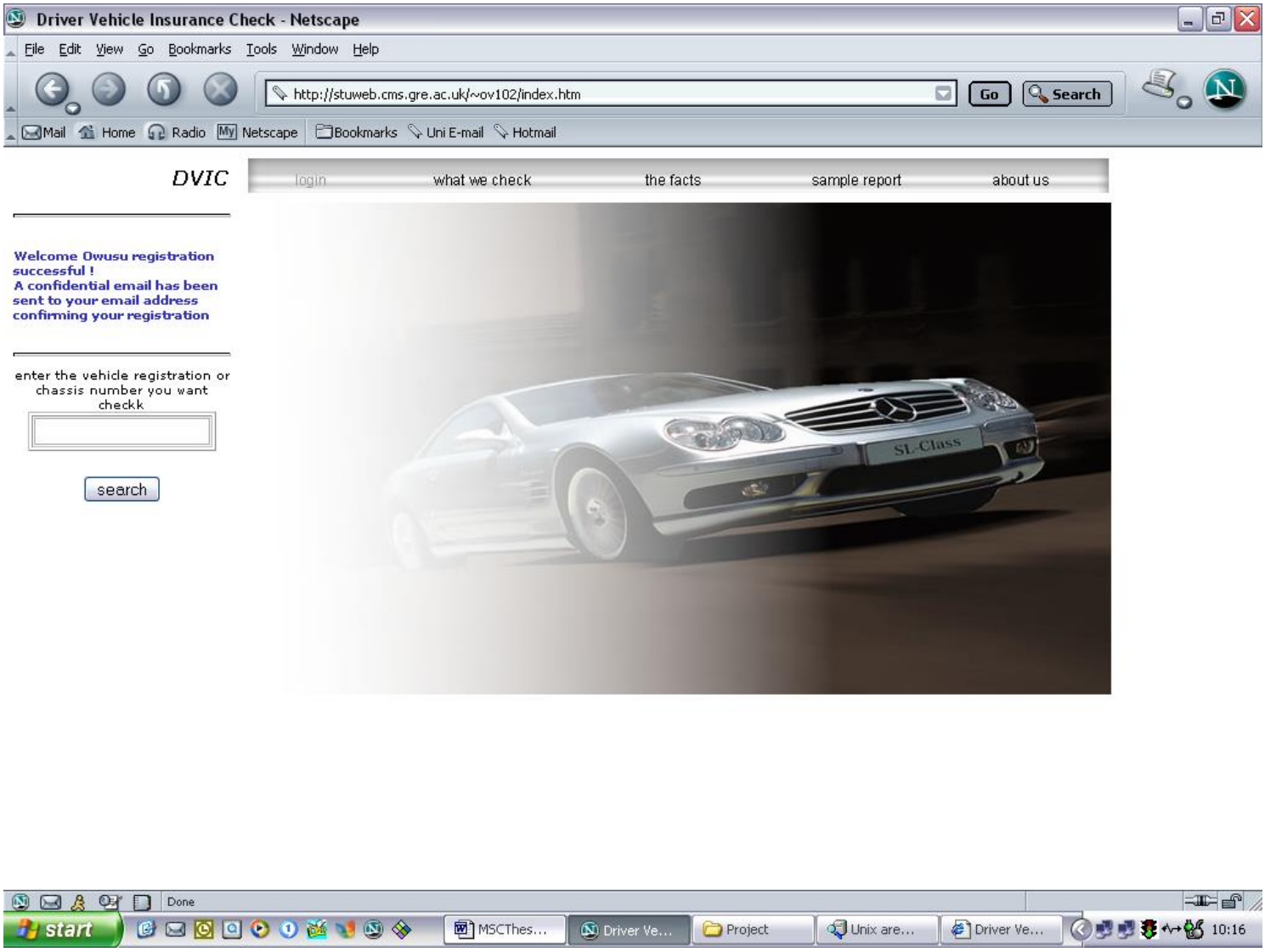

The registration page as displayed in Explorer



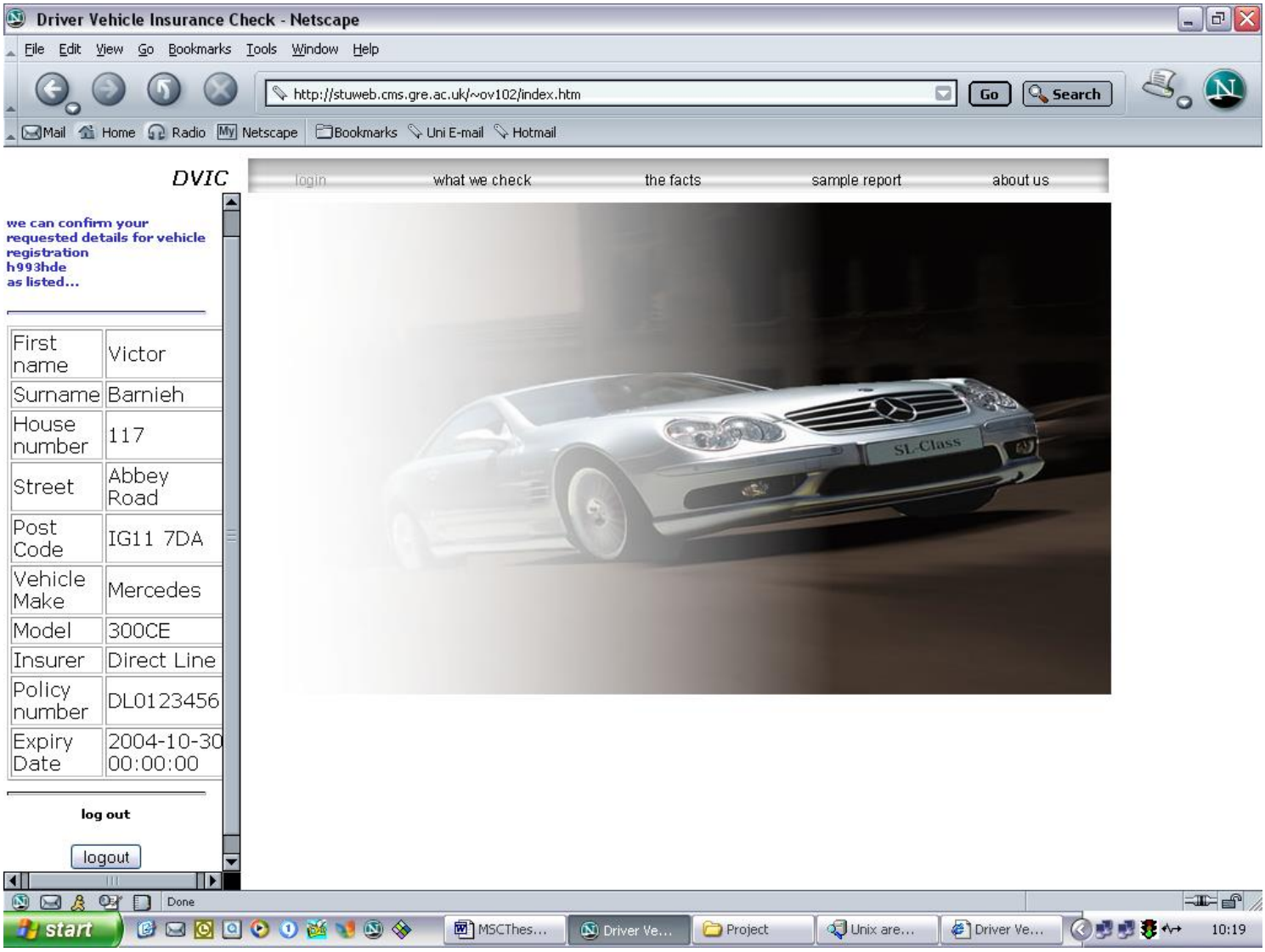

Netscape prompting the user about an invalid email address



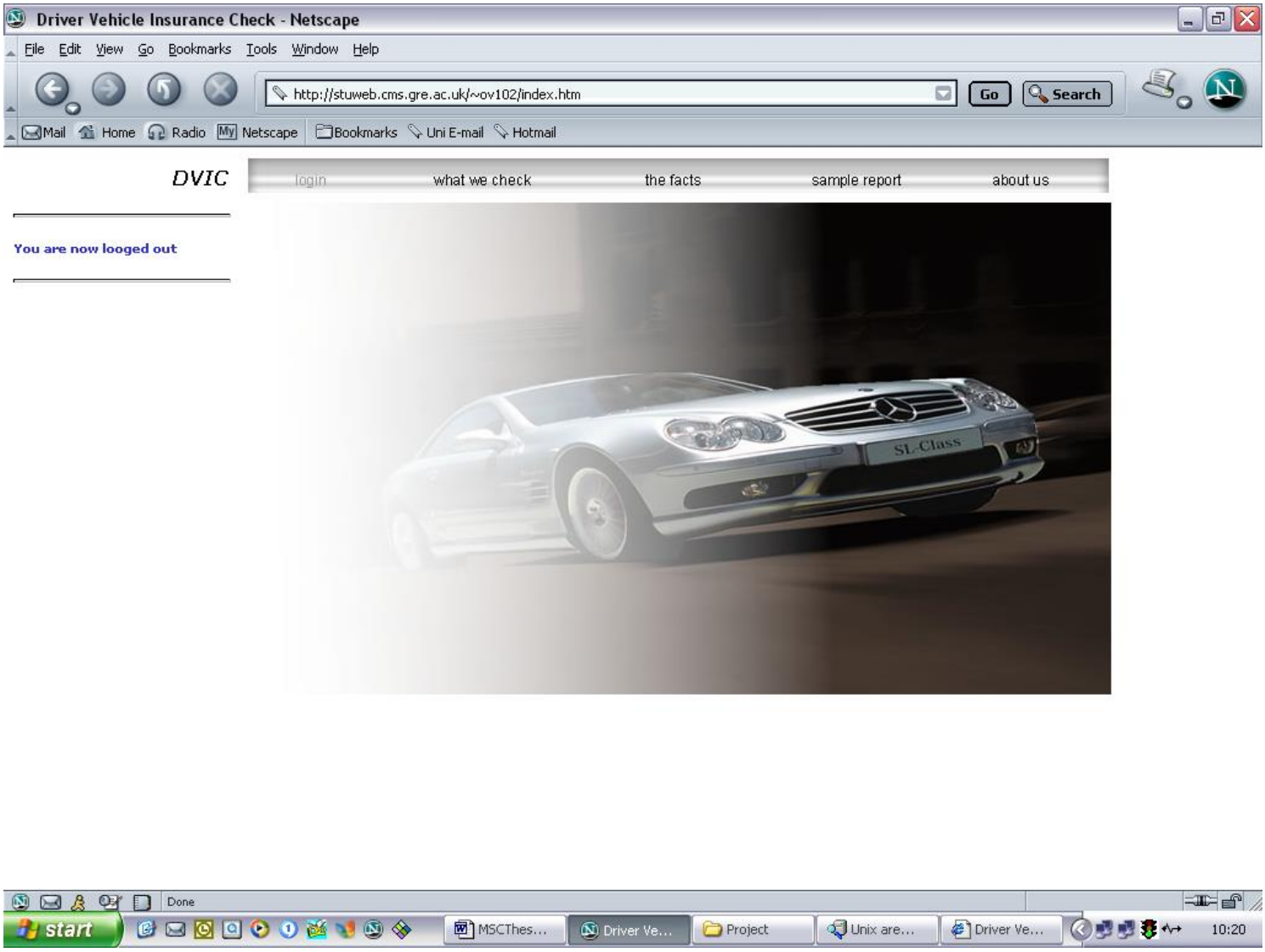

Explorer prompting the user about an invalid email address



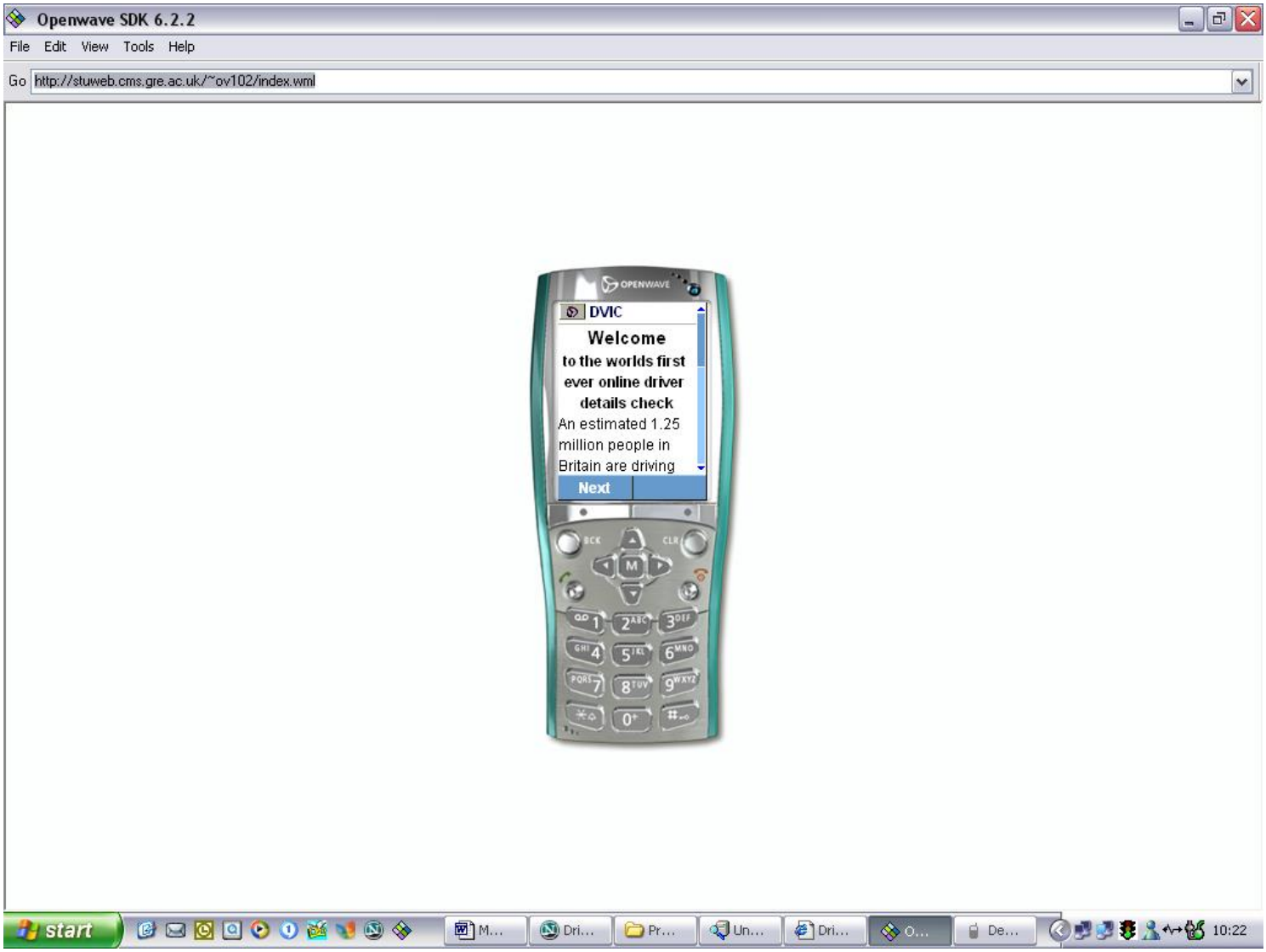

a welcome message after user registration



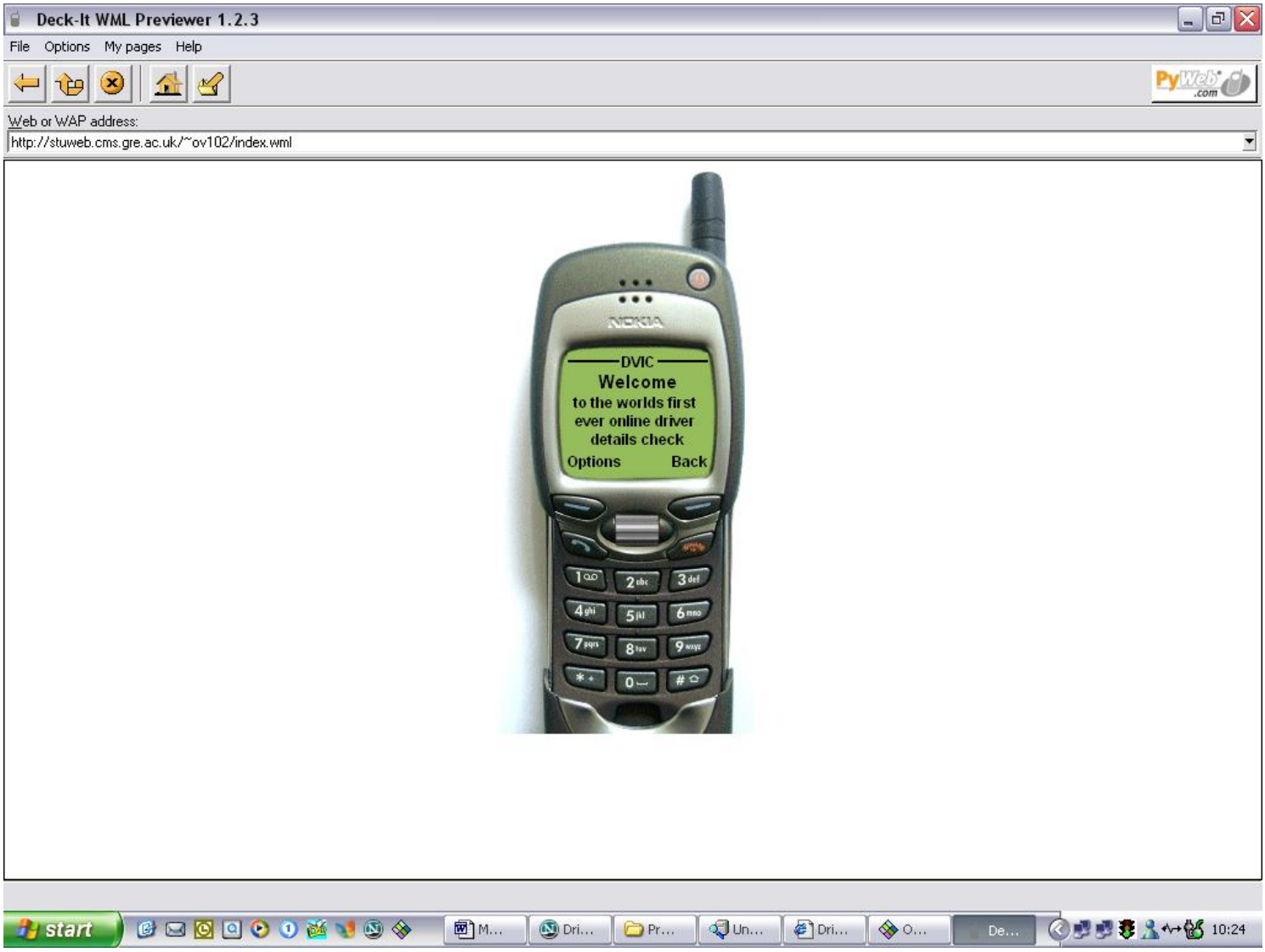

a view of the search results



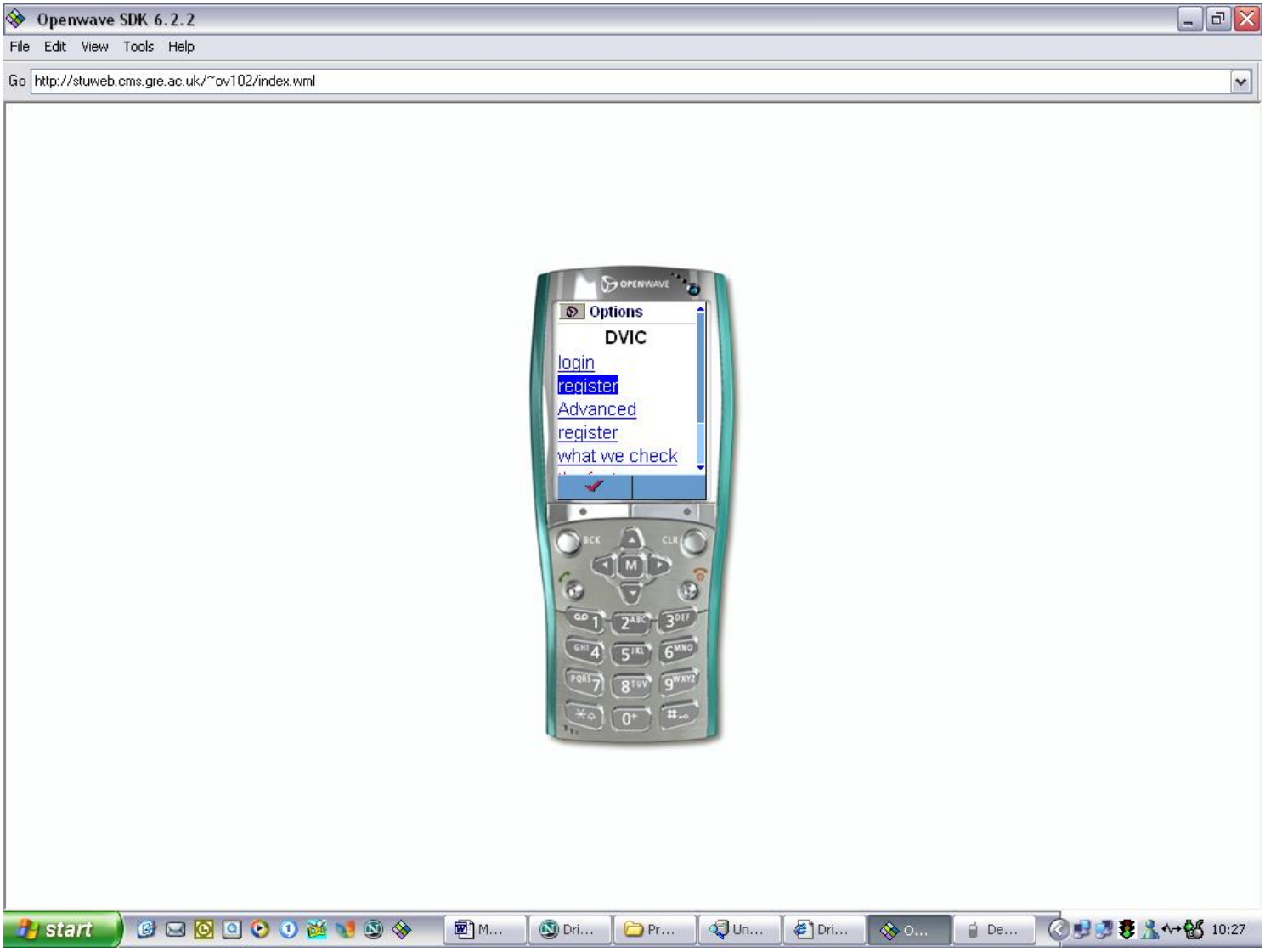

the logout screen



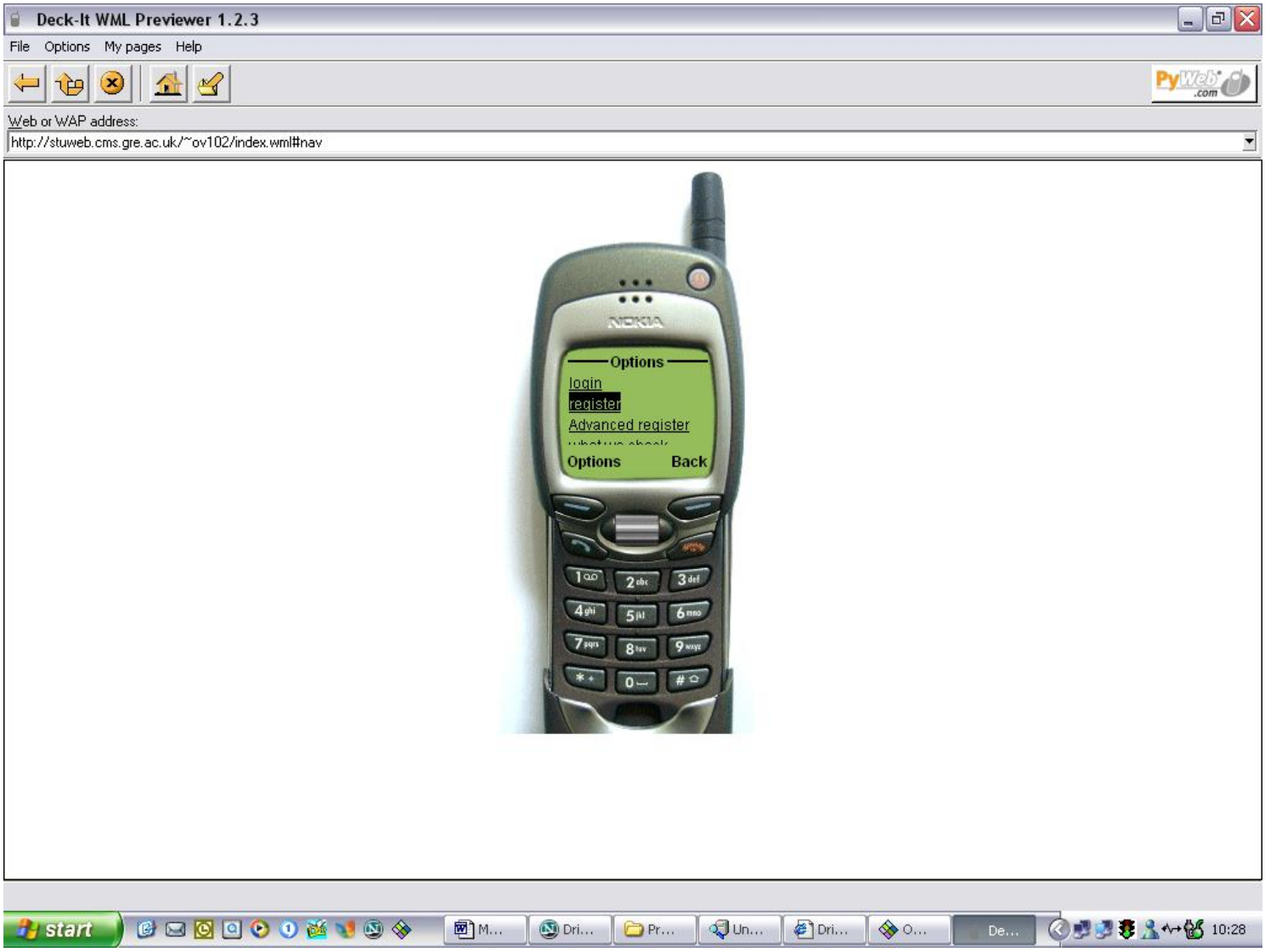

welcome screen as displayed in phone.coms's openwave wap emulator



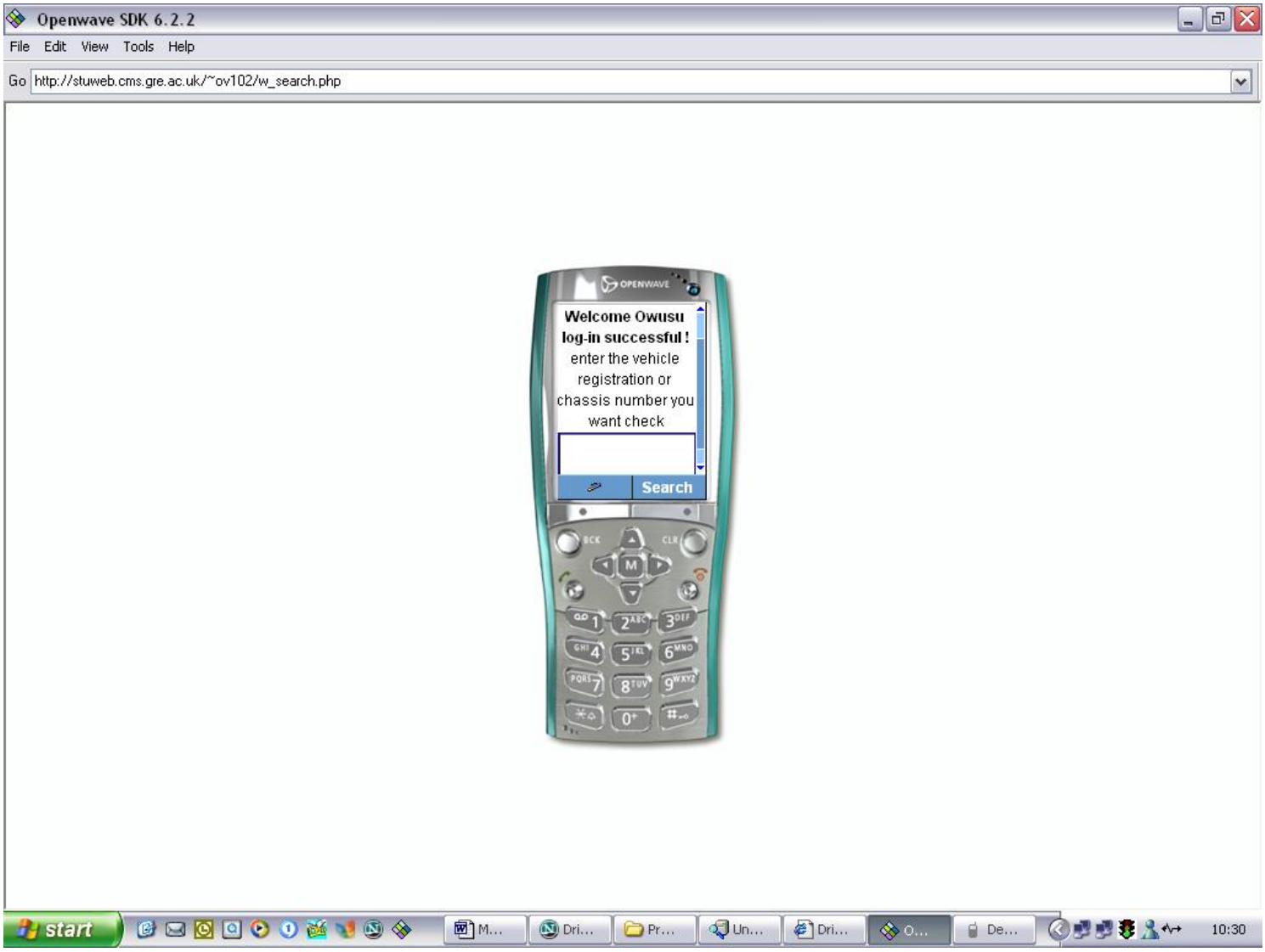

welcome screen as displayed in Deck-It previewer WAP emulator



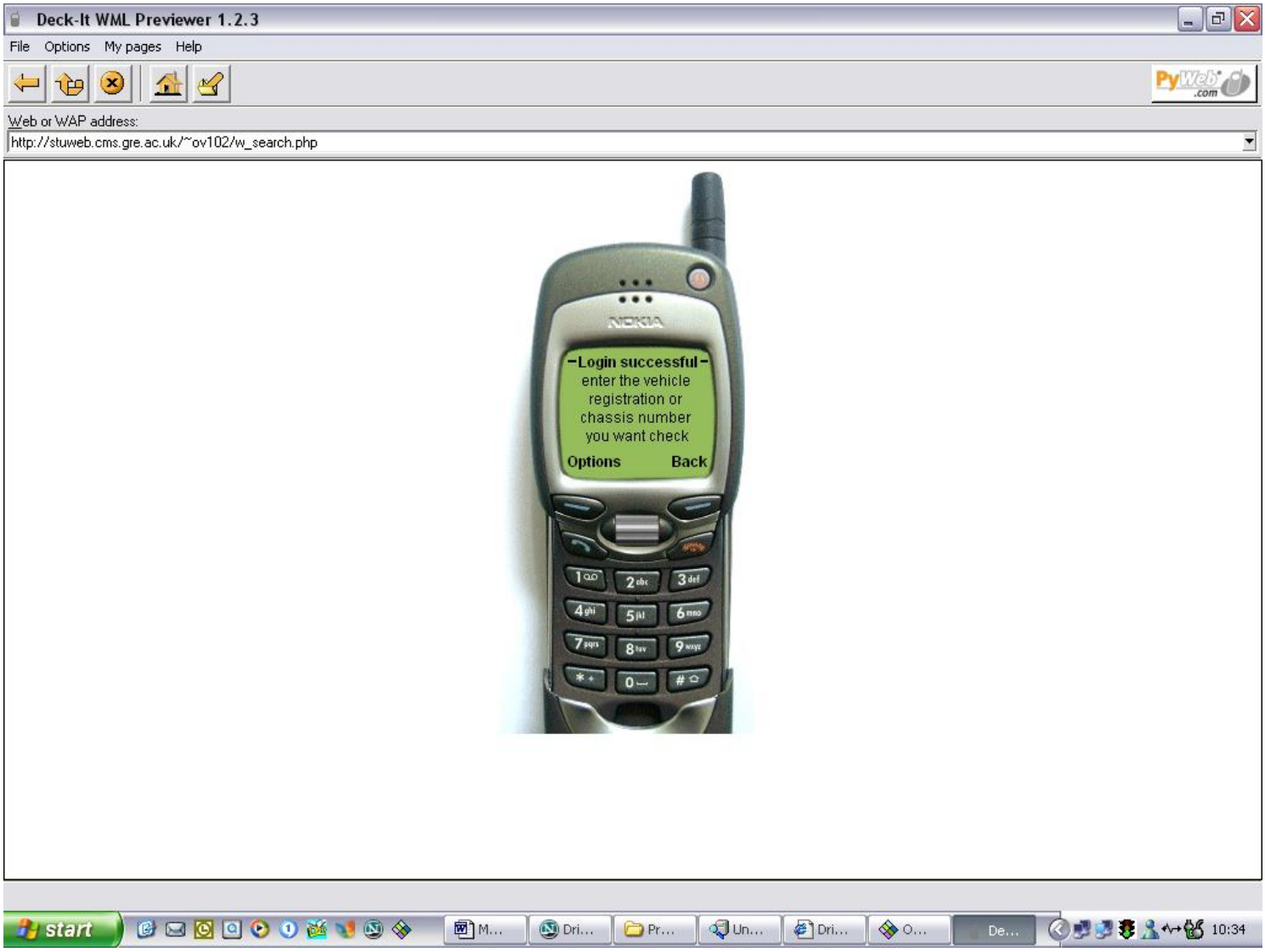

options screen as displayed in phone.coms's openwave WAP emulator



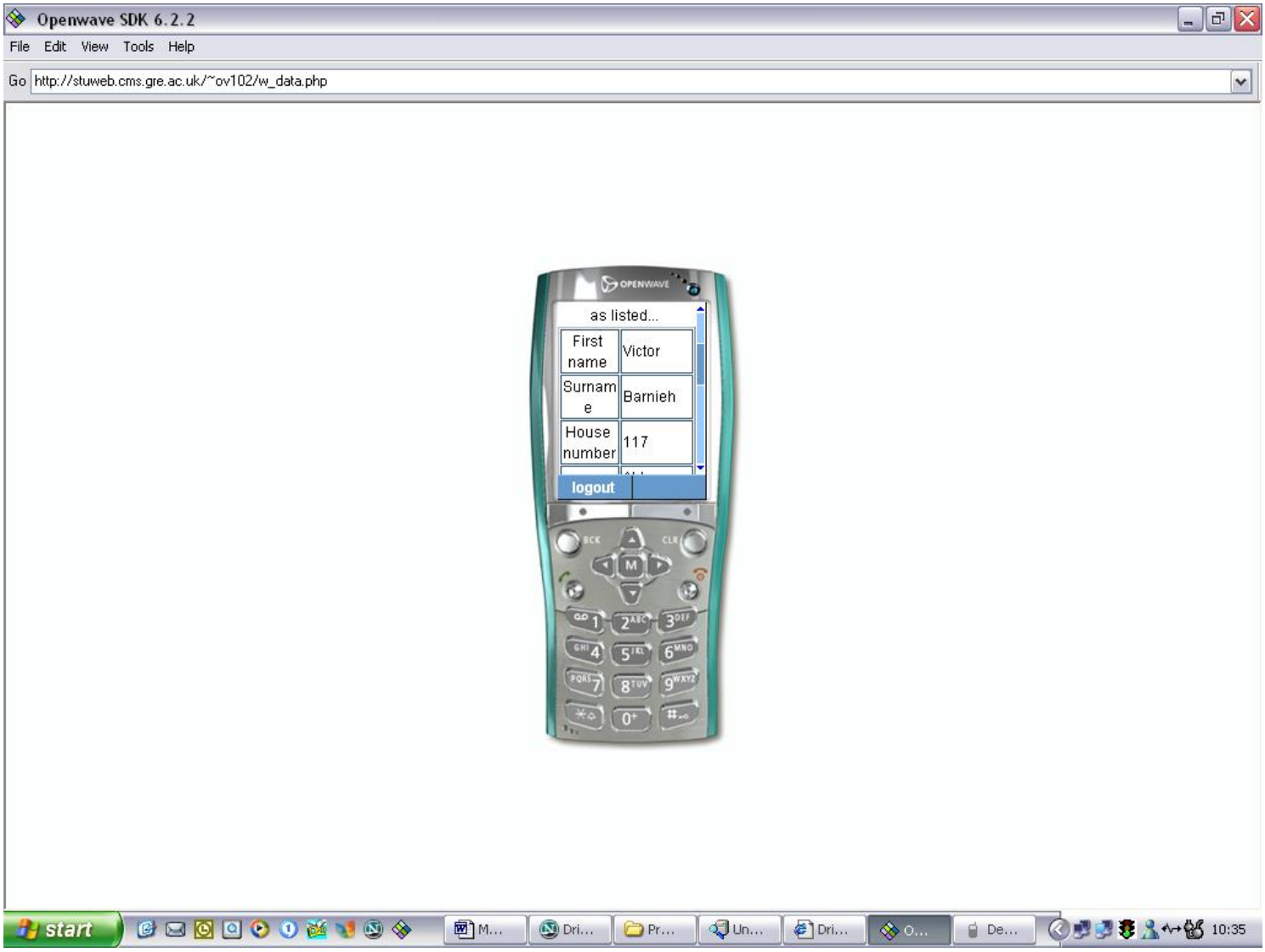

options screen as displayed in Deck-It previewer WAP emulator



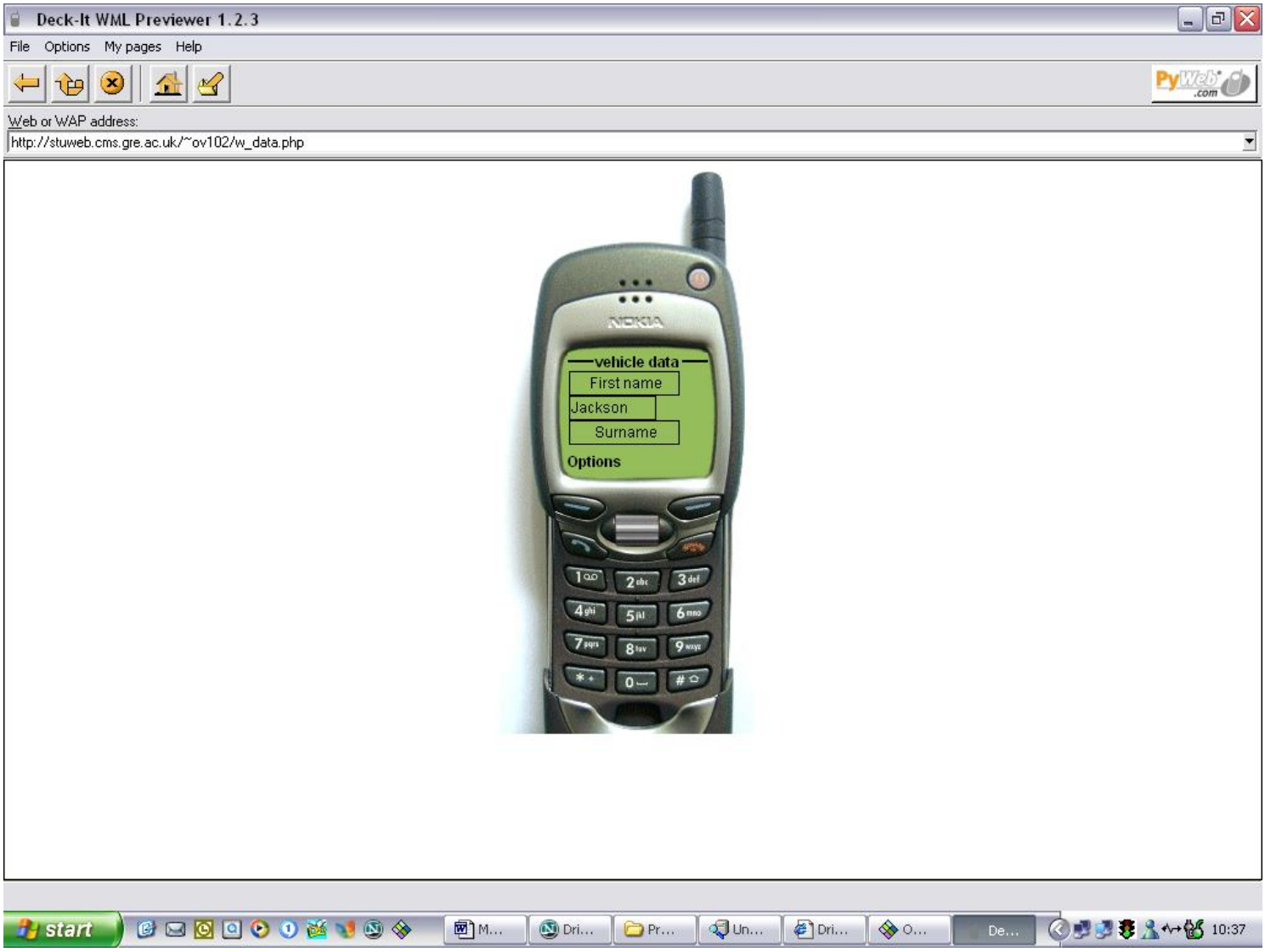

the welcome screen as displayed in phone.coms's openwave WAP emulator after a successful login awaiting vehicle registration or chassis number entry to be search



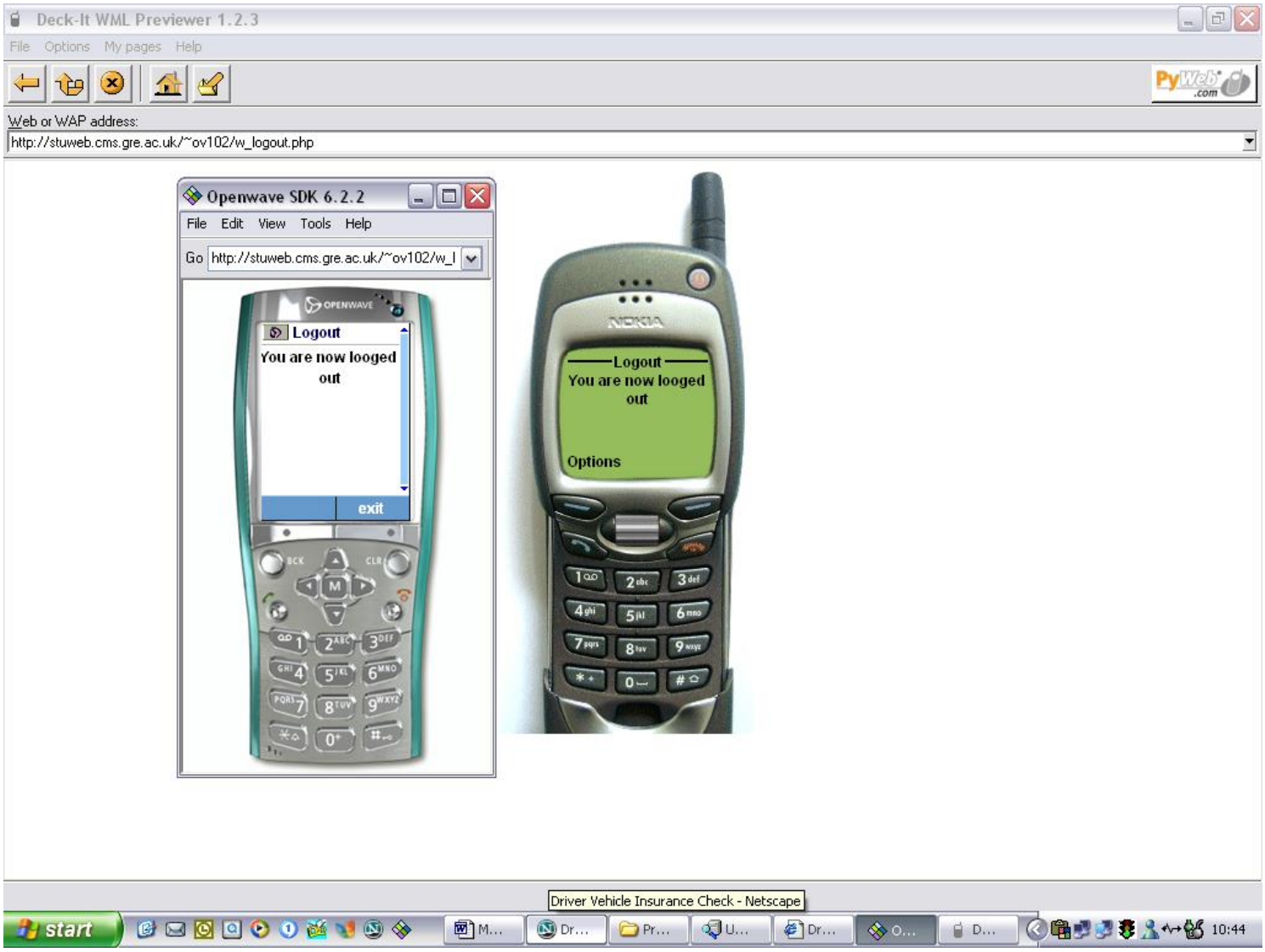

the welcome screen as displayed in Deck-It previewer WAP emulator after a successful login awaiting vehicle registration or chassis number entry to be search




the results of the search as displayed in phone.coms's openwave WAP emulator




the results of the search as displayed in Deck-It previewer WAP emulator




the logout screen as displayed in both

# REFERENCES

## Bibliography


Krithi Aiyappa, *WAP Vs i-Mode: The big fight*, http://www.ciol.com/content/technology/techbytes/100092101.asp

Tom Fitzpatrick, *WAP and i-mode: A Comparison*, http://www.allnetdevices.com/wireless/opinions/2000/09/15/wap_and.html

Tom Fitzpatrick, *WAP vs. i-Mode*, http://www.paralleldevices.com/archive/news/20000918419.asp

Elisa Batista, *WAP or I-Mode: Which Is Better?*, http://www.wirednews.com/news/technology/0,1282,38333,00.html

*Imode - Japanese version of WAP*, http://www.cellular.co.za/imode.htm

Niraj K. Gupta, *i-mode shows the way*, http://www.angelfire.com/nd/ramdinchacha/JUL00.html

Clifford Swift, *Will WAP and i-mode converge?*, http://www.itweb.co.za/sections/columnists/m-people/swift001026.asp

*The World Wide Web Consortium*, http://www.w3c.org

*Your guide to the wireless Internet, wap phones, wap services and PDAs*, http://www.wap.com


## Books


**PHP AND MySQL FOR DYNAMIC WEBSITES**

by Larry Ullman

**PHP and MySQL Web Development, Second Edition**

*by Luke Welling and Laura Thomson*

**PHP Functions Essential Reference**

*by Zak Greant, Graeme Merrall, Torben Wilson, Brett Michlitsch*

**Managing and Using MySQL (2nd Edition)**

*By Randy Jay Yarger et al*

**High Performance MySQL**

*by Jeremy D. Zawodny and Derek J. Balling*

**JavaScript: The Definitive Guide**

*by David Flanagan*

**JavaScript Bible, 4th Edition**

*by Danny Goodman Journals*

**WAP Development with WML and WMLScript**

*by Ben Forta, et al*

**WML & WMLScript: A Beginner's Guide**

*by Kris A. Jamsa*


## Journals


http://www.databasejournal.com/features/mssql/article.php/3087841

http://www.innodb.com/thirdparty.php