Honours Project Final Report

The implementation of mobile web HCI guidelines in the development of Business to Consumer retailing applications for PDAs.

Daffy Duck
2006xxxxx

BSc(Hons) Internet Software Development
BHWS4

Supervisor: Richard Foley
Second Marker: Brian Shields

Submitted for the Degree of BSc Internet Software Development 2007-2008

Except where explicitly stated all work in this report, including the appendices, is my own

Signed:

Acknowledgements

I would like to thank my supervisor, Dr. Richard Foley, for his on-going advice and support throughout course the project. I would also like to thank those who volunteered to participate in my research.

Abstract

The adoption of mobile devices for carrying out momentary transactions has been increasing significantly over the past decade. However, a poorly designed Business to Consumer (B2C) site may drive a potential customer away from one company and straight to a competitor. A prototype was developed using an established set of mobile web development guidelines and usability evaluation was carried out with 8 test participants of with varying amounts of computer experience. The results indicate that the some of the guidelines were applied successfully as participants commented positively on the aspects those guidelines cover. However, several participants said they would be unlikely to use the application as they did not like the device on which it was accessed. The results also showed an interesting trend where the participants with years of computer experience were having more problems than those with fairly little experience.

Contents

1. Introduction

1.1 Background

The past decade has seen a significant increase in the usage of mobile devices by both consumers and businesses. A survey carried out by Varian et al (2002) of 2,000 American businesses showed accumulated revenues of $443.9 billion between 1998 and 2001, which is expected to rise to $1551.9 billion by 2010 (also see United Nations, 2003). The past decade has also seen a significant increase in the sale of mobile devices. Figures provided by eTForecasts (2003) indicate a massive jump in Personal Data Assistant (PDA) sales between 1995 and 2000, but those sales have been surpassed by smart phones and convergent devices in recent years.

A significant number of these devices have wireless internet capabilities as shown by Marketing Week (2007), with 5.7 million Britons accessing the web by a mobile device in January 2007. These figures clearly show an increasingly large potential market for carrying out mobile commerce (M-commerce). M-commerce, as defined by Dholakia et al (2006), refers to monetary transactions conducted via a mobile telecommunications network using a mobile device.

A fundamental problem with browsing the web on a mobile device is that most websites are designed for usage on a desktop computer (Zhang, 2007). If a website has not got a mobile version then the mobile device user will be directed to the exact same site that is intended for desktop users. This may not be a problem depending on how the site was designed and how the device interprets the site but much of the time there will be issues. These sites usually will not scale well down to the small size of the screen and can cause difficulty in navigation; this applies particularly to websites which rely upon HTML tables for structure (Tarasewich, 2003). Some browsers will display the site in exactly the same form as it would have been on a desktop which will result in the user having to do a lot of vertical and horizontal scrolling. This can become very disorientating and the user will not be able to grasp the overall picture as they would have on a desktop (Buyukkokten et al, 2000), which may frustrate them to the point where they decide to just leave it and do business with a competitor instead.

There are numerous companies following the Business to Consumer (B2C) model that have mobile versions of their sites. A significant issue with many of them, indicated by Brown (2007), is that they lack the functionality of their desktop counterpart. The sites allowed the user to browse the products or services on offer but did not allow them to purchase. This resulted in lost revenue and customer dissatisfaction.

A field study carried out by Ramsay and Nielsen (2000) indicates that the devices themselves are not causing the usability issues as the users had no problem using them. The usability issues predominantly came about as a result of poor website design which was demonstrated when a user was able to pull the same information from one website quicker than another, giving the clear indication that there is a lack of standards between mobile websites in regards to interface design as designers are clearly doing different things. This project aims to compile a set of guidelines to be used for mobile application development, develop a mobile application using the guidelines and carry out a usability test to gauge the effectiveness of the development.

1.2 Project Outline and Research Question

This will be a "develop and test" type project. An application will be developed using mobile web usability guidelines and a usability test will be carried out to evaluate the success of the development. There are a wide variety of mobile devices available on the market but it is unfeasible for this project to cover them all. This project will focus on applications specifically for PDAs.

Although sales of PDAs have been declining for some time (IDC, 2007), several increasingly popular convergent devices, such as the Apple iPhone, share significant similarities with PDAs in terms of their user interfaces. Fitzsimmons (2007) predicts that devices like these will influence the design of future devices, meaning that the usability issues that exist with PDAs today would continue to exist in the future. The research question will attempt to address these issues, the question is as follows:

- To what extent will the application of mobile web HCI guidelines to a B2C retail application have an effect in the creation of an easily accessible and usable mobile equivalent for a PDA?

Before development can begin, a number of sub-topics must be addressed. This project is focussed primarily on the user interface. However, it is important to know what functionality a business to consumer application requires as the user interface must be designed to accommodate these requirements. Therefore, the first sub-topic to be covered in the literature review is as follows:

- What are the requirements of a Business to Consumer retailing application?

It will be necessary to gain an understanding of usability issues that affect both desktop web applications and mobile web applications. This will help in the development of a solution that takes into account both general usability issues as well as ones regarding mobile devices which should result in a better application**.** The second sub-topic to be covered in the literature review is as follows:

- What are the primary usability issues?

Mobile devices are inherently different from desktop computers and the vast the majority of the time they are less capable (hardware wise) in every respect besides portability. It will be necessary to gain knowledge about the constraints so that an application can be developed that will run effectively within these constraints. The third sub-topic to be covered in the literature is as follows:

- What are the constraints of developing applications aimed at a mobile platform?

While desktop and mobile applications share some usability issues, mobile applications face a myriad of additional issues which could arise from a number of factors including environment, attention levels and location (Tarasewich, 2003). These issues will need to be identified before application development commences. The fourth and final sub-topic to be covered in the literature is as follows:

- What are the specific guidelines within the usability issues that pertain to mobile application development?

The hypothesis is that using the mobile web development guidelines when creating the site should result in a site fulfils its requirements as a business to consumer site and is highly usable.

1.3 Report structure

1.3.1 Literature Review

The purpose of the literature review was to identify the requirements of a business to consumer application so that it was possible to check whether the application was successful in fulfilling its purpose as a business to consumer application. General usability issues pertaining to websites were identified under 4 main headings. Finally, the constraints of application development on a mobile device were investigated and the guidelines designed to assist the development of applications which work within these constraints were identified.

1.3.2 Problem and Systems Analysis

Using the information gathered from the literature review, a web application (which meets all the requirements of a business to consumer application based on the information gathered) was chosen to model the prototype upon. Using mobile web development guidelines and heuristics, the web application was analysed in order to find out which features of the site could be cut out from the mobile version. From this analysis, a set of functional and non-functional requirements were derived that the prototype had to meet.

1.3.3 Design and Implementation

The purpose of this section is to present the design decisions and their justification based on the functional and non-functional requirements derived from the systems analysis. In addition to this guidelines and heuristics used to implement those requirements are detailed with examples of where their effect can be seen in the application. The justification of the technology used is also in this section. The reasoning behind carrying out a Develop & Test type project was discussed and the data capture methods will be outlined and identified.

1.3.4 Testing and Evaluation

This section contains the development of the test procedures, their usage, and presentation of the data from the data gathering methods as well as some brief conclusions from the results presented.

1.3.5 Conclusions

This section contains an overview of the project as well as detailed conclusions and a word about potential future work in this area. The results were largely expected but there was a surprising trend in which the many of the participants with lots of experience with computers were having usability issues that the less experienced folk were seemingly unaffected by.

2. Literature Review

The first objective of this literature review was to identify the requirements of a B2C web application in order to derive a set of functional requirements for the prototype. The second objective was to identify the usability issues that affect web applications. The last two objectives were to identify the constraints of a mobile device and the guidelines that help make a usable website while working within those constraints.

2.1 Requirements of a Business to Consumer application

Each B2C application, regardless of the business for which it is being used, has the same basic requirements which will facilitate and protect both parties involved in a transaction. In order to test the hypothesis that the mobile web application can meet the requirements of a B2C web application using the guidelines, it was necessary to find out what those requirements are and gain an understanding their of purpose.

2.1.1 Security, authentication and payment systems

When beginning a monetary transaction, Van Thanh (2000, cited in Wang et al (2005)) indicates that both of the parties involved in the transaction must be authenticate each other to ensure that each party is who they claim to be. To facilitate this, encryption may be used to create a channel for both parties to communicate securely. Van Thanh (2000) also indicates that a user friendly payment should also be implemented to ensure each transaction is easily carried out.

Authenticating the user of a website is usually done by means of a username and password. A B2C application will inevitably have to capture the user's details at some point in order to complete a transaction. This may, for example, include the user's name, address and payment details. These details are usually captured by having the user complete a registration process. It is best to keep the amount of information required from the user to a minimum as Nielsen (1999) states that it takes the user's focus away from what they want to do.

2.1.2 Adaptability

Adaptability is a key requirement for e-commerce (Choren et al, 2005). The concept of adaptability covers the autonomous software that can deal with many different users simultaneously and in the same way. The information displayed on the web page is bespoke to the user that requested it as its content is influenced by the actions that user carries out while on the site.

2.1.3 User Interface

Olsen & Malizia (2005) state that the user interface ought to be simple but should give the user some flexibility. Having a good user interface is important as a poor one can result in users becoming frustrated and simply giving up. Najjar (2003) indicated that up to 50% may have been lost because the customer could not find what they wanted. There have been many examples of how much improvement in business, such as DEC's increased revenue of 80% (Wixon & Jones, 1995) or Dell increasing its

average daily online sales by $33 million. These two examples give a clear indication of the benefits of having a usable site and why it is so important to invest some resources into creating one.

2.2 Primary usability issues

The first objective of this literature review was to identify the requirements of a B2C web application in order to derive a set of functional requirements for the prototype. The second objective was to investigate the factors that effected usability. The last two objectives were to identify the constraints of application development and specific guidelines for development on a mobile device.

To identify the factors that affect usability, usability must be defined. The ISO 9241 definition of usability is the:

"effectiveness, efficiency and satisfaction with which a specified set of users can achieve a specified set of tasks in a particular environment."

For the purposes of this report, the definitions of effectiveness, efficiency and satisfaction provided by Koivunen & May (2002) will be used. Effectiveness is the accuracy and completeness with which specified users can achieve specified goals in particular environments. Efficiency is the amount resources expended in relation to the accuracy and completeness of goals achieved. Satisfaction is the comfort and acceptability of the system to its users.

2.2.1 Speed

Speed is a crucial issue as it determines the acceptance and therefore adoption rate (King, 2003 p5); if the web application fails to load within a reasonable amount of time then the user may simply go to another web page. King (2003, p3) states that a user is likely to wait approximately eight seconds before going elsewhere. A number of factors can cause a web site to slow down, a large one being unneeded or un-optimised content.

The user should also be able to allow the user to quickly find the information they are looking for, with Tracy (2002) stating that the user should be able to find any product within 5 clicks. This should improve efficiency as the user does not have to go through a lot of effort to find what they want. Webcredible (2007) back this up by recommending that contextual data be provided to the user based on location. This could be practical if the user had a data connection to a cellular or wifi network. The former can find information regarding what the user is interested in near the transmission tower and wifi could be used to find information regarding user interests near the origin of the wifi hotspot.

2.2.2 Structure & Navigation

Ramsay and Neilsen (2000) indicate that the successful implementation of a navigation system is crucial to the success of the system. Behind slow response times, Najjar (1998) indicates that navigation issues are the biggest problems which cause people to leave a site.

Several points are made, being that mobile systems share the same basic navigation. On a basic level, the user navigates through a hierarchy until they find what they are looking for. Either that or they can use a search facility. Once the user is at the desired

page they can then bookmark it. However, Longoria (2004, p29) reports that having a hierarchy too deep can frustrate the user and discourage them from re-using the application. Studies by Kaikonnen & Roto (2003) indicate the users would prefer to have more information on the one page and scroll as opposed to having to jump between many smaller pages that have all their information on the one page.

Webcredible (2007) recommend placing a back button on each page as mobile devices may not have one. The target device, a PDA, does have a back button built into its default web browser so implementation of basic navigation controls such as that were not deemed necessary.

2.2.3 Consistency

Having a consistent user interface means that the user can retain the skills that they have learned as well as reducing the time it takes to use the site in system in general (Newman & Lamming, 1995). This is improves the three aspects of usability identified earlier (effectiveness, efficiency and satisfaction) as the user will have a better idea of what to do, they will be able to carry a task out quicker as a result and will probably become more satisfied as it requires a smaller amount of effort. The Cascading Style Sheets (CSS) technology is a method for creating a consistent interface (Germonprez et al, 2006) as the device will interpret the CSS in its own way to produce what the designer intended on the small screen. A layout may be exactly the same on a small and large screen using the same code.

2.2.4 Relevance

Relevance in this case refers to the actions required of the user and the information they need to provide or are provided with.

Kaikonnen & Roto (2003) indicate that a login should only be implemented in sections of the site that deal with confidential information. This concept of delaying the login process is backed up Nielsen (1999) who indicates that the customer may simply leave the site if you ask them too early on before gaining their interest. At an early stage the user may not see the value of signing up to the site.

Nielsen (2005) recommends that only relevant information be displayed as more information reduces the visibility of what is important. This would mean cutting down all the information provided to only what the user requires to know in order to make the booking or purchase.

2.3 Constraints of development on a mobile platform

In order to make a mobile device more portable it is necessary to use fewer resources (Lee & Benbasat, 2004) which will result largely in degraded performance in comparison with a desktop computer. It is therefore important to design an application with these performance limitations in mind.

2.3.1 Cost

The cost of setting up a wireless infrastructure can be significant, with numerous European contractors bidding 46.2 billion US dollars in Germany for 3G licenses. This cost of this venture may then be passed on to the users (Brewin and Sayer, 2000, cited in Tarasewich, 2002). This can incur a large bill for the mobile device user if they use up a lot of bandwidth when accessing the internet via a wireless telecommunications network. Zoller (2007) indicates that, while mobile device users appear to be willing to pay for premium content such as ringtones and games, there is no indication as of yet whether they will be willing to pay high amounts to access a web sites. With the advent of mobile devices with built in wi-fi capabilities, this could change and perhaps more websites will become more content rich.

2.3.2 Screen size

PDAs generally have lower resolution screens than desktops which mean the size of items on the screen will be larger, typically 40% (Karkkainen and Laarni, 2002). Since the items are larger but the screen is smaller it is important to take into account the factors that can have an effect on the displayed items. Longoria's (2004, p114) principle of "every pixel counts" applies. Here, it is recommend that drop-down lists should be used to save space. This could be applied to cases where the user is given a choice as opposed to radio buttons for example.

Karkkainen and Laarni (2002) also recommend avoiding the use of images unless necessary. Depending on what is being sold on the site an image may be beneficial but may cause navigation and bandwidth issues. The benefit to cost ratio depends largely on what is being sold, the cost and speed of bandwidth and the device on which the site is being viewed.

2.3.3 Input

Mobile devices require different input mechanisms than their desktop counter parts as they do not use a full-sized keyboard and mouse. For a PDA a stylus is generally used to select objects on screen as well as manipulating a virtual keyboard. PDAs sometimes have handwriting recognition but Kärkkäinen & Laarni (2002) indicate this is usually inaccurate and a virtual keyboard is a better text input mechanism for PDAs with a stylus. Text input can be a fairly long process however so one should aim to reduce aim to reduce the amount of text input required by providing selections for the user (Longoria, 2004, p128) as opposed to the user having to type them out.

2.4 Specific guidelines within the issues relevant for mobile development

A simple user interface is vital as the user should be able use it easily at any time. As mobile devices as generally used for brief and routine tasks it is important to either hide or remove less important functions that could hinder these tasks; if the user cannot learn the interface within a few minutes they are likely to not use the system (Shneiderman & Plaisant, 2005, p305 & 393).

2.4.1 Text Size

A guideline set out by Kärkkäinen & Laarni (2002) indicates that the minimum text size used for websites aimed for PDAs should be 14pts due to variable lighting conditions as the device may be used in different contexts.

2.4.2 Buttons and links

Links should be used when the user is being taken to another page and buttons should be used if an action is being carried out (Cremin et al. 2007). The text of these links and buttons should be non-jargon terms that the user is familiar with (Macaulay, 1995, p52).

2.4.3 User input

Webcredible (2007) recommends that the user be given selections wherever possible so that they are not given the ability to cause an error. In this case a drop-down list could be used so that the user need not type out selections.

2.4.4 Images

As a general rule, Kärkkäinen & Laarni (2002) recommend that the pages be designed for a 56k modem but this applies especially to images. Images take up a lot of bandwidth which can cause a slow down and provide no benefit

2.4.5 Automate certain functions

Kärkkäinen & Laarni (2002) recommend that the website should automate any functions that it can. This may include tasks such as making calculations and working out prices. Doing this should improve the overall efficiency as the user does not have to put in additional effort into achieving their goal.

2.4.6 Overall consistency

CSS can be used to provide sections that span across the full width of the screen (Germonprez et al, 2006) which will provide as a consistent reminder to the user that they are still on the site (Kärkkäinen & Laarni, 2002). Having everything in the same place wherever possible means that the user will be able to recognise those elements quickly and spend little to no time learning the interface.

3. Problem and System Analysis

This section is an analysis of an existing B2C application which was used to base the prototype upon. Many sections of the B2C application are not incorporated into the application as they cannot be justified according to the guidelines identified in the literature review.

3.1 Systems analysis and requirements creation

Based on the literature review, it is necessary to implement a mobile web application that fulfils the requirements of B2C application taking into account the usability issues and the constraints of a mobile device. Coming up with an idea for an B2C e-commerce application was beyond the scope of this project so it was decided to model the prototype on an existing web application and adapt it for mobile use. The web application chosen to base the mobile web application on was lastminute.com. This site is a typical example of a B2C application. The user has the options to browse and search for various products (mostly travel related in this case), add products to a basket, create an account and complete a purchase.

For the mobile application being developed, Longoria's (2004) heuristic of feature shredding was applied. Longoria recommends that the system creator should not try to cram a desktop application into a mobile device but instead 'shred' some features depending on the form factor of the device. As a PDA has a fairly large form factor in comparison to other mobile devices it is not necessary to shred as many features from a web application targeted at a PDA as one may do so for a mobile phone for example.

For the prototype being developed, the product search was restricted only to flights and hotels. The other categories, cars and entertainment, have the user find bookings in the same way as the flight and hotel categories, so it was not necessary to have these features implemented as they would contribute nothing extra besides making the coding and testing processes longer. The first functional requirement is therefore to provide to the user the ability to look for two categories of item.

The search capability at the top of the page was not implemented as Webcredible (2007) recommend the user is given the option to browse wherever possible as opposed to searching on a mobile application wherever possible. Having the user browse as opposed to searching means the scope for errors is narrowed. The search box says "search site" which is fairly vague. It does not give an indication what will be searched for so implementing it may have caused problems, which goes against Nielsen's heuristic of Error Prevention, preventing errors before they happen.

The product selection process is inconsistent between the various sections. For example, under "Flights + Hotels" and "Holidays" the user can select both the destination and place of departure from a drop down box, yet if the user goes to "Flight only" or "Hotel only" then have to type the destination into a text box. This goes against two of Nielsen's (2005) heuristics, preventing errors before they happen and inconsistency. The user is given the ability to enter a destination or airport that is not available to them on the site, which would result in an error if they type in

somewhere not available, whereas they are given a selection on earlier pages. The first non-functional requirement is therefore to only allow the user to browse for what is available.

The focus of this project is usability. However, security, authentication and payment mechanisms have such a large impact on usability that they must be included in the prototype if it is to accurately represent a business to consumer web application. Programming all of these systems is beyond the scope of the project so it will be necessary to simulate them in order to include them in the usability test. The next functional requirements are therefore to simulate payment, authentication and security systems.

The B2C site the application is modeled on provides a checkout facility, much like many other B2C sites so the prototype will need to include provide that functionality. An order confirmation screen will be required as well, Ragus (2000) indicates that it should include the order number and instructions on how to cancel the order. Najjar (2003) indicates that, in addition to cancelling orders, the user should be to alter their own details.

Using contextual information in this case may be useful to check what hotels are nearby or what flights are available from the airport you happen to be in. However, technically this is beyond the scope of the project.

3.2 Functional requirements:

1. Capture customer information
2. Simulate security features
3. Automatically adapt to user actions and inputs
4. Basket system
5. Checkout system
6. Order confirmation
7. Order cancellation.
8. Self-user details alteration.

3.3 Non-functional requirements:

1. System for browsing selections as opposed to having to search
2. Consistent system throughout for browsing for items.

4. Design and Implementation

This section contains a justification as to why the research methodology used was chosen as well as providing in detail justification of technology choices and design issues in regards to security, authentication and payment systems. Each of the guidelines identified have been implemented in the application with screenshots to show where they have an effect.

4.1 Primary Research Method

The research methodology chosen for this project was "Develop and Test". It was felt this was the best method as it gives the ability to create a test that can tackle specific issues and have specific guidelines applied to it. An experiment may have required seeking out a site that takes all of the issues researched in this report into account. In addition, it would have be designed in accordance with the guidelines identified, such a site may not exist. A survey would have probably not have been useful as response rates tend to be low and the opinions on what constitutes a good list of guidelines would vary widely.

A case study of a company who have already created or overhauled a mobile version of their site may have been useful as it would give a real world indication as to what worked and what did not. The problem is finding a company who would be willing to share details on how their website became successful, or how badly it failed.

4.2 Development of prototype

4.2.1 Technology

The web site will be developed using the Microsoft ASP.NET (Active Server Pages.NET) framework. This framework was chosen for a number of reasons. ASP.NET is a framework designed for creating dynamic websites. This makes it ideal for this scenario (and e-commerce in general) as it fulfils the adaptability requirement of B2C applications.

A comparison study between the Java and .NET frameworks carried out by Wesson & Van Der Waalt (2005) indicated that, while the Java framework processed data faster and used less network bandwidth overall; the .NET framework was favoured by the test participants, possibly due to the better graphical user interface (GUI). The .NET framework is supported by a powerful IDE (Visual Studio) which also provides drag-and-drop interface creation functionality. This should enable the rapid development in addition to making the interface easily modifiable should the need arise as a result of user feedback.

Dynamic pages do not sit well with Longoria's (2004) heuristic of "every round trip counts". If the user is doing a lot of browsing or making alterations to their order the server requests are going to add up, which takes bandwidth and can cost money. However, this can be helped by making the purpose of links and buttons plainly obvious as per the guideline set out by Cremin et al (2007). This has the added benefit

of reducing the chance of ‘click-through disappointment’ as the user is more likely to know what to expect.

4.2.2 Security, authentication and payment systems

The development of security, authentication and payment systems were beyond the scope of this project. However, the presence of those systems on B2C applications has an influence on the interface so they had need to be simulated.

Security and authentication were simulated by means of implementing input validation on various web controls (such as length of credit card numbers) to make the application appear to be more realistic. The same applies to the payment system. The mobile web application will be linked to a Microsoft Access database to store details including:

- User account details
- Flight details
- Hotel details
- Bookings

This is so that the user can look up flight or hotel bookings when carrying out the usability test. This database also allows users who have an account authenticate and log on.

When a user logs on their ID number is written to a cookie stored on their computer. The web application looks for this cookie every time a page is loaded. If the value is zero the web application shows no one as logged on. If the value is something other than zero the system will match that user ID number with the user’s details on the database and show that user as logged in. This is, technically, a fairly easy system to circumvent as the user could simply open up the cookie in a text editor and change the user ID to another. This means that next time they visit the site they would be logged on as the user whose details match the ID in the cookie.

While the site does provide security in the form of user authentication, as stipulated by the B2C requirements, this method of security is too weak for a real-world B2C web application. However, there is no way for the user to know this just by looking at the website so for the purposes of the usability test this weakness can be overlooked.

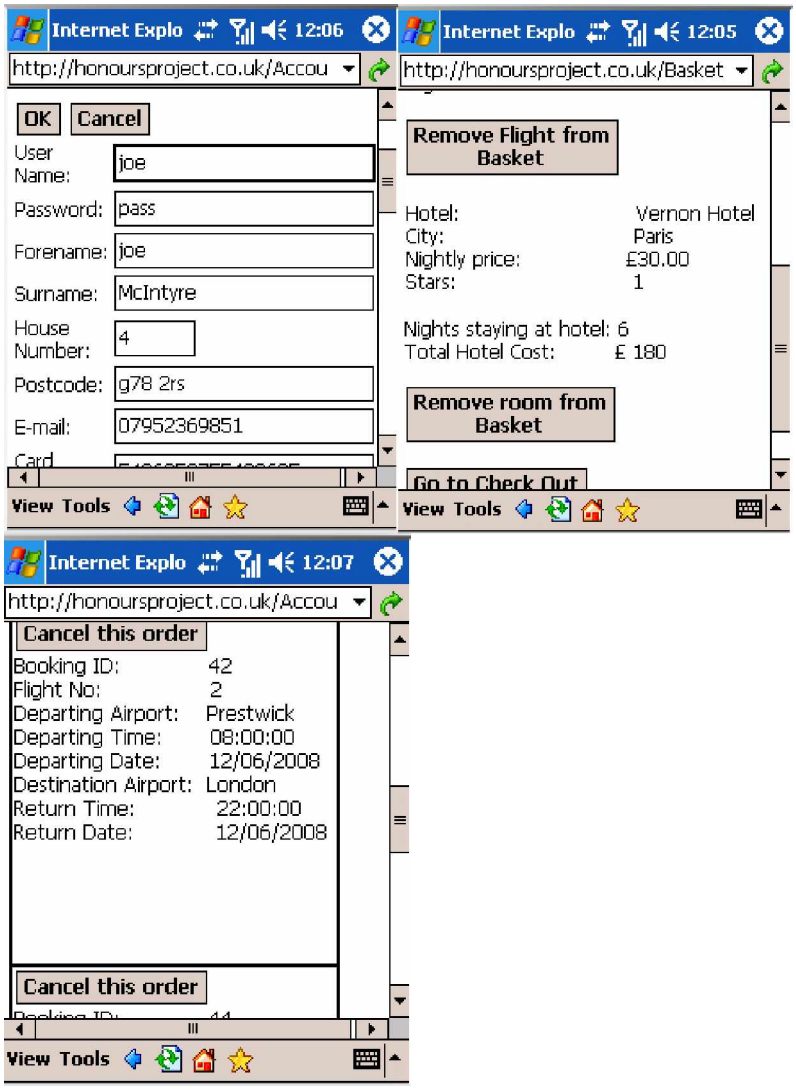
4.2.3 Heuristic testing

Throughout the web application development process general usability heuristics proposed by Nielsen, Appendix A, and mobile web development heuristics by Longoria, Appendix B were referred to in order to reduce the number of usability issues. This is being carried out to supplement the usability evaluation as Nielsen (1993, p156) states that one evaluator is likely to pick up only 35% of usability problems. However, carrying out a heuristic evaluation means that there will be less usability issues when test participants become involved (Nielsen 1993, p226). This means that the test participants will not get stuck on usability problems that the developer is already aware of and has already rectified prior to the usability test

4.3 Development of prototype – Implementation of guidelines

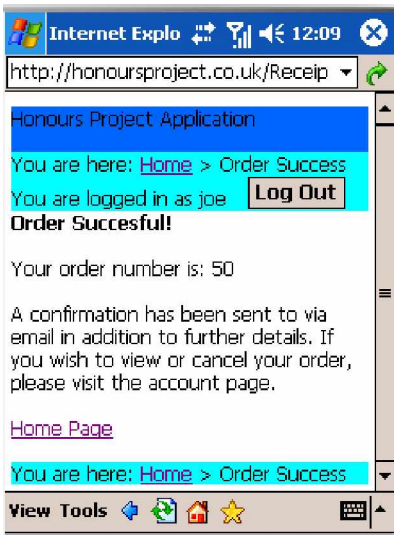
4.3.1 Cancel or edit details

The implementation of the guideline for editing details or cancelling orders is shown below. The first screenshot shows the user account screen. The user alters their details in the text box and clicks OK then their updated details are send to the database. The second screenshot shows the basket, if the user makes a mistake or simply changes their mind they are able to remove items from their basket. The last screenshot shows the cancellation functionality. In the account screen, under the heading of bookings, the user can view all of their bookings and cancel the ones they wish.



4.3.2 Don’t repeat navigation

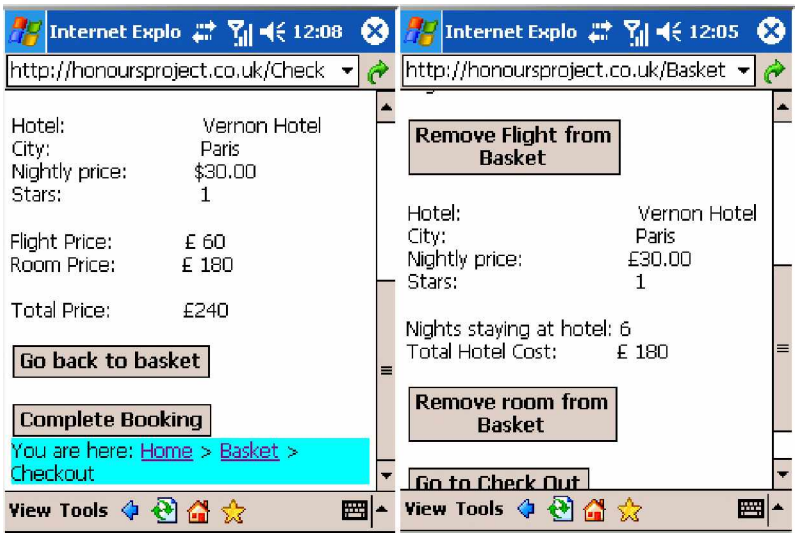
Here, the Webcredible (2007) recommendation of not repeating navigation elements can be seen. Since there are no repeated navigation elements, everything is kept on the one screen so the user will have a better picture. The recommended breadcrumb feature can be seen at the top and bottom of the page, giving the user the option go to the pages leading up to the current page as well as providing a “root” page so that the user can always get back to the home page.



4.3.3 Automation of site functions

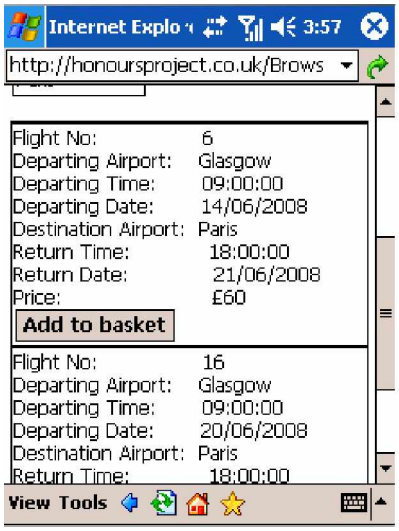
Kärkkäinen & Laarni's (2002) guideline of automation has been implemented on the basket and checkout page. On the basket page, the application automatically calculates how many nights the user will be staying at the hotel based on the flight that they have chosen. It then calculates the total price of the hotel stay and displays it to the user.

The checkout page takes the room price calculation from the basket page and adds it to the price of the flight and provides a final total for the user to review.



4.3.4 Show only essential info

The guidelines of showing only essential information have been implemented in the flight browsing section. Here, only the information that the user absolutely must know about the flight is displayed.

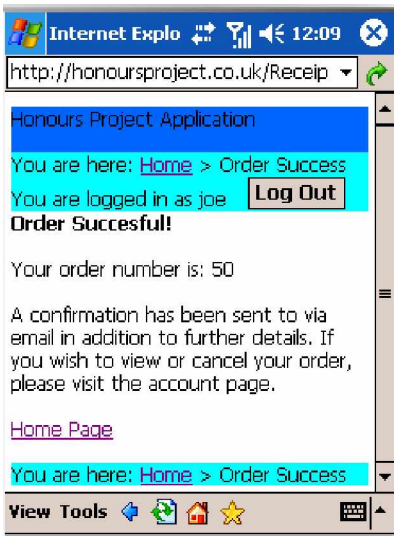


4.3.5 Simplify user input

Guidelines regarding simplification of input can be seen here. The browse flight page only gives the users the ability to choose between airports that are available, which reduces the margin for error by a large amount as it prevents the user from having to check whether an airport is available or not, only to be confronted by an error.

4.3.6 Provide confirmation

The checkout confirmation guidelines can be seen here. The user is provided with an order number and instructions on how to cancel. With that information they can easily cancel the order if the wish.



Daffy Duck
20060xxxx

4.3.7 Registration

An issue identified from the literature review was that users were leaving the site when they had to register as they felt there was no incentive at that point. Therefore the web site was designed so that a user can access and use every section of the site, with the exception of account details and checkout, without having to register an account. If the user tries to access those sections of the site they will be presented with the following screen:

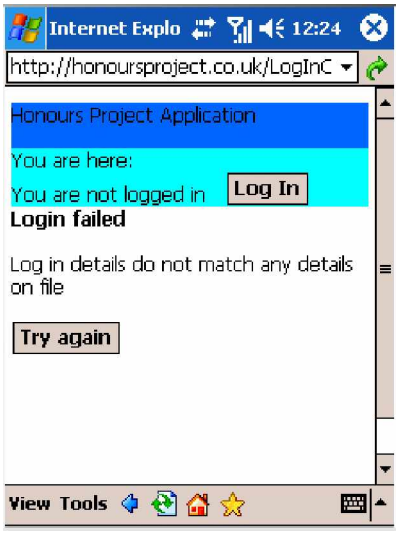
4.3.8 Text

A guideline identified in the literature review was that the minimum text size should be 14 pts as it may be difficult to read other wise on a small screen. Early in the development process this guideline was adhered to but it was found that much of the text would go onto the next line as the size was too large. No other literature was found to back this guideline up so it was decided to revert the text back to the default size (12pt) to test whether this was sufficient. From the previous screenshots it can be seen that the text is the same size throughout. As the resolution of mobile devices tend to be lower that will make the text larger than a non-mobile device by default, so an increase in size was not deemed necessary.

4.3.9 Error recovery

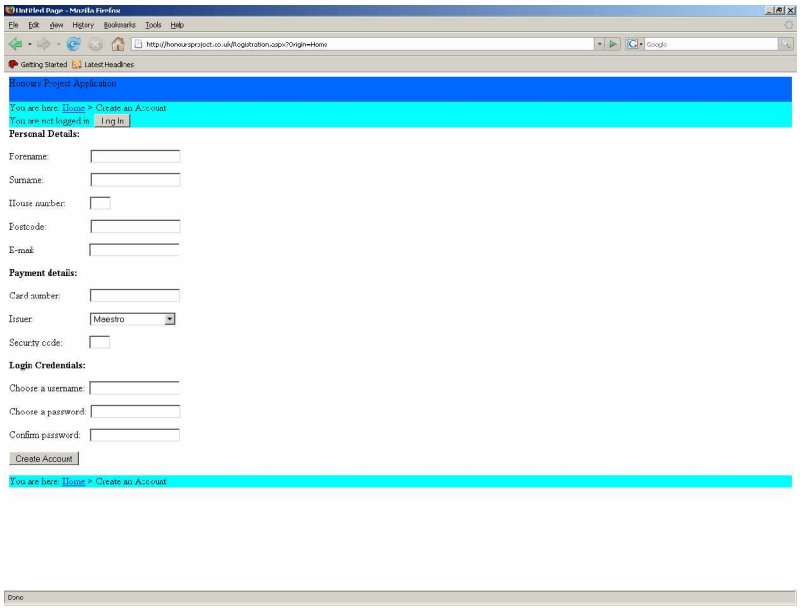
The following two screenshots show give examples of how Nielsen's (2005) heuristic of error recovery has been implemented. The first screenshot shows the account creation screen. In this case, the user has pressed the "Create Account" button without having filled out all the details. The error appears at the top of the screen and provides an explanation as to what happened, making it clear what the user must to do correct the error.

The screenshot below shows the screen the user would be taken to if they tried to log in with details that do not match up. Once again, the screen gives an explanation as to what happens and the button below lets the user attempt to try logging in again.



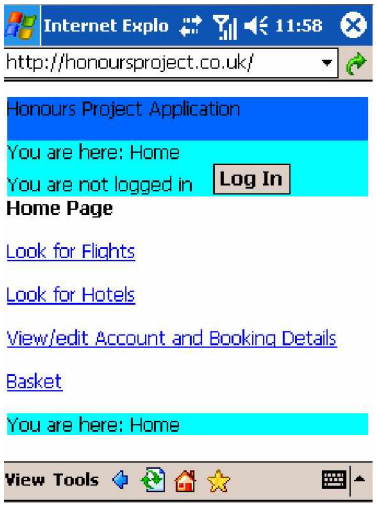
4.3.10 Allow for desktop based customisation

The screenshot shows how Longoria's (2004) heuristic of allowing desktop based customisation. Strictly speaking, the user cannot customise the site but they can carry out any action that they would on the mobile device. Carrying out the registration process (shown below) may be more desirable on a desktop computer as it is much more suited to typing, plus the user is able to see the whole screen as opposed to having to keep on scrolling down once they have completed a text field.



4.3.11 Consistency

Here, as well as on the other pictures, it can been how CSS has been used to provide a consistent interface. The bars across the top and bottom were generated using CSS code, having them there re-assures the user that they are still on the same site.



5. Testing and Evaluation

This section contains information regarding the data capture instruments and the results derived from using them. All of the participants did reasonably well, and the results showed some unexpected trends in regards to usability issues with more experienced users.

5.1 Usability Testing

5.1.1 Location

As mobile devices can be used practically anywhere the initial idea was to conduct field tests to give an accurate representation of the user's performance in such an environment. Kaikonnen et al (2005) indicate that a field test may not be worth the time and effort as they found no additional usability issues were found in a field test than a lab test. To facilitate the schedules of the test participants, it was decided to carry out the test in two places: one being the Saltire centre and the other being a quiet house. In addition to providing a test in multiple scenarios, using these places supports the work of Kim et al (2002) as they found that users are unlikely to use the mobile device everywhere but instead in a few contexts; they are likely to go somewhere that is out of the way in order to put more focus into using the device. The house is completely quiet and a relatively quiet section of the bottom floor of the Saltire was used where there was not a lot of foot traffic, both are out the way and provide different context to test the mobile device.

5.1.2 Recruitment of test participants

Eight test participants were acquired for the test, ages ranging from 18 to 48. They were acquired by means of a convenience sample. In order to represent the possible users of the application, people with prior experience using websites were selected. In addition to representing the likely users of the application, this will have the advantage of reducing frustration and time required for the test as the user will already have some degree of familiarity with the type of application.

Most of the test participants have had several years of experience with using a web site. A participant with limited experience was intentionally chosen as Rubin (1994, p129), who refers to them as 'Least Competent Users', indicates that they can provide valuable insight to usability issues which a more experienced participant may manage to overlook without thinking about it.

5.1.3 Data capture method

Two methods were used for data capture in this project. One method was observation. This method was used to keep note of how many typing and selection errors the user made as well as how long the user took to complete the test. The second method was a semi-structured interview. 11 questions were asked (Appendix E) and the user was encouraged to add any additional comments. The participant's age and years of experience of computers were learned so that any correlation between the results and the age or the experience of the participant could be observed. Brief notes were taken regarding user behaviour.

5.1.4 Scenario

Before the test, the test participants were told the following in order to give them an idea of what to expect and to hopefully help them relax a little:

"Thank you for coming along today. The purpose of this experiment today is to find flaws in this web site so please bear in mind you are not being tested, you are helping me improve this prototype. At the end of the testing I will ask you a set of 11 questions, please avoid saying what you think I want you to say but answer truthfully. The session is entirely anonymous; none of the information gathered today can be used to identify you. You can ask me questions at any time but while the testing is being carried out try and assume that I am not here. Do you have any questions"

The test participants were asked to carry out a set of tasks that were representative of typical tasks that would be carried out on a web site such as this one, for this purpose a scenario used. The user was asked to assume a role as Anschuetz (2006) indicates that having the user assume the role gives them more freedom for critique. In addition to this, having a role meant that the participant could be provided with details to use during the registration section of the usability test. This means that they wouldn't be put in the position of having to type in their own details or make up details on the spot, reducing reluctance to take part and speeding up the testing process.

The participants were given four tasks to complete. The scenario was set out to the user as they completed each task so that they would not have to memorise all the details. The first scenario was given to the user as follows:

You are a business person living in Scotland and you have a meeting to attend in London on the 10th of June. You want to fly down to the meeting and come back straight after. You are looking to book the cheapest possible flight from a Scottish airport, those being Glasgow and Prestwick, on that day. You can assume that travelling costs to either of the airports from your home would be the same.

Once the user had reached the point where they had to register, they were provided with the details in Appendix C to use. After the flight had been booked, the user was given the second scenario:

After your meeting, you have two weeks on holiday so you decide you want to get away for a while. You have decided that you want to go to Paris for a week shortly after the day of the meeting and stay at a four star hotel.

Once the booking was made, the user was given their third scenario:

You have been getting a lot of spam with to your current e-mail address so you set up a new one; you want to update your account information to reflect the change.

The user was given the e-mail address in appendix D to use. Once the user had completed the task they were given their final scenario:

The day after you made the bookings, something came up at work so you can no longer go to Paris and you have to cancel the booking you made.

After these tasks were completed, the users were asked a set of questions (Appendix E). The results of those questions are presented in the following section.

5.2 Evaluation

This section provides the results of the evaluation and initial conclusions derived from them. Some of the questions cross between two usability areas identified in the literature review but they have all been grouped into the one deemed to have had a bigger impact. As a result the questions are not in the order in this section as they are in the rest of the sections, but the question number is clearly indicated. The times are available in Appendix N.

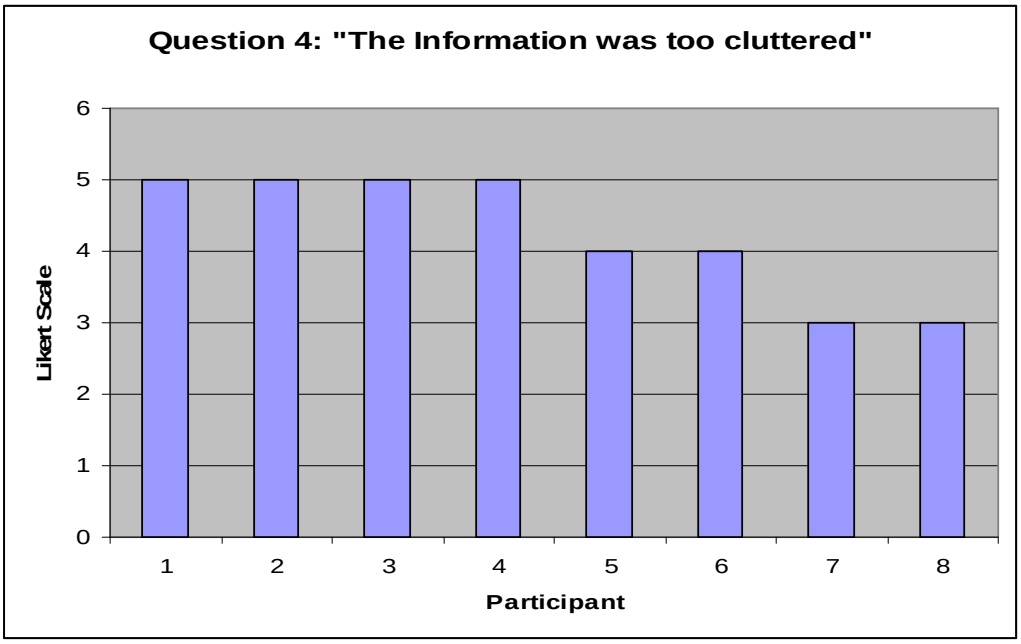
5.2.1 Speed

Some questions touched on this section but they pertained largely to other sections. Users were timed but the times do not show a pattern. The 'least competent user' took the longest time by a few minutes, besides that the times are fairly mixed up. Amount of experience does not seem to factor much into the times.

5.2.2 Structure & Navigation

5.2.2.1 Question 4

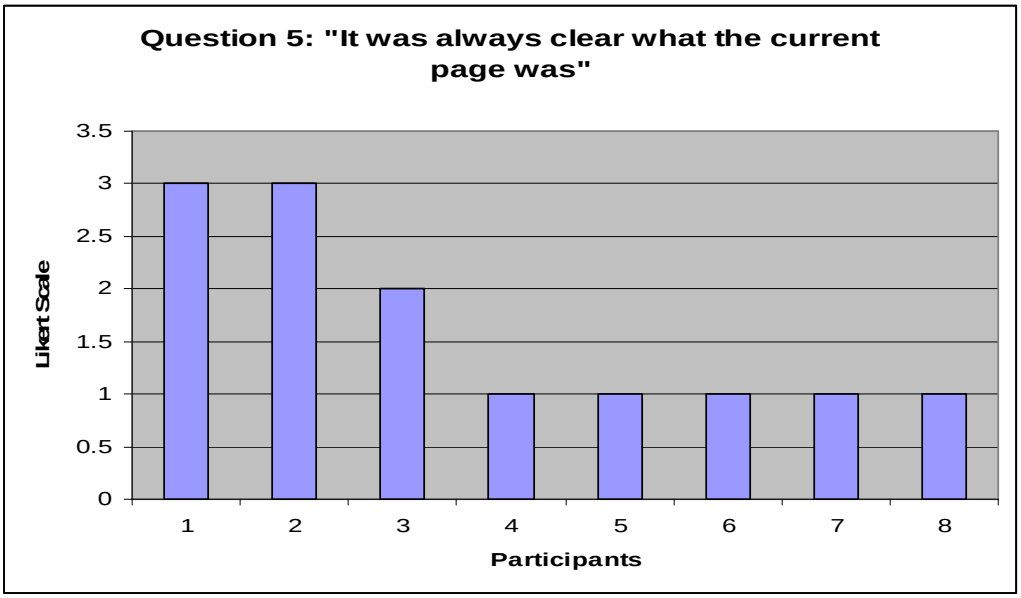
This question asked the participant whether they thought the information was too cluttered. This was to test guidelines in regards to structure of the page so that the information was laid out in a clean and clear manner. 5 is for Strongly Disagree, 1 is Strong Agree.



Most of the participants responded that they did not think it was cluttered at all, 1 commented it was extremely well laid out. Someone did comment that it was a bit cluttered but it was acceptable for a mobile device, if the screen was that busy on a desktop it would not be good. There did not appear to be any correlation between this and age or experience.

5.2.2.2 Question 5

This question asks the participant how much they agree, on a scale of 1 to 5, with the statement “It was always clear what the current page was”. 1 is strongly agree, 5 strongly disagree. This was to test guidelines in regards to having a consistent layout so that the user can look in the same places to find out where they are.



Most participants responded that they thought the page was always clear, referring to the title at each page. Some participants commented that they were not always sure where they were but took comfort in the fact that the breadcrumb navigation was available for them to use. There was a slight correlation with experience, with the more experienced users having trouble or claiming so but it wasn’t significant.

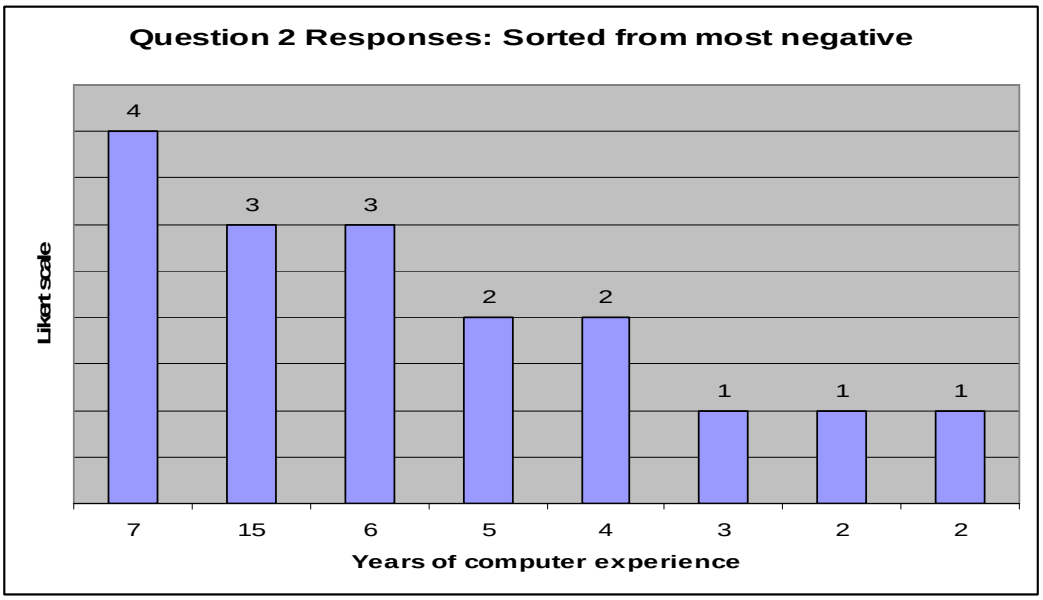
5.2.2.3 Question 6

This question asks the participant how much they agree, on a scale of 1 to 5, with the statement: “It was easy to navigate through the site” 1 is strongly agree, 5 strongly disagree. This was to test guidelines in regards to having a simple navigation system. 3 participants responded with strongly agree and 5 with agree. None of the participants made any specific comment in regards to this section.

5.2.3 Consistency

5.2.3.1 Question 2

This question asked the participants whether they always knew what was going to happen when they clicked on a link or a button.



Interestingly, participants with more years of experience responded that they were unsure of what to expect whereas those with less experience responded they had a good idea. One of the participants commented that it "just didn't seem right". Perhaps it did not respond in the way they are accustomed to whereas the less experienced users expected the response which the button hopefully indicated.

5.2.3.2 Question 11

This question asked the participant is the text always clearly readable to them. This was to test the hypothesis that Kärkkäinen & Laarni's (2002) guideline of using at least 14pt text was too large. 7 of the 8 participants responded that they had no problem reading the text. The last participant commented it was a little fuzzy because she did not have her glasses. 5 of the test participants had sun glaring off of the screen, which was unavoidable due to the lack of alternative accommodation nearby but this did not have a significant effect on the readability of the text.

5.2.4 Relevance

5.2.4.1 Question 7

This question asked the participants if they felt that if there was an image of each of the hotel rooms on the hotel browsing page would it have had an effect on their decision. This was to test the guideline that images should not be used unless they provide information. The absence of the image means that the user is probably more likely to think about why it is missing, as opposed to just glancing at it if it was there. As expected, 7 of 8 participants responded that having a picture would have been nice and have had an influence on their choice. This clearly indicates that the guideline is correct as an image can, in the right scenario, describe in a small space what would take several pages of text to write.

5.2.4.2 Question 8

This question asks the participant if they felt that in order to carry out any of the tasks they were asked to do did they feel there were any steps that were too long or could have been avoided. This was to test the guideline that the processes should be streamlined and only ask data of the user that is essential. All participants responded with a No. Some commented that the registration process was a bit long but understood that it was necessary. The guideline appears to have been successful in reducing the amount of work that the user has to carry out.

5.2.4.3 Question 9

This question asked if the participant felt that any of the information provided to them on the site was irrelevant. This was to test the guideline regarding only providing relevant information had worked effectively. All of the test participants responded saying that they felt that none of the information was irrelevant though one felt that the account and booking details should have not been on the same page. Another participant claimed the information was exactly what he would expect. This seems to indicate that all the information was relevant but it could have been structured better.

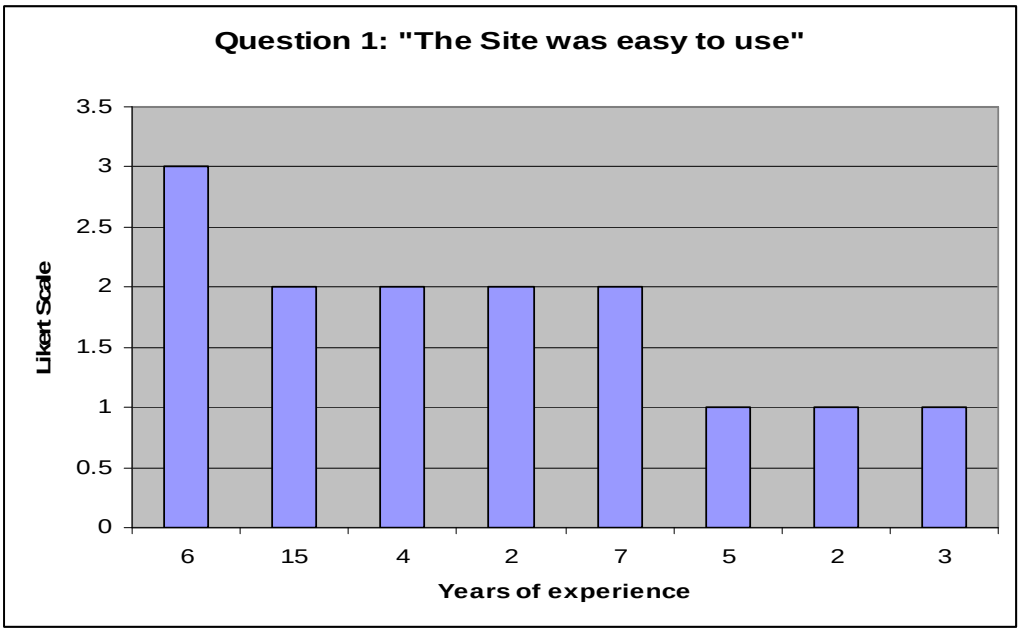
5.2.4.4 Question 10

The question asked if the participant felt that any of the information asked from them during the registration process was irrelevant. This was to test the guideline that stipulates you should only ask for the bare essential data required to complete an order. 7 of the 8 participants responded saying that they thought the data was definitely required with one saying he liked how you did not have to fill out your full address. The participant who responded negatively commented specific on the address part, saying that the site should have a pop-up allowing him to verify his address once he typed in his post code.

5.2.5 Satisfaction

5.2.5.1 Question 1

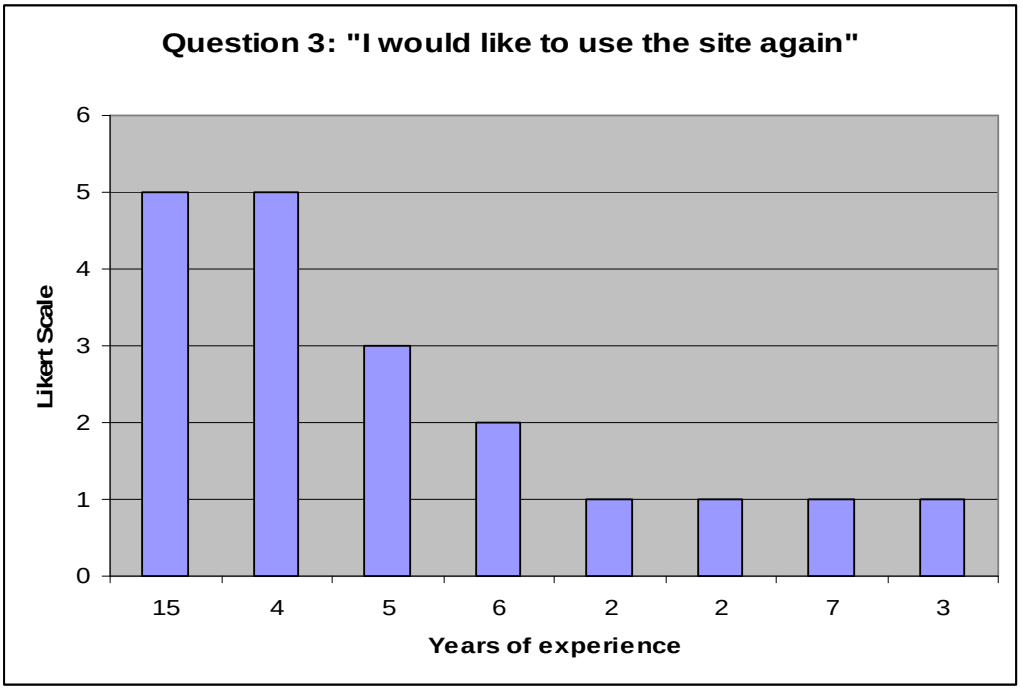
This question asked the participant how much they agree with the statement “The site was easy to use”. This question was not intended to test any specific guidelines but instead to act as an overall test as a primary goal of a usability test is to see if something is easy to use. 1 is Strongly Agree, in this case the most negative response was number 3, neither agree nor disagree.



Once again, participants with more years of experience generally responded that they found it more difficult to use than those with little experience. This may be due to the more experienced participants being ‘dug-in’ to using desktop machines and hope that the mobile device works in the same way whereas the less experienced participants do not have much to compare the device and application to.

5.2.5.2 Question 3

This question asks the participant to rate how much they agree with the statement “I would like to use this site again”. To gain an idea of how usable the prototype was, it was necessary to find out how satisfied the user was with the prototype, as per the usability definition in the literature review. 5 is Strongly Disagree going down to Strongly Agree at 1.



The results show that the participants with the most computing background in general did not wish to use the application. One participant explicit stated that he did not like using the mobile device and would only use it again out of desperation. It seemed that the more experienced participants in general did not like using mobile devices in general as they have been so used to using desktop applications for years.

6. Conclusions

This section of the report provides the detailed conclusions that can be gathered from use of the evaluation method. The report starts with an overview of the project and ends with a brief word on what work on this area could be carried out in future.

6.1 Project Overview

The purpose of this develop and test project was to test the effectiveness of a set mobile web usability guidelines applied to a prototype based on a typical Business to Consumer application. A prototype was developed and a usability evaluation was carried out on it with the assistance of 8 test participants.

The hypothesis which was being tested was that using a set of mobile web development guidelines when creating the site should result in a site fulfils its requirements as a business to consumer site and is highly usable.

To carry this out, it was necessary to identify the factors affecting usability as well as the constraints and guidelines for developing mobile web applications for mobile devices. Once these were identified, a systems analysis of a typical Business to Consumer application was carried out to remove unnecessary features and derive a set of functional and non-functional requirements.

The prototype was then put into development with adherence to the guidelines identified. A usability test was developed as the primary evaluation method and data capture methods were created in accordance to existing literature. Once the evaluation was carried out a set of conclusions were derived.

From these conclusions it has been found that the project was largely successful in developing a usable application. The following sub-section provides the main discussion of the results from this project.

6.2 Discussion of results

The results of the questions regarding relevancy 7, 8, 9 and 10 were extremely good, with most participants answering them positively. This gives a clear indication that the guidelines regarding relevancy are valid and can be used by future developers.

The results show that the more experienced test participants, in general, had more trouble using the device and less desire to use it again than the participants with less experience. The usability issues could be down to a few potential factors. Firstly, the more experienced users are not familiar with the mobile device but already had an idea in mind as to how it might have worked but got it wrong. Having the system constantly doing something other than what they expected could have become frustrating.

Another possibility is that all of the test participants were having issues but only the more experienced ones spoke up about the issues as they probably had a good idea of

what they were talking about. The less experienced participants may have had something to say but thought they would not be able to explain it so they simply graded the software good to avoid embarrassment. The notes I took during the testing process back this up. The less experienced users were fairly hesitant before carrying out an action, often asking if “they are doing it right” and, as Appendix N shows, they took slightly longer to complete the task

6.3 Future work

The prototype was designed primarily using guidelines that recommend almost everything be kept to a minimum, nothing that is unnecessary should be shown. Technology is always improving; 10 years ago the prototype may have been viewed as very advanced. With cheaper and faster bandwidth as well as more advanced devices it would be interesting to apply web technologies such as AJAX to mobile devices.

AJAX is a technology which refreshes only the part of the page that needs refreshed. This could be used when the user is filling out a textbox to provide suggestions while they type for example. As PDA usage is on the decline in general, it would be interesting to see if devices with similar major characteristics (such as the iPhone) can have the principles applied to the prototype in this project applied to them to see if they aid effectively in the development of a usable site to be viewed on such a device.

Contextual information would be a useful development as it would allow the user to see what is available nearby. That functionality was not included in this project due to time and logistical constraints. In addition to that it would have required further field testing as the user would have to be near a location that would be flagged up as relevant to what the user is looking for.

7. References

*7 Usability guidelines for websites on mobile devices*2007, 10/2007-last update [Homepage of Webcredible], [Online]. Available: http://www.webcredible.co.uk/user-friendly-resources/web-usability/mobile-guidelines.shtml [2007, 21/11/2007] .

*Handheld Device Vendors Benefit from Dell's Departure, But Overall Market Still in Decline*2007, 08/08/2007-last update [Homepage of IDC], [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=prUS20815207 [2008, 15/01/2008] .

"MOBILE MARKETING: Search nears breakthrough", 2007, *Marketing Week*, , pp. 38.

E-commerce and development report, 2003, United Nations, New York and Geneva.

Worldwide PDA & Smartphone Forecasts Report: 1998-2008, 2003 , eTForecasts.

*HTML 4.0 Guidelines for Mobile Access*1999, 15/03/1999-last update [Homepage of W3C], [Online]. Available: http://www.w3.org/TR/NOTE-html40-mobile/ [2008, 10/02/2008] .

Anschuetz, L. 2006, , *Creating Realistic Testing Scenarios* [Homepage of UPA], [Online]. Available: http://www.upassoc.org/usability_resources/conference/2006/im_anschuetz.html [2008, 01/03/2008] .

Brewin, P. & Sayer, P. 2000, 18/08/2000-last update, *Analysts Predict Euro-telcos Will Stick Users with High Cost of Spectrum Auction* [Homepage of Computerworld], [Online].

Brown, P. 2007, *M-Commerce: Search Efficiency & Results Customisation Critical*, Strategy Analytics.

Buyukkokten, O., Garcia-Molina, H. & Paepcke, A. 2000, "Focused Web searching with PDAs", *Proceedings of the 9th international World Wide Web Conference on Computer Networks : the international Journal of Computer and Telecommunications Networking*North-Holland Publishing Co., Amsterdam, The Netherlands, pp. 213.

Choren, R., Garcia, A. & Lucena, C. 2005, *Software Engineering for Multi-Agent Systems III: Research Issues and Practical Applications*, Springer.

Dholakia, N., Rask, M. & Dholakia, R. 2006, *M-commerce: Global Experiences and Perspectives*, Idea Group Inc.

Fitzsimmons, C. 2007, 15/11/2007-last update, *Who really wants a phone that plays music?* [Homepage of Guardian News and Media Limited], [Online]. Available: http://blogs.guardian.co.uk/digitalcontent/2007/11/who_really_wants_a_phone_that.html [2008, 15/01/2008] .

Germonprez, M., Avital, M. & Srinivasan, N. 2006, "The Impacts of the Cascading Style Sheet Standard on Mobile Computing", *International Journal of IT Standards & Standardization Research*, vol. 4, no. 2, pp. 55.

Jones, M., Marsden, G., Mohd-Nasir, N., Boone, K. & Buchanan, G. 1999, "Improving web interaction on small displays", *Computer Networks*, vol. 31, no. 11-16, pp. 1129-1137.

Kaikonnen, A., Kekäläinen, A., Cankar, M., Kallio, T. & Kankainen, A. 2005, "Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing", *Journal of Usability Studies*, vol. 1, no. 1, pp. 4-16.

Kaikonnen, A. & Roto, V. 2003, "Navigating in a Mobile XHTML Application", *CHI 2003*ACM, Ft. Lauderdale, Florida.

Kärkkäinen, L. & Laarni, J. 2002, "Designing for Small Display Screens", *Proceedings of the second nordic conference on human computer interaction* NordiCHI, Århus, Denmark, pp. 227.

Kim, H., Kim, J., Lee, Y., Chae, M. & Choi, Y. 2002, "An Empirical Study of the Use Contexts and Usability Problems in Mobile Internet", *Proceedings of the 35th Hawaii International Conference on System Sciences*IEEE, Hawaii.

King, A. 2003, *Speed Up Your Site: Web Site Optimization*, 1st edn, New Riders.

Lee, Y. & Benbasat, I. 2004, "A Framework for the Study of Commerce Interface Design for Mobile Commerce", *International Journal of Electronic Commerce*, vol. 8, no. 3, pp. 79-80-102.

Lim, E. & Siau, K. 2003, *Advances in Mobile Commerce Technologies*, Idea Group Inc.

Longoria, R. 2004, *Designing Software for the Mobile Context: A Practitioner's Guide*, Springer, New York.

Macaulay, L. 1995, *Human-Computer Interaction for Software Designers*, 1st edn, International Thompson Publishing, London, UK.

Newman, W. & Lamming, M. 1995, *Interactive System Design*, 1st edn, Addison Wesley, Harlow, England.

Nielsen, J. 2005, *Ten Usability Heuristics*. Available: http://www.useit.com/papers/heuristic/heuristic_list.html [2007, 01/12/2008] .

Nielsen, J. 1993, *Usability Engineering*, Morgan Kaufmann, San Francisco, California, USA.

Olsen, K.A. & Malizia, A. 2005, "Flexible User Interfaces for B2C Systems, Using Booking as a Case Study", *Proceedings HCI 2005*Las Vegas, USA, pp. 28.

Ramsay, M. & Nielsen, J. 2000, *WAP Usability Déjà vu: 1994 All Over Again. Report from a Field Study in London Fall 2000*, Nielsen Norman Group.

Ragus, D. (2000). *Best practices for designing shopping cart and checkout interfaces* [Online]. Available: http://www.dack.com/web/shopping_cart.html

Rubin, J. 1994, *Handbook of Usability Testing*, John Wiley & Sons, Canada.

Shneiderman, B. & Plaisant, C. 2005, *Designing The User Interface*, 4th edn, Addison-Wesley, Reading, Massachusetts.

Tarasewich, P. 2003, "Designing Mobile Commerce Applications", *Association for Computing Machinery.Communications of the ACM*, vol. 46, no. 12, pp. 57.

Tarasewich, P., Nickerson, R. & Warkentin, M. 2002, "Issues in Mobile E-Commerce", *Communications for the Association for Information Systems*, vol. 8, pp. 41-42-64.

Tracy, B. (2000). Easy net navigation is mandatory – Viewpoint: Online users happy to skip frills for meat and potatoes. *Advertising Age*, p. 38.

U.S. Census Bureau, International Data Base (IDB), last updated 17/07/2003. Available: http://www.census.gov/ipc/www/idbnew.html

Van Thanh, D. 2000, "Security Issues in Mobile eCommerce", *Proceedings of the 11th International Workshop on Database and Expert Systems Applications*IEEE Computer Society, Billingstad, Sweeden, pp. 412.

Varian, H., Litan, R., Elder, A. & Shutter, J. 2002, *The Net Impact Study: The Projected Economic Benefits of the Internet in the United States,United Kingdom, France and Germany.*, www.netimpactstudy.com.

Wang, J.J., Song, Z., Lei, P. & Sheriff, R.E. 2005, "Design and Evaluation of M-Commerce Applications", *Communications, 2005 Asia-Pacific Conference on*, ed. Z. Song, Perth, Western Australia, pp. 745.

Wesson, J. & Van Der Waalt, D. 2005, "Implementing Mobile Services: Does the Platform Really Make a Difference?", *Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*ACM International Conference Proceeding Series; Vol 150, White River, South Africa, pp. 208.

Zhang, D. 2007, "Web content adaptation for mobile handheld devices", *Association for Computing Machinery.Communications of the ACM*, vol. 50, no. 2, pp. 75.

Zoller, E. 2007, 13/02/2007-last update, *20% of UK Subscribers Search Internet Via Mobile Phones* [Homepage of 3G.co.uk], [Online]. Available: http://www.3g.co.uk/PR/Feb2007/4267.htm [2008, 09/02/2008] .

8. Appendices

Appendix A – Nielsen's Usability Heuristics (Nielsen, 2005)

Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Appendix B – Longoria's heuristics (Longoria, 2004)

1. There is a need.
2. Every pixel counts.
3. Every round trip counts
4. Employ "feature shredding"
5. Keep your navigation model simple
6. Think modular
7. Minimise data entry
8. Allow for desktop-based customisation
9. Fight the hype
10. Basic UI principles still apply.

Appendix C – Data provided to test participant for registration process

Name: Robert Wilson
Address: 10 Dunlop Street
Stewarton
KA5 1RS
E-Mail: rwilson@hotmail.com

Credit card: Visa
5484321566118165
Security Code: 551

User name: rwilson
Password: pass

Appendix D – New E-Mail address

E-mail: rwilson@gmail.com

Appendix E – Questions asked for usability test

The first 6 questions are statements rated by the user on a Likert Scale see whether they Strongly Agree, Strongly Disagree or anywhere in between with what was said. The rest are "Yes" or "No" questions, the participant were encouraged to add any extra comments if they wished.

1. Strongly Agree
2. Agree
3. Neither Agree nor Disagree
4. Disagree
5. Strongly Disagree

1. "The site was easy to use".

2. "It was always clear what was going to happen after I had clicked on a button or hyperlink"

3. "I would like to use the website again"

4. "The information was too cluttered"

5. "It was always clear what the current page was"

6. "It was easy to navigate through the site"

7. If there was an image of each of the hotel rooms on the hotel browsing page, do you think they would have had an influence on your choice?

8. In order to carry out any of the tasks I asked you to do, did you feel there were any steps that were too long or could have been avoided?

9. Did you feel that any of the information provided to you on the site was irrelevant?

10. Did you feel that any of the information asked from you during the registration process was irrelevant?

11. Was the text always clearly readable?

Appendix F – Interview transcript: Participant 1

Me: Thank you for taking part in the test, I am now going to ask you a few questions on your opinion of the prototype. Keep in mind that this is not a test, I am here to find flaws in the prototype. Please feel free to add any comments after you have answered each question.
Participant: Okay

Me: For the first few questions, I would like you to respond with a 1, 2, 3, 4 or 5. 1 meaning you strongly agree with what I have said, 5 if you strongly disagree, or in between.

Participant: Okay

Me: Alright, "the site was easy to use".

Participant: 2

Me: "It was always clear what was going to happen after I had clicked on a button or hyperlink"

Participant: 1

Me: "I would like to use the website again"

Participant: 1

Me: "The information was too cluttered"

Participant: 5

Me: "It was always clear what the current page was"

Participant: 1

Me: "It was easy to navigate through the site"

Participant: 1

Me: Thanks, for the next questions, please answer with a Yes or No. Once again, feel free to elaborate if you wish.

Participant: Okay

Me: Okay, if there was an image of each of the hotel rooms on the hotel browsing page, do you think they would have had an influence on your choice?

Participant: No

Me: In order to carry out any of the tasks I asked you to do, did you feel there were any steps that were too long or could have been avoided?

Participant: No

Me: Did you feel that any of the information provided to you on the site was irrelevant?
Participant: No

Me: Did you feel that any of the information asked from you during the registration process was irrelevant?

Participant: No

Me: Was the text always clearly readable?

Participant: No

Me: That's the last of the questions, thank you for taking part. Do you have anything you would like to add?

Participant: No, thank you.

Appendix G – Interview transcript: Participant 2

Me: Thank you for taking part in the test, I am now going to ask you a few questions on your opinion of the prototype. Keep in mind that this is not a test, I am here to find flaws in the prototype. Please feel free to add any comments after you have answered each question.
Participant: Okay

Me: For the first few questions, I would like you to respond with a 1, 2, 3, 4 or 5. 1 meaning you strongly agree with what I have said, 5 if you strongly disagree, or in between.

Participant: Okay

Me: Alright, "the site was easy to use".

Participant: 1

Me: "It was always clear what was going to happen after I had clicked on a button or hyperlink"

Participant: 2. Generally, I knew what was going to happen but I wasn't too sure at the part where I had to change my details though.

Me: "I would like to use the website again"

Participant: 5. I didn't like using the PDA in general, nothing really against the site.

Me: "The information was too cluttered"

Participant: 5

Me: "It was always clear what the current page was"

Participant: 1

Me: "It was easy to navigate through the site"

Participant: 1

Me: Thanks, for the next questions, please answer with a Yes or No. Once again, feel free to elaborate if you wish.

Participant: Okay

Me: Okay, if there was an image of each of the hotel rooms on the hotel browsing page, do you think they would have had an influence on your choice?

Participant: Yes

Me: In order to carry out any of the tasks I asked you to do, did you feel there were any steps that were too long or could have been avoided?

Participant: No

Me: Did you feel that any of the information provided to you on the site was irrelevant?
Participant: No, there were enough details.

Me: Did you feel that any of the information asked from you during the registration process was irrelevant?

Participant: No, didn't think so.

Me: Was the text always clearly readable?

Participant: Yes

Me: That's the last of the questions, thank you for taking part. Do you have anything you would like to add?

Participant: I think that at the account screen the change confirmation buttons should have been at the bottom

Appendix H – Interview transcript: Participant 3

Me: Thank you for taking part in the test, I am now going to ask you a few questions on your opinion of the prototype. Keep in mind that this is not a test, I am here to find flaws in the prototype. Please feel free to add any comments after you have answered each question.
Participant: Okay

Me: For the first few questions, I would like you to respond with a 1, 2, 3, 4 or 5. 1 meaning you strongly agree with what I have said, 5 if you strongly disagree, or in between.

Participant: Okay

Me: Alright, "the site was easy to use".

Participant: 2

Me: "It was always clear what was going to happen after I had clicked on a button or hyperlink"

Participant: 2

Me: "I would like to use the website again"

Participant: 3

Me: "The information was too cluttered"

Participant: 3

Me: "It was always clear what the current page was"

Participant: 1

Me: "It was easy to navigate through the site"

Participant: 1

Me: Thanks, for the next questions, please answer with a Yes or No. Once again, feel free to elaborate if you wish.

Participant: Okay

Me: Okay, if there was an image of each of the hotel rooms on the hotel browsing page, do you think they would have had an influence on your choice?

Participant: Yes

Me: In order to carry out any of the tasks I asked you to do, did you feel there were any steps that were too long or could have been avoided?

Participant: No

Me: Did you feel that any of the information provided to you on the site was irrelevant?
Participant: No

Me: Did you feel that any of the information asked from you during the registration process was irrelevant?

Participant: No, it was all needed.

Me: Was the text always clearly readable?

Participant: Yes

Me: That's the last of the questions, thank you for taking part. Do you have anything you would like to add?

Participant: I expected the account change buttons to be at the bottom. In addition, this is not a dig at your application but in general, the way for putting text in is too difficult. If it was an older person it would be impossible for them to type on it. I was confused as to what the two parts at the top and bottom were. Nice website though.

Appendix I – Interview transcript: Participant 4

Me: Thank you for taking part in the test, I am now going to ask you a few questions on your opinion of the prototype. Keep in mind that this is not a test, I am here to find flaws in the prototype. Please feel free to add any comments after you have answered each question.
Participant: Okay

Me: For the first few questions, I would like you to respond with a 1, 2, 3, 4 or 5. 1 meaning you strongly agree with what I have said, 5 if you strongly disagree, or in between.

Participant: Okay

Me: Alright, "the site was easy to use".

Participant: 1

Me: "It was always clear what was going to happen after I had clicked on a button or hyperlink"

Participant: 4. It didn't seem quite right, seemed vague.

Me: "I would like to use the website again"

Participant: 1

Me: "The information was too cluttered"

Participant: 3. For a mobile device it was nicely set out but on a regular PC it would be far too busy.

Me: "It was always clear what the current page was"

Participant: 1

Me: "It was easy to navigate through the site"

Participant: 2

Me: Thanks, for the next questions, please answer with a Yes or No. Once again, feel free to elaborate if you wish.

Participant: Okay

Me: Okay, if there was an image of each of the hotel rooms on the hotel browsing page, do you think they would have had an influence on your choice?

Participant: Yes, definitely.

Me: In order to carry out any of the tasks I asked you to do, did you feel there were any steps that were too long or could have been avoided?

Participant: No

Me: Did you feel that any of the information provided to you on the site was irrelevant?
Participant: No, all of it was required I thought.

Me: Did you feel that any of the information asked from you during the registration process was irrelevant?

Participant: No

Me: Was the text always clearly readable?

Participant: Yes

Me: That's the last of the questions, thank you for taking part. Do you have anything you would like to add?

Participant: No, thank you.

Appendix J – Interview transcript: Participant 5

Me: Thank you for taking part in the test, I am now going to ask you a few questions on your opinion of the prototype. Keep in mind that this is not a test, I am here to find flaws in the prototype. Please feel free to add any comments after you have answered each question.
Participant: Okay

Me: For the first few questions, I would like you to respond with a 1, 2, 3, 4 or 5. 1 meaning you strongly agree with what I have said, 5 if you strongly disagree, or in between.

Participant: Okay

Me: Alright, "the site was easy to use".

Participant: 2, the OK button on the account page threw me off though since I scrolled right by it, expecting it to be at the bottom

Me: "It was always clear what was going to happen after I had clicked on a button or hyperlink"

Participant: 1

Me: "I would like to use the website again"

Participant: 1. The fact that it calculated the hotel nights and price was excellent

Me: "The information was too cluttered"

Participant: 5

Me: "It was always clear what the current page was"

Participant: 3, generally. I knew I could refer to the home page though.

Me: "It was easy to navigate through the site"

Participant: 2

Me: Thanks, for the next questions, please answer with a Yes or No. Once again, feel free to elaborate if you wish.

Participant: Okay

Me: Okay, if there was an image of each of the hotel rooms on the hotel browsing page, do you think they would have had an influence on your choice?

Participant: Probably

Me: In order to carry out any of the tasks I asked you to do, did you feel there were any steps that were too long or could have been avoided?

Participant: No

Me: Did you feel that any of the information provided to you on the site was irrelevant?
Participant: No

Me: Did you feel that any of the information asked from you during the registration process was irrelevant?

Participant: No

Me: Was the text always clearly readable?

Participant: Yes

Me: That's the last of the questions, thank you for taking part. Do you have anything you would like to add?

Participant: No, thank you.

Appendix K – Interview transcript: Participant 6

Me: Thank you for taking part in the test, I am now going to ask you a few questions on your opinion of the prototype. Keep in mind that this is not a test, I am here to find flaws in the prototype. Please feel free to add any comments after you have answered each question.
Participant: Okay

Me: For the first few questions, I would like you to respond with a 1, 2, 3, 4 or 5. 1 meaning you strongly agree with what I have said, 5 if you strongly disagree, or in between.

Participant: Okay

Me: Alright, "the site was easy to use".

Participant: 1

Me: "It was always clear what was going to happen after I had clicked on a button or hyperlink"

Participant: 1, I really liked the fact it gave me a confirmation and a number after I completed my booking

Me: "I would like to use the website again"

Participant: 1

Me: "The information was too cluttered"

Participant: 5, it was well set out and straight forward

Me: "It was always clear what the current page was"

Participant: 1

Me: "It was easy to navigate through the site"

Participant: 2, I made a mistake with the hotel but I did manage to fix it fairly easily

Me: Thanks, for the next questions, please answer with a Yes or No. Once again, feel free to elaborate if you wish.

Participant: Okay

Me: Okay, if there was an image of each of the hotel rooms on the hotel browsing page, do you think they would have had an influence on your choice?

Participant: Yes

Me: In order to carry out any of the tasks I asked you to do, did you feel there were any steps that were too long or could have been avoided?

Participant: No, having used this kind of site before I found this to be easy in comparison

Me: Did you feel that any of the information provided to you on the site was irrelevant?
Participant: No, not at all

Me: Did you feel that any of the information asked from you during the registration process was irrelevant?

Participant: No

Me: Was the text always clearly readable?

Participant: Yes

Me: That's the last of the questions, thank you for taking part. Do you have anything you would like to add?

Participant: I liked the colour scheme and how it was consistent through out. I think it would definitely be worth further development, though having pictures of the hotel rooms would have been nice.

Appendix L – Interview transcript: Participant 7

Me: Thank you for taking part in the test, I am now going to ask you a few questions on your opinion of the prototype. Keep in mind that this is not a test, I am here to find flaws in the prototype. Please feel free to add any comments after you have answered each question.
Participant: Okay

Me: For the first few questions, I would like you to respond with a 1, 2, 3, 4 or 5. 1 meaning you strongly agree with what I have said, 5 if you strongly disagree, or in between.

Participant: Okay

Me: Alright, "the site was easy to use".

Participant: 3, it was not as bad as most mobile sites I have used but I just didn't like it in general, might just be an age thing.

Me: "It was always clear what was going to happen after I had clicked on a button or hyperlink"

Participant: 3, I was pretty uncertain what was going to happen

Me: "I would like to use the website again"

Participant: 5, I much prefer a desktop version. Would only use this out of shear desperation.

Me: "The information was too cluttered"

Participant: 4, it was laid out OK but it could have been a bit tidier.

Me: "It was always clear what the current page was"

Participant: 2, I liked the breadcrumb system though.

Me: "It was easy to navigate through the site"

Participant: 2, I liked having the home button always on display

Me: Thanks, for the next questions, please answer with a Yes or No. Once again, feel free to elaborate if you wish.

Participant: Okay

Me: Okay, if there was an image of each of the hotel rooms on the hotel browsing page, do you think they would have had an influence on your choice?

Participant: Yes, I chose one because it had a fancy name haha.

Me: In order to carry out any of the tasks I asked you to do, did you feel there were any steps that were too long or could have been avoided?

Participant: No

Me: Did you feel that any of the information provided to you on the site was irrelevant?
Participant: Having the account information and booking information on the same page was pretty confusing to me.

Me: Did you feel that any of the information asked from you during the registration process was irrelevant?

Participant: I think it would have been better if it popped up after you put in your post code and asked you to verify your address

Me: Was the text always clearly readable?

Participant: The departure and return sections were confusing. Maybe if there was a wee graphic next to those details they would be more clear

Me: That's the last of the questions, thank you for taking part. Do you have anything you would like to add?

Participant: Having airport codes as opposed to city names would have been less confusing. Having some clarification as to what a "security code" was would have been nice. Making the cancellation of a booking more obvious would be good.

Appendix M – Interview transcript: Participant 8

Me: Thank you for taking part in the test, I am now going to ask you a few questions on your opinion of the prototype. Keep in mind that this is not a test, I am here to find flaws in the prototype. Please feel free to add any comments after you have answered each question.
Participant: Okay

Me: For the first few questions, I would like you to respond with a 1, 2, 3, 4 or 5. 1 meaning you strongly agree with what I have said, 5 if you strongly disagree, or in between.

Participant: Okay

Me: Alright, "the site was easy to use".

Participant: 2

Me: "It was always clear what was going to happen after I had clicked on a button or hyperlink"

Participant: 3

Me: "I would like to use the website again"

Participant: 2

Me: "The information was too cluttered"

Participant: 4

Me: "It was always clear what the current page was"

Participant: 3

Me: "It was easy to navigate through the site"

Participant: 2

Me: Thanks, for the next questions, please answer with a Yes or No. Once again, feel free to elaborate if you wish.

Participant: Okay

Me: Okay, if there was an image of each of the hotel rooms on the hotel browsing page, do you think they would have had an influence on your choice?

Participant: Yes

Me: In order to carry out any of the tasks I asked you to do, did you feel there were any steps that were too long or could have been avoided?

Participant: No

Me: Did you feel that any of the information provided to you on the site was irrelevant?
Participant: No

Me: Did you feel that any of the information asked from you during the registration process was irrelevant?

Participant: No, I liked how you entered your postcode and didn't have to do the full address

Me: Was the text always clearly readable?

Participant: Yes

Me: That's the last of the questions, thank you for taking part. Do you have anything you would like to add?

Participant: Everything is pretty self-explanatory. Having the edit confirmation buttons at the top was confusing though. I expected them at the bottom

Appendix N – Raw Data

	1	2	3	4	5	6	7	8
Question 1	2	2	1	2	1	1	2	3
Question 2	1	2	2	4	1	1	3	3
Question 3	1	5	3	1	1	1	5	2
Question 4	5	5	3	3	5	5	4	4
Question 5	1	1	1	1	3	1	2	3
Question 6	1	1	1	2	2	2	2	2
Question 7	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Question 8	No	No	No	No	No	No	No	No
Question 9	No	No	No	No	No	No	No	No
Question 10	No	No	No	No	No	No	Yes	No
Question 11	No	Yes	Yes	Yes	Yes	Yes	No	Yes
Typing errors:	6	2	4	0	0	3	4	0
Age:	48	18	40	22	26	21	30	20
Time:	11	10	9	6	7	8	8	7
Experience:	2	4	5	7	3	2	15	6

Appendix O – Prototype source code

```
'Lines starting with a ' are comments. All of the comments are just
'above the lines they refer to. Page are in alphabetical order, with
'the exception of the master page.
```

Masterpage.masterpage

The purpose of this page is to provide a consistent template for all of the pages to be loaded into. When a page is requested, this page loads then the requested page will load within it.

```
<%@ Master Language="VB" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<script runat="server">
    Dim userID As Integer
    Dim currentPage As String

    'Five cookies are created to store the various pieces of data
    'required to be transferred between pages. stores the ID of the
    'user currently logged in. flightCookie stores the ID of the
    'selected flight. roomCookie stores the ID of the selected room.
    'roomCostCookie stores the cost of the total cost of the selected
    'room by taking the number of days staying into account.
    'roomNightsCookie stores the number of nights the customer is
    'staying in the room

    Dim loginCookie As HttpCookie = New HttpCookie("login")
    Dim flightCookie As HttpCookie = New HttpCookie("flight")
    Dim roomCookie As HttpCookie = New HttpCookie("room")
    Dim roomCostCookie As HttpCookie = New HttpCookie("roomCost")
    Dim roomNightsCookie As HttpCookie = New HttpCookie("roomNights")

    Sub Page_Load()
        'Every time the page loads carry out this sub-routine..
        'Gets the URL of the current page

      currentPage = Request.Url.ToString

        If Request.Cookies("login") Is Nothing Then

            'The values of all the cookies are set to 0 if there are
            no 'cookies on the users computer (would happen if it's
            the 'first time the web site has been used.

            loginCookie("UserID") = 0
            loginCookie.Expires = Now.AddDays(1)
            Response.Cookies.Add(loginCookie)

            flightCookie("flightID") = 0
            loginCookie.Expires = Now.AddDays(1)
            Response.Cookies.Add(flightCookie)

            roomCookie("roomID") = 0
            loginCookie.Expires = Now.AddDays(1)
            Response.Cookies.Add(roomCookie)

            roomCostCookie("cost") = 0
            roomCostCookie.Expires = Now.AddDays(1)
```

```
            Response.Cookies.Add(roomCostCookie)

            roomNightsCookie("nights") = 0
            roomNightsCookie.Expires = Now.AddDays(1)
            Response.Cookies.Add(roomNightsCookie)


            'Shows the user that they are not logged in and shows a '
            'button to allow them to login.

            lblLoginStatus.Text = "You are not logged in "
            btnLogin.Visible = True

        Else 'If the loginCookie is present

            'Assigns the value of the loginCookie to userID
            userID = Request.Cookies("login")("UserID")


            'If the userID is 0, no one is logged in, show the status
            'as not logged in

            If userID = 0 Then

                lblLoginStatus.Text = "You are not logged in "
                btnLogin.Visible = True
                lnkLogOut.Visible = False


            Else

              'If the userID is not 0, someone is logged in. Set the
            'invisible label Label1 to the userID. A GridView1 is
            'connected to a data source. It takes the value of the
            'userID label uses that to query the users table and
            'return the user name that corresponds with the user ID.

                Label1.Text = userID
                lblLoginStatus.Text = "You are logged in as " +
GridView1.Rows.Item(0).Cells(1).Text.ToString
                lnkLogOut.Visible = True
                btnLogin.Visible = False

            End If
        End If

      'If the current screen is the registration success screen, make
      'the login button invisible. Another button is provided on that
      'screen which returns the users to where they were.
        If Right(currentPage, 27) = "RegSuccess.aspx?Origin=Home"
Then
            btnLogin.Visible = False
        End If
    End Sub
    Protected Sub lnkLogOut_Click(ByVal sender As Object, ByVal e As
System.EventArgs)

        'When the logout button is clicked, reset the values of all
        the   'cookies to 0 and take the user to the home page.

        loginCookie("UserID") = 0
```

```
        loginCookie.Expires = Now.AddDays(1)
        Response.Cookies.Add(loginCookie)

        flightCookie("flightID") = 0
        loginCookie.Expires = Now.AddDays(1)
        Response.Cookies.Add(flightCookie)

        roomCookie("roomID") = 0
        loginCookie.Expires = Now.AddDays(1)
        Response.Cookies.Add(roomCookie)

        roomCostCookie("cost") = 0
        roomCostCookie.Expires = Now.AddDays(1)
        Response.Cookies.Add(roomCostCookie)

        roomNightsCookie("nights") = 0
        roomNightsCookie.Expires = Now.AddDays(1)
        Response.Cookies.Add(roomNightsCookie)

        Response.Redirect("Default.aspx")
    End Sub

    Protected Sub btnLogin_Click(ByVal sender As Object, ByVal e As
System.EventArgs)
       'When the login button is clicked, redirect the user to the
      `login page

        Response.Redirect("Login.aspx?TopLogin=Yes&Origin=" +
currentPage)
    End Sub
</script>

   <html>
   <head id="Head1" runat="server">
       <title>Master Page</title>
       <style type="text/css">
       div.Header { background-color: #0066FF }
       div.Nav { background-color: #00FFFF }
       div.LogInput { visibility :hidden }
       </style>
   </head>
   <body>
   <form id="Form1" runat="server">
   <div class="Header">
          Honours Project Application<br />
     <br />
   </div>
   <div class="Nav">
         You are here:
         <asp:SiteMapPath ID="SiteMapPath1" runat="server">
         </asp:SiteMapPath>
         <br />
         <asp:Label ID="lblLoginStatus" runat="server" Text="Log In
Status"></asp:Label>

          <asp:Button ID="btnLogin" runat="server" Text="Log In"
             onclick="btnLogin_Click" />
         <asp:Button ID="lnkLogOut" runat="server" Visible="False"
Text="Log Out"
             onclick="lnkLogOut_Click"/>
         </div>
```

```
      <div>
          <asp:contentplaceholder
        id="ContentPlaceHolder1"
        runat="Server">
     </asp:contentplaceholder>
    </div>

    <div class="Nav">

      You are here:
      <asp:SiteMapPath ID="SiteMapPath2" runat="server">
      </asp:SiteMapPath>
        <asp:Label ID="Label1" runat="server" Text="Label"
Visible="False"></asp:Label>
        <br />
        <asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False"
            DataKeyNames="UserID" DataSourceID="AccessDataSource1"
Visible="False">
            <Columns>
                <asp:BoundField DataField="UserID" HeaderText="UserID"
InsertVisible="False"
                    ReadOnly="True" SortExpression="UserID" />
                <asp:BoundField DataField="UserName"
HeaderText="UserName"
                    SortExpression="UserName" />
            </Columns>
        </asp:GridView>
        <asp:AccessDataSource ID="AccessDataSource1" runat="server"
            DataFile="~/SiteDataBase.mdb"
            SelectCommand="SELECT [UserID], [UserName] FROM [Users]
WHERE ([UserID] = ?)">
            <SelectParameters>
                <asp:ControlParameter ControlID="Label1" Name="UserID"
PropertyName="Text"
                    Type="Int32" />
            </SelectParameters>
        </asp:AccessDataSource>
    </div>
    </form>
        </body>
    </html>
```

Account.aspx

The purpose of this page is to display user account data and bookings, as well as giving the user the option to change them.

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
Title="Untitled Page" %>

<script runat="server">
    Sub Page_Load()
        'Execute this code each time the page loads…
        Dim userNo As String
        Dim cancel As String

       'This checks the URL to see if the user has just cancelled a
      ‘booking. The cancellation code will place a “Yes” in the URL
      ‘under the query value “Cancel”

        cancel = (Request.QueryString("Cancel"))

        'Sets the text value of the label below to what ever the
cancel
        ‘value is.
        lblCancelConf.Text = cancel

        'If an order has just been cancelled, then display a
        ‘confirmation that it has been cancelled.

        If lblCancelConf.Text = "Yes" Then
            lblCancelConf.Text = "Order cancelled"
            lblCancelConf.Visible = True
        End If

        'Get the userID of the user logged in from the login cookie
and
        ‘send that value to the userNo label.

        userNo = Request.Cookies("login")("UserID")
        lblUserNo.Text = userNo


        'Checks to see if there are any bookings for this user. If
        ‘there are not, display a message stating so.

        If DataList1.Items.Count.ToString = "0" Then
            lblNoBookings.Visible = True
        End If


    End Sub
    Dim totalItems As Integer




 Sub Item_Created(ByVal sender As Object, ByVal e As
DataListItemEventArgs)
        'This code is executed when the system is populating the list
```

```
        ‘of bookings. It is intended to hide the hotel fields if only
      a
        ‘flight has been booked.

        'Apply this code to all items..
        If e.Item.ItemType = ListItemType.Item Or e.Item.ItemType =
ListItemType.AlternatingItem Then

            'Get each of the values of the fields pertaining to hotel
            ‘data from the web control and give those fields a
shorter
            ‘programmatic name

            Dim hotel As Label =
CType(e.Item.FindControl("HotelLabel"), Label)
            Dim hotelDisp As Label =
CType(e.Item.FindControl("dispHotelLabel"), Label)
            Dim city As Label =
CType(e.Item.FindControl("cityLabel"), Label)
            Dim cityDisp As Label =
CType(e.Item.FindControl("dispcityLabel"), Label)
            Dim stars As Label =
CType(e.Item.FindControl("starsLabel"), Label)
            Dim starsDisp As Label =
CType(e.Item.FindControl("dispstarsLabel"), Label)
            Dim cost As Label =
CType(e.Item.FindControl("costLabel"), Label)
            Dim costDisp As Label =
CType(e.Item.FindControl("dispcostLabel"), Label)

            'When a booking is put through that does not include a
            ‘hotel, the system sets the hotel number to 22, which is
            a ‘number to indicate that no hotel is chosen. The name
            of ‘the hotel is “empty”. If this is the case for this
            ‘booking, hide all the hotel related fields.

            If hotel.Text = "Empty" Then

                hotel.Visible = False
                hotelDisp.Visible = False
                city.Visible = False
                cityDisp.Visible = False
                stars.Visible = False
                starsDisp.Visible = False
                cost.Visible = False
                costDisp.Visible = False

            End If

        End If

    End Sub



   Protected Sub btnCancel_Click(ByVal sender As Object, ByVal e As
DataListCommandEventArgs)
        'This code is executed when the cancel booking button is
        ‘pressed.
```

```
        Dim bookingID As Integer
        Dim comSource As String = e.CommandName.ToString

        'Checks to see what booking was cancelled by the user.
        If comSource = "Select" Then
            DataList1.SelectedIndex = e.Item.ItemIndex

            'Get the booking ID of the cancelled item
            bookingID =
Convert.ToInt32(CType(DataList1.SelectedItem.FindControl("BookingIDLa
bel"), Label).Text.ToString)
        End If

        'Set the value of the orderNo label to the booking ID
        lblOrderNo.Text = bookingID

        'Carry out the update query. This sets the values of all the
        'fields for the cancelled booking ID to 0, so that it will
        not 'appear again. This is a limitation, the record is not
        actually 'deleted but only set to blank. It works for the
        purposes of 'this test however.

        BookingDataSource.Update()

        'Reload the page, sending an indication that a booking has
      been
        'cancelled.
        Response.Redirect("Account.aspx?Cancel=Yes")

    End Sub

</script>



<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <b>Account</b>

 <br />
     <br />
     <asp:Label ID="lblCancelConf" runat="server" Text=""
Visible="False">
     <br />
     <br />
     <b>Personal Details:</b>
    </asp:Label><asp:Label ID="lblUserNo" runat="server" Text="Label"
Visible="False"></asp:Label>
      
    <br />
    <asp:DetailsView ID="DetailsView1" runat="server" Height="50px"
Width="240px"
        DataSourceID="PersonalDetailsDataSource"
        AutoGenerateRows="False" DataKeyNames="UserID"
GridLines="None" >
        <Fields>
           <asp:CommandField ButtonType="Button" EditText="Edit
Personal Details"
                ShowEditButton="True" UpdateText="OK" />
```

```
            <asp:BoundField DataField="UserName" HeaderText="User
Name:"
                SortExpression="UserName" />
            <asp:BoundField DataField="Password"
HeaderText="Password:"
                SortExpression="Password" />
            <asp:BoundField DataField="Forename"
HeaderText="Forename:"
                SortExpression="Forename" />
            <asp:BoundField DataField="Surname" HeaderText="Surname:"
                SortExpression="Surname" />
            <asp:BoundField DataField="HouseNumber" HeaderText="House
Number:"
                SortExpression="HouseNumber" />
            <asp:BoundField DataField="Postcode"
HeaderText="Postcode:"
                SortExpression="Postcode" />
            <asp:BoundField DataField="Email" HeaderText="E-mail:"
                SortExpression="Email" />
            <asp:BoundField DataField="CardNumber" HeaderText="Card
Number:"
                SortExpression="CardNumber" />

            <asp:BoundField DataField="Issuer" HeaderText="Issuer"
                SortExpression="Issuer" />
            <asp:BoundField DataField="SecurityCode"
HeaderText="Security Code:"
                SortExpression="SecurityCode" />

        </Fields>

    </asp:DetailsView>
    <br />

    <asp:Label ID="lblOrderNo" runat="server" Text="Label"
Visible="false"></asp:Label>
    <b>Bookings</b>

    <br />
    <br />
   <asp:Label ID="lblNoBookings" runat="server"
        Text="You currently have no bookings"
Visible="false"></asp:Label>

    <asp:DataList ID="DataList1" runat="server"
DataSourceID="BookingDataSource"
         OnItemDataBound="Item_Created"
OnItemCommand="btnCancel_Click"
        GridLines="Both">
        <ItemTemplate>
            <asp:Button ID="btnCancel" runat="server"
CommandName="Select"
                Text="Cancel this order" />
            <br />
            Booking
ID:           
 
            <asp:Label ID="BookingIDLabel" runat="server" Text='<%#
Eval("BookingID") %>' />
            <br />
```

```
            Flight
No:           
    
            <asp:Label ID="FlightNoLabel" runat="server" Text='<%#
Eval("FlightNo") %>' />
            <br />
            Departing Airport:   
            <asp:Label ID="DepartingAirportLabel" runat="server"
                Text='<%# Eval("DepartingAirport") %>' />
            <br />
            Departing Time:      
            <asp:Label ID="DepartingTimeLabel" runat="server"
                Text='<%# Eval("DepartingTime") %>' />
            <br />
            Departing
Date:       <asp:Label
ID="DepartingDateLabel" runat="server"
                Text='<%# Eval("DepartingDate") %>' />
            <br />
            Destination Airport: 
            <asp:Label ID="DestinationAirportLabel" runat="server"
                Text='<%# Eval("DestinationAirport") %>' />
            <br />
            Return
Time:          
 <asp:Label ID="ReturnTimeLabel" runat="server"
                Text='<%# Eval("ReturnTime") %>' />
            <br />
            Return
Date:          &nbs
p;
            <asp:Label ID="ReturnDateLabel" runat="server"
                Text='<%# Eval("ReturnDate") %>' />
            <br />

            <asp:Label ID="dispHotelLabel" runat="server"
Text="Hotel:"></asp:Label>

           &nb
sp;         <asp:Label
ID="HotelLabel" runat="server" Text='<%# Eval("Hotel") %>' />
            <br />
            <asp:Label ID="dispCityLabel" runat="server"
Text="City:"></asp:Label>

           &nb
sp;          
            <asp:Label ID="CityLabel" runat="server" Text='<%#
Eval("City") %>' />
            <br />
            <asp:Label ID="dispStarsLabel" runat="server"
Text="Stars:"></asp:Label>

           &nb
sp;         
            <asp:Label ID="StarsLabel" runat="server" Text='<%#
Eval("Stars") %>' />
            <br />
            <asp:Label ID="dispCostLabel" runat="server"
```

```
                Text="Cost: &amp;nbsp; &amp;nbsp; &amp;nbsp;
&amp;nbsp; &amp;nbsp; &amp;nbsp; &amp;nbsp; &amp;nbsp; &amp;nbsp;
&amp;nbsp; &amp;nbsp;£"></asp:Label>
            <asp:Label ID="CostLabel" runat="server" Text='<%#
Eval("Cost") %>' />
            <br />
            <br />
        </ItemTemplate>
    </asp:DataList>

    <br />
    <br />

    <br />
    <br />

    <asp:AccessDataSource ID="PersonalDetailsDataSource"
runat="server"
        DataFile="~/SiteDataBase.mdb"


        SelectCommand="SELECT UserID, UserName, [Password], Forename,
Surname, HouseNumber, Postcode, Email, CardNumber, Issuer,
SecurityCode FROM Users WHERE (UserID = ?)"

        UpdateCommand="UPDATE Users SET UserName =@UserName,
[Password] = @Password, Forename = @Forename, Surname = @Surname,
HouseNumber = @HouseNumber, Postcode = @Postcode , Email = @Email,
CardNumber = @CardNumber, Issuer = @Issuer, SecurityCode =
@SecurityCode WHERE UserID = @UserID">
        <SelectParameters>
            <asp:ControlParameter ControlID="lblUserNo" Name="UserID"
                PropertyName="Text" Type="Int32" />
        </SelectParameters>
        <UpdateParameters>

        <asp:Parameter Name="UserName" Type="String" />
        <asp:Parameter Name="Password" Type="String" />
        <asp:Parameter Name="Forename" Type="String" />
        <asp:Parameter Name="Surname" Type="String" />
        <asp:Parameter Name="HouseNumber" Type="Int32"  />
        <asp:Parameter Name="Postcode" Type="String" />
        <asp:Parameter Name="Email" Type="String" />
        <asp:Parameter Name="CardNumber" Type="String" />
        <asp:Parameter Name="Issuer" Type="String" />
        <asp:Parameter Name="SecurityCode" Type="Int32" />

        </UpdateParameters>
    </asp:AccessDataSource>
    <asp:AccessDataSource ID="BookingDataSource" runat="server"
        DataFile="~/SiteDataBase.mdb"




        SelectCommand="SELECT Flights.FlightNo,
Flights.DepartingAirport, Flights.DepartingTime,
Flights.DepartingDate, Flights.DestinationAirport,
Flights.ReturnTime, Flights.ReturnDate, Hotels.Hotel, Hotels.City,
Hotels.Stars, Bookings.Cost, Bookings.BookingID FROM ((Bookings INNER
```

```
JOIN Flights ON Bookings.FlightID = Flights.FlightNo) INNER JOIN
Hotels ON Bookings.RoomID = Hotels.RoomID) WHERE (Bookings.UserID =
?)"

        UpdateCommand="UPDATE Bookings SET UserID = 0, FlightID = 0,
RoomID = 0, Cost = 0 WHERE (BookingID = @orderNo)">
        <SelectParameters>
            <asp:ControlParameter ControlID="lblUserNo" Name="UserID"
PropertyName="Text"
                Type="String" />
        </SelectParameters>
        <UpdateParameters>
        <asp:ControlParameter ControlID="lblOrderNo" Name="orderNo"
PropertyName="Text"
                Type="Int32" />
        </UpdateParameters>
    </asp:AccessDataSource>
    <br />
    </asp:Content>
```

Basket.aspx

The purpose of this page is to provide a section of the site where the user can view selected items, change an order and check details before making a commitment to purchasing.

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
Title="Untitled Page" %>

<script runat="server">
    Dim flightNo As Integer
    Dim roomNo As Integer
    Dim flightCookie As HttpCookie = New HttpCookie("flight")
    Dim roomCookie As HttpCookie = New HttpCookie("room")
    Dim roomCostCookie As HttpCookie = New HttpCookie("roomCost")
    Dim roomNightsCookie As HttpCookie = New HttpCookie("roomNights")
    Dim firstDay As String
    Dim lastDay As Integer
    Dim dateDiff As Integer
    Dim roomPrice As Integer

    Sub Page_Load()
       'Execute this code each time the page loads.

        'Get the value of the selected flight number and room number
        'from their respective cookies and set the text values of two
        'labels to those numbers.

        flightNo = Request.Cookies("flight")("flightID")
        roomNo = Request.Cookies("room")("roomID")

        lblFlightNo.Text = flightNo
        lblroomNo.Text = roomNo

       'If no flight has been selected, ensure the button for
removing
       'a flight from the basket are invisible.

        If flightNo = 0 Then
            btnFlightRemove.Visible = False
        Else
            btnFlightRemove.Visible = True
            btnFlightBrowse.Visible = True
        End If









        'If a flight and a room have been selected. Calculate how
        many   'days a hotel room is required for. Get the date of
        departure 'and date of return, cut the string down to the
        first two 'characters so that only the day is displayed, ie
        24 as opposed 'to 24/05/08. Then calculate the difference
```

```
        between the two 'days and take away 1. Get the price per
        night of one room and 'multiply that by the amount of nights
        the user is going to 'stay for and display that to the user.
        A limitation of this is 'that it will only work if the user
        is departing and coming 'back within the same month.

        If flightNo > 0 And roomNo > 0 Then
            firstDay =
Convert.ToInt32(Left(CType(DataList1.Items.Item(0).FindControl("Depar
tingDateLabel"), Label).Text.ToString, 2))
            lastDay =
Convert.ToInt32(Left(CType(DataList1.Items.Item(0).FindControl("Retur
nDateLabel"), Label).Text.ToString, 2))

            dateDiff = (lastDay - firstDay) - 1

            lblDays.Text = dateDiff

            roomPrice =
CType(DataList2.Items.Item(0).FindControl("PricePerNightLabel"),
Label).Text * dateDiff

            lblHotelTotalCost.Text = roomPrice

        'If neither a room or flight has been selected, display a
        'message saying that the basket is empty and hide the
        checkout 'button.

        ElseIf flightNo = 0 And roomNo = 0 Then
            lblEmpty.Visible = True
            btnCheckOutGo.Visible = False
        End If

        'If only a flght has been selected, hide all the information
        'labels pertaining the hotel, and display a link to the
      browse
        'hotel page. If a room has been selected then display the
        'remove room button.

        If roomNo = 0 Then
            btnRoomRemove.Visible = False
            lblDays.Visible = False
            lblDaysDisplay.Visible = False
            lblHotelCostDisplay.Visible = False
            lblHotelTotalCost.Visible = False
        Else
            btnRoomRemove.Visible = True
            btnFlightBrowse.Visible = False
        End If


    End Sub

    Protected Sub btnCheckOutGo_Click(ByVal sender As Object, ByVal e
As System.EventArgs)
       'This code is executed when the user chooses to checkout. The
        'hotel information is written to a cookie and the user is
        'redirected to the login page. If they are already logged in
        'they will automatically be redirected to the checkout.

        roomCostCookie("cost") = roomPrice
```

```
        roomNightsCookie("nights") = dateDiff
        Response.Cookies.Add(roomCostCookie)
        Response.Cookies.Add(roomNightsCookie)
        Response.Redirect("Login.aspx?Origin=Basket")
    End Sub

    Protected Sub btnFlightRemove_Click(ByVal sender As Object, ByVal
e As System.EventArgs)
       'This is executed when the user chooses to remove a flight
       from 'their basket. It sets the value of the selected flight
       to 0 and 'writes that to a cookie, the page then reloads.

        flightCookie("flightID") = "0"
        flightCookie.Expires = Now.AddDays(1)
        Response.Cookies.Add(flightCookie)
        Response.Redirect("Basket.aspx")
    End Sub

    Protected Sub btnRoomRemove_Click(ByVal sender As Object, ByVal e
As System.EventArgs)
       'This code is executed when the user chooses to remove a room
       'from their basket. Same idea as the flight removal, but this
       'code has 'a few more details to set to 0.
        roomCookie("roomID") = "0"
        roomCostCookie("cost") = "0"
        roomNightsCookie("nights") = "0"

        roomCookie.Expires = Now.AddDays(1)
        roomCostCookie.Expires = Now.AddDays(1)
        roomNightsCookie.Expires = Now.AddDays(1)
        Response.Cookies.Add(roomCookie)
        Response.Cookies.Add(roomCostCookie)
        Response.Cookies.Add(roomNightsCookie)
        Response.Redirect("Basket.aspx")
    End Sub

    Protected Sub btnFlightBrowse_Click(ByVal sender As Object, ByVal
e As System.EventArgs)
        'Redirects the user to the browse hotel page.
        Response.Redirect("BrowseHotel.aspx")
    End Sub
</script>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">

    <b>Basket</b>
    <br />
    <asp:Label ID="lblEmpty" runat="server" Text="Basket is currently
empty"></asp:Label>
     <asp:HyperLink ID="HyperLink1" runat="server"
        NavigateUrl="~/Default.aspx">Go Back</asp:HyperLink>
     <asp:Label ID="lblFlightNo" runat="server" Text="Label"
Visible ="false"></asp:Label>
       
```

```
    <asp:Label ID="lblroomNo" runat="server" Text="Label"
Visible="false"></asp:Label>
    <br />
<asp:AccessDataSource ID="RoomDataSource" runat="server"
    DataFile="SiteDataBase.mdb"
    SelectCommand="SELECT [Hotel], [City], [PricePerNight], [Stars]
FROM [Hotels] WHERE ([RoomID] = ?)">
    <SelectParameters>
        <asp:ControlParameter ControlID="lblroomNo" Name="RoomID"
PropertyName="Text"
            Type="Int32"  />
    </SelectParameters>
</asp:AccessDataSource>
    <asp:AccessDataSource ID="flightDataSource" runat="server"
        DataFile="~/SiteDataBase.mdb"


        SelectCommand="SELECT [DepartingAirport], [DepartingTime],
[DepartingDate], [DestinationAirport], [ReturnTime], [ReturnDate],
[Price], [FlightNo] FROM [Flights] WHERE ([FlightNo] = ?)">
        <SelectParameters>
            <asp:ControlParameter ControlID="lblFlightNo"
Name="FlightNo"
                PropertyName="Text" Type="Int32" />
        </SelectParameters>
    </asp:AccessDataSource>
    <asp:DataList ID="DataList1" runat="server"
DataKeyField="FlightNo"
        DataSourceID="flightDataSource" Width="283px">
        <ItemTemplate>
            Departing Airport:    
            <asp:Label ID="DepartingAirportLabel" runat="server"
                Text='<%# Eval("DepartingAirport") %>' />
            <br />
            Departing Time:       
            <asp:Label ID="DepartingTimeLabel" runat="server"
                Text='<%# Eval("DepartingTime", "{0:t}") %>' />
            <br />
            Departing Date:       
            <asp:Label ID="DepartingDateLabel" runat="server"
                Text='<%# Eval("DepartingDate", "{0:d}") %>' />
            <br />
            Destination Airport:  
            <asp:Label ID="DestinationAirportLabel" runat="server"
                Text='<%# Eval("DestinationAirport") %>' />
            <br />
            Return
Time:          &nbs
p;
            <asp:Label ID="ReturnTimeLabel" runat="server"
                Text='<%# Eval("ReturnTime", "{0:t}") %>' />
            <br />
            Return
Date:          &nbs
p;
            <asp:Label ID="ReturnDateLabel" runat="server"
                Text='<%# Eval("ReturnDate", "{0:d}") %>' />
            <br />

Price:          &nb
sp;          
```

```
            <asp:Label ID="PriceLabel" runat="server"
                Text='<%# Eval("Price", "{0:C}") %>' />
            <br />
            Flight
No:         
      
            <asp:Label ID="FlightNoLabel" runat="server" Text='<%#
Eval("FlightNo") %>' />
        </ItemTemplate>
    </asp:DataList>
    <br />

    <asp:Button ID="btnFlightRemove" runat="server"
        Text="Remove Flight from Basket"
onclick="btnFlightRemove_Click"
        Visible="False" Width="195px" />
    <br />
    <br />
    <asp:LinkButton ID="btnFlightBrowse" runat="server"
        Text="Browse for a hotel room" visible="false"
        onclick="btnFlightBrowse_Click" Width="195px"/>
    <asp:DataList ID="DataList2" runat="server"
DataSourceID="RoomDataSource"
        Height="16px" Width="309px">
        <ItemTemplate>

Hotel:         &nb
sp;          
    
            <asp:Label ID="HotelLabel" runat="server" Text='<%#
Eval("Hotel") %>' />
            <br />

City:         &nbs
p;          &
nbsp;     
            <asp:Label ID="CityLabel" runat="server" Text='<%#
Eval("City") %>' />
            <br />
            Nightly
price:         &nb
sp;     £<asp:Label ID="PricePerNightLabel"
runat="server"
                Text='<%# Eval("PricePerNight")%>' />.00
            <br />

Stars:         &nb
sp;          
    
            <asp:Label ID="StarsLabel" runat="server" Text='<%#
Eval("Stars") %>' />
        </ItemTemplate>
</asp:DataList>
    <br />
    <asp:Label ID="lblDaysDisplay" runat="server" Text="Nights
staying at hotel: "></asp:Label>
    <asp:Label ID="lblDays" runat="server" Text="Label"></asp:Label>

    <br />
    <asp:Label ID="lblHotelCostDisplay" runat="server" Text="Total
Hotel
```

```
Cost:         £"></asp:L
abel>
    <asp:Label ID="lblHotelTotalCost" runat="server"
Text="Label"></asp:Label>

    <br />
    <br />
    <asp:Button ID="btnRoomRemove" runat="server"
        Text="Remove room from Basket" onclick="btnRoomRemove_Click"
        Visible="False" />
    <br />
    <br />
<asp:Button ID="btnCheckOutGo" runat="server" Text="Go to Check Out"
        onclick="btnCheckOutGo_Click"/>
<br />


     
</asp:Content>
```

BrowseFlight.aspx

This page allows the user to choose a flight based on where they choose to depart from and where they choose to arrive.

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
Title="Untitled Page" EnableViewStateMac="false"%>
<script runat="server">
        'There are two drop down boxes with the available
        destinations ‘available. When they are changed, they cause
        the page to ‘automatically refresh to display the flights
        that match the ‘criteria the user selected.


    Dim loginID As String

    Protected Sub Page_Transfer(ByVal sender1 As Object, ByVal e As
EventArgs)
        'This subroutine is executed when the user choose a flight.
        It ‘grabs the flight number from the web control, writes that
        ‘number to a cookie and redirects the web browser to the
        basket ‘page.

        Dim flightNo As Integer

        flightNo =
CType(DataList1.SelectedItem.FindControl("FlightNoLabel"),
Label).Text

        Dim flightCookie As HttpCookie = New HttpCookie("flight")

        flightCookie("flightID") = flightNo
        flightCookie.Expires = Now.AddDays(1)
        Response.Cookies.Add(flightCookie)

        Response.Redirect("Basket.aspx")
    End Sub

</script>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
<b>Browse for Flights</b>
    <br />
    <br />

    Choose place of departure:<br />
    <asp:DropDownList ID="DropDownList1" runat="server"
AutoPostBack="True">
        <asp:ListItem>Glasgow</asp:ListItem>
        <asp:ListItem>Manchester</asp:ListItem>
        <asp:ListItem>Dublin</asp:ListItem>
        <asp:ListItem>London</asp:ListItem>
        <asp:ListItem>Paris</asp:ListItem>
        <asp:ListItem>Prestwick</asp:ListItem>
    </asp:DropDownList>
    <br />
    <br />
    Choose destination:<br />
    <asp:DropDownList ID="DropDownList2" runat="server"
AutoPostBack="True">
        <asp:ListItem>Glasgow</asp:ListItem>
        <asp:ListItem>Manchester</asp:ListItem>
        <asp:ListItem>Dublin</asp:ListItem>
        <asp:ListItem>London</asp:ListItem>
        <asp:ListItem>Paris</asp:ListItem>
        <asp:ListItem>Prestwick</asp:ListItem>
    </asp:DropDownList>



    <asp:AccessDataSource ID="AccessDataSource1" runat="server"
        DataFile="~/SiteDataBase.mdb"

        SelectCommand="SELECT FlightNo, DepartingAirport,
DepartingTime, DepartingDate, DestinationAirport, ReturnTime,
ReturnDate, Price FROM Flights WHERE (DepartingAirport = ?) AND
(DestinationAirport = ?)">
        <SelectParameters>
            <asp:ControlParameter ControlID="DropDownList1"
Name="DepartingAirport"
                PropertyName="SelectedValue" Type="String" />
            <asp:ControlParameter ControlID="DropDownList2"
Name="DestinationAirport"
                PropertyName="SelectedValue" />
        </SelectParameters>
    </asp:AccessDataSource>
    <br />

    <br />
    <asp:DataList ID="DataList1" runat="server"
DataKeyField="FlightNo"
        DataSourceID="AccessDataSource1" Font-Bold="False" Font-
Italic="False"
        Font-Overline="False" Font-Strikeout="False" Font-
Underline="False"
        ForeColor="Black" GridLines="Horizontal"
        OnSelectedIndexChanged="Page_Transfer" Width="226px">
        <ItemTemplate>
```

```
            Flight
No:           
     
            <asp:Label ID="FlightNoLabel" runat="server" Text='<%#
Eval("FlightNo") %>' />
            <br />
            Departing Airport:   
            <asp:Label ID="DepartingAirportLabel" runat="server"
                Text='<%# Eval("DepartingAirport") %>' />
            <br />
            Departing Time:      
            <asp:Label ID="DepartingTimeLabel" runat="server"
                Text='<%# Eval("DepartingTime", "{0:t}") %>' />
            <br />
            Departing Date:      
            <asp:Label ID="DepartingDateLabel" runat="server"
                Text='<%# Eval("DepartingDate", "{0:d}") %>' />
            <br />
            Destination Airport: 
            <asp:Label ID="DestinationAirportLabel" runat="server"
                Text='<%# Eval("DestinationAirport") %>' />
            <br />
            Return
Time:          &nbs
p;
            <asp:Label ID="ReturnTimeLabel" runat="server"
                Text='<%# Eval("ReturnTime", "{0:t}") %>' />
            <br />
            Return
Date:          &nbs
p; 
            <asp:Label ID="ReturnDateLabel" runat="server"
                Text='<%# Eval("ReturnDate", "{0:d}") %>' />
            <br />

Price:          &nb
sp;           
£<asp:Label ID="PriceLabel" runat="server" Text='<%# Eval("Price")
%>' />
            <br />

                <asp:Button id="SelectButton"
           Text="Add to basket"
                CommandName="Select"
                runat="server"
                />
        </ItemTemplate>
        <ItemStyle BorderStyle="Solid" />
    </asp:DataList>
    <br />

    </asp:Content>
```

BrowseHotel.aspx

The purpose of this page is to display a list of hotels based on what destination city the user chooses.

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
Title="Untitled Page" EnableViewStateMac="false" %>

<script runat="server">
    'There is a drop down box with a list of cities, when the user
makes a choice the page automatically refreshes and displays the list
of hotels that pertain to the city choice.

    Sub Page_Transfer(ByVal sender As Object, ByVal e As
System.EventArgs)
        'This code fires when the user has chosen a room. It writes
        the 'room number of the selected room to a cookie and
        redirects the 'user to the basket page.

        Dim roomCookie As HttpCookie = New HttpCookie("room")
        Dim roomNo As Integer

        roomNo =
CType(DataList1.SelectedItem.FindControl("lblRoomNo"), Label).Text

        roomCookie("roomID") = roomNo
        roomCookie.Expires = Now.AddDays(1)
        Response.Cookies.Add(roomCookie)

        Response.Redirect("Basket.aspx")
    End Sub
</script>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
<b>Browse for Hotels</b>
<br />
<br />
Choose destination:
<br />
    <asp:DropDownList ID="DropDownList1" runat="server"
AutoPostBack="True">
        <asp:ListItem>Glasgow</asp:ListItem>
        <asp:ListItem>Manchester</asp:ListItem>
        <asp:ListItem>Dublin</asp:ListItem>
        <asp:ListItem>London</asp:ListItem>
        <asp:ListItem>Paris</asp:ListItem>
        <asp:ListItem>Prestwick</asp:ListItem>
    </asp:DropDownList>
    <br />
    <br />
    <asp:AccessDataSource ID="AccessDataSource1" runat="server"
        DataFile="~/SiteDataBase.mdb"

        SelectCommand="SELECT [Hotel], [City], [PricePerNight],
[Stars], [RoomID] FROM [Hotels] WHERE ([City] = ?)">
        <SelectParameters>
            <asp:ControlParameter ControlID="DropDownList1"
Name="City"
                PropertyName="SelectedValue" Type="String" />
```

```
        </SelectParameters>
    </asp:AccessDataSource>
    <asp:DataList ID="DataList1" runat="server"
DataSourceID="AccessDataSource1"
        OnSelectedIndexChanged="Page_Transfer" Width="229px">
        <ItemTemplate>
            <asp:Label ID="lblRoomNo" runat="server" Text='<%#
Eval("RoomID", "{0}") %>'
                Visible="False"></asp:Label>
            <br />

Hotel:         &nb
sp;  
            <asp:Label ID="HotelLabel" runat="server" Text='<%#
Eval("Hotel") %>' />
            <br />

City:         &nbs
p;    
            <asp:Label ID="CityLabel" runat="server" Text='<%#
Eval("City") %>' />
            <br />
            PricePerNight: £<asp:Label ID="PricePerNightLabel"
runat="server"
                Text='<%# Eval("PricePerNight") %>' />
            <br />

Stars:         &nb
sp;  
            <asp:Label ID="StarsLabel" runat="server" Text='<%#
Eval("Stars") %>' />
            <br />
           <asp:Button id="LinkButton1"
           Text="Add to basket" CommandName="Select" runat="server"/>
            <br />
        </ItemTemplate>
        <ItemStyle BorderStyle="Solid" />
    </asp:DataList>
</asp:Content>
```

Checkout.aspx

The purpose of this page is to allow the user to review their choice, the total price and allow them to back out if they made a mistake. The last button on this page allows them to commit to the booking.

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
Title="Untitled Page" EnableViewStateMac="false" %>

<script runat="server">
    Dim roomNo As Integer
    Sub Page_Load()
        'Execute this code each time the page loads.
        Dim flightNo As Integer

        Dim userNo As Integer
        Dim roomCost As Integer
        Dim flightPrice As String
        Dim totalCost As Integer

      'Grab the userID, flightID, roomID and room cost from their
      'resepective cookies and set the values to text on specific
      'labels.
        userNo = Request.Cookies("login")("userID")
        flightNo = Request.Cookies("flight")("flightID")
        roomNo = Request.Cookies("room")("roomID")

        roomCost =
Convert.ToInt16(Request.Cookies("roomCost")("cost"))

        lblFlightNo.Text = flightNo
        lblroomNo.Text = roomNo
        lblUserNo.Text = userNo

        lblRoomPrice.Text = roomCost

        'If a room was not chosen, hide all the labels pertaining to
        'room booking data.

        If roomNo = 0 Then
            lblRoomPrice.Visible = False
            lblRoomPriceDisp.Visible = False
            lblFlightPrice.Visible = False
            lblFlightPriceDisp.Visible = False
        End If

        'Calculate the total cost by getting the flight cost from the
        'flight web control, add that value to the hotel cost and
        'display the total price at the bottom.

        flightPrice =
CType(DataList3.Items.Item(0).FindControl("PriceLabel"), Label).Text
        lblFlightPrice.Text = flightPrice
        totalCost = roomCost + flightPrice
        lblTotalPrice.Text = totalCost
    End Sub
    Protected Sub btnComplete_Click(ByVal sender As Object, ByVal e
As System.EventArgs)
        'This code is fired when the user chooses to complete the
        'booking
```

```
        Dim roomCookie As HttpCookie = New HttpCookie("room")

        'If a room was no selected, set the value of the room number
        to '22. This is place holder hotel record. It prevents the
        account 'page from crashing so that it doesn't have to try
        and display 'a hotel value that does not exist. The account
        page will 'detect that it is a place holder and hide it
        automatically. 'This number is then written to a cookie. A
        new database record 'is created with the userID, the selected
        flightID, roomID and 'total cost. The user is then redirected
        to the receipt page 'with an order confiramtion.

        If roomNo = 0 Then
            roomNo = 22
            lblroomNo.Text = roomNo
            roomCookie("roomID") = roomNo
            roomCookie.Expires = Now.AddDays(1)
            Response.Cookies.Add(roomCookie)
        End If
        BookingDataSource.Insert()

        Response.Redirect("Receipt.aspx")
    End Sub

    Protected Sub btnBack_Click(ByVal sender As Object, ByVal e As
System.EventArgs)
        'This sub takes the user back to the basket.
        Response.Redirect("Basket.aspx")
    End Sub
</script>



<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <b>Checkout</b>

    <asp:AccessDataSource ID="BookingDataSource" runat="server"
        DataFile="~/SiteDataBase.mdb"
        InsertCommand="INSERT INTO Bookings(UserID, FlightID, RoomID,
Cost) VALUES (@userID,@flightID,@roomID,@totalPrice)"
        SelectCommand="SELECT * FROM [Bookings]">
        <InsertParameters>
            <asp:ControlParameter DefaultValue="0" Name="userID"
ControlID="lblUserNo"
                PropertyName="Text" />
            <asp:ControlParameter ControlID="lblFlightNo"
DefaultValue="0" Name="flightID"
                PropertyName="Text" />
            <asp:ControlParameter ControlID="lblroomNo"
DefaultValue="22" Name="roomID"
                PropertyName="Text" />
            <asp:ControlParameter ControlID="lblTotalPrice"
Name="totalPrice" DefaultValue="0"
                PropertyName="Text" />
        </InsertParameters>
    </asp:AccessDataSource>
<asp:AccessDataSource ID="RoomDataSource" runat="server"
    DataFile="~/SiteDataBase.mdb"
    SelectCommand="SELECT [Hotel], [City], [PricePerNight], [Stars]
FROM [Hotels] WHERE ([RoomID] = ?)">
```

```
    <SelectParameters>
        <asp:ControlParameter ControlID="lblroomNo" Name="RoomID"
PropertyName="Text"
            Type="Int32" />
    </SelectParameters>
</asp:AccessDataSource>
    <asp:AccessDataSource ID="flightDataSource" runat="server"
        DataFile="~/SiteDataBase.mdb"


        SelectCommand="SELECT [DepartingAirport], [DepartingTime],
[DepartingDate], [DestinationAirport], [ReturnTime], [ReturnDate],
[Price], [FlightNo] FROM [Flights] WHERE ([FlightNo] = ?)">
        <SelectParameters>
            <asp:ControlParameter ControlID="lblFlightNo"
Name="FlightNo"
                PropertyName="Text" Type="Int32" />
        </SelectParameters>
    </asp:AccessDataSource>
        <asp:Label ID="lblFlightNo" runat="server" Text="Label"
Visible="false"></asp:Label>
      
    <asp:Label ID="lblroomNo" runat="server" Text="Label"
Visible="false"></asp:Label>
    <br />
    <br />
    <asp:Label ID="lblUserNo" runat="server" Text="Label"
Visible="false" ></asp:Label>

    <asp:DataList ID="DataList3" runat="server"
DataKeyField="FlightNo"
        DataSourceID="flightDataSource">
        <ItemTemplate>
            Departing Airport:   
            <asp:Label ID="DepartingAirportLabel" runat="server"
                Text='<%# Eval("DepartingAirport") %>' />
            <br />
            Departing Time:       
            <asp:Label ID="DepartingTimeLabel" runat="server"
                Text='<%# Eval("DepartingTime", "{0:t}") %>' />
            <br />
            Departing Date:       
            <asp:Label ID="DepartingDateLabel" runat="server"
                Text='<%# Eval("DepartingDate", "{0:d}") %>' />
            <br />
            Destination Airport: 
            <asp:Label ID="DestinationAirportLabel" runat="server"
                Text='<%# Eval("DestinationAirport") %>' />
            <br />
            Return
Time:          &nbs
p;
            <asp:Label ID="ReturnTimeLabel" runat="server"
                Text='<%# Eval("ReturnTime", "{0:t}") %>' />
            <br />
            Return
Date:          &nbs
p;
            <asp:Label ID="ReturnDateLabel" runat="server"
                Text='<%# Eval("ReturnDate", "{0:d}") %>' />
            <br />
```

```
            Flight
No:         
     
            <asp:Label ID="FlightNoLabel" runat="server" Text='<%#
Eval("FlightNo") %>' />
            <br />
            <asp:Label ID="PriceLabel" runat="server" Text='<%#
Eval("Price") %>'
                Visible="False" />
            <br />
        </ItemTemplate>
    </asp:DataList>



    <asp:DataList ID="DataList2" runat="server"
DataSourceID="RoomDataSource">
        <ItemTemplate>

Hotel:          &nb
sp;         
            <asp:Label ID="HotelLabel" runat="server" Text='<%#
Eval("Hotel") %>' />
            <br />

City:          &nbs
p;           
            <asp:Label ID="CityLabel" runat="server" Text='<%#
Eval("City") %>' />
            <br />
            Nightly
price:         
            <asp:Label ID="PricePerNightLabel" runat="server"
                Text='<%# Eval("PricePerNight", "{0:C}") %>' />
            <br />

Stars:          &nb
sp;         
            <asp:Label ID="StarsLabel" runat="server" Text='<%#
Eval("Stars") %>' />

        </ItemTemplate>
</asp:DataList>
    <br />
    <asp:Label ID="lblFlightPriceDisp" runat="server" Text="Flight
Price:          
£"></asp:Label>
    <asp:Label ID="lblFlightPrice" runat="server"
Text="Label"></asp:Label>
    <br />
    <asp:Label ID="lblRoomPriceDisp" runat="server" Text="Room Price:
         £" />
    <asp:Label ID="lblRoomPrice" runat="server"
Text="Label"></asp:Label>
    <br />
    <br />
    Total
Price:          &nb
sp;£<asp:Label ID="lblTotalPrice" runat="server"
Text="Label"></asp:Label>
    <br />
```

```
    <br />
    <asp:Button ID="btnBack" runat="server" onclick="btnBack_Click"
        Text="Go back to basket" />
    <br />
    <br />
    <asp:Button ID="btnComplete" runat="server"
onclick="btnComplete_Click"
        Text="Complete Booking" />
    <br />
    </asp:Content>
```

Default.aspx

The purpose of this page is to provide a home page for the user.

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
Title="Untitled Page" EnableViewStateMac="false"%>

<script runat="server">

    Protected Sub lnkLogin_Click(ByVal sender As Object, ByVal e As
System.EventArgs)
        'This takes the user to the login page, indicating to the
        login 'page that the user came from the home page.
        Server.Transfer("~/Login.aspx?Origin=Account")
    End Sub
</script>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <b>Home Page</b><br />
    <br />
    <asp:HyperLink ID="lnkFlights" runat="server"
    NavigateUrl="~/BrowseFlight.aspx">Look for
Flights</asp:HyperLink>
 <br />
<br />
<asp:HyperLink ID="lnkHotels" runat="server"
NavigateUrl="~/BrowseHotel.aspx">Look
for Hotels</asp:HyperLink>
<br />
<br />
    <asp:LinkButton ID="lnkLogin" runat="server"
onclick="lnkLogin_Click">View/edit
    Account and Booking Details</asp:LinkButton>
<br />
 <br />
<asp:HyperLink ID="lnkBasket" runat="server"
NavigateUrl="~/Basket.aspx">Basket</asp:HyperLink>
    <br />
<br />
</asp:Content>
```

Login.aspx

Other pages re-direct to this page if they require credentials and the user is not logged in. This page also gives the user the option to create an account.

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
Title="Untitled Page" EnableViewStateMac="false"%>

<script runat="server">
    Dim origin As String
    Dim topLogIn As String

    Sub Page_load()
        'When the page loads…
        Dim loggedIn As String

        'If the login cookie exists, set the value of loggedIn to the
        'userID stored in the cookie.
        If (Request.Cookies("login") IsNot Nothing) Then

            loggedIn = Request.Cookies("login")("UserID")

            'If the userID is not 0, which means someone is logged
            in, 'send the user to the account page if that was their
            'destination. Otherwise send them to the checkout. Note
            'that if this code fires the user will not see the login
            'page at all but will be sent to their destination
            straight 'away.

            If loggedIn <> 0 Then

                If Request.QueryString("Origin") = "Account" Then
                    Response.Redirect("Account.aspx")
                Else
                    Response.Redirect("Checkout.aspx")
                End If

            End If
        End If
    End Sub

    Sub reg_link()
        'This is fired when the user clicks on the registration link.
        'If they came from the basket page, send them to registration
        'but send them back to the basket when they are done.
        'Otherwise, send them to the home page when they are done.

        If Request.QueryString("Origin") = "Basket" Then
            Response.Redirect("Registration.aspx?Origin=Basket")
                Else
            Response.Redirect("Registration.aspx?Origin=Home")
        End If


    End Sub
    Protected Sub Page_Transfer(ByVal sender As Object, ByVal e As
System.EventArgs)
        'This code fires when the user hits the login button. The
        'details entered into the username and password fields as
```

```
        well 'as where the user came from. The page is then
        redirected to a 'verification page.

        Dim username As String
        Dim password As String

        origin = Request.QueryString("Origin")
        topLogIn = Request.QueryString("TopLogIn")

        username = txtUserName.Text
        password = txtPassword.Text

        Response.Redirect("LogInCheck.aspx?TopLogIn=" + topLogIn +
"&Origin=" + origin + "&Username=" + username + "&Password=" +
password)

    End Sub
</script>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <b>Log in</b><br />
    <br />
    User name:    <asp:TextBox ID="txtUserName"
runat="server"></asp:TextBox>
<br />
    Password: 
<asp:TextBox ID="txtPassword" runat="server" TextMode="Password"
></asp:TextBox>
    <br />
<br />
<asp:Button ID="btnLogin" runat="server" Text="Login"
onclick="Page_Transfer" />
    <br />
    <br />
 <asp:linkButton ID="lnkLoginRegister" runat="server"
        onclick="reg_link">Create an Account</asp:linkButton>
 - An account is required to make bookings. Creating an account
allows you
to create bookings in future without having to re-enter all of your
details.<br />
    <br />
    </asp:Content>
```

LoginCheck.aspx

The purpose of this page is to verify login details. If the login was successful the user will never see this page. If it was not then this page will tell them to try again.

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
Title="Untitled Page" %>

<script runat="server">
    Dim loginCookie As HttpCookie = New HttpCookie("login")
    Dim origin As String
    Dim username As String
    Dim password As String
    Dim userID As Integer

    Sub Page_Load()
        'When the page loads, get where the user came from and the
        'login credentials they entered, and set the values of labels
        'to those credentials.
        origin = Request.QueryString("Origin")

        username = Request.QueryString("Username")
        password = Request.QueryString("Password")

        lblUsername.Text = username
        lblPassword.Text = password

        'When this page loads it grabs the values of the login
        'credentials and carries out a database query to determine if
        a 'record exists. If a record does not exist then text
        appears 'telling them so. If the record does exist, the
        loginCookie has 'the userID of the record written to it and
        the user is 'redirected to a page depending on where they
        came from.

        If GridView1.Rows.Count = 0 Then
            lblNotFound.Visible = True
            btnPreviousPage.Visible = True

        Else
            lblNotFound.Visible = False
            btnPreviousPage.Visible = False

            userID = GridView1.Rows.Item(0).Cells(0).Text

            loginCookie("UserID") = userID
            loginCookie.Expires = Now.AddDays(1)
            Response.Cookies.Add(loginCookie)

            If Request.QueryString("Origin") = "RegBasket" Then
                Response.Redirect("Basket.aspx")
            End If

            If Request.QueryString("Origin") = "Basket" Then
                Response.Redirect("Checkout.aspx")
            ElseIf Request.QueryString("Origin") = "Account" Then
                Response.Redirect("Account.aspx")
            ElseIf Request.QueryString("Origin") = "Home" Then
                Response.Redirect("Default.aspx")
```

```
            Else
                Response.Redirect(origin)
            End If

            End If

    End Sub


    Protected Sub btnPreviousPage_Click(ByVal sender As Object, ByVal
e As System.EventArgs)
        'This code fires when the user clicks the Try again button
        'after a failed login, it takes them back to the login page.
        Response.Redirect("Login.aspx?Origin=" + origin)
    End Sub
</script>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <b>Login failed</b>
    <br />
    <br />
    <asp:Label ID="lblNotFound" runat="server" Text="Log in details
do not match any details on file"
        Visible="False"></asp:Label>
    <br />
    <br />
    <asp:Button ID="btnPreviousPage" runat="server" Text="Try again"
        Visible="False" onclick="btnPreviousPage_Click" />
    <p>
        <br />
        <asp:Label ID="Label1" runat="server" Text="Label"
Visible="false"></asp:Label>
    </p>
    <asp:GridView ID="GridView1" runat="server"
DataSourceID="LoginDataSource"
        AutoGenerateColumns="False" DataKeyNames="UserID"
Visible="false">
        <Columns>
            <asp:BoundField DataField="UserID" HeaderText="UserID"
InsertVisible="False"
                ReadOnly="True" SortExpression="UserID" />
            <asp:BoundField DataField="UserName"
HeaderText="UserName"
                SortExpression="UserName" />
            <asp:BoundField DataField="Password"
HeaderText="Password"
                SortExpression="Password" />
        </Columns>
    </asp:GridView>
    <div>

        <asp:AccessDataSource ID="LoginDataSource" runat="server"
            DataFile="~/SiteDataBase.mdb"
            SelectCommand="SELECT [UserID], [UserName], [Password]
FROM [Users] WHERE (([UserName] = ?) AND ([Password] = ?))">
            <SelectParameters>
                <asp:ControlParameter ControlID="lblUsername"
Name="UserName"
                    PropertyName="Text" Type="String" />
```

```
                <asp:ControlParameter ControlID="lblPassword"
Name="Password"
                    PropertyName="Text" Type="String" />
            </SelectParameters>
        </asp:AccessDataSource>

    </div>
    <asp:Label ID="lblUsername" runat="server" Text="User Name"
Visible="False"></asp:Label>
    <br />
    <asp:Label ID="lblPassword" runat="server" Text="Password"
Visible="false"></asp:Label>
    <br />
    <br />

    <br />

</asp:Content>
```

Receipt.aspx

The purpose of this page is to provide a confirmation to the user that their booking has gone through successfully.

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
Title="Untitled Page" %>

<script runat="server">
    Sub Page_load()
        'When the page loads…
        Dim userID As Integer
        Dim roomID As Integer
        Dim flightID As Integer

        'Grab the values of the userID, roomID and flightID from
their
        'respective cookies and set them to a label. A query is then
        'run and the database returns the bookingID that matches all
      of
        'those details.

        userID = Request.Cookies("login")("userID")
        roomID = Request.Cookies("room")("roomID")
        flightID = Request.Cookies("flight")("flightID")

        lblUserNo.Text = userID
        lblRoomNo.Text = roomID
        lblFlightNo.Text = flightID

        lblBookingID.Text =
CType(DataList1.Items.Item(0).FindControl("BookingIDLabel"),
Label).Text

        Dim roomCookie As HttpCookie = New HttpCookie("room")
        roomCookie("roomID") = 0
        roomCookie.Expires = Now.AddDays(1)
        Response.Cookies.Add(roomCookie)

    End Sub
</script>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <b>Order Succesful!<br />
    </b>
    <asp:Label ID="lblUserNo" runat="server" Text="Label"
Visible="false"></asp:Label>
 <asp:Label ID="lblRoomNo" runat="server" Text="Label"
Visible="false"></asp:Label>
 
    <asp:Label ID="lblFlightNo" runat="server" Text="Label"
Visible="false"></asp:Label>
 <asp:AccessDataSource ID="BookingDataSource" runat="server"
        DataFile="~/SiteDataBase.mdb"
        SelectCommand="SELECT [BookingID] FROM [Bookings] WHERE
(([FlightID] = ?) AND ([RoomID] = ?) AND ([UserID] = ?))">
        <SelectParameters>
            <asp:ControlParameter ControlID="lblFlightNo"
Name="FlightID"
```

```
                PropertyName="Text" Type="Int32" />
            <asp:ControlParameter ControlID="lblRoomNo" Name="RoomID"
PropertyName="Text"
                Type="Int32" />
            <asp:ControlParameter ControlID="lblUserNo" Name="UserID"
PropertyName="Text"
                Type="Int32" />
        </SelectParameters>
    </asp:AccessDataSource>
    <br />
    Your order number is:
    <asp:Label ID="lblBookingID" runat="server"
Text="Label"></asp:Label>
    <br />
    <br />
    A confirmation has been sent to via email in addition to further
details. If you
    wish to view or cancel your order, please visit the account
page.<br />
    <br />
    <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/Default.aspx">Home
    Page</asp:HyperLink>
    <br />
    <asp:DataList ID="DataList1" runat="server"
DataKeyField="BookingID"
        DataSourceID="BookingDataSource" Visible="False">
        <ItemTemplate>
            BookingID:
            <asp:Label ID="BookingIDLabel" runat="server" Text='<%#
Eval("BookingID") %>' />
            <br />
            <br />
        </ItemTemplate>
    </asp:DataList>
    <br />
</asp:Content>
```


Daffy Duck
20060xxxx

Registration.aspx

The purpose of this page is to allow the user to create an account, as well as providing some basic validation.

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
Title="Untitled Page" EnableViewStateMac="false" %>

<script runat="server">


    Protected Sub cmdRegister_Click(ByVal sender As Object, ByVal e
As System.EventArgs)
        'This code is fired when the user clicks the register button.
The code carries out validation on the text fields, checking to see
if they fall within certain parameters (like the credit card number
being the correct length or some of the fields missing data.). If any
of these checks fail a red error message will appear.
        If txtCardNo.Text.Length <> 16 Then
            lblCardNo.Visible = True
        ElseIf txtSecurity.Text.Length <> 3 Then
            lblSecurity.Visible = True
        ElseIf txtCardNo.Text = "" Or txtForename.Text = "" Or
txtHouseNo.Text = "" Or txtPassword.Text = "" Or
txtPasswordConfirm.Text = "" Or txtEmail.Text = "" Or
txtPostcode.Text = "" Or txtSecurity.Text = "" Or txtSurname.Text =
"" Or txtUserName.Text = "" Then
            lblCompleted.Visible = True
        Else
            lblSecurity.Visible = False
            lblCardNo.Visible = False
            lblCompleted.Visible = False

            'Once all the details have validated, create a new
            database 'record with those details and redirect the user
            to the registration confimration page.
            AccessDataSource1.Insert()

            lblUserID.Text = txtUserName.Text

            If Request.QueryString("Origin") = "Basket" Then
                Response.Redirect("RegSuccess.aspx?Origin=Basket")
            Else
                Response.Redirect("RegSuccess.aspx?Origin=Home")
            End If

        End If
    End Sub

</script>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">

        <b>Personal Details:</b><p>
            <asp:Label ID="lblCompleted" runat="server"
ForeColor="Red"
                Text="*All fields must be completed."
Visible="False"></asp:Label>
        </p>
```

```
        <p>

Forename:          
      
            <asp:TextBox ID="txtForename"
runat="server"></asp:TextBox>
        </p>
    <p>

Surname:          &
nbsp;               <asp:TextBox
ID="txtSurname" runat="server"></asp:TextBox>
        </p>
    <p>
        House number:       
         <asp:TextBox ID="txtHouseNo" runat="server"
Width="30px"></asp:TextBox>
        </p>
    <p>

Postcode:          
               <asp:TextBox
ID="txtPostcode" runat="server"></asp:TextBox>
         </p>
    <p>
        E-
mail:          &nbs
p;          
        <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
        </p>
    <p>
        <b>Payment details:</b></p>
    <p>
        Card
number:          &n
bsp;
        <asp:TextBox ID="txtCardNo" runat="server"></asp:TextBox>
        <asp:Label ID="lblCardNo" runat="server" ForeColor="#FF3300"
            Text="*Card number must be 16 digits long"
Visible="False"></asp:Label>
        <p>

Issuer:          &n
bsp;           
;             <asp:DropDownList ID="cmbIssuer" runat="server">
                <asp:ListItem>Maestro</asp:ListItem>
                <asp:ListItem>Visa</asp:ListItem>
                <asp:ListItem>MasterCard</asp:ListItem>
                <asp:ListItem>American Express</asp:ListItem>
            </asp:DropDownList>
        <p>
        Security
code:          
<asp:TextBox ID="txtSecurity" runat="server"
Width="30px"></asp:TextBox>
         <asp:Label ID="lblSecurity" runat="server"
                Text="*Security code must be 3 digits long"
ForeColor="Red"
                Visible="False"></asp:Label>
                  <p>
                      <b>Login Credentials:</b><p>
```

```
                    Choose a username:  <asp:TextBox
ID="txtUserName" runat="server"></asp:TextBox>
        <p>
            Choose a password:  <asp:TextBox ID="txtPassword"
TextMode="Password" runat="server"></asp:TextBox>
        <p>
            Confirm password:    <asp:TextBox
ID="txtPasswordConfirm" TextMode="Password"
runat="server"></asp:TextBox>
        <p>
            <asp:Button ID="cmdRegister" runat="server" Text="Create
Account"
                onclick="cmdRegister_Click" />
          <asp:Label ID="lblUserID" runat="server"
Text="UserID" Visible ="false"></asp:Label>
        <p>
            <asp:AccessDataSource ID="AccessDataSource1"
runat="server"
                DataFile="~/SiteDataBase.mdb"
                InsertCommand="INSERT INTO Users(UserName,
[Password], Forename, Surname, Postcode, HouseNumber, Email,
CardNumber, SecurityCode, Issuer) VALUES
(@userName,@password,@foreName,@surname,@postCode,@houseNumber,@email
,@cardNumber,@securityCode,@issuer)"
                SelectCommand="SELECT UserID ">
                <SelectParameters>
                    <asp:ControlParameter ControlID="txtUserName"
Name="?" PropertyName="Text"
                        DefaultValue="0" />
                </SelectParameters>
                <InsertParameters>
                    <asp:ControlParameter ControlID="txtUserName"
Name="userName"
                        PropertyName="Text" />
                    <asp:ControlParameter ControlID="txtPassword"
Name="password"
                        PropertyName="Text" />
                    <asp:ControlParameter ControlID="txtForename"
Name="foreName"
                        PropertyName="Text" />
                    <asp:ControlParameter ControlID="txtSurname"
Name="surname"
                        PropertyName="Text" />
                    <asp:ControlParameter ControlID="txtPostcode"
Name="postCode"
                        PropertyName="Text" />
                    <asp:ControlParameter ControlID="txtHouseNo"
Name="houseNumber"
                        PropertyName="Text" />
                    <asp:ControlParameter ControlID="txtEmail"
Name="email"
                        PropertyName="Text" />
                    <asp:ControlParameter ControlID="txtCardNo"
Name="cardNumber"
                        PropertyName="Text" />
                    <asp:ControlParameter ControlID="txtSecurity"
Name="securityCode"
                        PropertyName="Text" />
                    <asp:ControlParameter ControlID="cmbIssuer"
Name="issuer"
                        PropertyName="SelectedValue" />
```

```
        </InsertParameters>

    </asp:AccessDataSource>

</asp:Content>
```

RegSuccess.aspx

This page lets the user know their registration was successful and tells them to login to continue.

```
<%@ Page Language="VB" MasterPageFile="~/MasterPage.master"
Title="Untitled Page" %>

<script runat="server">

    Protected Sub btnLogin_Click(ByVal sender As Object, ByVal e As
System.EventArgs)
        'This code fires when the user presses the login button on
        this 'page as opposed to the main one. Using this button
        allows them 'to return to where they were. It checks where
        they came from 'and sends them to another page based on where
        they came from.
        If Request.QueryString("Origin") = "Basket" Then

Response.Redirect("Login.aspx?TopLogin=Yes&Origin=RegBasket")
        Else
            Response.Redirect("Login.aspx?TopLogin=Yes&Origin=Home")
        End If
    End Sub
</script>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <b>Account Creation Successful!</b><br />
    <br />
    Please log-in, using the button below, to return to where you
were.
    <br />
    <br />
    <asp:Button ID="btnLogin" runat="server" Text="Log In"
        onclick="btnLogin_Click" />
    </asp:Content>
```