UNIVERSITY OF HERTFORDSHIRE
Faculty of Science, Technology and the Creative Arts

Modular BSc Honours in Information Technology

6COM0285 – Information Technology Project

Final Report
April 2012

Air Masriya: A Web-Based Airline Reservation System

N.Al-Aidroos

Acknowledgments

I would like to thank the following people for their help in the production of this project:

All of the following testers and validators and my Tutor, without whom it would not have been possible to find the errors in the system and my report:

Mr Faisal Rezwan
Arqam Mohammed
Bagashy Yussuf
Priscilla Thompson
Cleon Davidson
Sangar Agha
Agha Saam
Alex Christou

And lastly my parents: Habib Ibrahim & Saba Ibrahim; without whom I would not be where I am today.

Abstract

Trends in the airline industry are forcing changes upon airlines to be part of a competition of low cost carriers. Hence why there is a growth within the e commerce airline industry. To survive this I felt that airlines will need to focus on costs, price and profit, therefore to meet these requirements using IT infrastructure to transform the airline company Air Masriya I felt the need to create a low cost e commerce Airline Reservation System (ARS) to act as a transformation for companies to become more agile and profitable businesses.

The web based airline reservation system project is going to be my attempt to create the basic concepts of airline reservation system. My main aim was to create this reservation system with a competitive advantage of allowing customer to see statistical information on the flight they are about to book. E.g. How busy the flight will be and how booked it is.

Although I was not able to accomplish that particular task; I was still able to create this reservation system with some functionality and an appealing design.

Table of Contents

Appendices

Figures Table

Chapter 1 – Introduction

Introduction

Airline Reservation System used to be standalone systems, Each Airline company has its own system and these were disconnected from other airlines and or ticket agents, which meant only a designated administration staff were able to use it.

Companies in the air service sector are continuously working closely to manage costs while at the same time trying to improve their operational performance and overall customer service experience online. Which is why I have decided to create Air Masriya; an Airline Reservation System (ARS) for a small company which have yet to create e commerce – reservation system enabling customer to be able to reserve and potentially buy their bookings online.

Consolidation and modernization are seen as a key requirement for all web applications that are seeking to transform their online businesses into more competitive and agile organisations. For why, this is an additional reason as I want to incorporate my study to create a modernised web application. I intend to incorporate and enhance the outlook of flexibility and interoperability within my application, in terms of the airline industry context.

The service is realized through web services implementation using php as the programming language, published into PHPmyadmin as the registry using WAM P Server, and then to be composited using Dreamweaver, into a complete web-based airline reservation services.

The web based airline reservation system project is going to be my attempt to create the basic concepts of airline reservation system whilst adding on the competitive advantage of allowing customer to see statistical information on the flight they are about to book. E.g. How busy the flight will be and how booked it is.

The system will enable the customer to do the things such as search for flights for two cities on a specified date; choose a flight based on the details, reservation of the flight and then view there booking through a booking reference number which will be shown to them at the end of their transaction.

The system allows the airline passenger to search for flights that are available between the two travel cities, named the Departure City and Arrival city for a particular departure and arrival date.

The system displays all the flight details such as flight no, name, price and duration of the journey. If the customer will be travelling with extra passengers such as additional adults, children or infants this will be taken into account by the system and their cost details will also be shown.

Once the search is complete the system will display the list of available flights and allows the user to choose a particular flight. The system will then check for the availability of seats on the flight. If the seats are available then the system allows the passenger to book a seat; Otherwise it will ask the user to choose another flight.

To book a flight the system asks the customer to enter their login details. If the user has not already registered with the airline they will firstly have to do this and then book the flight. Once the booking is confirmed the user is also allowed view his/her reservation.

Aim of Project

The main aim of the project is to develop a website which would facilitate the reservation of online air tickets through an effective interface for a customer intending to travel on Air Masriya. Apart from reserving tickets from the customer's side; Administration will also have access to the back end of the application and be able to manage areas of the website that the customer will be able to see, for example the flights that are available, and also the customer information including reservations that have been made.

Scope of Project

The Reservation System is an application that assists the airline - Air Masriya with transactions relating to making ticket reservations, which includes reserving and cancelling tickets, whilst also providing all information for flights.

System Objectives

The reservation system is a software application that will assist an airline with transactions related to making ticket reservations which will include blocking, reserving and cancelling tickets.

Objectives of the system from the airlines point of view are:

1. To minimize any repetitive work undertaken through administrators.
2. Maintain customer information in case of emergency E.g. If a flight is cancelled.
3. The application will also be used by the administrators to track patterns in flights in order to distinguish peak and off peak periods.
4. To aware customers of offers and discounts
5. Customers should be able to choose departure date and be shown available flights for a day nearest to that date for the particular city they have chosen to fly to.
6. To show that fare stays consistent during entire transaction. No additional charges are added.
7. The price of the ticket should depend on the arrival place and departure date.
8. Show all possible combinations and itineraries available for a pair of origin-destination cities

From an IT specialists perspective the following objectives should be undertaken:

9. Widget's such as buttons, menus, scroll bars and text boxes should allow the user to be able to anticipate its behavior.
10. Users must be able to anticipate the behavior of the application by using their own gained knowledge from other applications e.g. Good GUI design.

Report Structure

Chapter 2 – gives background information and research carried out on similar Airline Reservation Systems (ARS)

Chapter 3 - defines the design of the ARS, displaying the Use Case and ER diagrams that were used to create the system

Chapter 4 – describes the implementation of the ARS in addition to the challenges that occurred whist developing the system.

Chapter 5 – describes the testing process that was carried out, ensuring the system is error free and successfully works.

Chapter 6 – evaluates the system and the work that has been completed. This chapter will contain information on what I have achieved in my aims and objectives and describes the challenges I faced and the experience I have gained in completing the system.

Chapter2 – Background Research and Analysis

Introduction

In this section I will explain the background research that I have conducted which will help me to come to a conclusion of what factors I need to take into consideration whilst implementing my Airline Reservation System. The topics will discuss the Research and Investigation that took place, background research on similar systems which are current in the airline industry and how they differentiate, accessibility and usability issues which are regarded in the web industry. In addition to these, I will also examine a literature review and discuss the importance of the methodology I have chosen to use in order to develop my system.

Research and Investigation

Whilst planning this project, I have been carrying out background research into online airline reservation systems to see what functionality they contain, what their user interface is like and how the processes are carried out. Also to investigate in what ways they have made their websites usable and accessible to the end user I conducted my research by looking at the following companies:

- Egyptair
- Emirates

As an IT specialist I will be focusing on the usability and accessibility of the websites named above and how they have made their system user friendly; whilst also concentrating on the consistency levels of the website. This will help me in considering what the user interface of my web application will be like and how I can make my application have a competitive advantage from the others in terms of the user interface

Literature Review

Egyptair

Egyptair's web application system is at a very high standard. When the URL for the website has been entered the user is taken to a welcome page where they are able to choose what language they want to view the website in and also what country they are viewing the website from.

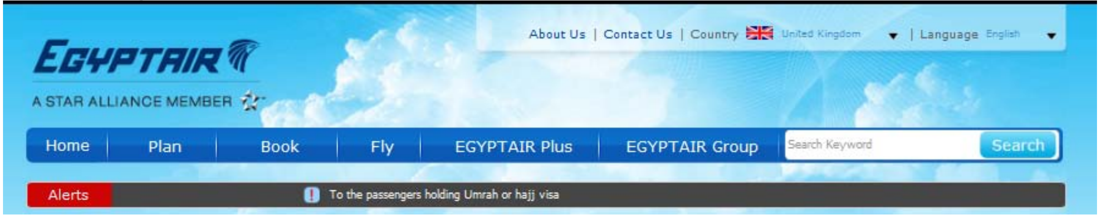

About Us | Contact Us | Country United Kingdom | Language English
EGYPTAIR
A STAR ALLIANCE MEMBER
Home
Plan
Book
Fly
EGYPTAIR Plus
EGYPTAIR Group
Search Keyword
Search
Alerts
To the passengers holding Umrah or hajj visa

Figure 1 - Egyptair Header - Usability

This is a very good tool that the company has used as it allows the users accessibility and usability of the website and targets a broad range of users. Once the user has selected their country and language they are taken to the main home page of the website where they are able to search for available flights through a search form by selecting details such as:

Return Flight → One-way Departure Date → Return Date To → From

Emirates

Emirates web application system is impeccable. They have a very welcoming feeling about their system and have made it possible for the user, as soon as they enter the website to be able to carry out the necessary search functions that a user would want to.

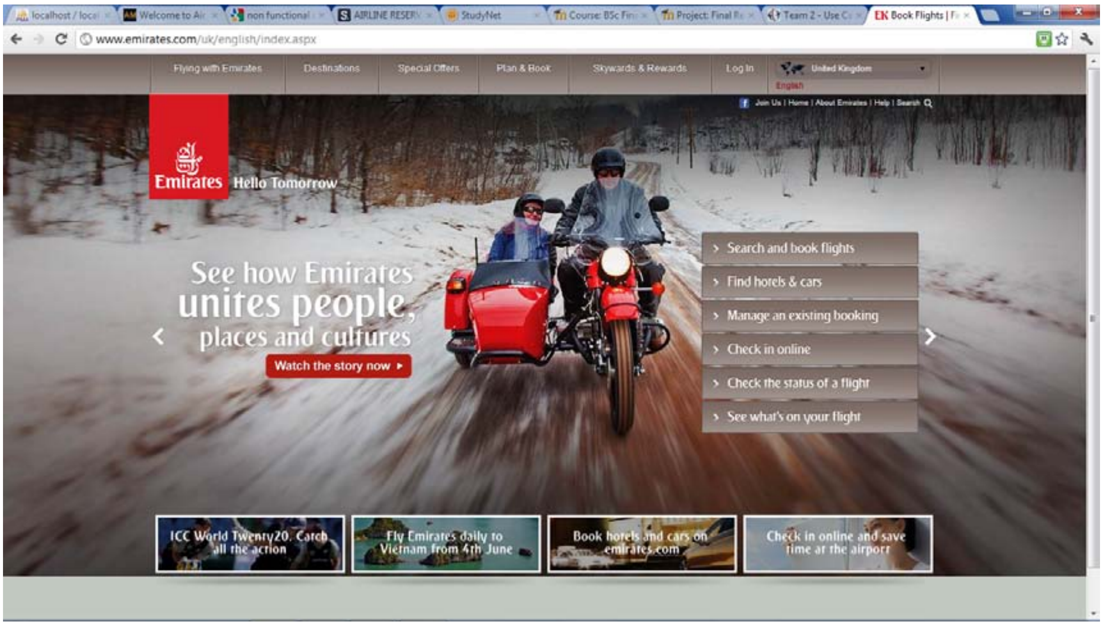

www.emirates.com/uk/english/index.aspx
Flying with Emirates
Destinations
Special Offers
Plan & Book
Skywards & Rewards
Log In
United Kingdom
English
Join Us | Home | About Emirates | Help | Search
Emirates Hello Tomorrow
See how Emirates unites people, places and cultures
Watch the story now ▸
Search and book flights
Find hotels & cars
Manage an existing booking
Check in online
Check the status of a flight
See what's on your flight
ICC World Twenty20. Catch all the action
Fly Emirates daily to Vietnam from 4th June
Book hotels and cars on emirates.com
Check in online and save time at the airport

Figure 2 - Emirates website Home Page - Literature Review

The way Emirates have designed there website allows the user access to every main functionality on the main page. On the right hand side of their homepage they have another navigation structured menu with all the functionality related to holidays and flight.

Emirates website is one that has caught my eye. And has inspired me to design my web application in the same way; to be clear and precise. And to have the main functionality within 'arms reach' of the user.

Why is Accessibility and Usability Important?

Ethical Perspective

As society of today is tending to interact more and more with the internet and its services, usability and accessibility of these services need to be at its highest standard to carry on reducing the digital divide - by making these services accessible it is a way of helping people achieve their independence.

According to (Zaphiris, 2003) the number of people with disabilities is expected to increase significantly as the world's population is rapidly growing older. The elderly population is becoming increasingly important for the tourist industry as they often have more time and the finance to spend as tourists.

Business Perspective

Ensuring that all users have access to websites and making sure that they are usable; is likely to reach specified audiences and also likely to increase a higher percent in market share. It is important to take into account that the reason for building the website in the first place is so that it reaches a larger audience; if accessibility and usability aren't taken into account then you could say that there is no need for the website itself. This is a factor which should be considered by all companies in usually the design phase.

Technology Perspective

Accessibility is something which should be tested at a much larger scale than just the usual browsers such as Microsoft Internet Explorer or the newer edition of Google Chrome. Not only are browsers something which need to be taken into consideration but also with the latest technology out; hand held Personal Digital Assistants and Smart phones should also be thought of when in the design process. As technology advances it may be possible to be allow websites to reach yet even more people and to develop even better solutions for accessibility.

Methodologies

Choosing the right software development methodology is a very important aspect of the entire project. These methodologies are considered as a very crucial step in recognising different system behaviors and also a way of simplifying the analysis required to create/check or improve quality project/software.

There are 4 things which need to be taken into account when choosing a methodology:

- Risk
- Time to market
- Budget
- Stability of the requirements

The most common approach of building a website is to create individual web pages using programming languages such as PHP and HTML.

In order for me to begin developing my project I need to establish which sort of methodology will be of best use for production of this web application. I have decided to go with the waterfall model as my methodology. This will make the understanding of my project and design a simpler procedure as documentation is a key property whilst also checking the application coding; testing will be carried out at the end of every major stage.

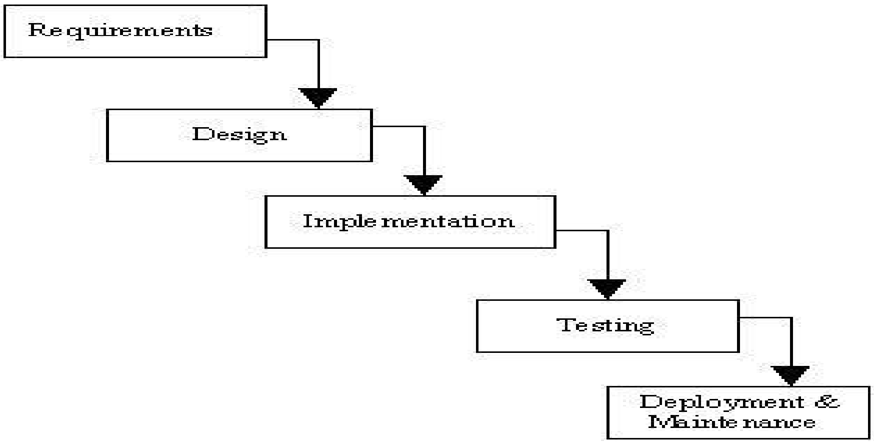


Figure 3- Waterfall Diagram

Requirements Analysis

This section is going to help me to collect and analyse the ideas gathered for the project.

Overview

The Airline Reservation System project is an implementation of an Airline Ticketing website for Air Masriya which enables customers to search the availability and prices of various flight fares, along with the different packages available for bookings for return flights. This project also covers various features for example a secure login for administration/staff to be able to update/delete or add customer details or flights information. In general, this website would be designed to perform like any other airline ticketing website available online but hopefully with a competitive advantage.

Features

The Airline Booking System has the following features:

This application is intended for two types of audiences. One is the customer/ the end user and the other is the administrator/management staff of the website. Some of the major functions of the application can be put under two different categories that are for the administrator and the user.

Home Page: Like all the other airline websites available online, the user can access the user home page of the Airline Booking System website, after he/she logs into the application. Here they will be able to look up latest information regarding flight offers and their fares.

Login and Register: The Airline Booking System also comes with the customer registration details page, where the customer can enter their details and register. They can also create a username and password. Additionally they will also be able to modify their personal information in case of a change in for example their e-mail address or telephone number, etc.

Booking Flights: The customer can also search for the flights available and reserve their place on the flight by making a booking.

Contact the Company: The Customer can also call the company if they have any queries or questions related to the bookings they have made online.

Booking Instructions: The website also provides instructions to the customers on how to book airline tickets.

Some of the functions of the System, such as creating, maintaining and updating the database are available only to the administrator/ management staff.

The functions of the administrator are explained in detail are as follows:

ADMINISTRATORS ACTIVITIES

Login/Logout: The administrator has to login first in order to be able to make changes to the Airline Booking System; adding, deleting or modifying the data in the Airline Reservation System database.

Add/Delete/Modify Customer Information: The Airline Booking System will have many customers registering with the website every day and many of them will most probably want to 'unsubscribe'. Only the administrator or management staff will be allowed to modify the database accordingly.

Add/Delete/Modify Flight Information: The Administrator will also be able to add, delete or modify the flight information. Sometimes, flights get cancelled, so those particular flights would need to be removed from the list of flights available to the customer on the customer interface. Similarly whenever any flight information has to be modified or if any new flights need to be added to the database, the administrator will need to perform the above stated activity.

Cancellation of Bookings: Sometimes, after making a booking, a customer might want to cancel the booking they have made. In a real life situation this will not be allowed by most airlines as a business rule.
Therefore, I have only made it possible for the administrator to carry out this function so they can deal with these sorts of situations and send the customer an e-mail confirmation after deleting the specific booking or transaction, if needs be.

E-mail confirmations: Whenever a customer makes or cancels a booking, administration will be responsible for sending confirmation e-mails to the customer, confirming the transaction.

CONSTRAINTS - CORRECT THIS

When changes are made to the database, the application should be able to show the updated information on the website, without much delay. The database for the project is designed to be of moderate size. The application is going to be designed to be able to run in all popular web browsers such an Internet Explorer, Google Chrome and Firefox.
I will be using PHP, HTML and CSS technology to code the project and PhpMyAdmin will act as the database for the project.

Requirements

Functional Requirements

The functional requirements of the Airline Reservation System are divided among the customer and the administrator of the application. These functional requirements can be explained in detail as follows:

Use Case Diagrams can also be seen in **Appendix C**

1. **Use Case name: User Registration**
   Description: This use case describes the scenario where the user registers with the application by providing all the necessary details, in order to make a booking for flights.

   Actor: User or the Customer

   Input: The user or the customer will have to provide all the necessary details present in the customer registration form of the application.

   Output: All the details entered in the customer registration page will be verified and accepted by the system into the database.

2. **Use Case name: User Login**
   Description: This use case describes the scenario where the user logs into the application, with the username and password he has provided while registering with the system.

Actor: User or the Customer

Input: The user or the customer creates a username and password at the time of registering with the system. He then uses them to logon to the system and make bookings or view any previously entered information.

Output: The application then verifies the authenticity of the username and password that the customer has provided and allows the user to view the information available on the system, if the username and password are valid.

3. **Use Case name: Contact the company**
   Description: This use case describes the scenario where the user contacts the company for any information.

   Actor: User or the Customer

   Input: The customer can contact the airline company, requesting them for any information he needs.

   Output: The application verifies the authenticity of the username and password that the customer has provided and allows the user to view the contact information for the company.

4. **User Case name: Booking Instructions**

   Description: This use case describes the scenario where the user views the instructions for booking flights.

   Actor: User or the Customer

   Input: After the customer logs onto the application with his username and password, he can look up the instructions posted on the website for booking flights, packages or motels.

   Output: The application verifies the authenticity of the username and password and displays the how to book instructions page

5. **Use Case name: Book Flight**
   Description: This use case describes the scenario where the user books airline tickets.

   Actor: User or the Customer

   Input: After logging into the application, the customer looks up the information related to various airlines and checks the availability of seats on flights. If he finds that there are any available tickets, he then purchases them and has the option of choosing a specific seat.

Output: The application verifies the authenticity of the username and password and then displays information related to various flights to the customer

The administrator activities use cases will be described here:

6. **Use Case name: Login/Logout**
   Description: This use case describes the scenario where the administrator of the application, logs into the system and logs out after the work is done.

   Actor: Administrator

   Input: The administrator of the website logs into the application with the username and password provided to him.

   Output: The application verifies the authenticity and displays the home page of the administrator.

7. **Use Case name: Add/Delete or Modify Customer information**

   Description: This use case describes the scenario where the administrator adds, deletes or modifies customer information in the system database

   Actor: Administrator

   Input: The administrator of the applications logs onto the system with his username and password.

   Output: The application authenticates the administrator, and then displays the page where the administrator can add new customers to the database, or delete existing customers or modify details of customers in the database.

8. Use Case name: Add/Delete or Modify flight information

   Description: This use case describes the scenario where the administrator adds, deletes or modifies flight information in the application database

   Actor: Administrator

   Input: The administrator logs onto the system with the username and password provided to him.

Output: The application authenticates the administrator, by verifying the username and password. Then the application displays the page where the administrator can add new flights to the database, delete the flights that have been cancelled or modify information for the flights.

9. **Use Case name: Cancellation of Reservations**

   Description: This use case describes the scenario where the administrator handles the cancellation of reservations by the customers.

   Actor: Administrator

   Input: The administrator logs onto the system with the given username and password.

   Output: The application authenticates the administrator and then displays the page where the administrator looks up the id of the customer who has requested cancellation of booking. After cancelling the booking, the administrator then sends a confirmation e-mail to the customer.

10. **Use Case name: E-mail confirmations**

    Description: This use case describes the scenario where the administrator sends e-mail confirmations to the customers of the application.

    Actor: Administrator

    Input: The administrator logs onto the application with the username and password provided.

    Output: The application then authenticates the administrator and displays the page where the administrator can send e-mail confirmations to the customer. These e-mail confirmations may be sent in cases where the customer has cancelled a reservation or changed the personal information available on the website.

11. **Use Case name: Modifying details of webpage**

    Description: This use case describes the scenario where the administrator logs onto the application to modify the details of the airline website

    Actor: Administrator

    Input: The administrator logs onto the application with the username and password provided to him

    Output: After verifying the username and password of the administrator, the application then allows the administrator to login. The administrator can then browses through the website and change the details of any webpage in the Airline Booking application.

What it does. - Login Search Book View booking

Chapter 3 – Design

User Interface Design

EXTERNAL INTERFACES

The different types of interfaces that we would come across while developing the Airline Reservation System application are User, Hardware Interface and Software Interface.

USER INTERFACE

Initial Design

There are two types of users for the Airline Reservation System project. One is the customer and the other is the administrator. Both the customer and administrator user interface would be a graphical user interface. The graphical user interface for the customer home page would be as follows:

Figure 4 - Initial User Interface Design - Home Page

The graphical user interface for administrators would be as follows:

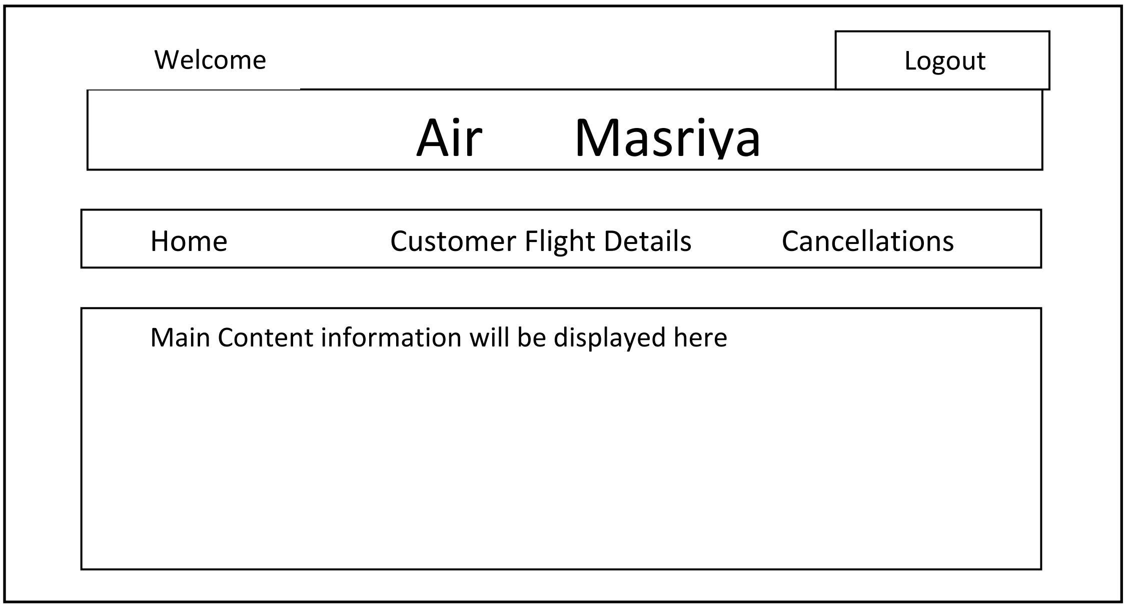


Figure 5 - Initial Administration Home Page Design

Final Design

Once I started implementing the project together I wasn't too keen on the design that I originally had. I decided to come back to the design phase to work out other alternative design approaches I could think of.

I went over my background research and came up with the idea of having a welcome page where the user would have access to a search facility for flights. This would lead them to the main purpose of the web application.

Whilst going over my background research another issue came to my thoughts that In many airline reservation systems, the user does not have to register or sign to be able to book a flight.

This made me question my initial requirements for the project. I then decided to remove the requirements of the user to register and login their details to book a flight and have no decided to allow the user to be able to book a flight but whilst coming to the end of their transaction they have to enter their main details such as name, email address and telephone number. This way the user will receive an email with their confirmed booking and they will also receive a booking reference number. This way the booking is on our database as well as in the hands of the customers via email.

Below in **Figure 6** is what I plan to have implement as part of my main page for my website:

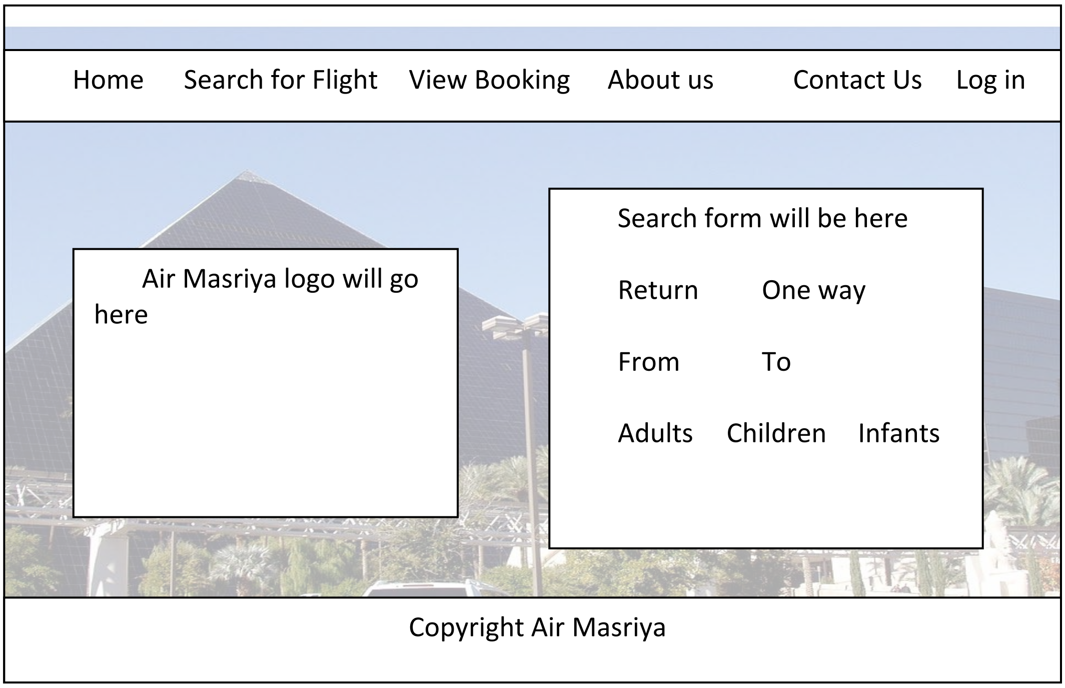
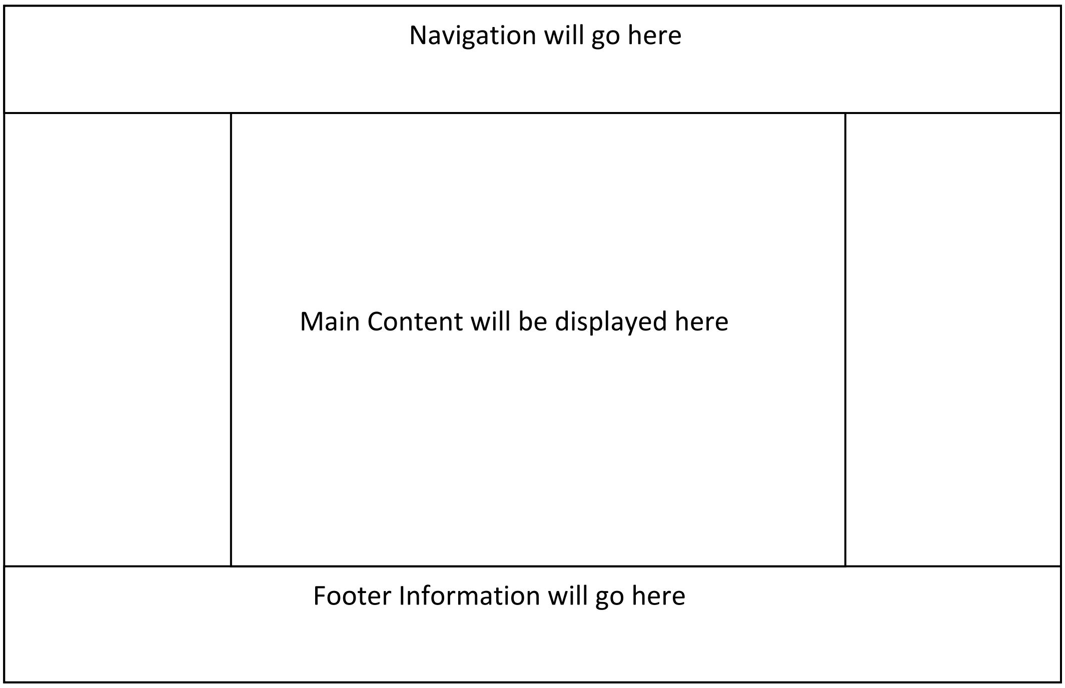


Figure 6 - Final User Interface Design Home Page

I have planned to use a color scheme consisting of browns and oranges as these are the colors that represent Egypt which is what Air Masriya stands for.

The main page is the only page that will have an image in the background. The rest of the pages will have a centered main area where all the content will be displayed. The consistency will lie between the rest of these pages whilst the navigation bar and footer will remain to be the consistent factor in all the web pages throughout the web application.

Below is the design for the rest of the user pages on the web application.


Navigation will go here
Main Content will be displayed here
Footer Information will go here

In the above design I am hoping to have the main content pop out with a sort of postcard effect look. This is because it all goes with the theme of flying abroad and matches the concept of the project.

The is another way of making the main information stand out.

The administration interface will also need to be re designed. Although I was not able to do any background research on this sort interface I felt that I should still use the same concept of the navigation bar being held at the top of the page with the usability and accessibility factors.

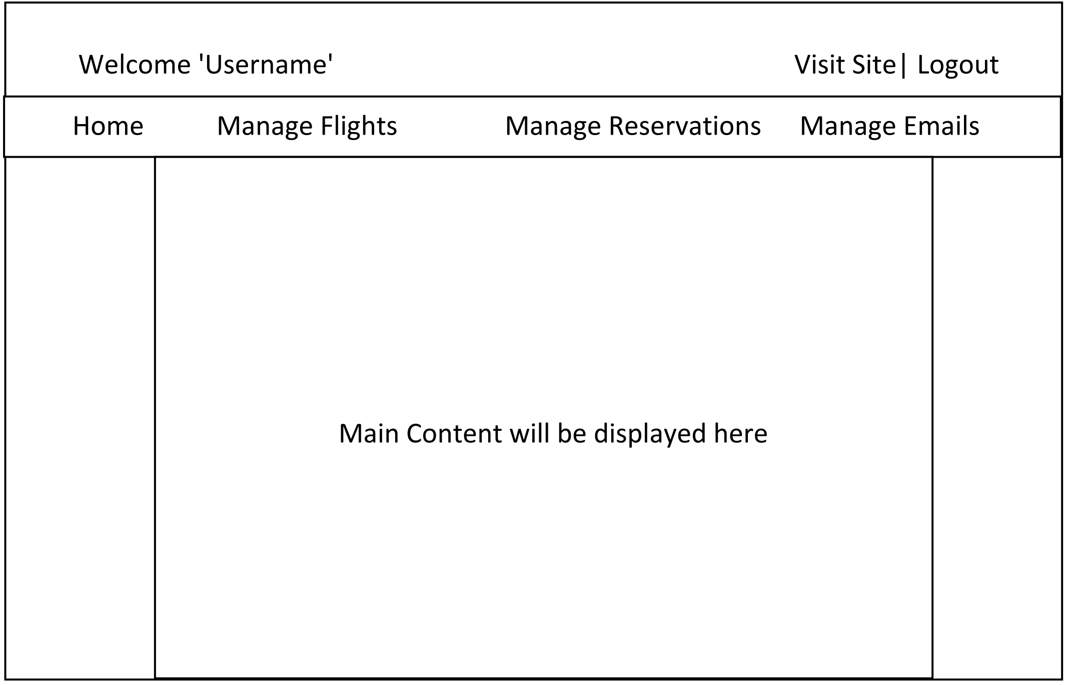


Figure 7 - Administration Home Page User Interface

I am going to use a completely clear white and blue theme for this interface as it doesn’t need to be designed in a modernised manner. It isn't being made to be pretty but to be functional and clear to use.

Database Design

Mission Statement - A Reservation system is needed to allow customers to search and book flights from one city to another, once booking has been made customer needs pay for the reservation; at the end of the transaction a booking reference should be shown to customer to allow them to view their booking.

In order for me to begin implementing my project I had to ask myself the following questions:

- What data will go into the database and what data will come out?

Whilst also taking into consideration the following:

- Everything in the database must have a specific use for it.
- Everything in the input documentation must go somewhere.
- Everything in the output documentation must come from somewhere.

Initial Database Design - ERD

Initially when designing the database I had a lot of trouble getting to know what tables to use to link the entire application together. I researched on search engines to see if anyone else has the same problem when acquiring this project.

Final Database Design - ERD

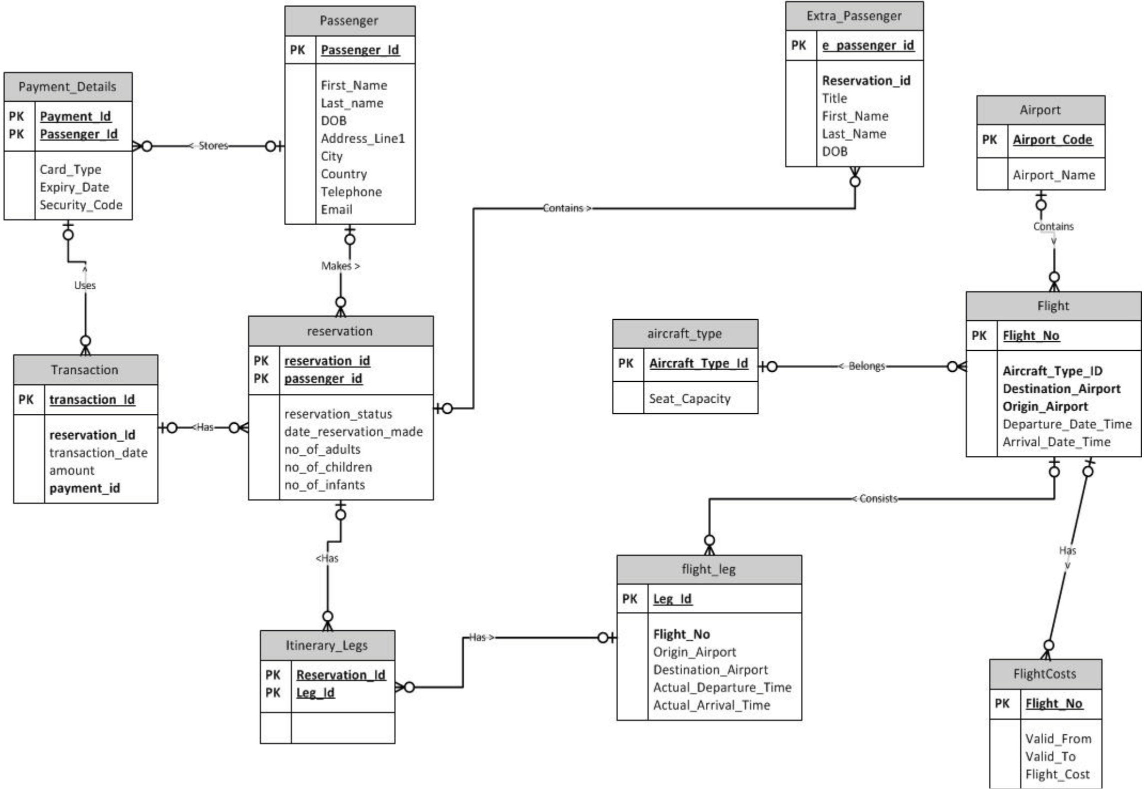
I began conducting the process of establishing what tables I would need by going through the requirements analysis. I went through this document step by step to figure out how the main people for- example the customers and administrators – would interact with the system and what tables would be needed.

When creating the ERD I began by separating the table relations into two separate ERD diagrams; the passengers interaction vs. the flight interaction with the airport .

I then linked these two ER Diagrams through **itinerary leg** table. This is where the tables are connected and the 3rd normal form is established.

Below is an image displaying the tables I have created; in accordance to the requirements analysis and by using the knowledge I gained in researching the aspects of the airline industry**.**

Figure 8 - Entity Relationship Diagram Can also be seen in Appendix A



As I am specializing in an IT degree, I know the importance that an ERD with holds. If the correct tables aren't made; whilst also having the correct relationships; when it comes to implementing the system I will have many problems in terms with the data from different tables will be connected and also to minimize data duplication. This is called normalization.

My ERD needs to be in 3rd normal form in order for it to be successful. Entities such as Extra Passengers had to be thought of carefully. The reason to why I have created this Entity is due to the reason that the Passenger table has a primary key called Passenger_ID.

If a customer chooses to make a booking for more than 1 person; we need the details of the extra passengers. Although these details cannot be entered into the passenger table as there needs to be a way to differentiate who the main passengers are who have made the booking.

Security

SQL Injection

As my system has a bank end database for storing data such as the customer details and administrative accounts; the interaction that I have with the database will be through the means of SQL queries. Quite a few of the queries I have used in the system have been assembled by the program which joins together predefined SQL language with user input.

For example the query below *(originally in the login.php file)*:

```
"SELECT ID, username FROM admin WHERE username='$u' AND password= ";
```

Figure 9 - Showing SQL query - Unsecured

```
'x' or 'a' = 'a
```

Figure 10 - Showing example injection phrase

The above code will **inject** an extra phrase into the SQL code which will then generate the code below:

```
SELECT ID, username FROM admin WHERE username = '$u' and PASSWORD = 'x'
or 'a' = 'a'
```

Figure 11 - Showing SQL injection

This will allow the potential hackers to enter the admin section of my application without the necessary authorization means.

There are several ways to prevent this from happening. The technique that I have used is to escape the username and password using the function as shown below:

Figure 12 - Showing PHP Script used in Login.php - AirMasriya

In addition to the above method I have also employed an authentication method that has password protection during input and when the connection to the database is being made.

MYSQL employs a stronger authentication method that has better password protection during the connection process than in earlier versions. It is secure even if TCP/IP packets are sniffed or the mysql database is captured. (In earlier versions, even though passwords are stored in encrypted form in the user table, knowledge of the encrypted password value could be used to connect to the MySQL server.

(dev.mysql.com)

Below is the SQL query which prevents SQL injection from taking place, to do this I have declared the password variable as a PASSWORD and also given the password field in the database a PASSWORD function.

```
        "SELECT ID, username FROM admin WHERE username='$u' AND
password=PASSWORD('$p')"
```

Figure 13 - Showing SQL code used in login.php to prevent SQL injection

Data Dictionary

Table aircraft_type

Field	Domain	Constraint	Description
aircraft_type_id	Varchar(10)	Primary Key	Type of plane. E.G. Boeing
aircraft_seat_capacity	Int(3)	Primary Key	Number of seats on aircraft.

Table airport

Field	Domain	Constraint	Description
airport_code	Varchar(3)	Primary Key	Unique ID
airport_name	Varchar(25)	Not Null	Name of Airport

Table flight

Field	Domain	Constraint	Description
Flight_no	Varchar(25)	Primary Key	Unique Identifier
Aircraft_type_id	Varchar(5)	Foreign Key	
Destination_airport	varchar(25)	Not Null	
Origin_airport	Varchar(25)	Not null	
Departure_date_time	Datetime	Not Null	
Arrival_date_time	Datetime	Not null	

Table flight_costs

Field	Domain	Constraint	Description
Flight_no	Varchar(5)	Primary Key	
Valid_from	Date	Unique Index	
Valid_to	date	Unique Index	
Flight_cost	Int(11)	Not Null	

Table Itinerary_legs

Field	Domain	Constraint	Description
Reservation_id	Varchar(11)	Composite key	
Leg_id	Int(11)	Composite key	

Table Leg

Field	Domain	Constraint	Description
leg_id	int (11)	Primary key	
Flight_no	Varchar(5)	Foreign key	
Origin_airport	Varchar(25)	Not null	
Destination_airport	Varchar(25)	Not null	
Actual_departure_time	datetime	Not null	
Actual_arrival_time	datetime	Not null	

Table Passenger

Field	Domain	Constraint	Description
passenger_id	int (25)	Primary key	
Title	Varchar(5)	Not null	
First_name	Varchar(25)	Not null	
Last_name	Varchar(25)	Not null	
Telephone	Int(25)	Not null	
Email_address	varchar(25)	Not null	

Table Payment_Details

Field	Domain	Constraint	Description
Payment_id	int (11)	Primary key	
passenger_id	Int(25)	Foreign key	
Card_type	Varchar(25)	Not null	
Expiry_date	Date	Not null	
Security_code	Int(3)		

Table Reservation

Field	Domain	Constraint	Description
Reservation_id	Varchar(11)	Primary key	
passenger_id	Int(25)	Foreign Key	
Reservation_status	Varchar(25)	Null	
Date_reservation_made	Datetime	Not null	
No_of_adults	Int(3)	Not null	
No_of_children	Int(3)	Not null	
No_of_infants	Int(3)	Not null	

Table Transaction

Field	Domain	Constraint	Description
transaction_id	Int(11)	Primary key	
Reservation_id	Varchar(11)	Foreign key	
Transaction_date	Date	Not null	
Amount	Int(11)	Not null	
Payment_id	Int(11)	Not null	

Chapter4 –Implementation

Database Implementation

Connection between database and Web tool

For the connection between the database and the Web tool I am going to be using a server called WAMP. This is in a form of a mini server than runs on any windows operating systems.

WAMP stands for Windows, Apache, MySQL , PHP.

PhpMyAdmin is a software that allows the managing of MYSQL databases in a more pleasing way. This will make it a lot less challenging for me to use query databases using phpmyadmin rather than using command line.

Creation of Tables

To create the tables in PHPMYADMIN I initially had to ensure that I have created a database, which I did and have called 'Masriya'. Once doing this I used the ERD diagram that I had created to guide me in creating the tables in phpmyadmin.

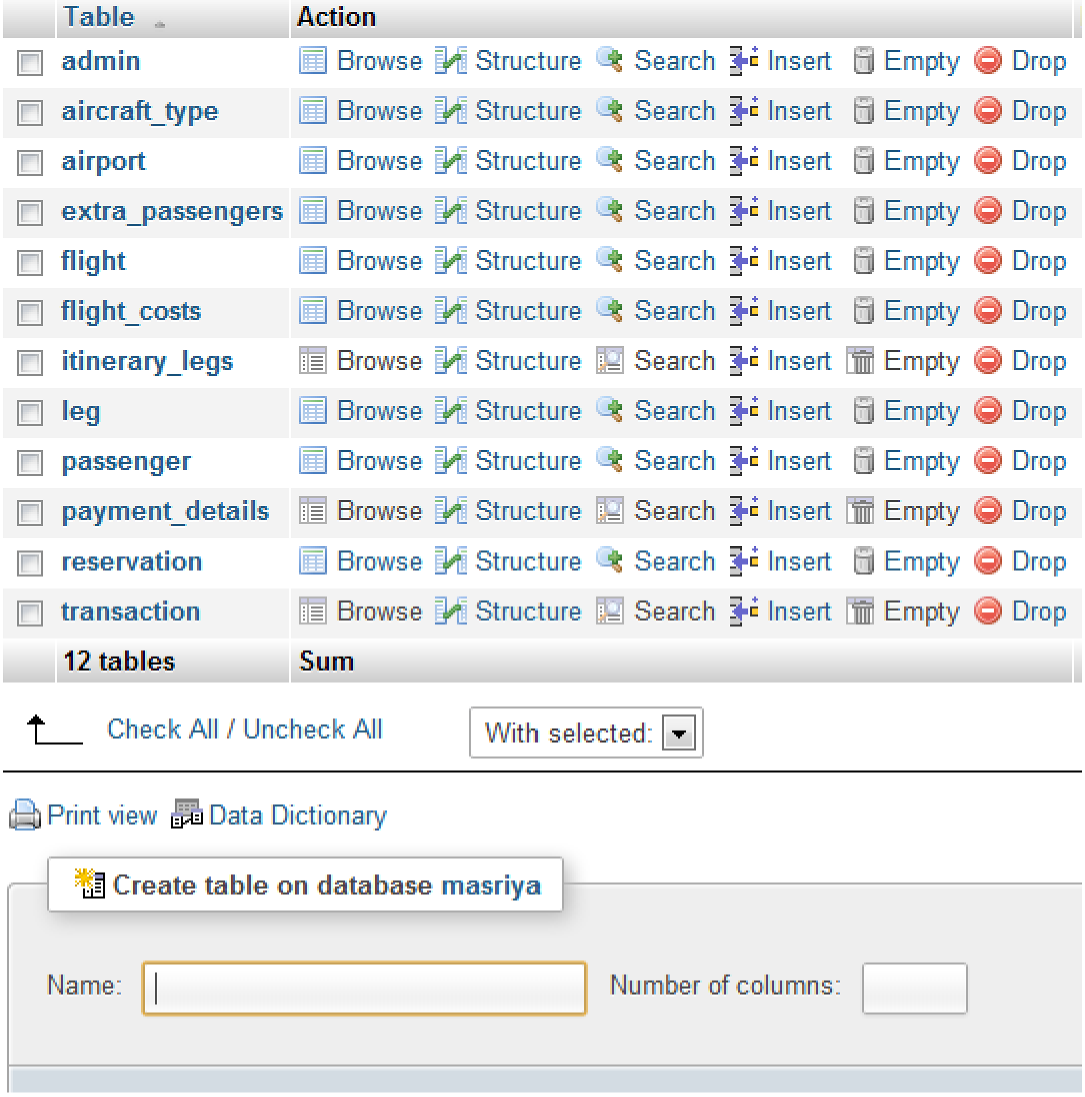


	Table	Action					
□	admin	Browse	Structure	Search	Insert	Empty	Drop
□	aircraft_type	Browse	Structure	Search	Insert	Empty	Drop
□	airport	Browse	Structure	Search	Insert	Empty	Drop
□	extra_passengers	Browse	Structure	Search	Insert	Empty	Drop
□	flight	Browse	Structure	Search	Insert	Empty	Drop
□	flight_costs	Browse	Structure	Search	Insert	Empty	Drop
□	itinerary_legs	Browse	Structure	Search	Insert	Empty	Drop
□	leg	Browse	Structure	Search	Insert	Empty	Drop
□	passenger	Browse	Structure	Search	Insert	Empty	Drop
□	payment_details	Browse	Structure	Search	Insert	Empty	Drop
□	reservation	Browse	Structure	Search	Insert	Empty	Drop
□	transaction	Browse	Structure	Search	Insert	Empty	Drop
	12 tables	Sum					

In phpmyadmin you are able to add tables through forms or you also have the option of using SQL queries.

To create my tables I used the forms and then I assigned primary and foreign keys to specific fields by using the index option. This is so tables are linked together so there aren't any duplicate information. This is called normalization. As explained in the Database Design.

Web application Implementation

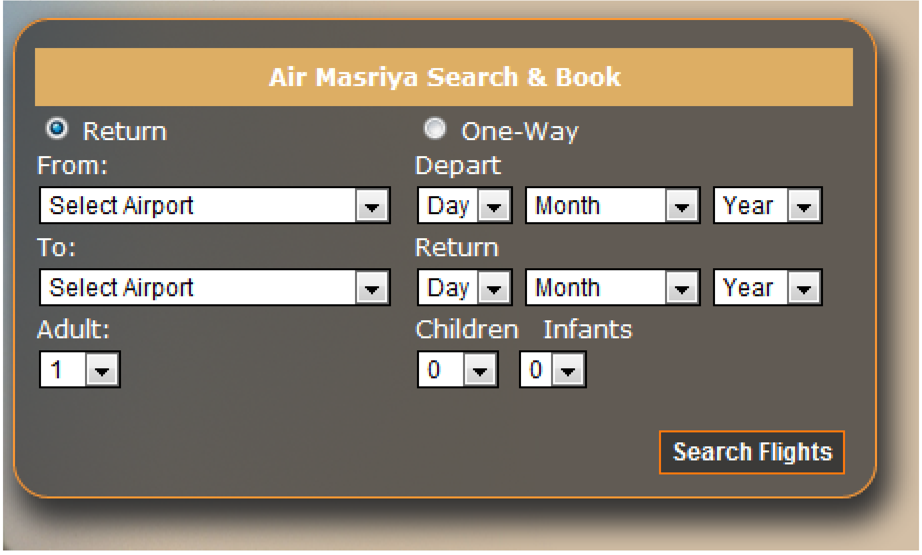
Implementation of any project is something which needs to be planned out very thoroughly.

In this section I will be going through the main functionalities which I have implemented within my project:

1. Search Form
2. Cookie Login
3. Search Function
4. View Booking
5. Administration Log in / Log out

Creating the Home page - index.php with Search functionality

In order for me to create the index.php I had to design the search form and get it to successfully operate as this is going to be the main functionality that will be displayed on this page.



Air Masriya Search & Book

● Return ○ One-Way

From: Select Airport

Depart: Day Month Year

To: Select Airport

Return: Day Month Year

Adult: 1

Children: 0

Infants: 0

Figure 14 - Search Form - Index.php

As stated in the design chapter:

The fields that I am going to include in this search form are as follows:

- Whether the search is for a Return or a One-way flight
- The origin airport(from) and the destination airport(to)
- Departure date and return date (if return search is being made)

- How many passengers are travelling. These need to be differentiated into Adult, Child and Infant fields.

  To create the above stated search form I had to create functions which would allow the system to display a drop down menu for the user to be able to choose a specific date.
  The PHP code that was used to create this function specified the number of days in a month validated from 1-31, the month names from January to December and the year numbers starting from 2012.
  This will allow the user to select a date in the correct format.

  I have created two date functions one for the depart date and one for the return date so they can be differentiated when I want to use that information to search for a flight.
  The next function which is needed is the function that will allow users to select the number of passengers whether they are adult child or infant.
  For the maximum number of adults allowed to be 10; I had to think of a way for this to be displayed appropriately. The best way to do this was to create a while loop.
  Firstly I made a $a variable and set it equal to 1.
  The while loop condition statement was to check whether the variable - $a - was less than or equal to ten.
  If this condition is true then the value will be incremented by 1 until the value reaches 10.
  Then the select option is echoed, which is how the user will be able to select an option from 1 - 10.
  I have used this function to constrain the data, this is so that anything that comes from this field is a number and has already been constrained to its domain to avoid customer input errors. This maintains integrity and consistency.
  Later on in this chapter I will explain how this information will be concatenated and trimmed down to a format that I will need in order to perform the search and book flight task.

  The same while loop has been created for the number of children and number of infants field in the search form. The only difference is that the number of infants has been limited to two as per the business rule of Air Masriya.

  Once I created the necessary functions I then moved onto creating the rest of the form in a file called index.php.

  The next step was to create the radio buttons by an input type of radio, this allowed me to setup up a way for the user to differentiate between what sort of flight they are searching for; whether it be a return flight or a one way flight.

  This file contains all the necessary coding to allow the customer to search for a flight and also navigate through the site by the use of a navigation bar.

  The next part of the search form I implemented is the section where the airport names are displayed in the two drop down boxes. ('From' and 'To') I had to pull the

data from my database to be display such as the list of airports in the 'To' and 'From' drop down box.
In order for me to do this I used the mysql_fetch_array to return rows from the record airport_name from my database table Masriya.

Creating the Header

The Header.php file contains the link to the style sheet and also includes all the navigation bar links which, will allow the user access to other pages on the web application.

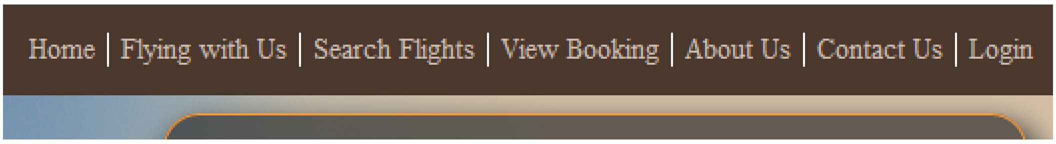


Figure 15 - Navigation bar before user Log in.

Within the navigation bar div I have used the if **isset** function to determine whether the **$_COOKIE array** has been set which is retrieving the cookie which has been defined in the login.php file. This allows the system to check that IF there a cookie is set then to display the a logout hyperlink on the navigation bar ELSE to display a Login hyperlink.

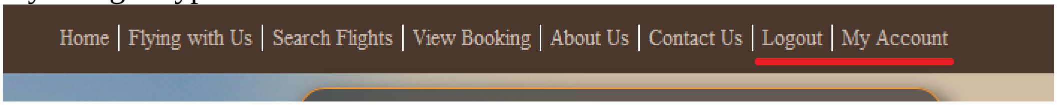


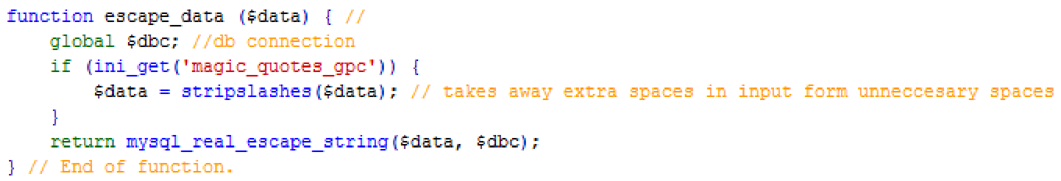
Figure 16 - Navigation bar after User Log in.

As the Login function will only be used by administrators or staff of Air Masriya I have also included a 'My Account' hyperlink to allow the user to be redirected back to the main page for Administration purposes, so the administration page is still accessible. As shown in the image above.

Below is the code used to retrieve the cookie on the navigation bar.

```
    <?php
     if (isset($_COOKIE['username'])) {?>
          <li><a <?php if (strpos($_SERVER['PHP_SELF'], 'logout.php')) echo 'class="current"
style="color:#0CF"';?>href="admin/adminIndex.php">My Account</a></li>
          <?php }
    else {}?>
            <?php
     if (isset($_COOKIE['username'])) {?>
           <li><a <?php if (strpos($_SERVER['PHP_SELF'], 'logout.php')) echo 'class="current"
style="color:#0CF"';?>href="logout.php">Logout</a>
          </li>
             <?php }
    else{ ?>
           <li><a <?php if (strpos($_SERVER['PHP_SELF'], 'login.php')) echo 'class="current"
style="color:#0CF"';?>href="login.php">Login</a>
           </li>
    <?php } ?>
```

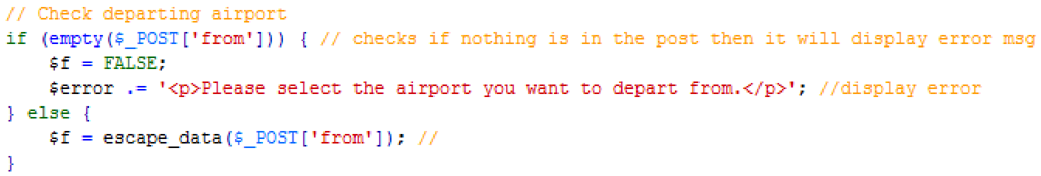
Figure 17 - Script to show cookie being retrieved



```
function escape_data ($data) { //
    global $dbc; //db connection
    if (ini_get('magic_quotes_gpc')) {
        $data = stripslashes($data); // takes away extra spaces in input form unneccesary spaces
    }
    return mysql_real_escape_string($data, $dbc);
} // End of function.
```

Figure 18 - escape_data function (search.php - Line 5)

This function contains the striplashes() function which removes backslashes and any extra spaces from a string input. I have created this function so when I collect information from the form in search.php I can use this function to take away any extra spaces that have been input - For example in the script below.



```
// Check departing airport
if (empty($_POST['from'])) { // checks if nothing is in the post then it will display error msg
    $f = FALSE;
    $error .= '<p>Please select the airport you want to depart from.</p>'; //display error
} else {
    $f = escape_data($_POST['from']); //
}
```

Figure 19 - IF statement collecting value using escape_data function (search.php Line18)

I have used the same principle as above to collect information for the all the values in the search form. These were:

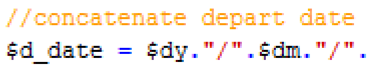
- Whether the flight was a return or one way
- Departing airport value
- Destination airport value
- Depart date
- Return date
- Number of passengers travelling (Adults, children & infants)

Earlier on in this chapter I explained how I created the date_function.

In order for me to retrieve this information in this section of my project for the depart and return date I had to check whether each individual field had been input; if not, then an error message for that particular field would be displayed. If the entire date has not been displayed then a separate error message informing the user of this mistake will be displayed.

Once the system collects the information for the depart and return date. (day, month, year) .

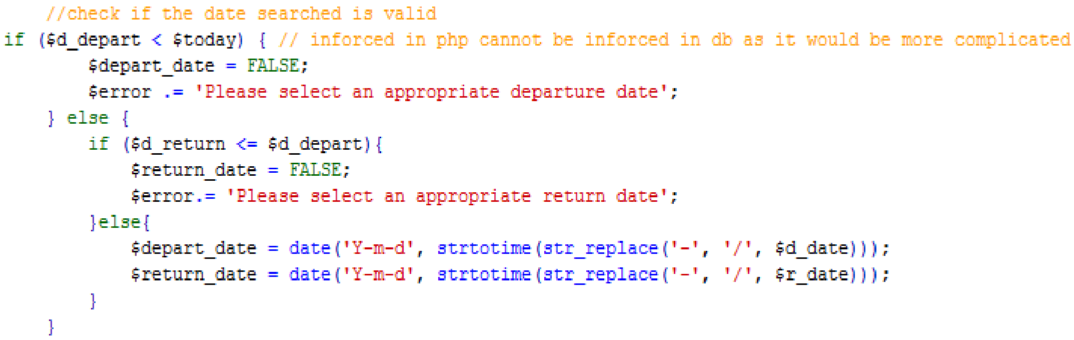
I concatenated the fields to equal to one variable as shown below:



```
//concatenate depart date
$d_date = $dy."/".$dm."/".$dd ;
```

Figure 20 Concatenate depart date (Search.php Line 60)

I did this so I could then change the departure date and return date from string to datetime values by using the strtotime() function.



```
    //check if the date searched is valid
if ($d_depart < $today) { // inforced in php cannot be inforced in db as it would be more complicated
        $depart_date = FALSE;
        $error .= 'Please select an appropriate departure date';
    } else {
        if ($d_return <= $d_depart){
            $return_date = FALSE;
            $error.= 'Please select an appropriate return date';
        }else{
            $depart_date = date('Y-m-d', strtotime(str_replace('-', '/', $d_date)));
            $return_date = date('Y-m-d', strtotime(str_replace('-', '/', $r_date)));
        }
    }
```

Figure 21 - If statement - Validation for date being searched

By changing the value of the date to a unix timestamp I then have the ability to check if the search criteria by the user is valid by using an IF statement to enforce the condition that:

if the depart date is 'less than' today's date the system will display an error to the user to enter an appropriate departure date. In addition to this, IF the return date is 'less than' the departure date again an error message will be displayed. Or else the validity of the dates entered are correct. (as shown above in Fig 17)

Once all the collected values have been validated and checked I used these to perform an SQL query in order to display the origin airport and destination airport name with the closest date requested by the user.

I use this information later on in the file to be shown in a table format.

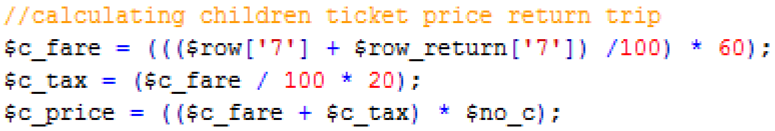
I repeated this query whilst changing the necessary fields required to find the same information for the return flight.

Fare Prices

I had to take into consideration that I need to display the different prices for each flight and the different passengers. For example an Adult would have a different price to pay and this would be considerably higher than of course a child that would be travelling.

The method I used for calculating the fare for an adult was to use the normal fare price and then to work out 60% of that which would be the price for the child. I had to do this as all fares are different for each flight therefore to create a separate pricing table in the database for a child and adult would have cause the procedure to take up a lot more time than this method.

Fig.18 shows the php script which enabled me to be able to do the above.



```
//calculating children ticket price return trip
$c_fare = ((($row['7'] + $row_return['7']) /100) * 60);
$c_tax = ($c_fare / 100 * 20);
$c_price = (($c_fare + $c_tax) * $no_c);
```

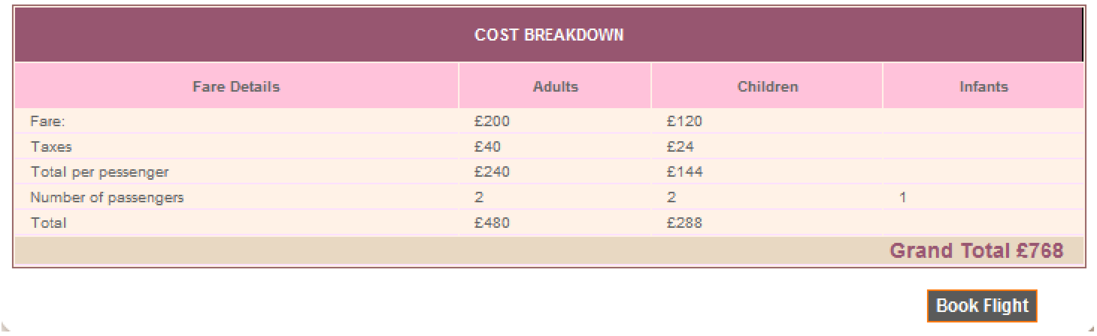
Figure 22 - Method to create child fare - search.php

After being able to see that I have all the information I need to be displayed in the results for the search form I began to create a booking results form, where all the information I have collected will be displayed. This information will include the departure time and arrival time for both trips; Departing and Return.

The next part of the results page will be the Cost break down. This will include how much each passenger will be paying according to type of passenger. For example Adult, Child and/or Infant.
The cost is broken down into:

Fare - How much the flight costs per passenger
Taxes - per passenger how much tax will be paid
Total per passenger - Total costs per passenger
Number of Passengers - No of Adult, Child And/or Infant.
Total - the total per passenger multiplied by the number of passengers.

To conclude the above cost break down I have also made to display the grand total of the entire flight for that booking. As seen in Fig 19.



COST BREAKDOWN

Fare Details	Adults	Children	Infants
Fare:	£200	£120	
Taxes	£40	£24	
Total per pessenger	£240	£144	
Number of passengers	2	2	1
Total	£480	£288	

Grand Total £768

Figure 23 - Cost Break down PrtSc

While trying to get the above Cost breakdown table to work successfully and display the correct prices for each passenger; I used trial and error to keep checking whether the tax for the children would come up correctly. At many points I was displayed incorrect figures this was because I was not coding the calculation properly. I resolved this issue but unfortunately do not have a screen shot to prove this.

In conjunction with getting the above calculations to work I also implemented the interface for the table.

As in my original design I did not indicate that I would be creating this table I came up with the color scheme by using trial and error to see which colors and format would look best. To stand out from the rest of the tables I choose a plum and cream theme so the user knows that this information is different from the flight information.

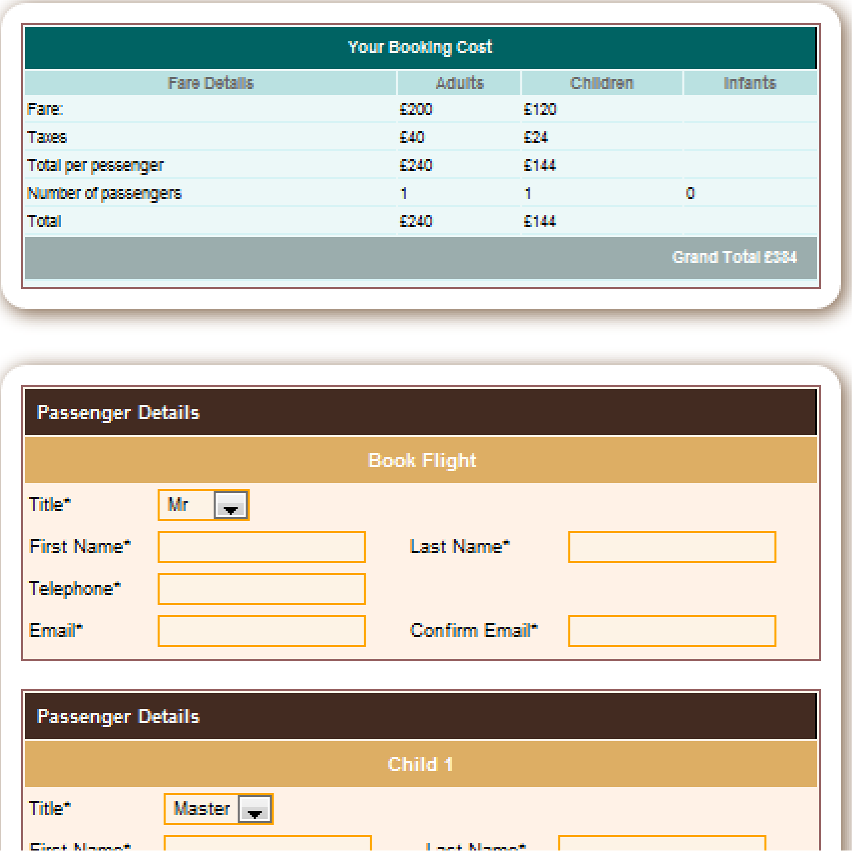
The last part of this section consists of the Book Flight submit button at the end of the form. This submit button takes all the information from the form labeled as book_flight and then sends the user to the passenger.php page where the user will be able to see the final break down of their booking and also be allowed to enter information such as their name, email address, date of birth and telephone number. In addition to this they will also need to enter the details of the extra passenger so the passenger booking can be differentiated.

Viewing And Confirming Booking - Passenger details being entered

This section is going to be talking about the code in the passenger.php file.

In order for me to retrieve information from the previous file; which was the search.php file; I have to collect the information needed to show the passenger exactly what they are booking to keep the layout of the website consistent, using the $_POST function. I will be retrieving the same information which was displayed in the cost breakdown section from the previous page.

This information will now be displayed as 'Total Booking Cost' to confirm to the user that this is what they will be booking . As seen below in Figure 20. Below the 'Your Booking Cost' Table.



Your Booking Cost

Fare Details	Adults	Children	Infants
Fare:	£200	£120	
Taxes	£40	£24	
Total per passenger	£240	£144	
Number of passengers	1	1	0
Total	£240	£144	

Grand Total £384

Passenger Details

Book Flight

Title*	Mr		
First Name*		Last Name*	
Telephone*			
Email*		Confirm Email*	

Passenger Details

Child 1

Title*	Master		
First Name*		Last Name*	

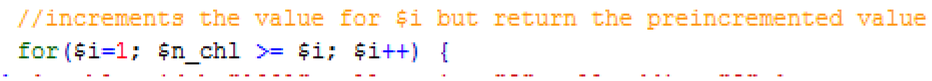
Figure 24 - Enter Passenger Details - passenger.php

The next part of the passenger.php file was for me to be able to collect details of all the passengers whom will be travelling on one particular booking.

I created forms so I could be able to collect this information and store it in the database for each passenger.

In order for me to do this I had to firstly create three separate forms one a for the main passenger making the reservation and then for additional passengers such as Adults and one for Children.

The above image displays the code on how using ' i++' will increment the value for 'i' but will return the pre-incremented value.



```
//increments the value for $i but return the preincremented value
for($i=1; $n_chl >= $i; $i++) {
```

Figure 25 - Incrementing passenger forms - passenger.php

One of the challenging parts of this process was too connect the number of extra passengers with the number of forms needed to be filled out.

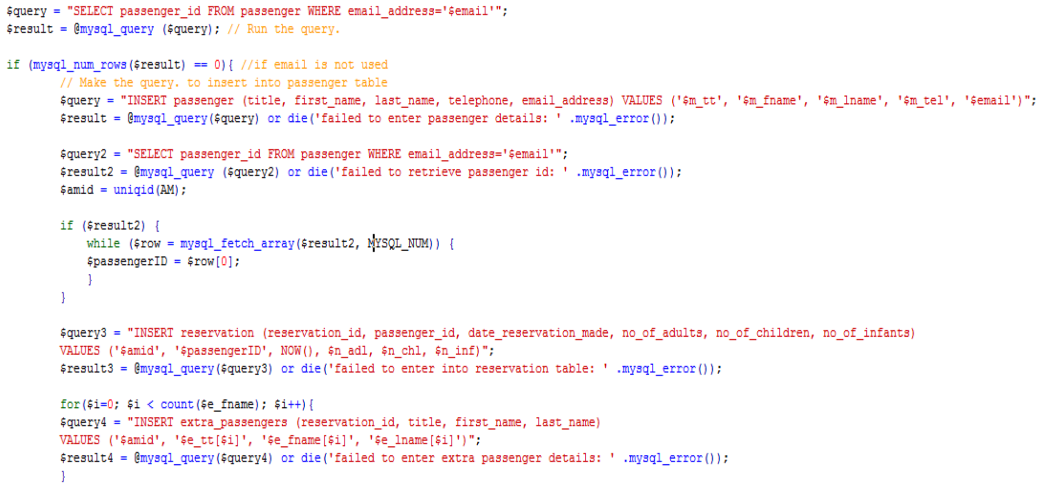
For example if 2 extra passengers were also part of the booking that would mean that there should be 3 forms to be filled out. One by the main passenger then 2 more for the extra passengers.

I needed to increment the number of forms but also make the forms relevant to the passenger. Hence the extra adults have a different form to the extra children's booked as the children's booked will need a date of birth to be entered as validation for the price that they are paying.

Once the user has entered the necessary information and selects the book flight button, the information entered is all taken to the enter_passenger.php file.

This is where all the information entered from the form is put into the database by using an SQL query, and the customer is also here told their unique booking reference number whilst also being shown a confirmation.

I separated the queries in to the sets of tables that the information will be going into. So for example the passenger information will be entering the passenger table, where as the reservation information such as passenger_id, the date the flight is booked for and so on so forth is being entered into the reservation table.



```
$query = "SELECT passenger_id FROM passenger WHERE email_address='$email'";
$result = @mysql_query ($query); // Run the query.

if (mysql_num_rows($result) == 0){ //if email is not used
        // Make the query. to insert into passenger table
        $query = "INSERT passenger (title, first_name, last_name, telephone, email_address) VALUES ('$m_tt', '$m_fname', '$m_lname', '$m_tel', '$email')";
        $result = @mysql_query($query) or die('failed to enter passenger details: ' .mysql_error());

        $query2 = "SELECT passenger_id FROM passenger WHERE email_address='$email'";
        $result2 = @mysql_query ($query2) or die('failed to retrieve passenger id: ' .mysql_error());
        $amid = uniqid(AM);

        if ($result2) {
            while ($row = mysql_fetch_array($result2, MYSQL_NUM)) {
            $passengerID = $row[0];
            }
        }

        $query3 = "INSERT reservation (reservation_id, passenger_id, date_reservation_made, no_of_adults, no_of_children, no_of_infants)
        VALUES ('$amid', '$passengerID', NOW(), $n_adl, $n_chl, $n_inf)";
        $result3 = @mysql_query($query3) or die('failed to enter into reservation table: ' .mysql_error());

        for($i=0; $i < count($e_fname); $i++){
        $query4 = "INSERT extra_passengers (reservation_id, title, first_name, last_name)
        VALUES ('$amid', '$e_tt[$i]', '$e_fname[$i]', '$e_lname[$i]')";
        $result4 = @mysql_query($query4) or die('failed to enter extra passenger details: ' .mysql_error());
        }
```

If all validation checks are successful then a confirmation message will be shown to the user, otherwise an error message will be displayed. Within the booking confirmation message the user will also be shown their unique booking reference number where they will be able to check the details of their booking.

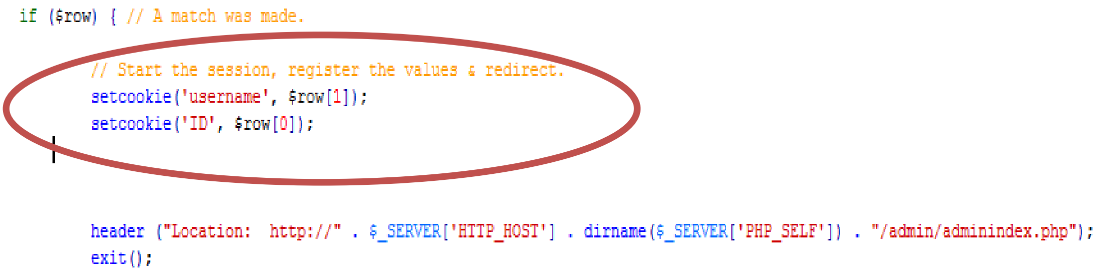
Log in/ Log out

Logging into the system has been made for administration purposes only in this application. Due to a time constraint I was only able to create this function for one part of the system.

Administration are require to login to maintain information such as flights and passenger details.

For this reason the access required to the administration pages needs to be secured.

The script used in the login.php page has been taken from Larry Ullman - *PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide*

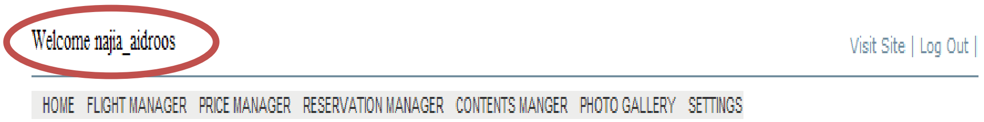


```
if ($row) { // A match was made.

        // Start the session, register the values & redirect.
        setcookie('username', $row[1]);
        setcookie('ID', $row[0]);


        header ("Location:  http://" . $_SERVER['HTTP_HOST'] . dirname($_SERVER['PHP_SELF']) . "/admin/adminindex.php");
        exit();
```

Figure 26 - Using Cookies - Login.php

In this script I have enabled cookies to be used so that when a member of staff logins into their user area they are greeted with their name and once they are on their part of the system they are able to logout when going on to the main website.

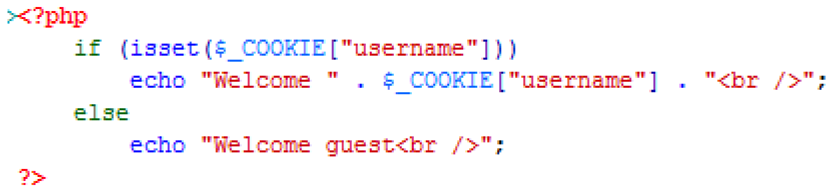
As you can see in Figure 23. Once the user has signed in they are welcomed by their unique username. In the far right hand corner of the navigation bar they are also given the option to logout or to visit the main Air Masriya website.



Welcome najia_aidroos

Figure 27 - Showing Administration log in.

Below is the code I have used to enable the system to carry out this action



```
<?php
     if (isset($_COOKIE["username"]))
         echo "Welcome " . $_COOKIE["username"] . "<br />";
     else
         echo "Welcome guest<br />";
 ?>
```

Figure 28 - Code showing administration cookie log in.

Once the administrator selects to 'Visit Site' they are displayed the same user site but they are able to also have the option to logout or to return back to their account. This allows the user access back into the administration side of the application as seen below.

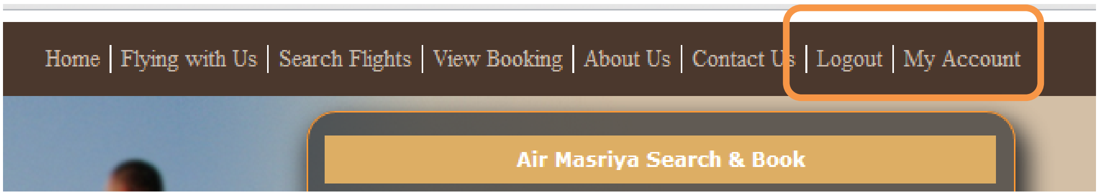

Home | Flying with Us | Search Flights | View Booking | About Us | Contact Us | Logout | My Account
Air Masriya Search & Book

Figure 29 - Displaying website through Administration Log in

In order for me to be able to do this I had to insert code into the navigation div that I have used in the index.php file and use an if statement. I have spoken about this earlier on the chapter in the **Creating the Header** chapter.

Administration Implementation

For the Administration page As per my design I used the navigation from within Dreamweaver's own navigation bar templates.

In the administration web application you can find the following functions which have been implemented.

1. Add
2. Edit
3. Delete
4. View

This script has been taken from an online forum where people have discussed problems with their php scripts and they have come up with their own script. The website name is www.killersites.com

Chapter 5 – Testing

Introduction

The main purpose of the test plan for the Airline Reservation System is to discuss the testing details of the use cases of the Airline Reservation System. The software project
test plan also describes the objective, scope and approach of the software testing effort
for the Airline Reservation System project. The test plan for the Airline Reservation System also indicates the personnel responsible for each task and also specifies the risks
associated with the test plan.

Testing Plan

The purpose of testing is to find as many bugs in the system as possible and to see if the end user will be able to use the system to its best ability.

I will be testing this project in 3 different ways.

Unit Testing - This particular test deals with testing the functionality of the system

User Acceptance Testing -

System Testing- This will examine the final system with different browsers and different browser settings,

Accessibility Testing - I will use an online web application called WAVE which checks the accessibility websites.

Unit Testing

This section will consist of the tasks that I will carry out to test the behaviour of the functionality to ensure that it gives the correct outputs.

Unit Testing is a test which covers to test every single module or area of the software to check it for errors. The main purpose of this test if so it can segregate each part of the application and to check how correct the code actually is.

In terms of the Airline Reservation System all the forms and functionality will be tested.

There are many benefits for this unit testing:

- The unit testing facilitates change in the code.
- It allows testing to be done in a bottom up fashion.

At the same time, unit testing has some disadvantages such as, it might not identify each

and every error in the system hence why I will also be carrying other forms of testing as mentioned above.

The Unit Testing is a test that tests each single module of the software to check for errors.

Please refer to appendix B for screenshots of test data.

Unit Testing

No	**Task**	**Result**	**Problem Occurrence**	**Correction Made**
1	Logging into the system using the correct password	Pass	n/a	
2	Logging into the system using incorrect password	Pass	n/a	
3.1	Searching for return flight using valid search dates with 1 of each passenger type.	Pass	n/a	
3.2	Book flight with search criteria from No.3	Pass	n/a	
3.3	Enter Details for passenger with search criteria from No.3	Pass	n/a	
3.4	Confirm Booking from 3.3	Fail	Error message from system is shown. AM	I went back into the file and checked what the

			variable has not been declared	problem was and the variable had not been declared the problem has now been rectified.
4	Check Book reference number is shown once booking is confirmed	Pass	n/a	n/a
5	View booking using Booking reference from task 3.4	Pass	n/a	n/a
Administration Functionality Testing				
6.1	View all Flights in 'View all' Mode	Pass	N/a	n/a
6.2	View all Flights in 'View Paginated Mode'	Pass	N/a	N/a
7	Add a New Flight	Fail	The value cannot be updated as the field aircraft type is a foreign key in the flight table. In order to allow this to work I need to create a drop down menu for the user to be able to select this information from the aircraft type table.	N/a Time constraint

`

User Acceptance

User acceptance testing is when the system is tested by actual users of the system. This is vital to determine the success of the system.

I will carry this out to receive feedback on the system and to determine what parts of the system according to users are successful.

Chapter 6 – Evaluation

In this chapter I will be evaluating the following aspects of the project.

- The Reservation Booking System as a whole.
- Challenges regarding the aims of the project

When looking through the final outcome of Air Masriya's Reservation System I myself am not content with the outcome. There are requirements which have not been yet made due to reasons such as time constraints which have led me to criticize my own work and feel frustrated about.

In terms of design, consistency, usability and accessibility I feel content due to the feedback received from my user acceptance testing. Although there are factors of usability and accessibility which I have covered in my background research which I feel I could have implemented in my application which could have made a huge difference. For example allowing the user to change the size and color of text on the web application. This would improve user's experience.

I also feel that the view booking feature of the system should be better. Currently only a basic view booking function has been implemented for users. From my research I gathered that almost all other airline reservation system have a view function which displays information which is far more advanced and detailed; All of which helps to improve the user's experience.

A feature of the system which I was very determined to have in this project was to allow the user to be able to select which seat they want to book and have different allocated prices for seats in terms of classes. For example Economy, Business or First Class.

Unfortunately due to time constraints I was unable to implement this feature into my system. I did initially do research on this task and I found it to be a very challenging task. In order for me to be able to carry out this task I would have had to you arrays and assign each block a different price. The coding behind it was far too complex and would have taken me a longer time to get round to understand, than the time I was actually given.

A feature that I am proud of completing but am still not 100% satisfied with is the search function which allows users to search and book for flights. Again due to a time constraint I was unable to finish this functionality of the system.

The user is only able to search for a return flight and not for a one way flight. I could have achieved this by adding another IF statement within the code and using the same condition as the return flight to making the entire search form work.

I was happy with the way the flight information was displayed to the user.

I would have also liked to improve this section by allowing the user two select firstly the departing flight and then be shown the return flight rather than having multiple combinations of flight shown together.

Perhaps one of the areas which I feel my system has excelled in is its usability. The testers that I had been able to agree to test out my project felt that the site was very easy to use due to its clear menus and plain interface system.

What was even more pleasing was that they found my interface easier to use that some of the websites that I had attempted background research on.

One thing I am very happy about is that as being a student specializing in IT, my main focuses were on the design, usability, consistency and accessibility of the project, which I feel had improved from the first initial design I had made.

Overall I feel that If I had more time to be able to complete this project I would have been able to achieve a lot more of what I am truly capable of. Leaving things unfinished is something which I have never agreed with hence why I will continue to improve this project till I am content.

Perhaps one of the most important areas that I feel my system excels in is its usability. All of my testers felt that the site was very easy to use, as its plain interface and clear menus, reduce the complexity and confusion often found in e-commerce systems. What is even more pleasing about this is that one of the key requirements from my client was that they wanted the system to be easy to use and it seems that this has been met.

I feel that my time management for the project could have been a lot better than what it was. Initially I should created my Gantt chart with more thought and made it more realistic for me to follow.

The time it took me to implement the system was actually a lot longer than expected as I did experience a few problems which were not taken into consideration at the beginning of the planning stage.

A problem I encountered when trying to put the system on the universities server; so that I could have people test the website through their own computers; was that when I tried to restore my database from my local server to the 'Masriya' server, I received errors and problems regarding to permission to access the site. This was later on resolved as the problem turned out to be that my version of the SQL server wasn’t the same as the one the university has.

Another problem

A problem which held me back a few days was the function where an email is sent to the user with their booking reference number.

This was something which I had never done before and even though there were plenty of tutorials around for me to look at, the file that I had implemented was just not working. I later on found out that this was due to a part of the code that I had not implemented properly. Unfortunately as I found this out at a later stage I was not able to implement this as part of my project.

There are a number of things which could be added on to the system for future work.

As I mentioned earlier on in the project; I really wanted my system to have a competitive advantage against all the other Airline Reservation systems. I was going to do this by having a statistical chart when the user had searched for their particular flight. This chart would have contained information as to how busy the flight is and if there are any other flights available nearer to the date to when they are travelling that are not as busy to travel on.

This is one of the biggest regrets that I had as I was very excited about this particular function but again do to time constraints I was not able to even begin planning this task.

I have gained a lot of experience by undertaking this project. It has broadened the way I think and approach things not just in education but in normal activities in life. It has taught me that planning is the key thing and is essential in creating a successful system. To make notes as you go along is also very helpful when undertaking a project. It helps in the long run when you forget particular information; you can then look back at your notes and help you to carry on completing tasks.

In the future when I undertake projects I will know how to assess and manage them due to the experience gained.

Bibliography

References

1. Script Used in Administration Functions has been taken from this forum.- [Online]< http://www.killersites.com/community/index.php?/topic/1969-basic-php-system-vieweditdeleteadd-records/>

2. Larry Ullman - Guide to php and mysql for dynamic websites. - [Online]<http://www.larryullman.com/books/php-and-mysql-for-dynamic-web-sites-visual-quickpro-guide/>

3. Connolly, Y. Begg, C. 2008. Database systems: a practical approach to design, implementation and management. Fifth edition. Boston: Pearson Education.

4. Stephens, R. 2008. Beginning database design solutions. Indianapolis:Wrox.

5. David Saywer Mc Farland.2004.Dreamweaver MX 2004: The Missing Manual. Fourth Edition

6. Thomas A.Powell.2000.Web Design:The Complete Reference.

7. Thomas A.Powell - 1998. The Complete Reference: HTML

8. Kumar,Waters,Gianotto,Anton,Stephens,Solin. 2002.PHP4 web Development Solutions

9. Connolly Begg. 2005. Fourth Edition. Database Systems.

10. Luke Welling and Laura Thomson. 2003. PHP and MYSQL Web Development

11. http://coredogs.com/lesson/vertical-navigation-bars [Online] Navigation bar tutorial

12. http://tutvid.com [online] Forms and email Tutorial.

13. Images used from:
    1. www.lufthansa.com/uk/en/Homepage - Lufthansa
    2. www.**emirates**.com/ - Emirates website
    3. www.egyptair.com/ - Egyptair website