

Pooja Gautam

ONLINE AIRLINE TICKETING SYSTEM

Thesis

CENTRIA UNIVERSITY OF APPLIED SCIENCES

Degree Program in Information Technology

September 2015

ABSTRACT

| | | |
|---|------------------------|------------------------|
| Unit Kokkola - Pietarsaari | Date September 2015 | Author Pooja Gautam |
| Degree Program Information Technology | | |
| Name of thesis ONLINE AIRLINE TICKETING SYSTEM | | |
| Instructor Kauko Kolehmainen | Pages 44 | |
| Supervisor Kauko Kolehmainen | | |
| <p>The aim of this thesis was to explain the system of online ticketing used by airlines by explaining the mechanism and presenting a sample application.</p> <p>The thesis describes the evolution and working of airline reservation system. Various aspects related to online flight booking are explained in the thesis. The main focus of the thesis was developing a working application. A basic application similar to the models used by airlines companies today was developed with the use of various software and programs. Research in sources such as e-books, web articles, documentations and tutorial was completed for the development.</p> <p>The thesis describes how the online ticket booking works with the help of different diagrams and illustrations. The thesis concluded that with the progress of ever-changing technology, the trend of booking online is likely to develop with more features. While there are a few limitations, the convenience of booking tickets online and the possible enhancements will lead to an even advanced form of booking tickets online.</p> | | |

Keywords

Web applications, web framework, databases, Online Reservation System

ABSTRACT

TABLE OF CONTENTS

| | | |
|----------|--|-----------|
| 1 | INTRODUCTION | 1 |
| 2 | ONLINE TICKET BOOKING | 3 |
| | 2.1 Development of e-commerce | 3 |
| | 2.2 Airline reservations system | 4 |
| | 2.3 Working mechanism of flight booking | 5 |
| 3 | SOFTWARE AND PROGRAMS USED | 6 |
| | 3.1 Python | 6 |
| | 3.2 Django | 7 |
| | 3.3 HyperText Markup Language | 8 |
| | 3.4 Jinja | 8 |
| | 3.5 Javascript | 9 |
| | 3.6 Cascading Style Sheet (CSS) | 9 |
| | 3.7 SQLite Database | 10 |
| 4 | FUNDAMENTAL ASPECTS USED IN APPLICATION | 11 |
| | 4.1 Website | 11 |
| | 4.2 Web application | 12 |
| | 4.3 Web designing | 13 |
| | 4.4 Web development | 13 |
| 5 | THE DESIGN STAGE | 14 |
| | 5.1 System Design | 14 |
| | 5.2 Database design | 15 |
| | 5.3 The website and its logic | 16 |
| 6 | The DEVELOPMENT STAGE | 21 |
| | 6.1 Setting up Django framework | 21 |
| | 6.2 Setting up the database | 23 |
| | 6.3 Creating admin user | 23 |
| | 6.4 Making static web pages | 24 |
| | 6.5 User-interactive pages | 28 |
| | 6.5.1 Use of Django forms | 29 |
| | 6.5.2 Views and templates | 30 |

| | | |
|---------------|---|----|
| 6.5.3 | Javascript functions for conditions | 33 |
| 6.5.4 | Interactive subpages | 34 |
| 6.5.5 | Preparing for sending e-ticket | 38 |
| 6.5.6 | E-ticket | 39 |
| 7 | ADVANTAGES AND CHALLENGES | 41 |
| 7.1 | Advantages | 41 |
| 7.2 | Disadvantages | 42 |
| 7.3 | Challenges faced during application development | 42 |
| 8 | CONCLUSIONS | 44 |
| | | |
| GRAPHS | | |
| GRAPH 1 | Database tables | 16 |
| GRAPH 2 | General working flow | 17 |
| GRAPH 3 | Web pages and subpages | 19 |
| GRAPH 4 | Django Architecture (Adapted from MyTardis 2013.) | 22 |
| GRAPH 5 | Running local server in Django | 22 |
| GRAPH 6 | Admin view | 24 |
| GRAPH 7 | Designing the index | 25 |
| GRAPH 8 | Defining main pages within index | 25 |
| GRAPH 9 | Homepage | 26 |
| GRAPH 10 | About page | 27 |
| GRAPH 11 | Fares and regulations page | 28 |
| GRAPH 12 | Declaring feedback form | 29 |
| GRAPH 13 | Declaring flight search form | 30 |
| GRAPH 14 | Flight search function | 30 |
| GRAPH 15 | Displaying search results in form | 31 |
| GRAPH 16 | Flight search page | 32 |
| GRAPH 17 | Contact page with feedback form | 33 |
| GRAPH 18 | Defining JavaScript functions | 34 |
| GRAPH 19 | Appending correct data to departing flights | 35 |
| GRAPH 20 | Choosing from the flight search results | 36 |
| GRAPH 21 | Booking the selected flight | 37 |
| GRAPH 22 | Thank you message after booking | 38 |
| GRAPH 23 | Preparing the e-ticket | 39 |
| GRAPH 24 | E-ticket received in customer email | 40 |

Abbreviations

| | |
|------|------------------------------------|
| ARS | Airlines Reservation System |
| CRS | Computer Reservations System |
| GDS | Global Distribution System |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheet |
| WWW | World Wide Web |
| CCVS | Credit Card Verification System |
| API | Application Programming Interfaces |

1 INTRODUCTION

World has seen the most technological boom in the last fifty years, with the innovations in every field made possible now for making the human life easier and more comfortable. Airlines industry has evolved into one of the most sophisticated and fascinating industries of today. Airlines industry has made the dream of flying of the humans into a reality in less than a century. Today, millions of people fly every day. This has strengthened not only the economies of places but also connected people and cultures. The advancement of technology has led to big progress in the system of flight ticket booking over the years.

Tickets are the documents that confirm purchase and guarantee a seat on the airplane for the chosen journey. Tickets are required as a proof to get the boarding pass at the airport, which is needed in order to board the aircraft. The traditional tickets on the early days of air travel were made of paper and were to be collected from the travel agencies or airline office after purchasing. Along with globalization and the development of aviation industry, the process of ticket purchasing has also changed. Since the rapid growth and use of internet since the 2000s, ticket purchasing has been possible online. Decreasing number of people are now using the traditional paper ticket while almost all major airlines have given the possibility of online tickets, commonly known as e-ticket. A ticket today contains the information of the passenger's name, date of travel, the flight number, destination and origin of travel, fare, taxes, baggage information, rules on changes and refunds, form of payments and the validity of the ticket.

In this thesis, the system behind online ticket booking especially used by airlines is going to be explained along with a demonstration of a working application to show sample online booking. Various sources such as e-books, online articles and websites have been referred to gather the materials for the thesis. The thesis has been completed with the co-operation with the supervisor to take in his valuable

ideas and suggestions for the improvement of the work. The paper has been written with the help of information absorbed from the sources mentioned above and displayed with own understanding. The topic has been chosen as a result of author's interest in the mechanism of modern flight booking process.

After introduction, chapter two of the thesis named "Online Ticket Booking" is discussed. In it, a brief introduction of online ticket booking, the evolution of e-commerce, airline reservation system and working mechanism of flight booking are described. From chapter three onwards, the thesis is focused on the development process of online ticket booking application that has been developed. In chapter three, different programs and software that are used in the development are briefly described. The key aspects that are needed in the development of the working application are described in chapter four. Those aspects are briefly described and explained why they were necessary in developing the project.

Chapter four is followed by chapter five on the topic, which is about the design aspect of the application. Under this chapter, design of the database used, system design and logic of the website are explained and illustrated. Chapter six is dedicated to the various stages of development progress. This is the biggest chapter of thesis where the process of development are described and shown with graphs. The chapter is divided into many sub-chapters, where each sub-chapter focuses on different aspects of the development. This chapter is accompanied with many graphs and screenshots of the code used in the making of application. The last chapter of the thesis, chapter seven, is about the advantages, disadvantages and the challenges faced during the making of application. The advantages and disadvantages sections are about the general pros and cons of the popular online flight booking system that we use today while the challenges is about the personal challenges faced during the development of the project. The final part of the thesis will present the conclusions.

2 ONLINE TICKET BOOKING

Over the years with the advent of internet, online ticket booking for airline travel has been increasingly popular. The main focus of airline companies today has been in the satisfaction of customers. The companies are doing this by making journeys possible in a fully mobile and social environment with the intelligent use of vast quantities of data to deliver real service and operational improvements. The personalized use of mobile phones and web by customers is expected to increase drastically, with the estimation of 90% airline companies offering mobile check-in by 2015. (Travelmole 2013.)

2.1 Development of e-commerce

E-commerce is the business transaction of goods and services done over the internet. With the advent of internet, e-commerce has been growing rapidly in the past few years. It is becoming impossible to imagine a world without e-commerce today. Many major business companies around the world now offer their customers choice of selecting and buying the goods and services at the comfort of their homes. Various sets of communication and interaction rules are used in e-commerce in the form of file transfer, email and shopping carts. E-commerce makes it possible to buy goods and services round the clock which is possible from any part of the world through a computer connected to the internet. Online shopping is not only more convenient than the traditional style of physically going to the stores, but also offers a wider selection of products and good accessibility. (Rouse 2015.)

E-commerce utilizes the electronic payment system for the transactions. The monetary paperless system has completely changed the face of global trade

because of its easy and convenient facilities, minimal paperwork on invoices and minimal labor and administration costs. The most common methods used to pay online are credits cards, debit cards, bank transfer and via other companies such as PayPal. Due to the increase in e-commerce and its easy, hassle-free e-payment system, many businesses, including aviation industry, have consistently grown over the years. (Ueland 2013.)

Online ticket booking is one of the many features of e-commerce. More people are flying than ever, and easy online ticket purchase system is one of the major contributors in the increase of the passengers using air travel. Today, to book an airline ticket for anywhere in the world, one just has to log onto the web address of the airlines company (sometimes other websites which searches the best flights for client), search and purchase. The check-in can also be done online in many major airlines and the boarding pass can be directly printed before going to the airport. (Atkinson 2015.)

2.2 Airline reservations system

Evolving from manual records and logs in the early 1930s, Airlines Reservations System (ARS) is the improved, computerized feature of airline reservations. ARS helps in systematic and effective organization of bookings, prices, schedules and customer data. Airlines reservations system has today evolved into Computer Reservations System (CRS). ARS, when integrated with Global Distribution System (GDS), can be used by multiple distribution channels such as travel agencies, which can then use it for hotel rentals, flight booking, car hires as well as activities and tours via single system. ARS consists of several areas such as the inventory management, availability display and reservation and fare quotes and tickets. (Winston and Morrisson 1995.)

American Airlines introduced the first automated ARS called the Electromechanical Reservisor in 1946. It was followed by a new machine called Magnetronic Reservisor. In 1959, to improve the existing Reservisor, an improved automated booking system called SABRE was introduced. Similarly, other airlines created their own systems for ticket booking and management. Today, many brands co-operate with the world airlines companies for user-friendly direct systems, increased productivity and efficiency. Some of the major ARS brands today are Abacus, Amadeus, Navitaire, Sabre and TravelSky. (Ross 2013.)

2.3 Working mechanism of flight booking

The users can easily purchase an e-ticket by going to the ticket sale website, searching and selecting the destination, entering the details such as name, way of travel, luggage information and dates and finally making the payment via bank cards, bank transfer or through online payment companies. The e-ticket is then emailed or texted on the telephone of the customer. While previously travel agents and airlines assisted customers in making ticket purchases, today with the improved internet system, it is getting easier and easier to book the flights on your own. After the customer makes the purchase, the electronic record and the details of the ticket are saved into airline's database. The database is integrated with the passenger service system, which is then connected to the airports, airlines, travel agencies for sharing real time information. (Crosby 2007.)

3 SOFTWARE AND PROGRAMS USED

A number of software and programs have been used in order to develop the sample online booking system. The types of programs and software used were programming languages such as Python, development platform such as Eclipse, script language such as Javascript, markup language such as HTML and database program such as SQLite, among others. Proper installation of all the programs used were required and basic understanding of the programs were utilized in the completion of the project. Sources such as tutorial videos and official documentations were referred to in the case of insufficient knowledge during the development process.

3.1 Python

Python is a script programming language. Although system programming languages such as C++, Java etc. are highly popular in school and college curriculums, Python is an advanced script programming language, which can be used in many kinds of applications, is easy to learn and has simpler syntax and semantics. Scripting languages are especially well-suited for the introductory programming sequence. Scripting languages generally also have simpler syntax and semantics than system languages. Since Python contains hassle-free syntax rules, it encourages creation of advanced applications within short time. (Laird and Soiraz 1998.)

Python combines the features of programming paradigms such as object-oriented, imperative and procedural style. Especially for new programs, Python spares the use of curly braces, semicolons and other met characters required for syntax in other programming languages. Python is created to be easily readable and look

meticulous to the user and uses simple keywords in English language. Python uses space for indentation of blocks. Use of functions and classes are not mandatory, and hence simple programs can be written simply with fewer lines of codes. Loops, control-structures and use of variables are also considerably easier in Python. There are handful of Python interpreters available today which can be operated in various operating systems. In the recent years, Python is a popular programming languages and there are various reading and exercising materials available. (Mertz 2001.)

3.2 Django

Django is a web framework designed for the development of easy and yet also advanced web applications. Django is invented to encourage rapid development and clean, pragmatic design. Some of the key features of Django are its facilitation of rapid web development, reliable security and impressive scalability. Django was the brainchild of World Online developers Adrian Holovaty and Simon Willison in 2003. As they began to develop websites using Django, they constantly added new features which helped in developing rich, interactive websites quicker and quicker. (Kaplan-Moss and Holovaty 2007.)

Django combines the use of other open-source projects such as Apache, Python and PostgreSQL. Django is an effective programming tool to make rich websites. Since Django is a web framework, it saves the developer time of making bits and pieces from the scratch. Django is a collection of libraries written in Python language, and hence coding in python is required for development in Django. Django is an open-source software, and today there is a community of volunteers who run Django. Users can continually contribute and improve Django and hence it's very popular today. It has been used by notable web applications such as Instagram, Pinterest, Mozilla and Pitchfork. (Django Software Foundation 2013.)

3.3 HyperText Markup Language

HyperText Markup Language (HTML) is a popular markup language used in web pages. HTML can be simply written in a text editor and tested through web browser. Writing in html is easy; with html it's also possible to add media and images to the web page. HTML contains special markup tags like <title>, <h>, <p> etc. (Encyclopedia Britannica 2015.) To declare the title of the page for example, the title has to be included in the title tags. Similarly, the paragraphs, headings and other different contents in website pages have to be included inside the respective HTML tags.

It is easy to store HTML code; in a simple text file with filename followed by .html or .htm extension. HTML gives developers the possibility of creating sections in the document. As <title> tag gives the title for the web page, <H1> for example specifies the main content of the web page. Similarly, with H2, H3 and so on tags, HTML creates minor contents. There are tags for other features such as paragraphs (<p>), style of font (bold) and tables (<table>) etc. (Shannon 2012.)

3.4 Jinja

Jinja is a user-friendly, template language that uses Python. Jinja is used as the base for highly interactive webpages. It offers user-friendly features such as sandboxed execution, template inheritance and configurable syntax. Syntax of Jinja can be written within HTML template. Jinja is derived similarly as Django. Simplicity is one of the best features of Jinja. Notable companies such Mozilla have used Jinja for their development. Jinja can be simply written in a text file, in any format. The variables and expressions in template get replaced with values when template is rendered. (Ronacher 2008.)

3.5 Javascript

Javascript is one of the most popular script languages for webpages today. Javascript (JS) is an object-oriented language that supports features such as imperative and functional programming. The syntax is similar to other object-oriented languages such as Java and C++, hence making it easy to learn for people who know these languages. JS is used to add interactive features such as buttons, animation, games etc. JS was invented by Brendon Eich. Javascript is capable of creating many different features ranging from beginner to advanced such as 2D and 3D features on a website. (JavaScript Basics 2015.)

3.6 Cascading Style Sheet (CSS)

CSS is a language that was developed in 1992-1993. CSS helps design each and every element of the markup elements language such as HTML by giving complete control to the designer. While the HTML elements enables the web page designers to add what content they want, CSS makes it possible how to display the content to the user. CSS covers the areas such as colors, layout, advanced positions of elements, fonts and also allows the content to adapt the content to different devices such as phones, tablets, bigger screens and printers. CSS can operate independently as well as be used with any markup languages based in XML. (Lie and Bos 1999.)

CSS uses simple, everyday English words and has an easy syntax. CSS is crucial in advanced web designing as it gives control to the layout and offers numerous techniques to make the web page look sophisticated. Currently, the basic features of CSS are supported by all main browsers such as Internet Explorer, Safari,

Opera, Chrome and Firefox. (UKOLN 2006.) CSS has been used in the web pages for the development of sample online booking system for this thesis for positioning, layout, margins and colors for the HTML elements.

3.7 SQLite Database

Databases are collections of similar data. Databases are used for organized collection and storing of similar data, to be later used for specific purposes. A database contains tables with rows and columns populated with objects, which displays connection between them. A database acts as a shared resource for the programs which can use the information from the database. Many enterprises rely on databases today to store a wide range of information systematically. Databases are used almost everywhere: in small companies which can use database to save customers' information and also for more advanced scientific and military areas. Databases facilitate the task of searching hundreds and thousands of records much simpler by storing them in an organized manner. (Encyclopedia Britannica 2015). Among many database programs available today, SQLite Database has been used in the development work for this thesis.

4 FUNDAMENTAL ASPECTS USED IN APPLICATION

The development of application required to use many aspects such as website designing and development, working with databases, working with logics in programming languages such as Python and Javascript as well as using local server in Django to run, test and modify the application. Basic knowledge in such features as well as more learning along the development process were needed to write the functionalities. These basic features are needed to make any kind of websites or web applications. Some fundamental features that are used in the application development are briefly described.

4.1 Website

A website is a collection of web pages. The information that is generated by many sources is carefully organized, and made into files and web pages. It is then merged together to form a single entity called website. (Rouse 2005.). The data that is stored in the website is displayed to the user with the help of different programming languages like HTML, CSS and PHP. Websites are used in online business, as sources of information, marketing, for developing relationships and for entertainment among others.

There are special techniques to rendering the web elements to the user to make it look nicer and more organized which fall under the web designing area. For loading websites, hosted servers are required. The servers make it possible to transfer the pages to any web browser through HTTP, the main protocol of WWW (World Wide Web). Website can be easily accessed by simply typing the address at the address bar. Most website names are easy to remember and begin with

'www'. Although there are countless varieties of websites out there today, there are common elements. Most, if not all, websites have a homepage, the first page that user sees after the page has loaded a navigation menu which contains other sub-pages and a footer area with information such as contact, copyright information and others. (Skutin 2014.)

4.2 Web application

A web application, as the name suggests, is an interactive application in web. It is a collection of many components of HTML and other programming languages such as servlets, html pages, classes and other resources. The application can be inside a website which is loaded via servlet when the address is entered. (Chaffe 2012.). Web applications are mainly used for action rather than information. Web applications have capabilities of utilizing advanced device features that it is used with. Web applications can often also work offline. Web applications are important features of web technology. (Borodescu 2013.)

Anything that interacts with the user through the web page can be classified as a web application. Web applications give numerous possibilities to businesses and similar pages to interact with who is viewing their page. A simple survey embedded in webpage, or a newsletter signup form, a login form and a shopping basket are all examples of web applications. Web applications essentially capture all the data, store and process them for further use. Web apps operate within layers: the first layer is usually a web browser; the second layer is the tool that generates content or servlets and the third layer is the database that has the content as well as customer data. (Aceuntix 2015.)

4.3 Web designing

Web designing is an essential part in the development of website and web applications. Web designing specifically is about the front-end of the page that the users view. The basic aspect of any web pages is with two different things: one is with the front-end, meaning what the users can see and how they can interact while the other one is the back-end of the page where all the data, functionalities and information are stored. A web designer can design how the page looks to the user using basic markup languages such as HTML among others. There are also many other aspects associated with the vast area of web designing. A designer has to keep in mind many things such as appearance of the web page which may or may not look the same on different browsers, the knowledge of monitor sizes, display settings and the interactivity and dynamism of the website. Although it is a tedious and relatively hard job, web designing offers limitless opportunities on the development of web pages. (McGuigan 2015.)

4.4 Web development

Web development, or web programming, is an important aspect of building websites and web applications by programming a set of logical instructions to the computer. Programming languages such as Java, Perl, PHP, C, C++, ASP.NET are used in the development of web applications. While web design takes care of the front-end of the website, web programming is responsible for the stuff that goes behind the view of the user. The back-end features of any websites generally include databases, content, logics, interaction of the pages and the development of applications. When any page is opened in the browser, the web server parses and reads the HTML page step by step. When it comes across programming languages, it will then implement the code. Web programming is essential as it has the power of turning simple and static HTML pages into a dynamic, user-interactive website with many features. (Holiday 2014.)

5 THE DESIGN STAGE

The design plan of the application was important for fast and efficient application development. A planned layout of the website was prepared at first. Features such as system design, database design and the way application was intended to work had to be kept in mind before starting the implementation. After downloading the necessary platforms and programs for implementation and learning about the features that were planned to incorporate in the application, the implementation process started.

5.1 System Design

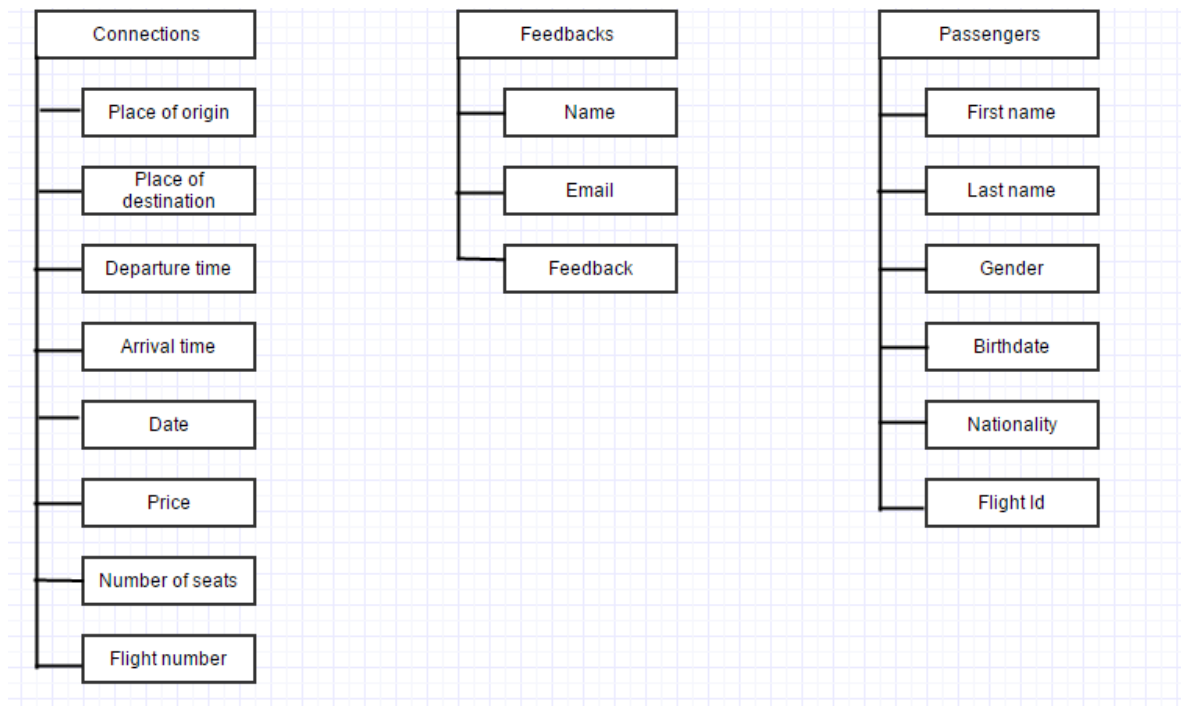
The web-based ticket system uses client/server architecture. In client/server computing model, a server hosts, provides and manages the resources and services made for the client. This kind of model usually has more than one client computers connected to central server over internet connection. All requests and services are delivered over the network in this model and the system shares resources. The internet itself is also based on this architecture. (Techopedia 2015.)

In online ticket booking model, a client can use a web browser to access the website and book online tickets. It is very easy to make the booking as the system is user-friendly. The user does not have to worry about the operating system either as the web page is supported in all popular browsers on all platforms. For the server, Django server is used in the application. Most of the code is written in Python as Django platform is used. At the flight search query, when the user searches preferred choices, the web server searches the database for similarities and returns the result in user-readable form. After the customer books the selection and the payment is verified, the data of client is saved to the company's

database. Depending on the selection, users are sent an e-ticket or SMS confirmation. In the sample application created for this thesis, users are simply sent the e-ticket to their email address after booking.

5.2 Database design

The database used in the application is SQLite which comes installed with Django package. The data is divided into four main parts named Connection, Feedback and Ticket. In the tables under Connection, there is data such as place of origin, place of destination, departure time, arrival time, date, price, seats available and flight name. The information in Connection was manually coded. The Feedback section is filled with the customer's name, email address and design when s/he leaves a feedback or message on the page through the contact form in the website. The ticket section contains the itinerary of the client after s/he has made the booking. This section contains information like the personal details of customer such as name, last name, birth date, nationality and the booked flight details. This information is later used to make ticket and send it via email to the client. Graph 1 shows the entries under Connection, Feedback and Passenger tables used in the database of the application.



GRAPH 1 Database tables

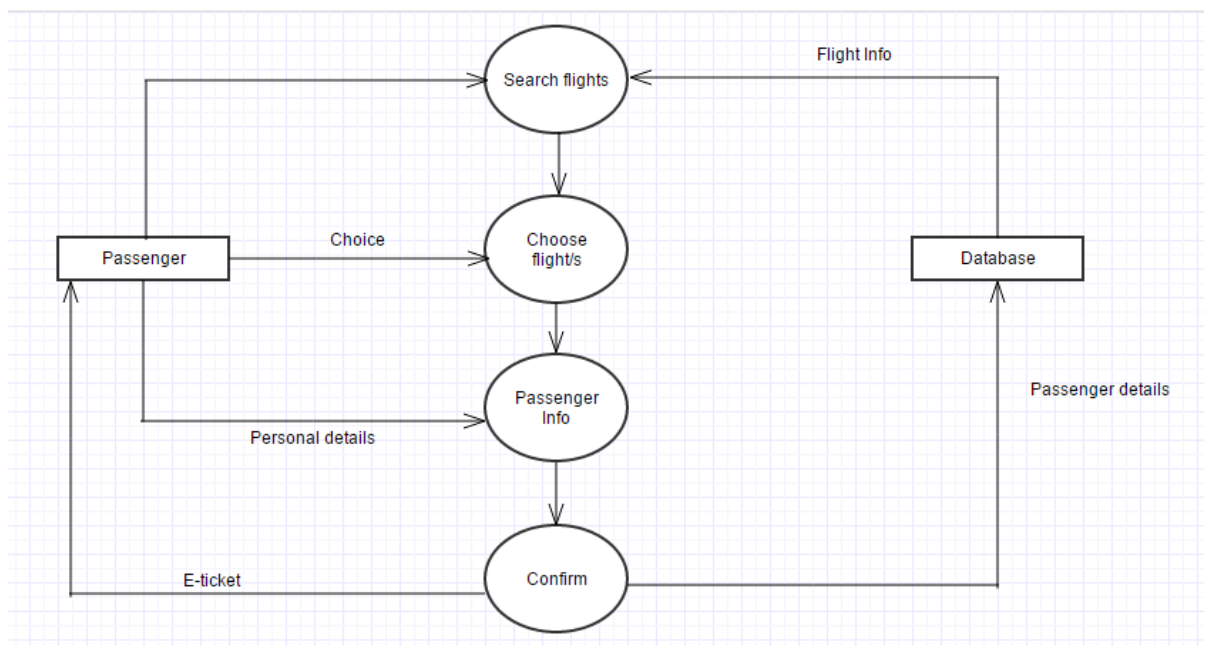
On the admin side of the database, the admin can simply manage the content of database manually. Admin has access to all the data in the database. The admin user is first created by creating username, entering email address and desired password in Django shell. After this, the admin can log in to the admin site of the webpage and make edits to the content. The admin can do any changes such as updating the tables in database, adding new data to it, creating new groups and users etc. on the admin part of the website.

5.3 The website and its logic

The website is based on an imaginary company called 'FlyHigh'. The imaginary airlines company is based in Nordic region and operates budget flights between Nordic cities. The website of the company is simple. There are five subpages which are easy to navigate through the index. The background and simple font style is the same in all the pages. Once the user goes into the web page, s/he can

find the home page, where there is some recent news about the company as well as the rules. Besides home, the other subpages are named "Flight Search", "About Us", "Fares and Regulations" and "Contact".

The main functionality of the application lies in the page called "Flight Search" where the user can make flight bookings. The layout is very simple: user can select the place of origin and place of destination along with the preferred date of departure. There is option to check the box for one way flight or enter the return date for two way flight. The user also has to specify the number of adult tickets and/or child and infant tickets. All the fields except one way checkbox, children and infant ticket are mandatory.



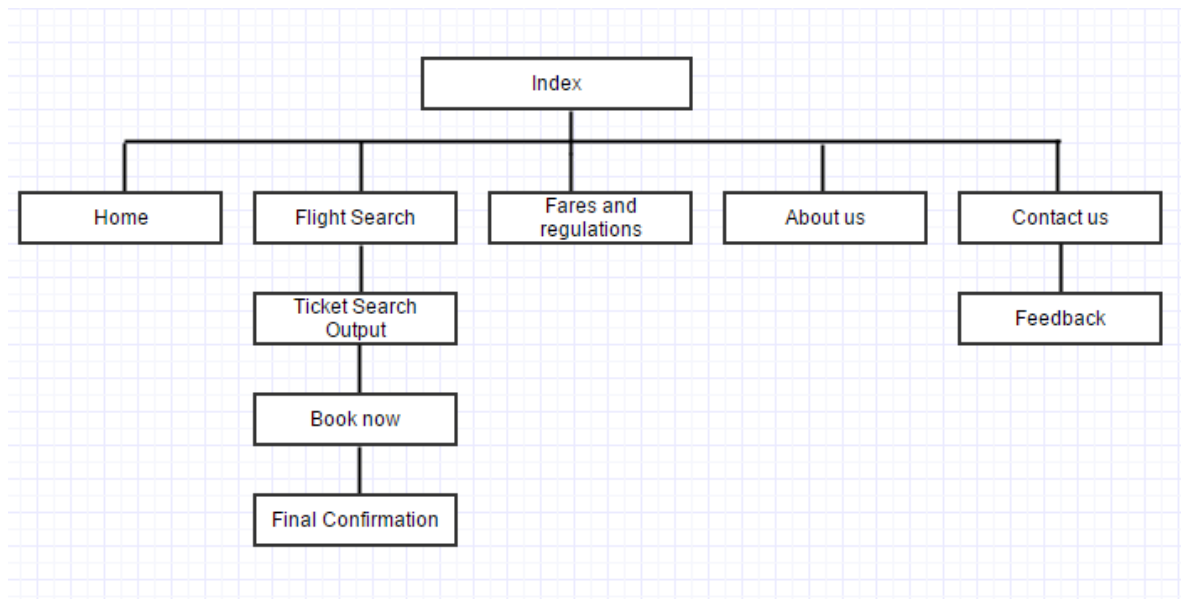
GRAPH 2 General working flow

Graph 2 presents the general working flow of the application where the user can first search flights. After the search query is passed, the data from database is fetched. If the exact date of departure or/and arrival is not found, the nearest available dates are shown. In the search results, flight route, date of travel, time of

departure and arrival and the price is shown. The price for children and infants are programmed to be a third and sixth of the adult price respectively. When displaying to the user, just the total price is shown. After this the user can proceed to select the desired flight via radio button. On the next page, the user is asked to enter details such as first name, last name, nationality, birthdate, gender and contact email. After this step, the user can finally confirm the booking by clicking 'Confirm Reservation' button. At the back-end of the website, after the button is pressed, the data is copied to database. The database then uses the data to email the ticket with the passenger and booked flight details to the email address of the user.

5.4 The website pages

The website has five main pages which can be navigated by the simple index. The user first lands on the home page. Home page has content like recent news about the company, and links to other subpages. Similarly, pages 'About' and 'Contact' hold the information about the company and contact details respectively. In 'Contact' page, the user can make use of the feedback form to send message to the company. In 'Fares and regulations' page, the user can find information about the policies of the company regarding fares and regulations. There is 'Flight Search' page from which the user can search flights and proceed to make reservations. Except 'Contact' and 'Flight Search' pages, the remaining three main pages are static. Graph 3 shows the main web pages and sub-pages of the application.



GRAPH 3 Web pages and subpages

The application is developed with the aim of displaying and learning how the flight booking system works. SQLite database is used for storage of data. Upon entering the homepage of the website, the customer has option to search flights. The user has to simply select the place of origin and destination, along with date and number of passengers. The user also has option to search return flight for the same path. If no departure or return for the exact same date is found, the ones that are available are displayed in the results. Users can select the connections of their choice from the results, enter their contact email address and proceed to booking. In booking page, users can enter their personal details such as name, birthdate, nationality and gender. Alongside the booking form, the flight details that they have booked is displayed. After filling out the details, the users can simply click the "Confirm reservation" button after which they will receive an eticket on their email address.

On the backend of the application, when a search request is submitted for flight search, the information from Django form is stored and checked in database. If the selected flightpath for the exact date is not found, then the connection for available dates is displayed. When the users have confirmed the reservation and entered their email address, the eticket with the flight details that was saved on session

along with the personal details received through Django form is sent by email to the users. The personal details entered by the users are saved in the database of the company admin page.

6 THE DEVELOPMENT STAGE

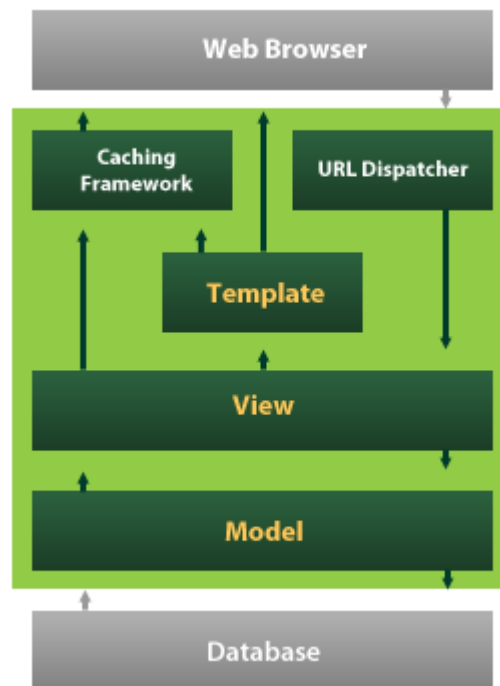
The development stage was the most crucial part of the thesis. Although it is a simple web application, it involved working with different programs and their features. After deciding on the general format of the application, all the programs that were necessary were installed correctly. The development process was carried out step-by-step and many new techniques and features were learned during the process from various sources. Along with programming, the process also involved frequent testing and modifying to produce the desired version in application. The fictional airlines company is named as “FlyHigh” which is a Nordic-based airlines offering cheap connection between Nordic cities.

6.1 Setting up Django framework

The application was developed with Django web framework and was written in Python programming languages. Django has components that work together in different areas. The SQLite database that is used in the application is supported by Django and comes installed with the packages. To start the project, it was important to have configured Django modules properly.

Graph 4 shows the Django architecture. The `urls.py` module, which is the URL dispatcher, tracks the requested URL in the view function, which then does some operations and returns value through defined template. Views function often involves reading, writing or fetching from the database, but can also perform other actions. The model module, which is `models.py`, defines the data in database and interacts with it. In Django, databases such as MySQL and SQLite are often used but Django is capable of supporting other databases too. Templates are usually

HTML pages. When views module runs the function, it returns result in template, which is then displayed in the defined html page to the user.



GRAPH 4 Django Architecture (Adapted from MyTardis 2013.)

For setting up the project, a few commands on command prompt in Django location with "manage.py" application is run. Django automatically generates the modules and collection of settings once the project is set up. To view the empty project on browser, Django server has to be run locally. Once the server is running, the project can be viewed by typing "127:0:0:1:8000" or "localhost:8000" on the address bar. Graph 5 shows the running of local server in Django with simple one line command.

```
C:\Users\Naczelny\workspace\flyhigh_project\mysite_flyhigh>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
August 20, 2015 - 19:11:35
Django version 1.8.2, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

GRAPH 5 Running local server in Django

6.2 Setting up the database

The `models.py` module is responsible for populating the database in Django. The fields in models were defined for three tables named Connection, Feedback and Passenger. All the fields in respective tables are the initialized and stored as the right type such as IntegerField for data holding integer values, CharField for data holding character values, DateField for values with date, EmailField for email data etc. After this was done, it was then required to activate the models. For this, the name of on Django shell, on the project location with `manager.py`, command “`python manage.py makemigrations`” is run. This Django command is for storing the changes in database done through models as migration. Then, “`python manage.py migrate`” to create the models in database. It is necessary to run these two commands every time a change has been made in `models.py`.

6.3 Creating admin user

Django gives the possibility of creating an admin user, who has access to all the content of the website. This permission is only for the site managers who have the authority to modify, add and delete content. The admin user can also see and modify the database. Admin access is created in Django shell. User and password are set up which is later used to log in to the admin part of the page. The admin page can be opened by simply typing “`localhost:8000/admin`” in the address bar and logging in. Graph 6 shows the admin panel. Graph 6 shows the admin view of the application where the admin can modify, add and delete content manually.

Django administration

Site administration

| Authentication and Authorization | |
|----------------------------------|--|
| Groups | + Add Change |
| Users | + Add Change |
| Flyhigh_App | |
| Connections | + Add Change |
| Feedbacks | + Add Change |
| Passengers | + Add Change |

GRAPH 6 Admin view

6.4 Making static web pages

An index with five main pages: “Home”, “Flight Search”, “About”, “Fares and Regulations” and “Contact us” is created with uniform design and CSS in all pages. A code snippet of index.html is shown in Graph 7. The background color, title, font and style of the web pages are defined in the code. This design is uniform in all the other web pages.

```

<html>
<body bgcolor="#CDE18C">
<h1><font face="arial, sans-serif" size="+1" color="#6663366">Welcome to FlyHigh!</font></h1>
</body>
</html>

<title>FlyHigh Homepage</title>

<style type="text/css">
#tabContainer {
padding: 25px 15px 0 15px;
background: #67A897;
}

ul#topTab {
list-style-type: none;
width: 100%;
position: relative;
height: 27px;
font-size: 13px;
font-weight: bold;
margin: 0;
padding: 11px 0 0 0;
}

```

GRAPH 7 Designing the index

The index page acts as master page for all the other subpages. Graph 8 shows the code snippet where the other main pages are defined within index. The design and layout of the sub pages are the same as index page. The five main pages named “Flight Search”, “About”, “Contact Us” and “Fares and Regulations” are hence embedded with the index page.

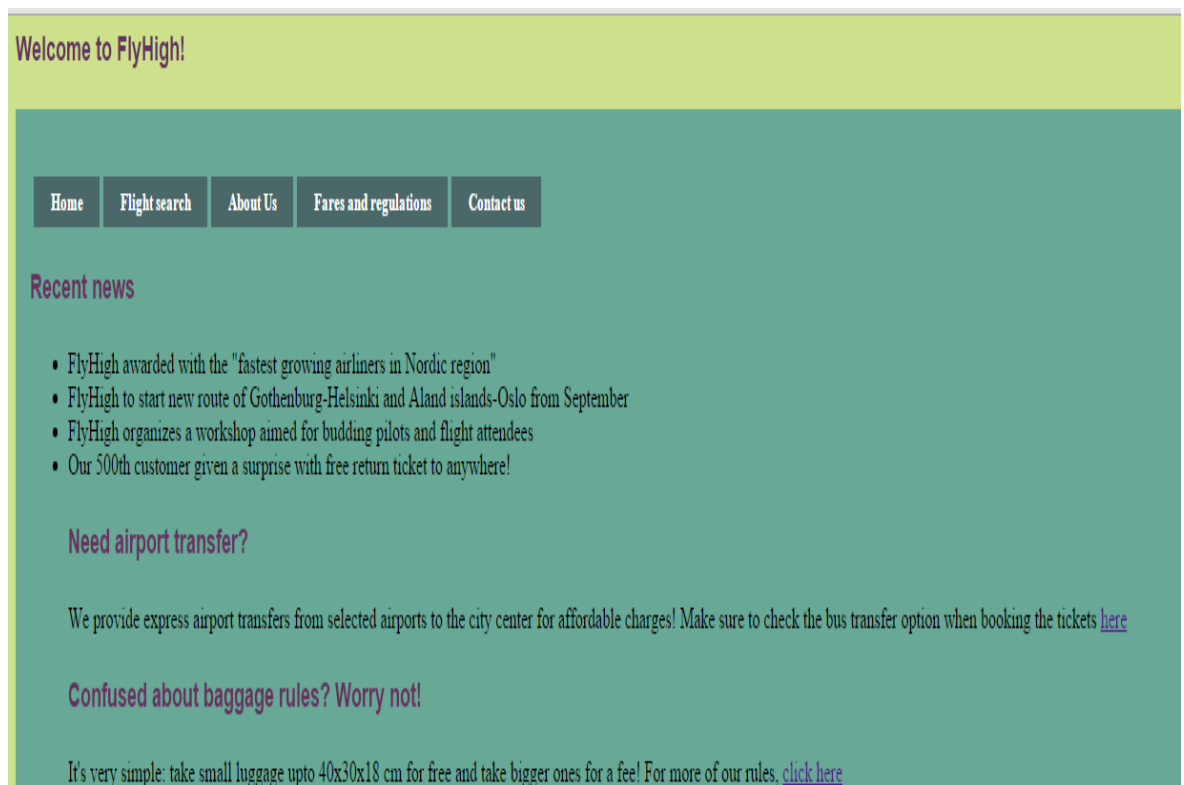
```

<body>
  <div id="tabContainer">
    <ul id="topTab">
      <li class="active"><a data-toggle="tab" href="{% url 'home_page' %}">Home</a></li>
      <li><a href="flight_search" title="Search flights">Flight search</a></li>
      <li><a data-toggle="tab" href="about_page" title="About us">About Us</a></li>
      <li><a data-toggle="tab" href="fares_page" title="Fares and regulations">Fares and regulations</a></li>
      <li><a data-toggle="tab" href="contact_page" title="Contact us">Contact us</a></li>
    </ul>
    {% block content %}{% endblock content %}
  </div>
</body>
</html>

```

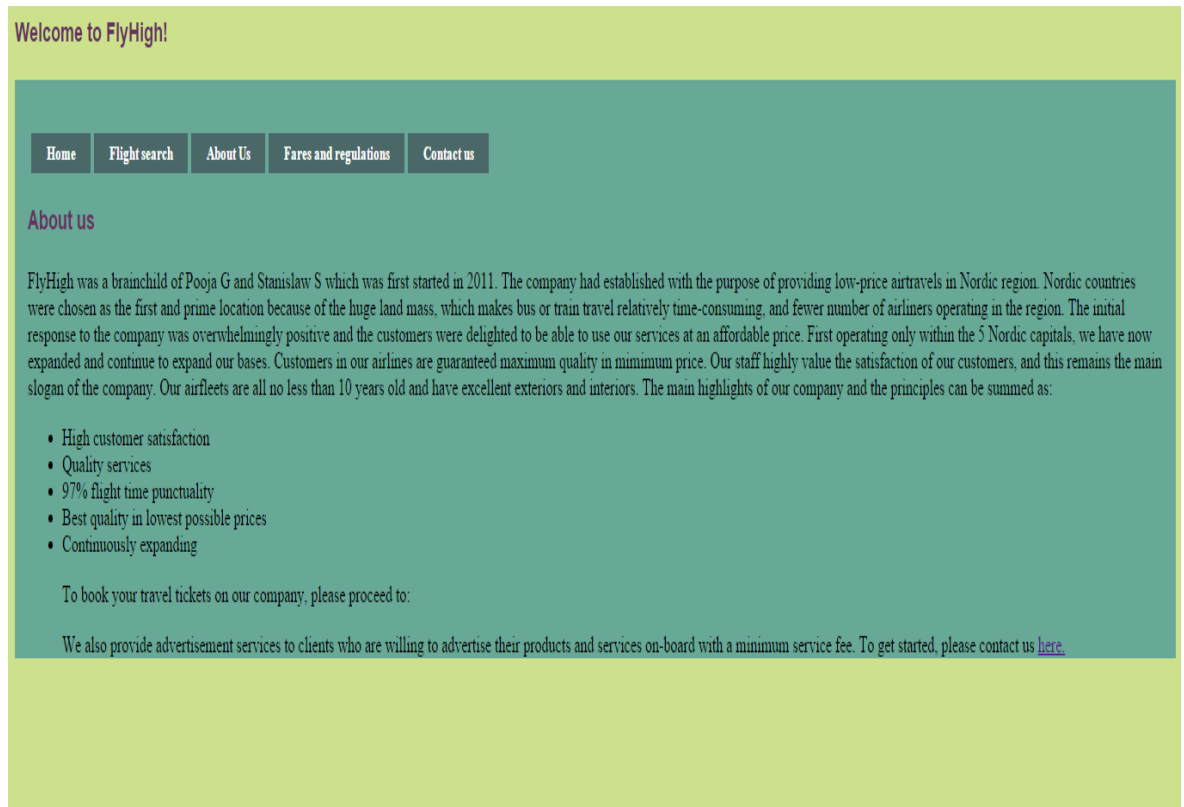
GRAPH 8 Defining main pages within index

The home page displays static content about recent news of the company, short information about airport transfer and baggage rules. The homepage is the first page that the user sees when opening the website. Under the information about airport transfer and baggage rules, there are links directing to respective pages. Graph 9 shows how the homepage of “FlyHigh” company looks.



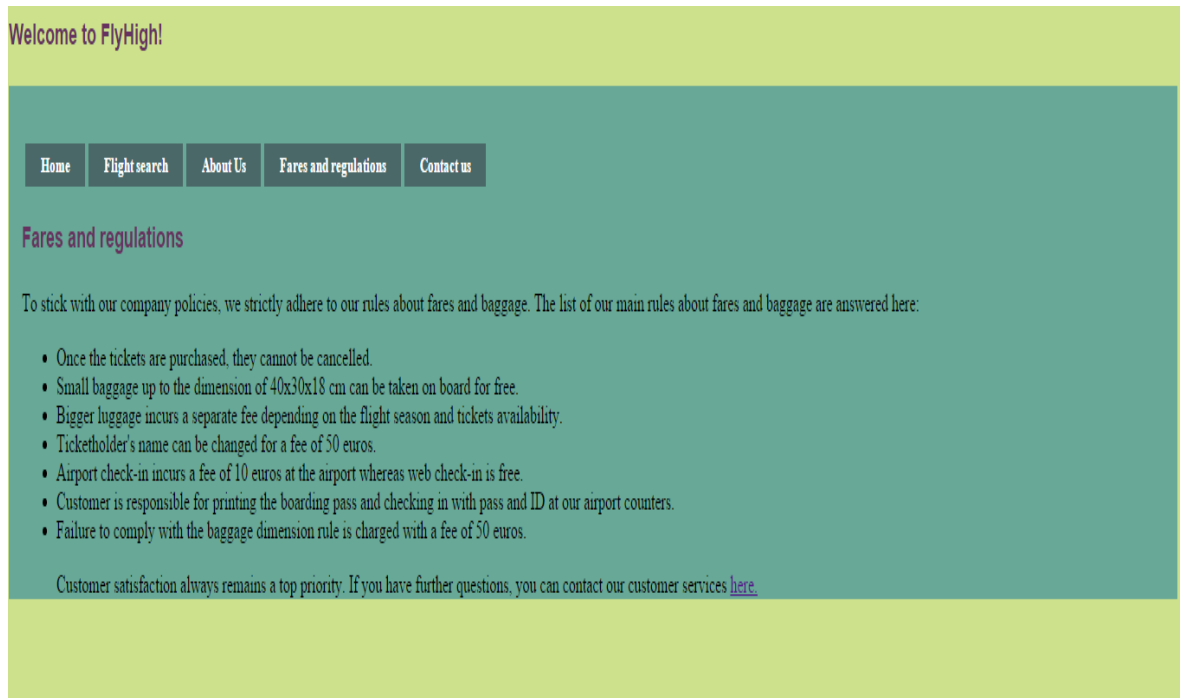
GRAPH 9 Homepage

Three of the five main pages at index are static pages. For adding any web pages, they have to be defined in `urls.py`. Then in `views.py`, the functions which perform functions and return them via templates, have to be defined. For the pages “Home”, “About” and “Fares and Regulations”, some information in text is added. In `views.py`, the functions just render the template defined for these pages.



GRAPH 10 About page

Graph 10 shows the “About” page of the application. “About” is static page which displays content about the “FlyHigh” company. Some features of the company are displayed in points and there is a link direction to “Contact” page at the bottom. The layout is simple. Graph 11 shows the “Fares and Regulations” page which is similar to “About” page. The fares and regulations policy of the company are displayed as static text in points in this page.



GRAPH 11 Fares and regulations page

6.5 User-interactive pages

The two main pages that are user-interactive are "Flight Search", where the users can search and book flights and "Contact Us" page where the user can view the contact information about the company as well as send feedback through feedback form. Through "Flight Search" page, other interactive pages open. The first interactive sub-page is for flight selection, where the users can select the flight from results and enter their email address. The second interactive sub-page is flight booking, where the users have to enter their personal details in order to confirm the booking.

6.5.1 Use of Django forms

Django forms are very useful for user interaction. Django form library contains many features such as displaying form in HTML, checking submitted data against validation conditions, converting form data into correct types for Python, which can all be easily used. (Django Documentation, 2015) . Django forms are used in at least three places in the application to fetch the data from the user. Graph 12 shows the code snippet of feedback form that's embedded in "Contact Us" page. The feedback form is declared and the fields are defined.

```

1 from django import forms
2 from flyhigh_app.models import Connection
3 from django.forms.extras.widgets import SelectDateWidget
4 from django.core.exceptions import ValidationError
5
6 from datetime import date
7
8 class FeedbackForm(forms.Form):
9     name = forms.CharField(label = "name", required = True, max_length = 50)
10    email = forms.EmailField(label = "email", required = True)
11    feedback = forms.CharField(label = "feedback", required = True,max_length = 500)
12

```

GRAPH 12 Declaring feedback form

For flight search functionality, the selection of user has to be saved to later fetch data from database. A form called FlightSearchForm is hence defined where the fields are declared. Condition for checking form validation are also defined. Graph 13 shows the declaration of FlightSearchForm and its fields.

```

13 class FlightSearchForm(forms.Form):
14     flights = Connection.objects.order_by('date')
15     origins = set()
16     destinations = set()
17     for flight in flights:
18         origins.add(flight.place_origin)
19         destinations.add(flight.place_destination)
20     flight_from_choices = [(c, c) for c in origins]
21     flight_to_choices = [(c, c) for c in destinations]
22     numbers_choices = [(n, n) for n in range(21)]
23     origin = forms.ChoiceField(label="From:", choices=flight_from_choices, widget=forms.Select(attrs={'onchange': 'updateDestinations();'}))
24     destination = forms.ChoiceField(label="To:", choices=flight_to_choices)
25     one_way = forms.BooleanField(required=False, label='One way:', widget=forms.CheckboxInput(attrs={'onclick': 'validate();'}))
26     depart_date = forms.DateField(label="Date of departure:", initial=date.today(), widget=SelectDateWidget())
27     return_date = forms.DateField(required=False, label="Date of return:", initial=date.today(), widget=SelectDateWidget())
28     adult_ticket = forms.ChoiceField(label='Adult', choices=numbers_choices[1:])
29     child_ticket = forms.ChoiceField(label='Children[2-16years]', choices=numbers_choices)
30     infant_ticket = forms.ChoiceField(label='Infant[0-2years]', choices=numbers_choices)
31
32     def clean(self):
33         flights = Connection.objects.order_by('date')
34         cleaned_data = self.cleaned_data
35         if not cleaned_data["one_way"] and cleaned_data["return_date"] < cleaned_data["depart_date"]:
36             raise ValidationError("Return date cannot be before selected flight.")

```

GRAPH 13 Declaring flight search form

6.5.2 Views and templates

In views.py, it's checked if the form is valid and the data from form is saved. For Flight Search form, the function in views is defined as shown in Graph 14. Here the choices displayed in the choice fields for the place of origin and place of destination are synchronized according to the entry in database. This function returns the suitable origins and destinations which is rendered via "Flight Search" page.

```

10 |
11 def flight_search(request):
12     template_name = "flight_search.html"
13     flights = Connection.objects.order_by('place_origin')
14     origins = set()
15     destinations = {}
16     for flight in flights:
17         origins.add(flight.place_origin)
18         if not flight.place_origin in destinations:
19             destinations[flight.place_origin] = set()
20         destinations[flight.place_origin].add(flight.place_destination)
21     for key in destinations:
22         destinations[key]=list(destinations[key])
23     flightsearch = FlightSearchForm()
24     return render(request, template_name, {'origins': origins, 'destinations': json.dumps(destinations),
25                                           'flights': flights, 'form':flightsearch})
26

```

GRAPH 14 Flight search function

To display the form elements on HTML template, the form is simply called from inside Jinja tags. Django gives possibilities of displaying the form elements in different styles. In the application, the form elements are displayed as paragraph. Graph 15 shows the code where the flight search results are displayed as paragraph in a form via “Search Results” page.

```

26 <title> FlyHigh flight search </title>
27 <html>
28   <h2>Search flights: </h2>
29 <form id="search_form" class = 'form-search' action="ticketsearch_output" method="POST" enctype="multipart/form-data">
30 <div class = "input-append">
31   {% csrf_token %}
32   {{form.as_p}}
33
34 <input class = "oblique" id="submit_button" name = "submit_button" type="submit" value="Search" />
35
36 </div>
37 </form>

```

GRAPH 15 Displaying search results in form

The flight search page contains form of elements. The user can search flight connections by filling out the fields with desired entries. The form is displayed as a paragraph in a simple layout. The data that users enter is sent to the server and some part of it is saved in sessions which is used later for booking and preparing e-ticket. Graph 16 shows the flight search page where the user can search flights. “From”, “To”, “Date of departure” and “Adult” fields are mandatory and the other fields are optional.

Welcome to FlyHigh!

[Home](#) [Flight search](#) [About Us](#) [Fares and regulations](#) [Contact us](#)

Search flights:

From:

To:

One way:

Date of departure:

Date of return:

Adult:

Children[2-16years]:

Infant[0-2years]:

GRAPH 16 Flight search page

A small form called Feedback Form can be found in the “Contact Us” page. The users can send in their feedbacks to the company using this form. The form contains three fields where the users can enter their name, email address and message. The data from Feedback Form after sending is saved into company’s database. Graph 17 shows the view of Contact Page with Feedback Form.

Welcome to FlyHigh!

[Home](#) [Flight search](#) [About Us](#) [Fares and regulations](#) [Contact us](#)

Contact us!

FlyHigh Airlines Company Ltd
Pengerkatu 21, Helsinki 4531
Email address: flyhigh@flyhigh.com
Customer service: (+358) 456-875-67
Hours: Mo-Fri from 9:00-17:00 (Eastern European Time)

Feedback form

Your name:

Your email:

Your message:

GRAPH 17 Contact page with feedback form

6.5.3 Javascript functions for conditions

In flight search page, there are certain conditions when searching the flight. The destinations have to be updated on the basis of selected place of departure, the return flight connections have to be made inactive if the one-way check box is

selected. This is all done by defining functions in Javascript which is shown in Graph 18.

```

6  function validate() {
7      if (document.getElementById('id_one_way').checked) {
8          document.getElementById('id_return_date_day').disabled=true
9          document.getElementById('id_return_date_month').disabled=true
10         document.getElementById('id_return_date_year').disabled=true
11     } else {
12         document.getElementById('id_return_date_day').disabled=false
13         document.getElementById('id_return_date_month').disabled=false
14         document.getElementById('id_return_date_year').disabled=false     }
15     }
16
17     function updateDestinations(){
18         document.getElementById('id_destination').length = 0 //makes drop down menu for 'to' empty
19         var destinations = {{destinations|safe}} //value of destinations from jinja, which is passed in views
20         console.log(document.getElementById('id_origin').value)
21         for (var i=0; i<destinations[document.getElementById('id_origin').value].length; i++){
22             document.getElementById('id_destination')[document.getElementById('id_destination').length] =
23                 | new Option(destinations[document.getElementById('id_origin').value][i],
24                             destinations[document.getElementById('id_origin').value][i]);
25         }
26     }
27 </script>

```

GRAPH 18 Defining JavaScript functions

6.5.4 Interactive subpages

When the search query is requested, result is displayed after checking in database for the searched connections. In views.py, the data from form is fetched, a connection to database is made and a comparison is made for the query and records. A part of function which appends data to departing flights is shown in Graph 19.

```

D4
65 if form.is_valid():
66     t_origin = form.cleaned_data["origin"]
67     t_destination = form.cleaned_data["destination"]
68     t_oneway = form.cleaned_data["one_way"]
69     t_departuredate = form.cleaned_data["depart_date"]
70     t_returndate = form.cleaned_data["return_date"]
71     t_adultticket = form.cleaned_data["adult_ticket"]
72     t_childrenticket = form.cleaned_data["child_ticket"]
73     t_infantsticket = form.cleaned_data["infant_ticket"]
74     flights = Connection.objects.filter(place_origin=t_origin, place_destination=t_destination)
75     departures_same_date = []
76     departures_other_dates = []
77     for flight in flights:
78         print("from database: "+str(flight.date)+" , from form: "+str(t_departuredate))
79         if flight.date == t_departuredate and int(flight.seats_available) >= int(t_adultticket)+int(t_childrenticket)+int(t_infantsticket):
80             print(flight.seats_available)
81             departures_same_date.append({'flight': flight, 'total_price': int(flight.price)*int(t_adultticket)+0.6*int(flight.price)*
82                                     int(t_childrenticket)+0.3*int(flight.price)*int(t_infantsticket)})
83         elif int(flight.seats_available) >= int(t_adultticket)+int(t_childrenticket)+int(t_infantsticket):
84             print(flight.seats_available)
85             departures_other_dates.append({'flight': flight, 'total_price': int(flight.price)*int(t_adultticket)+0.6*int(flight.price)*int(t_childrenticket)
86                                         +0.3*int(flight.price)*int(t_infantsticket)})

```

GRAPH 19 Appending correct data to departing flights

The requested query from flight search form is taken and checked with the database. If the connection for the same date is found then it is displayed. If not, then the available dates for the selected route is displayed in departing flights. The same is true for the return flights. The results are shown and can be selected with radio buttons. The selection via radio button is mandatory to proceed to booking. Graph 20 shows the result page where the user can make selection via radio button and proceed to booking.

Welcome to FlyHigh!

Home Flight search About Us Fares and regulations Contact us

AVAILABLE DEPARTURE DATES:

Helsinki -> Oslo

- July 11, 2015
6 a.m. -> 7:30 a.m.
Price (in Euros): 70.0
- Aug. 18, 2015
10:25 a.m. -> noon
Price (in Euros): 120.0

AVAILABLE RETURN DATES:

Oslo -> Helsinki

- Aug. 24, 2015
6 p.m. -> 7:30 p.m.
Total price (in Euros): 80.0

Contact Email:

GRAPH 20 Choosing from the flight search results

After the users have selected the connections and submitted through “Book Now” button, they are directed to a page named “Book now”. In this page, the users have to enter their personal details before booking the flight. The selected flight details made by the users are displayed on the top. The personal details submitted by the user are saved into the company’s database. Graph 21 shows the booking page.

Welcome to FlyHigh!

[Home](#) [Flight search](#) [About Us](#) [Fares and regulations](#) [Contact us](#)

Selected departure flight:
From: Helsinki
To: Oslo
Travel date: July 11, 2015
Departure: 6 a.m.
Arrival: 7:30 a.m.
FH100

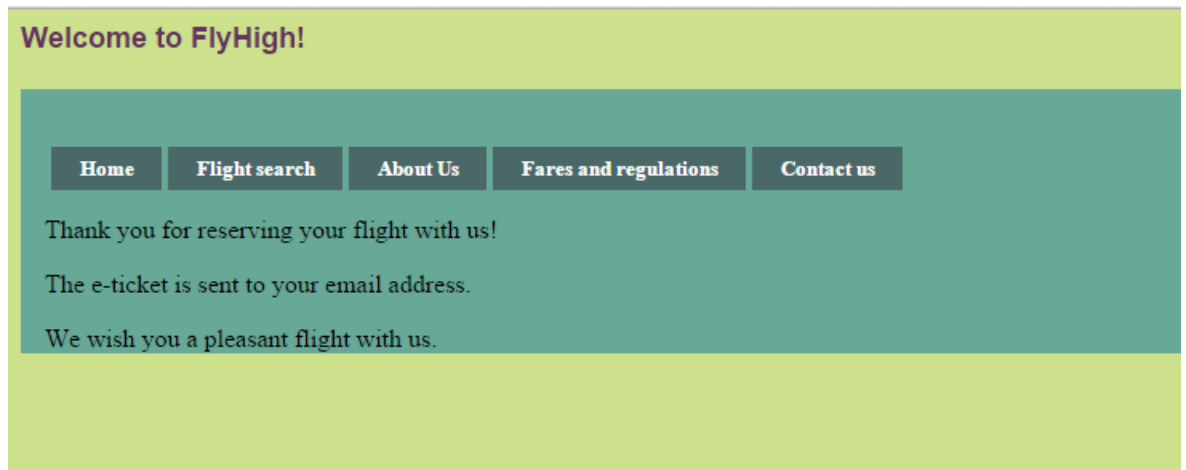
Selected return flight:
From: Oslo
To: Helsinki
Travel date: Aug. 24, 2015
Departure: 6 p.m.
Arrival: 7:30 p.m.
FH100
Grand total: 150.0 euros

Passengers details:

First name: Last name: Gender: Birthdate: Nationality:

GRAPH 21 Booking the selected flight

After clicking the "confirm reservation" in "Book Now" page, they users are directed to a page which displays a short thank-you message. The e-ticket is prepared and sent to the email address of the user immediately after the "Confirm reservation" is clicked in the "Book now" page. Graph 22 shows the short message displayed after confirming the reservation.



GRAPH 22 Thank you message after booking

6.5.5 Preparing for sending e-ticket

In `views.py`, the function for sending e-ticket to the email address submitted by the user is executed after the submission button. The personal data from the user through formsets is saved in the database and the flight details saved in session are used for preparing the e-ticket. The email to the user is sent via python's 'smtplib' module. In `settings.py`, email host, email host user, email password and email port are defined which is needed to send e-ticket to the email address. Graph 23 presents a part of code used in preparing the content of the e-ticket.

```

if request.method == 'POST':
    formset = FinalConfirmation(request.POST)
    passengers = []
    if formset.is_valid():
        for form in formset:
            passenger = {}
            passenger['first_name']=form.cleaned_data["first_name"]
            passenger['last_name']=form.cleaned_data["last_name"]
            passenger['gender']=form.cleaned_data["gender"]
            passenger['birthdate']=form.cleaned_data["birthdate"]
            passenger['nationality']=form.cleaned_data["nationality"]
            passenger['flight_id'] = request.session["flight_id"]
            passenger['return_id'] = request.session["return_id"]
            passengers.append(passenger)
        flight_object = Connection.objects.get(id=request.session['flight_id'])
        flight_object.seats_available = flight_object.seats_available - len(formset)
        flight_object.save()
        return_flight_object = Connection.objects.get(id=request.session['return_id'])
        return_flight_object.seats_available = flight_object.seats_available - len(formset)
        return_flight_object.save()

```

GRAPH 23 Preparing the e-ticket

6.5.6 E-ticket

The e-ticket contains the passenger details fetched from database and the flight connection that they chose. The function for sending e-ticket is automatically triggered after the user presses “Confirm Reservation” button in “Book now” page. The e-ticket is sent as plain text as shown in Graph 24 to the email address of the user.

The screenshot shows an email client interface. On the left is a sidebar with a search bar and a list of folders: 'Inbox 78', 'Archive', 'Spam 64', 'Drafts 27', 'Sent', 'Deleted', and 'New folder'. The main area displays an email titled 'Your travel ticket'. The email content is as follows:

*****Automatically generated by FlyHigh*****

Travel itinerary for departure flight:

Full name: Pooja Gautam
Gender: Female
Nationality: Nepali
Route: Odense -> Stockholm
Traveling on date: 2015-08-17
Flight departure at: 18:30:00
Flight arrival at: 19:50:00

Travel itinerary for return flight:

Full name: Pooja Gautam
Gender: Female
Nationality: Nepali
Route: Stockholm -> Odense
Traveling on date: 2015-08-24
Flight departure at: 08:30:00
Flight arrival at: 10:00:00

GRAPH 24 E-ticket received in customer email

7 ADVANTAGES AND CHALLENGES

There are little doubts about the convenience of online ticket booking. Flights are very important in today's modern world: whether it be for personal, business or other purposes. Clients want to have the best possible deal when booking their flight tickets. While the ticket booking system has advanced in a lot of ways and has simplified air travel, there do remain some challenges and limitations as well. However, the continuous advancement in technology has positively impacted the online flight booking system as well and we can be sure that this system is here to stay for some time.

7.1 Advantages

The reason why more and more people are purchasing their flight tickets online is because of the many advantages and conveniences online flight booking system has to offer. Opposed to the traditional way of purchasing tickets through travel agents office, online booking today provides the access to hundreds of flight paths, their prices and other services at the click of a button. This can be very important for the customer in order to find the best possible deal and the most convenient connection. Online system of booking flight tickets gives more control to the customers in terms of their holiday planning and helps them make informed decisions. The system of searching the flights is so easy, that anybody with a computer and internet access can easily search connections, compare prices and purchase any connection they wish to. Not only this, but this method is much more faster, easier, hassle-free and convenient. Many flight companies these days also provide other services such as airport transfer, accommodation services, car hire and airport parking. The customers can also take advantages of frequent discounts, promotions and other offers from the airlines through subscribing to their email or simply by viewing their website. The clients can also fully be aware of the various policies of the company such as cancellation, baggage rules and others. and can easily make use of those services. Online flight booking not only

provides easy and fast services to the customer from the convenience of computer from anywhere in the world, but often also provides services at more reasonable prices. (GazetteLive 2011.) Majority of airlines companies also provide the services of online check-in after which the customer can easily print the boarding pass at home and save considerable time and hassles at the airport.

7.2 Disadvantages

Online flight booking system has its own set of limitations as well. Prospective clients in Asia and other parts of the world where the connection speed and availability is poor may find it hard to access the website of flight companies and book tickets. In much of the less developed part of the world, besides internet connectivity, the limited knowledge and access to technology can also hinder the chances of using this service. As a result, many people in the developing world still use the traditional method of purchasing tickets from the offices of travel agents. Machine failures such as lost connection or unresponsive program can sometimes also cause the disappearance of all the flight itinerary. Furthermore, the customer might not be able to get correct or enough information from the website alone as it is not done face to face with a person who knows everything about the connections, offers and company policies. Then there is also the issue of security: it is sometimes possible that somebody can steal your credit card information but developers are improving the security of the system more and more with the advancement of technology.

7.3 Challenges faced during application development

The application was created in a short time of one month. A lot of compromises had to be made in the features of the application because of this time constraint. Good amount of time was consumed in installing and setting up all the programs

and software used. A considerable amount of time was also used in learning everything from the scratch. As there were countless possibilities of enhancing the application and adding new features, limitations had to be made. The main motive was developing a working web application through its main functionality of searching and booking flights. Besides the main web application embedded in a HTML page called "Flight search" and the interactive contact page, the rest of the web pages were static, just developed to demonstrate the general view of real airlines websites. Advanced ideas such as creating e-ticket in PDF format, using credit card authentication for payment etc. were thought of but could not be implemented due to lack of enough knowledge in the area and the time constraint. Further upgrades remain a possibility for the enhancement of the application. The basic working application for flight booking was successfully developed but I left room for upgrades and improvements because of various mechanical and physical hindrances involved. (Crosby 2007.)

8 CONCLUSIONS

The world has advanced into the era of internet technology. Even mundane chores such as paying bills, making purchases and booking holidays are increasingly being done on the internet. The airline industry made use of the internet boom by developing airlines reservation system (ARS), which has now evolved into Computer Reservation System (CRS). Using the CRS technology now, it is easily possible to book flights online through the company websites or through third-party websites. It is predicted that the CRS, with features such as online reservation and check-in, will be used by the vast majority of aircraft companies in the world in the coming years. The competition among the airlines companies, the drop in fares over the years, and the availability and accessibility of the flight deals have led to the soar in number of people flying. Nowadays, the websites of airlines companies provide all kinds of facilities such as package tours, accommodation services, airport transfer, car rental as well as 3D illustrations through websites and pictures of the destinations. These features and the convenience of booking from your home makes online flight booking very attractive to the customers.

There is no doubt that the online flight booking system is here to stay. Although only the areas in the world with high speed connection and internet accessibility are the largest customers of this system currently, the internet reach throughout the world is rapidly growing. It can be expected of not only the airlines industries but also other travel and tourism industries to add even more features and enhance the online system of booking and other services. The internet security and the limited customers from only the developed side of the world are currently the biggest general challenges. But since various research and results show that the business of airlines has increased by folds upon the introduction of online flight booking system, it is almost certain that there will be further development and enhancement in the already user-friendly online system of most airlines today with the rapid growth of technology.

REFERENCES

- Atkinson, B. 2015. How does online check in work? Available:
<http://www.travelsupermarket.com/blog/how-does-online-check-in-work/>.
Accessed: 17th September, 2015.
- Borodescu, C. 2013. Web Sites vs Web Apps: What The Experts Think. Available:
<http://www.visionmobile.com/blog/2013/07/web-sites-vs-web-apps-what-the-experts-think/>. Accessed: 17th September, 2015.
- Chaffee, A. 2012. What is a web application? Available:
<http://www.jguru.com/faq/view.jsp?EID=129328/>. Accessed 30th July, 2015.
- Crosby, T. 2007. How Airline E-Tickets Work. Available:
<http://adventure.howstuffworks.com/destinations/travel-guide/tips/how-airline-e-tickets-work.htm/>. Accessed 5th August, 2015.
- Django Software Foundation. 2013. Django Documentation, FAQ: General. Available:
<https://docs.djangoproject.com/en/1.8/faq/general/>. Accessed August 10th, 2015.
- Encyclopedia Britannica. 2015. HTML. Available:
<http://www.britannica.com/technology/HTML/>. Accessed 18th September, 2015.
- Encyclopedia Britannica Online. 2015. Website. Available:
<http://www.britannica.com/topic/Web-site/>. Accessed 15th September, 2015.
- GazetteLive. 2011. The benefits of booking flights online. Available:
<http://www.gazettelive.co.uk/news/local-news/benefits-booking-flights-online-3693877/>. Accessed 5th August, 2015.
- Holiday, L. 2014. Web design vs web development: what's the difference? Available:
<http://www.skilledup.com/articles/web-design-vs-web-development-whats-difference/>. Accessed 10th August, 2015.
- Johanson, M. 2014. How The Airline Industry Has Evolved In 100 Years of Commercial Air Travel, Available:
<http://www.ibtimes.com/how-airline-industry-has-evolved-100-years-commercial-air-travel-1524238/>. Accessed 10th July, 2015.

- Kaplan-Moss, J., Holovaty, A. 2007. The Definitive Guide to Django. New York. Apress.
- Laird, C., Soraiz, K. 1998. Getting Started with Python. SunWorld Online.
- Lie, H., Bos, B. 1999. Cascading Style Sheets, level 1. Available: <http://www.w3.org/TR/1999/REC-CSS1-19990111/>. Accessed: 17th September, 2015.
- McGuigan, B. 2015. What is web design? Available: www.wisegeek.org/what-is-web-design.htm/. Accessed 2nd August, 2015.
- Mertz, D. 2001. Functional Programming in Python. Available: http://gnosis.cx/publish/programming/charming_python_13.html/. Accessed: 1st August, 2015.
- Mozilla Developer Network. 2015. Available : https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/JavaScript_basics/. Accessed 2nd August, 2015.
- Ross, G. 2013. Airline Reservation Systems History 101. Available: <https://www.cbtravel.com/blog/2013/11/airline-reservation-systems-history-101/>. Accessed: 17th September, 2015.
- Rouse, M. 2015. e-commerce (electronic commerce or EC) definition, Available: <http://searchcio.techtarget.com/definition/e-commerce/>. Accessed 10th August, 2015.
- Rouse, M. 2005. Website definition. Available: <http://searchsoa.techtarget.com/definition/Web-site/>. Accessed 16th August, 2015.
- Shannon, R. 2012. What is HTML? Available: <http://www.yourhtmlsource.com/starthere/whatishtml.html/>. Accessed: 5th August, 2015.
- Skutin, A. 2014. What Every Frontend Developer Should Know About Webpage Rendering. Available: <http://frontendlabs.com/articles/webpage-rendering-101/>. Accessed: 18th September, 2015.

Techopedia. 2015. Client/Server Architecture. Available:
<http://www.techopedia.com/definition/438/clientserver-architecture/>. Accessed 2nd
August, 2015.

Ueland, S. 2013. 11 Alternative Payment Options. Available:
<http://www.practicalecommerce.com/articles/61562-11-Alternative-Payment-Options/>. Accessed 11th August, 2015.

UKOLN. 2006. Use of Cascading Style Sheets(CSS). Available:
<http://www.ukoln.ac.uk/qa-focus/documents/briefings/briefing-34/html/>. Accessed:
18th September, 2015.

Winston, C., Morrison, S. 1995. The Evolution of the Airline Industry. Brookings
Institution Press.