ZEPPELIN - CTRS

Cognizant Technology Solutions, Bangalore.

PROJECT REPORT

Submitted in partial fulfillment of the
requirements for the award of the degree of
M.Sc - SOFTWARE ENGINEERING
of Bharathiar University, Coimbatore..

Submitted by
S.NARMATHA VANI
9837S0054

K C T
LIBRARY
COIMBATORE 641 006

Under the Guidance of

Internal Guide
Mrs. D. Chandrakala (Senior Lecturer),
CSE Department,
Kumaraguru College Of Technology,
Coimbatore.

External Guide
Dr. M. Sai Krupa Sagar (Asst Manager- Projects),
Microsoft Team,
Cognizant Technology Solutions,
Bangalore.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE- 641 006
APRIL 2003

CERTIFICATE

Department of Computer Science and Engineering

Kumaraguru College of Technology

Coimbatore-641006.

This is to certify that the project work entitled "Zeppelin- CTRS"
Has been submitted by

Ms. S. Narmatha Vani

In partial fulfillment of the award of the degree of
Master of Science in Applied Science- Software Engineering of
Bharathiar University, Coimbatore
during the academic year 2002-2003.

20/3/'03
Guide

28/3/'03
Head of the Department

Certified that the candidate was examined by us in the Project Work Viva Voce Examination held on 5.4.2003 and the University Register Number was 9837S0054.

Internal Examiner

External Examiner

Cognizant
Technology
Solutions

LIBRARY
ATORE - 641 006

DECLARATION

I hereby declare that this project work titled ‘ Zeppelin- CTRS’ is a record of original project work done by me under the guidance of Mrs. B . ChandraKala , as internal guide and Dr. M. Sai Krupa Sagar , as external guide, and this project work has not formed the basis for the award of any Degree/ Diploma / Associate ship/ Fellowship on similar titles to any other candidates of any university.

S. Narmatha Vani

Date 05 · 04 · 2003

S. Narmatha Vani.

D. Chandrakala

M Sai Krupa Sagar

Internal Guide,
Mrs. B. ChandraKala(Senior Lecturer)
Dept CSE,
Kumaraguru College of Technology,
Coimbatore.

External Guide,
Dr. M. Sai Krupa Sagar (APM),
Cognizant Technology Solutions,
Bangalore.

Cognizant Technology Solutions India Ltd.
1/1 to 1/15, Silver Jubilee Block,
3rd Cross, Mission Road,
Bangalore-560 027. India
Phone : (+91-80) 227 3100
Fax : (+91-80) 229 1020

www.cognizant.com

ognizant
chnology
olutions

CERTIFICATE

This to certify that the project report titled **“ZEPPELIN-CTRS’**, which is being submitted by **Ms. S. Narmatha Vani** in partial fulfillment of the requirements for the award of the degree of **Master of Science in Applied Science – Software Engineering** of Bharathiar University, Coimbatore is a bonafide work carried out by her under my guidance, at **Cognizant Technology Solutions**, Bangalore during the period of January 06, 2003 to March 14, 2003 to my satisfaction.

For **Cognizant Technology Solutions India Limited**

Srikanth Srinivasan
Manager-Human Resources

Date : March 14, 2003

Ability is what you're capable of doing…
Motivation determines what you do…
Attitude determines how well you do it.

Dedicated to my Parents
S. Narmatha Vani.

Acknowledgement

Acknowledgements

"By sending our robust thoughts of faith, hope, hard work and optimism, we can surely achieve our goals"

I would like to express my heart felt thanks to our Principal Dr.K.K.Padmanaban B.Sc (Engg), M.Tech, Ph.D and our HOD Prof.. Dr. S.Thangaswamy Ph.D for giving me the needed encouragement in starting this project and carrying out successfully.

I also take an immense pleasure in thanking our class coordinator Mrs. S. Devaki (Asst Professor) and my internal project guide Mrs.D. Chandrakala (Senior Lecturer) who have taken up keen interest in the success of this project.

I would like to begin with a special note of thanks to Mr. Kalyan Mohan , Director, Cognizant Technology Solutions, Bangalore, for providing me an opportunity to carry out this project work at this center.

My grateful thanks goes to Mr. Vincent . T. P (Project Manager) and Dr. M. Sai Krupa Sagar (Asst Projects- Manager) for rendering their excellent guidance for successful timely completion of the project.

I also Extend my thanks to Mr. M . Girish for his valuable suggestions and advices.

I am externally indebted to my parents for their unending support, love and understanding throughout the endeavor.

My acknowledgement would be incomplete without thanking my friends for encouraging me to finish this project in a successful way.

Above all, I owe my gratitude to God Almighty ,for showering abundant blessings on me.

Synopsis

Synopsis

The project entitled ' Zeppelin – CTRS' is being developed for Clients in Cognizant Technology Solutions, Bangalore.

Computers play a vital role in Highwire Booking Engine. At present CTS are developing their projects for Highwire Booking Engine. The main purpose of this booking engine is to book the tickets via online and issuing the E-tickets.

The project given to me is "Zeppelin" developed for Corporate Travel Reservation System in Highwire Booking Engine. This project is divided into three main modules they are Air ticket reservation system, Hotel reservation system and Car reservation system. This document currently focuses on Air Ticket reservation System. This software could be used only by the employees of the particular company who has registered themselves with the highwire booking engine. The main reason for getting into the Zeppelin project is in the existing system the travel agent has to do the ticketing, which in turn consumes more time , expensive and done manually. Therefore in order to overcome these difficulties the proposed system eliminates the travel agent and does both Booking and Ticketing in the same Booking engine.

The functioning of this software is discussed in brief in this synopsis. The software starts with a user interface web application where only registered users can access this system. Therefore the Shopping and Booking is done with the help of User interface. If the user selects the Zeppelin option then Ticketing is done by interacting with GDS mainframe via XML Select. Finally an E-ticket is issued to the users via E-mail.

The technology used for this application is VC++ - ATL COM & XML Select.

INDEX

Topic **Page No**

Topic **Page No**

Topic **Page No**

Introduction

1. INTRODUCTION

1.1 About Zeppelin:

This project 'Zeppelin' for Corporate Travel Reservation System will provide an alternative, automated distribution channel for end-to-end transaction processing that needs no human intervention for completion. Project Zeppelin's goal is to create a new distribution channel that drives cost out of the air ticketing process to reduce the viability of competitive GDS bypass strategies. The traditional GDS model is under threat from alternative distribution channels. Many efforts are currently underway across the industry to bypass the GDS due to perceived cost savings and a desire for increased span of control. In order to meet these competitive challenges, Galileo is undertaking this project.

1.2 Organization Profile:

Cognizant Technology Solutions (CTS) is involved in various Highwire Projects. The Highwie projects are developed for the use in Corporate Travel Reservation Systems in United States. At present CTS is doing their project called Zeppelin . This project has team size of about 20 members.

1.3 Zeppelin in CTS:

Once complete, Project Zeppelin will provide an infrastructure to facilitate a "touch less" process for the procurement, pricing and fulfillment of air travel reservations. Rather than an end-to-end booking tool, it is envisaged that the transaction processing functionality will be front-end agnostic to the extent that various required subsets of the product will be exposed via an API, thus making it transportable between online and off-line booking channels.

Further, most of the functionality required to achieve the lower cost transaction model described above already exists in the GDS today. Additional development and enhancement to existing and downstream systems will be needed to ensure that bank settlement handoff requirements dictated by ARC and the suppliers can be achieved along with suppression of full service document issuance. Galileo will also need to create a method to track and distinguish between full service and low cost bookings to suppliers and subscribers along with a pricing model that creates incentives to participate in this distribution option.

1.4 About the project:

Project Zeppelin will investigate the development of an automated distribution channel for end-to-end transaction processing. By developing a lower cost distribution channel for

simple transactions, Galileo will be able to drive cost savings to suppliers, travel agencies, and corporations for any transaction that can be automated through this process.

1.4.1 Existing system and its limitations:

During the life of a PNR there are many potential events that constitute the need for manual intervention by the supplier, subscriber or agent. This PNR servicing necessitates a financial assistance element that contributes significantly to the overall cost of the booking.

Schedule Changes, Seat and Special Service Request confirmations from the vendors are internally applied to the PNR which is subsequently placed on queue for manual processing. The Exchange and Refund process includes Fill-in Format screens. Therefore it consumes more time, more money and needs man power to reserve the tickets.

1.4.2 Proposed system and its advantages:

The primary consideration for the proposed system is that to eliminate the travel agent. Where possible, the most simple change option has been selected, rewriting only where there is no alternative. There will be no changes to Web Services, XML Select, Availability, Pricing and Sell. Profile Management will be provided by the Booking Tool.

The Booking Tool will request the streamlined Zeppelin End Transaction (including instant Etkt) using the current ET data structure but with a new character 'Z' in the type field. Providing the AAT security, standard ET police and new Zeppelin content validation is passed, the PNR is flagged as Zeppelin. The process will then wait internally for the Vendor Locator and subsequently call Ticketing. Ticketing will provide the appropriate handoff and also ensure that manual reject screens are bypassed and errors returned.

The offline elements of Billing and Reporting will be modified to accommodate the new Zeppelin Pricing and Billing model. The online Subscriber and Ticketing handoffs will be modified accordingly.

System Requirements

2. SYSTEM REQUIREMENTS

2.1 Product Definitions

2.1.1 Problem Statement:

The product is concerned with Booking of Air tickets for the employees of the particular company who has registration with Highwire Booking Engine. The product should have the capability to Book as well as to Ticket for multiple users who log into the web application built in High wire booking Engine. Thus the product is named as **' Zeppelin - Corporate Travel Reservation System'** .

2.1.2 Functions to be Provided:

The function of this product is to eliminate Travel agency and do Booking and Ticketing with the help of Highwire Engine. The product is mainly based on 3 tier architecture. The inputs are retrieved with the help of User Interface using ASP pages and then Processed. To achieve various functions like Availability check, Booking, issuing E-tickets are involved in Zeppelin project.

2.1.3 Processing Environment:

Hardware Specifications:

Processor - Intel Pentium IV

Hard disk - 40 GB

Ram - I GB

Operating System - Windows 2000

Compiler - VC++ 6.0

Web Browser - IE 5.5

Technologies Used:

VC++ 6.0/ ATL COM / XML

Microsoft IIS 5.0.

Microsoft Exchange Server.

Front End: ASP.

Back End: SQL Server 2000

2.1.4 User Characteristics:

This product can be used only by the employees who has the registered with Highwire booking engine. The software is designed to be user friendly web application interface product. The user can interact with the system by using several web pages by using its links. The result is also given either by web page or by issuing E-tickets.

2.1.5 Solution Strategy:

The problem was approached n a step by step fashion. First and foremost the functions of the existing Web application and their purpose were learnt. Then analysis is done on the problem definition given by the Project Manager. Then the rough draft is made on the analysis and finally ends up in a solution by breaking down the defined problem into five Modules.

The solution is to have a Zeppelin technique to be employed. With the help of Zeppelin technique Man power could be reduced, Consumes less time and Reduces Costs.

2.1.6 Product Features:

The product is developed mainly for Highwire Booking Engine. The product has got its own features like accessing inputs from the user interface and then processing the inputs.

The product has got its own security. The console user can use the product only after registering their details to the web application. The security cannot be broken unless the user knows the password.

External representation of Zeppelin:

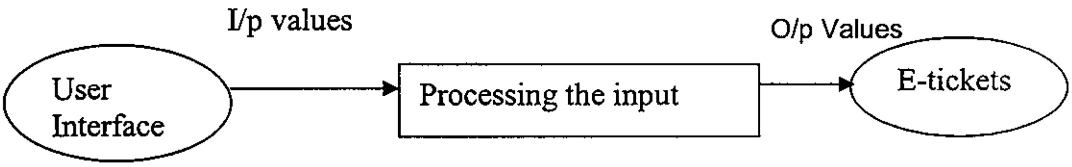


Fig . 1

2.1.7 Acceptance Criteria:

The product has to accept some input values, depending on the inputs supplied processing is done. Finally the product should issue E-ticket or display unavailability form.

2.1.8 Source of information:

- Books:

  ATL 3 COM Programming, Richard Grimes, Shroff Publishers & Distributors Pvt Ltd, 2000.

- Discussion with other team members working within Zeppelin team.

- Product definition given by the Project Leader.
- Web Sites:

  www.travelocity.com

  www.expedia.com

2.1.9 Glossary of Terms:

ATL - Active Template Library.

STL - Standard Template Library.

TMC – Travelers Management Companies.

2.2 Project Plan:

2.2.1 Life Cycle Mode

The Spiral Model is proposed to be life cycle mode followed while developing the product software. It provides the potential for the rapid development of incremental version of the software. The software is developed in the series of incremental releases. The spiral model has six task region.

Task Region 1:

- Terminology: Customer Communication.
- Milestones: Jan 6th – Jan 7th.
- Work Product: The Project Manager defines about the Corporate Travel Reservation System. Referring some books and Websites are done. Some rough documentations are also done here.

Task Region 2:

- Terminology: Planning
- Milestones: Jan 8th – Jan 15th .
- Work Product: Analysis of the product definition. What function the product has to perform, Processing environment, Product features, Programming language and development tools to be implemented are all decided in this stage.

Task Region 3:

- Terminology: Risk Analysis.
- Milestones: Jan 15th – Jan 17th.
- Work Product:

  Technical Risk:

Since Design Document plays a very important role in coding. Preparation of design documentation consumes more time. Therefore coding is based on how we design , what are all the logics and concepts used here.

Managerial Risk:

For every module time limits are set to be fixed for its completion. The project duration is between Jan 6th – Mar 15th. Within the given time slot for each module it has to be completed.

Task Region 4:

- Terminology: Engineering.
- Milestones: SRS Document .- Jan 20th – Jan 22nd .
- Work Product: Based on the needs of the Customer, the software Specification is prepared. SRS includes Product overview, Processing Environment, External interface and Dataflow, Functional Specifications, Performance requirements, Exception condition & handling. Early subsets , Foreseeable modification, Acceptance criteria, Design guidelines.

- Milestones: Design Document – Jan 22nd – Feb 4th.
- Work Product: Based on the needs of the Customer, the design document is prepared. Designing plays an important role during coding. Design document includes external design specification, Architectural design overview and detailed design specification.

Task Region 5:

- Terminology: Construction and release
- Milestones: Feb 5th – Mar 12th.
- Work Product: Once the inputs are retrieved from the User interface , it Is taken for its processing and finally an E-ticket is issued.

Mar 14th	Installing the product.
Mar 15th	Releasing the product to the customer.

Task Region 6:

- Terminology: Customer Evaluation.
- Milestones: Mar 16th
- Work Product: On seeing the performance of the web application , the

Highwire people gave the feedback. The feedback was
The web application works very well and it can be extended in
future for Car and Hotel reservation system.

2.2.2 Team Structure

The web application is concerned with corporate travel reservation system for Highwire booking engine. There are totally 20 members in a team.

Cognizant Resource	Roles
Cognizant Client Partner (Client Specific)	1. Client relationship management 2. Overall project delivery
Account Manager Onsite (Client Specific)	1. Coordinate interaction between Highwire and Cognizant 2. Ensures pending issues, if any are resolved at the earliest 3. Periodically reviews project schedules and deliverables 4. Provide a point of escalation for Highwire regarding any issues with the project/team
Project Manager Offshore/Offshore Project Lead(s) (Project Specific)	1. Ownership for project delivery, quality and timely execution of entire engagement 2. Participates in providing technical leadership and direction to sub-teams 3. Takes responsibility for the application architecture and design 4. Prioritize and assign tasks vs. resources 5. Estimation, planning, monitoring and reporting progress 6. Manage Cognizant resources offshore 7. Ensure quality procedures are complied in the project as per standards 8. Approval authority for deliverables, documents and procedures 9. Overall coordination of project related issues
Onsite Coordinator (Project Specific)	1. Act as single point of contact for Highwire 2. Report status to Highwire and Offshore Project Manager 3. Interact with Highwire Technical Staff and Product Managers to obtain requirements, analysis and supporting documents to be communicated for offshore development 4. Assure the quality of all deliverables to Highwire 5. Co-ordinate with offshore delivery manager/team 6. Manage Cognizant onsite team 7. Obtain acceptance from Highwire for all deliverables 8. Review with Highwire all work-in–progress, participate in status meetings
Onsite Team	1. Clarify all the doubts of the offshore team

(Project Specific)	2. Ensure that all the infrastructure code, standards etc., from Highwire are available for the offshore team 3. Help Highwire in setting up, testing and deployment of the solution. 4. Undertake high level and detail design tasks 5. Coding and testing 6. Unit Test Plan and Unit Test Case
Development Team (Project Specific)	1. Undertake detailed analysis and documentation of existing system functionality 2. Undertake high level and detail design tasks 3. Coding and testing 4. Unit Test Plan and Unit Test Case 5. Conduct peer reviews 6. Documentation 7. Technical support during acceptance testing and implementation

Cognizant Client Partner(Client Specific) : 1
Account Manager Onsite(Client Specific) : 1
Project Manager offshore/ Offshore Project leader (Project Specific) : 3
Onsite Coordinator (Project Specific) :1

Onsite Team (Project Specific):5
Development Team (Project Specific) :9

2.2.3 Development Schedule

In order to complete the project in time , the development schedule is framed and based on the time slots, the product is developed. The development schedule consists of Milestones and Reviews.

Milestones	**Reviews**
Jan 7th - Product Definition	A rough draft is made to product definition and the definitions are reviewed.
Jan 10th - Product Analysis	A rough draft is made to product analysis and the review is made on the analysis to do step by step fashion.

Jan 15th – Programming Language	The programming language decided. The problem of selecting development tool was also decided to have ATL COM.
Jan 17th – Risk Analysis	A rough draft is made to Risk Analysis. The draft was reviewed and there was two types of risks involved. They are Technical Risk and Managerial Risk
Jan 22nd –SRS	SRS general formats was reviewed.
Feb 4th – Design Documentation	In design document the type of techniques to be used was Reviewed.
Feb 7th – Reviewing Code	The code that already existed in Front end and back end were reviewed and document is prepared for them.
Feb 11th – Reviewing Code	The code that already existed in Middle Tier were reviewed and document is prepared for them.
Mar 5th – Implementation	The reviews were made on the Implementations.
Mar 7th – Test Plan	Reviews were made on the Test Plan.
Mar 11th – Testing	Based on the test plan testing is done to the application.

Mar 12th – User's Manual	The reviews were made on User's Manual.
Mar 14th – Installing	No Reviews
Mar 15th – Releasing	No Reviews.
Mar 16th – Customer Evaluation	No Reviews.

2.2.4 Programming Language:

Hardware Specifications:

Processor - Intel Pentium IV

Hard disk - 40 GB

Ram - I GB

Operating System - Windows 2000

Web Browser - IE 5.5

Technologies Used:

- VC++ 6.0/ ATL COM / STL

Microsoft IIS 5.0.

Microsoft Exchange Server.

Front End: ASP.

Back End: SQL Server 2000

2.2.5 Documents to be prepared:

It is suggested that the following documents can be prepared during the time of the project

- A System Definition consisting of a Product Definition and a Project Plan.
- A Software requirements Specification.
- A detailed document consisting of external design, architectural design and detailed design specification.

- A test plan.
- A User's manual.
- A properly documented, debugged and tested program.
- A project legacy document.

2.2.6 Manner of Demonstration:

Reviews:

Every week end the finished modules are explained to the Project Manager, reviews are then made on it.

Documents:

Drafts of every document is received by the Project Manager before it is finalized. If there is any changes to the draft they are incorporated in the module.

Product:

Demo of each module is given to the Project Manager as and when the module is completed. If any changes are required, they are incorporated in the module after the review.

2.2.7 Source of Information

- Books:

  ATL 3 COM Programming, Richard Grimes, Shroff Publishers & Distributors Pvt Ltd, 2000.
- Discussion with other team members working within Zeppelin team.
- Product definition given by the Project Leader.
- WebSites:

  www.travelocity.com

  www.expedia.com

2.2.8 Glossary of Terms

COM: Component Object Model.

ATL: Active Template Library.

GDS: Global Distribution System

PNR: Passengers Name Record..

TMC: Travelers Management Companies

CRS: A Computer Reservation System

Software Requirement Specifications

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 Introduction:

This document defines the high-level requirements for Project Zeppelin in Corporate Travel Reservation System, which is based on 3 tier architecture.

3.1.1 Purpose

The purpose and intent of this document is:

- To provide an overview of the project.
- To define the "relative" scope
- To provide high-level requirements which will drive a technical approach

3.1.2 Scope

The project is subdivided in to three main modules they are

- Air/Flight Reservation
- Hotel Reservation
- Car Reservation

The current document focuses on Air Reservation System .

Project Zeppelin encompasses functionality, both existing and yet to be developed/acquired, to facilitate a "touch-less" process to create and price an air travel reservation using the Apollo and Galileo reservations systems and to provide fulfillment services of such reservations by means of an electronic ticket (or more than one electronic ticket).

The scope of effort for Project Zeppelin will increase with phased rollouts of the functionality. For the purposes of defining the initial phase of the effort, the following conditions define the universe of bookings eligible for Zeppelin inclusion:

- Online travel reservations using the Highwire self-booking tool on the Apollo reservations system in the United States only .
- Electronic tickets only
- US point of sale and bank settlement only (ARC – Airlines Reporting Corporation)

- Reservations containing book-able air content of currently available and supported Apollo participation supplier airlines only
- Use of currently available shopping and pricing tools within the Apollo system only
- Not all transactions are targeted to be Zeppelin transactions.

3.1.3 Definition, Abbreviation, Acronyms:

COM: Component Object Model.

Com helps to assume application from pre built parts or components irrespective of location Or the development language used to implement the components.

ATL: Active Template Library.

ATL is an excellent alternative to MFC for writing COM components with a small memory and code footprints

GDS: Global Distribution System - a computer reservation and information system that is often operated by multiple airlines, used by travel agents & other travel professionals and contains information on all types of travel products

PNR: Passengers Name Record..

TMC: Travelers Management Companies

CRS: A Computer Reservation System for an ***individual*** airline containing information on that airline only and used by that airlines' employees.

Booking: Booking is just blocking the tickets.

Ticketing: Ticketing is a process of issuing E-tickets after its confirmation.

Clients: The Employees of the particular companies.

3.1.4 References:

Books:

- Rodger .S. Pressman, Software Engineering, MC-Grawhill publications,1997.
- ATL 3 COM Programming, Richard Grimes,Shroff Publishers & Distributors Pvt Ltd, 2000

Websites:

- MSDN.Microsoft.com – 10-Jan-2003
- www.travelocity.com – 16-Jan-2003
- www.expedia.com – 16-Jan-2003

3.2 General Description

3.2.1 Product Perspective

The impetus for Project Zeppelin arises from the need to leverage the resources and technology created and refined over several years within the Galileo automation infrastructure into a cost model that is sustainable in an era where the value of the traditional GDS model is coming under increased threat of dis-intermediation by alternate, lower cost distribution channels. The full service fulfillment paradigm of the GDS is geared primarily toward a highly managed, transaction intensive booking process in which several factors, human and technical, have input and impact on the reservation creation and fulfillment process. In this model, it is commonplace for bookings and their resulting tickets to be manipulated and changed several times over their life cycles, with each transaction representing incremental cost for the supplier in segment fees and for the end user in TMC-imposed management fees.

3.2.2 Existing System:

Disadvantage:

- Consumes more time.
- Consumes more money.
- Needs man power.

The existing system consists of travel agency for ticketing.

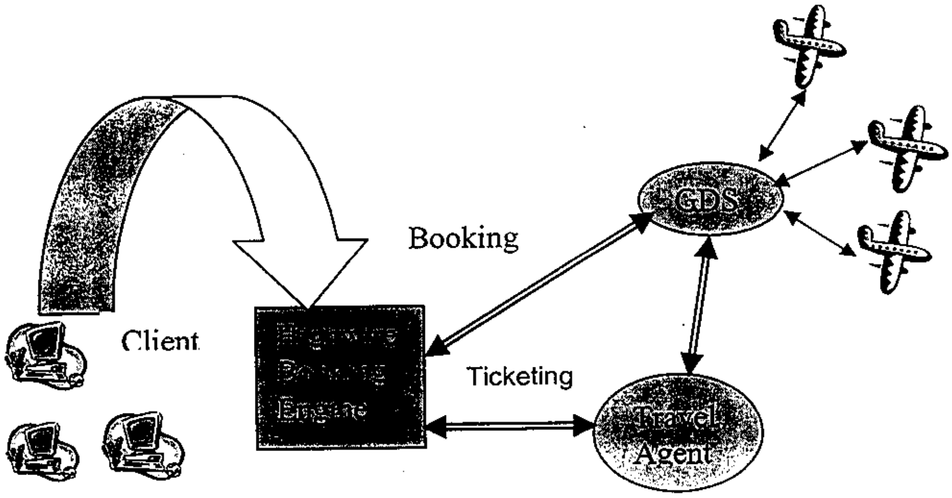


Fig . 2

3.2.3 Proposed System:

In proposed system the travel agency is not present. Ticketing and Booking is done automatically by the Highwire booking engine.

Advantages:

- Consumes less time.
- Reduces cost.
- Fully Automated

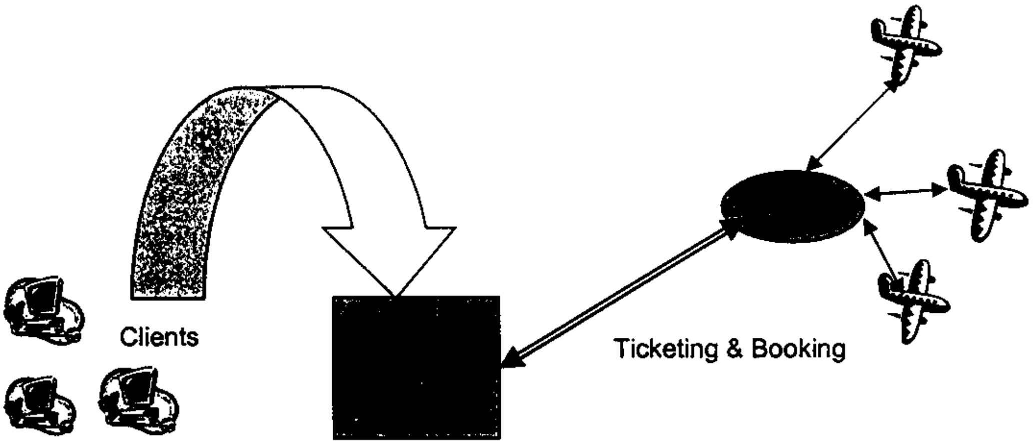


Fig . 3

The business hypothesis of Project Zeppelin is that there is a considerable share of these transactions, both booking and ticketing, currently fulfilled using only a small subset of the otherwise feature-rich functionality of the GDS. This creates an opportunity to segregate these transactions that do not require intricate servicing or manipulation into a separate and distinct market channel set aside specifically for the purposes of re-invention of the existing cost model. By limiting the scope and transaction type subsets of the low-cost model, the GDS has the opportunity to price such transactions for its suppliers at a lower-than full service price, thus making it attractive for suppliers to increase their share through this channel and away from the higher-cost, full service model. Scale economies would ensue with sufficient market penetration into distribution channels where Galileo today enjoys no cost-competitive advantage and overall margins would increase due to increase booking volumes movement of market share toward the lower cost model.

3.2.4 Product Functions

Once complete, Project Zeppelin will provide an infrastructure to facilitate a "touch-less" process for the procurement, pricing and fulfillment of air travel reservations. Rather than an end-to-end booking tool, it is envisaged that the transaction processing functionality will be front-end agnostic to the extent that various required subsets of the product will be exposed via an API, thus making it transportable between online and offline booking channels.

Further, it is envisaged that the majority of the functionality required to achieve the lower cost transaction model described above already exists in the GDS today. Development efforts to lock-down the scope of the low cost booking transaction subset should be confined mostly to business logic enhancements to core systems to enable a recognition and distillation of factors and attributes of a given booking to qualify it for inclusion in the low cost process model.

Additional development and enhancement to existing and downstream systems will be needed to ensure that bank settlement handoff requirements dictated by ARC and the suppliers can be achieved along with suppression of full service document issuance. Galileo will also need to create a method to track and distinguish between full service and low cost bookings to suppliers and subscribers along with a pricing model that creates incentives to participate in this distribution option.

Its Actually a 3 tier Architecture .

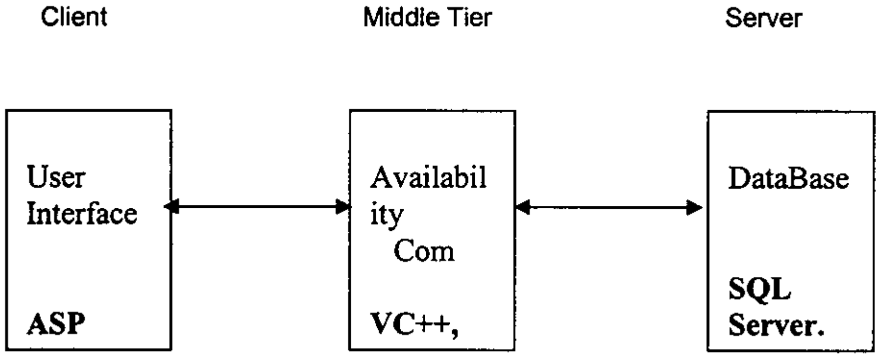


Fig . 4

Implementations to be done:

The presentation tier will include changes to the Profile and the Booking Engine Screens. In addition to that, few other screens for Zeppelin Ticketing, Status Messaging etc. have to be included in the presentation tier.

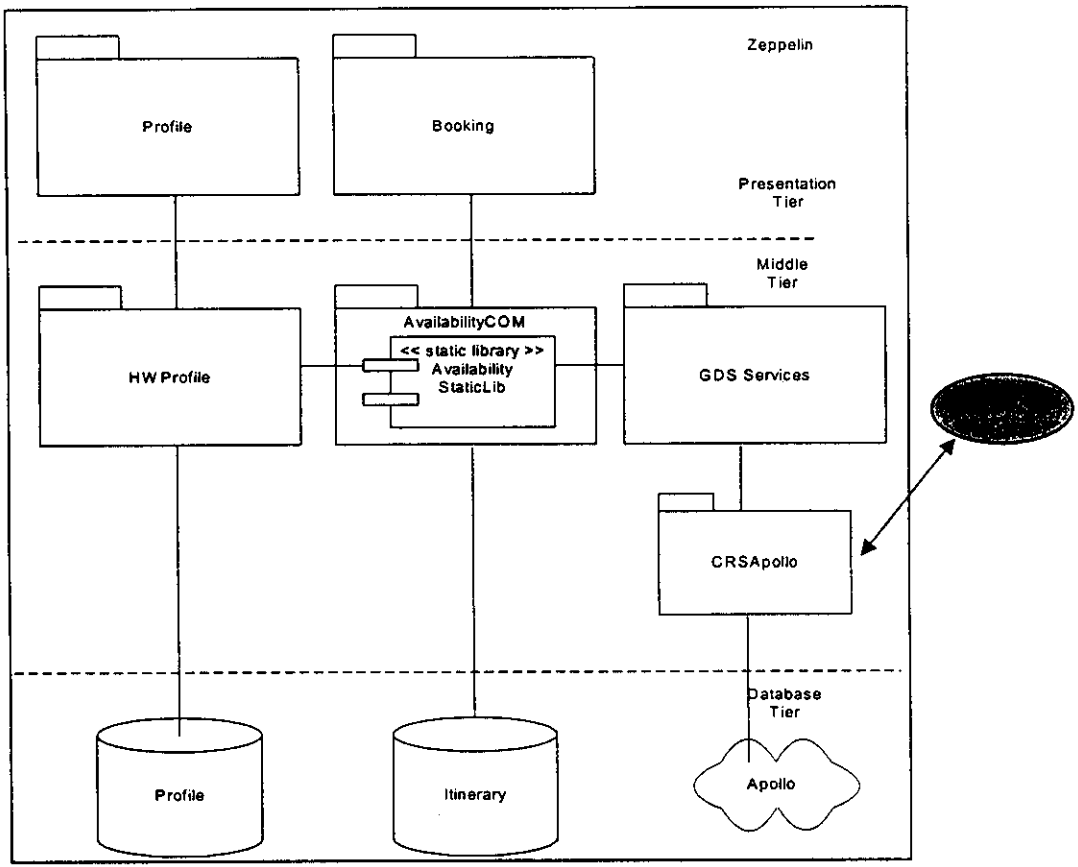


Fig . 5

Correspondingly some of the middle tier components will also get affected. Highwire Profile and related classes will be modified to provide capability of policy handling relevant to Zeppelin.

The booking component will use this policy for deciding to apply zeppelin feature. There are some more validations regarding the E-ticket, One carrier etc. that need to be done before the decision to apply Zeppelin ticketing.

GDS Services or lower level CRS Apollo component will provide low-level PNR calls to the GDS regarding the automated fulfillment (PNR End Transact/ Ticketing). If this call fails even after a predefined number of attempts (configurable), then it has to send the PNRBF End call with the normal queue place functionality. If successful, then the ticketing details page need to be shown to the user. Otherwise the normal booking confirmation page will be shown. This will involve some changes at the middle tier component level to accept the additional information from GDS and some presentation tier changes to show it to the user.

Existing Utility classes will be used for e-mail notifications, Error handling and XML logging process.

Data Tier changes will include changes to Booking Engine and Profile databases.

The architectural view of this project is shown below:

Changing the data model for pseudo city codes will require regression testing of all UI and middle
tier components that use the fulfillment service.

Currently my module is CRS Apollo.

Technology Used:

Modules	Technology
Availability COM	VB, VC++- COM.
Booking:	ASP,VC++ 6.0.
HW Profile:	VC++ 6.0, ASP.
CRS Apollo :	VC++, XML.

DataBase used:

Modules	DataBase
Profile	SQL Server 2000.
Itineraries	SQL Server 2000.

3.2.5 User Characteristics

The primary target distribution channel for low-cost Zeppelin processing is the online travel reservations market. This encompasses both the traditional high-volume corporate travel sector (Highwire, Get There) as well as the one-off, cost-conscious consumer sector (Trip.com, Expedia, Travelocity). Rather than relying on a characterizing the user, the product caters to a transaction model in which many online consumers can be classified. This transaction model, at a high level, contains the following attributes:

- Simple one way, roundtrip or open jaw itineraries
- Single surname
- Credit card form of payment
- Use of negotiated or net fares and the need for their online tool to locate and correctly apply these fares
- Electronic ticketing – a paperless transaction
- Corporate Travel - TMCs provide their value-add to corporate accounts (and collect their travel management fees) through management reports and financial tracking services. Transactions of all types, including those fulfilled online though the low cost channel, will need to feed financial and MI data to downstream systems in the same way that full service transactions do today.
- Leisure Travel – The online leisure sector (direct to consumer) requires the ability to shop both published and negotiated rates and fulfill their transactions in a fluid process. While the need for down line reporting is less complex in the sector, market share tracking and other statistics are normally tracked and reported upon to leverage volume against net fare deals.

Simply stated, the transaction subset that this product accommodates is straightforward, lightly managed and highly automated.

3.3 Specific Requirements:

3.3.1 Functional Requirements:

3.3.1.1 Introduction:

The Highwire engine gets input from the clients, process the information and checks the availability of flights and then books the tickets.

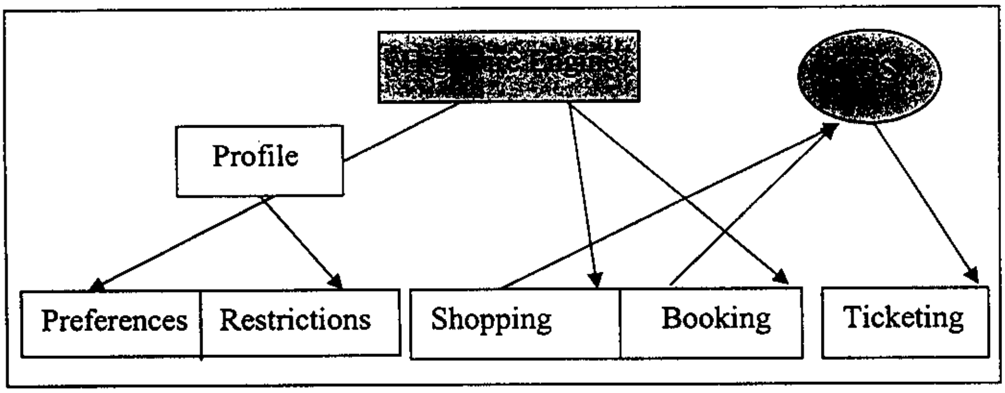


Fig . 6

The Airline reservation is divided into four main subdivisions they are

- Preferences
- Shopping
- Booking.
- Ticketing.

Functionality – shopping Process:

Graphical User Interface (GUI) / Front-end Booking Tool

This project is not dependent (long-term) on any specific GUI / Front-end Tool, it is front-end agnostic. However, short-term deliverables will depend on the Highwire Corporate Booking Tool (HW) as the main Front-end GUI Booking Tool.

Shopping Tools

Initial deliverables for this project must be delivered using existing Galileo shopping functionality. This functionality includes access to Super Best Buy transaction, also known as Shopper Fare Quote (or SuperBB) in XML Select. This requirement has been defined early in the requirements cycle and will be relieved when Galileo and/or Highwire provide functionality using the Shopping DB from ITA.

Functionality – End Transact

PNR/BF End

Functionality must be provided to Front-end Booking Tool to either initiate the original (higher cost model) End Transact process model or the Zeppelin (lower cost model) End Transact process model.

Automatic/Instant E-Ticketing

If PNR qualifies for the Zeppelin process, the End Transact process should validate the PNR based on Zeppelin process criteria. After PNR is qualified, the End Transact process should automatically/instantly call or initiate the E-Ticketing process.

E-Ticketing Failure

Should e-ticket issuance fail, or is rejected by the airline, PNR defaults to original (higher cost model) processing.

Success Reporting

The Zeppelin End Transact process should return a Galileo Record Locator and an E-Ticket number back to the calling application when successful. Otherwise Error Reporting should occur by the End Transact process.

Error Reporting

End Transact process should return existing End Transact errors as well as any errors reported by E-Ticketing process.

Functionality – PNR

"Zeppelin" PNR Attributes

The "Zeppelin" PNR (also known as a "ZPNR") contains the following attributes. References to these attributes and how these attributes are set, modified, etc... may also exist in other requirement definitions herein, otherwise they are contained with in the standard host system processing documentation.

Attributes are:

- Single Adult Passenger – The ZPNR is limited to 1 adult passenger.
- Origin/Destination (O/D) Pairs – ZPNR are limited to 4 O/D Pairs and eight (8) total segments.

- Single Carrier – ZPNR is limited to 1 carrier
- E-Ticket Eligible – ZPNR is limited only to itineraries that can be E-Ticketed. International segments originating in the US on Apollo (1V) that are e-ticket eligible are also acceptable.
- Email Address – ZPNR must include passengers Email address.
- US Point of Sale only – Point of Sale of a ZPNR must originate in United States on Apollo core (1V).
- Zeppelin Indicator – ZPNR will contain an indicator that denotes PNR as a Zeppelin PNR.
    - Show header line on face of ZPNR to indicate that this is a Zeppelin PNR for troubleshooting and servicing by GI staff.
    - Indicator should not be visible to United Airlines staff.
- If a ZPNR is retrieved and updated from 1G core via Global Access, once it is synchronized back to Apollo it should be billed and notified accordingly.
- ZPNR is only declared on a PNR if the above attributes are validated thru the End Transact process and the PNR passes thru E-Ticketing process and **is** E-Ticketed.

3.3.1.2 List of Inputs:

The inputs are accepted with the help of dialog boxes. The inputs are as follows

For Profiles:

Employee Name.

Company Name

Employee Mail ID

City

For Itineraries :

Origin

Destination

Date

Time

3.3.1.3 Information Processing Requirements

Hardware Specifications:

Processor - Intel Pentium IV

Hard disk - 40 GB

Ram - I GB

Operating System - Windows 2000

Web Browser - IE 5.5

Technologies Used:

VC++ 6.0/ ATL COM / STL

Microsoft IIS 5.0.

Microsoft Exchange Server.

Front End: ASP.

Back End: SQL Server 2000

3.3.2 Performance Requirements:

3.3.2.1 Security:

Only registered users can use this application.

3.3.2.2 Availability:

This web application can be accessed ay any time (ie) 24/7.

3.3.2.3 Capacity:

Simultaneously more than 100 users can use this system

3.3.2.4 Response Time:

Expected online response should be within 1 – 2 minutes in time.

3.3.3 Design Constraint:

3.3.3.1 Standard Compliance:

The standard compliance used here is Microsoft Specification for COM, ASP and Open standard HTML & XML.

3.3.3.2 External Interface Requirements:

The External Interface used here is GDS.

3.3.4 Other Requirements:

3.3.4.1 Operation required by the user:

The user just needs to get into the web application and go for reserving flights. To access the system the user should have web browser like Internet Explorer 5.5 and the above version.

3.3.4.2 Site Adaptation Requirements:

The software which is being developed is available to registered users for the websites from any where

Design Documentation

4. DESIGN DOCUMENTATION

4.1 Scope:

4.1.1.System Objectives:

Project 'Zeppelin' encompasses functionality, both existing and yet to be developed/acquired, to facilitate a "touch-less" process to create and price an air travel reservation using the Apollo and Galileo reservations systems and to provide fulfillment services of such reservations by means of an electronic ticket (or more than one electronic ticket).

The main objectives is to provide ticketing and booking via Highwire booking engine without human interface.

The primary target distribution channel for low-cost Zeppelin processing is the online travel reservations market. This encompasses both the traditional high-volume corporate travel sector (Highwire, Get There) as well as the one-off, cost-conscious consumer sector (Trip.com, Expedia, Travelocity). Rather than relying on characterizing the user, the product caters to a transaction model in which many online consumers can be classified. This transaction model, at a high level, contains the following attributes:

- Simple one way, round-trip or open jaw itineraries.
- Single surname.
- Use of negotiated or net fares and the need for their online tool to locate and correctly apply these fares
- Electronic ticketing - a paper-less transaction

An e-ticket (electronic ticket) is a paperless electronic document used for ticketing passengers, particularly in the commercial airline industry. Virtually all major airlines now use this method of ticketing.

4.1.2. Major Software Requirements:

For the current module which is in middle tier architecture
the major software required is VC++ 6.0. which is mainly used to develop ATL components and Win 32 dlls. The mode of data transfer between Mid tier components and GDS wrapper component is in the form of XML.

4.2 Data Analysis:

4.2.1 Data objects & Resultant Data Structure

Group of related properties are stored in a class as private members .These classes have Get and Set methods to access these private member variables

4.2.2 Global Data

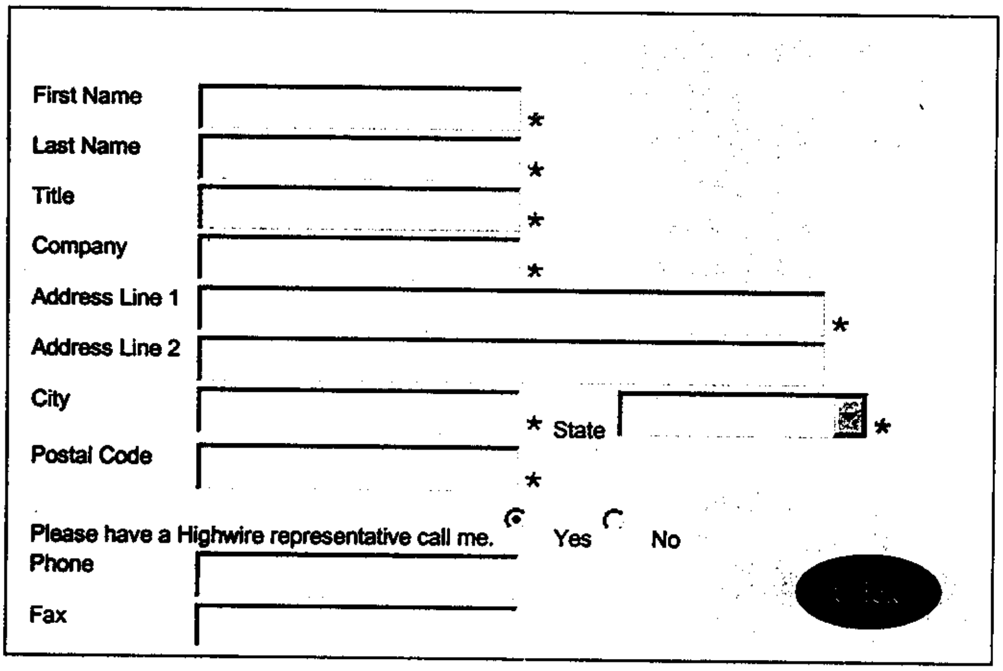
Global data are from the user interface where the classes are been created for each data for interacting with GDS.

4.3 Architectural design:

4.3.1 Review of data & control flow

Step 1: First the input is got through User interface .For the new employees the profile details has to be filled primarily by the user.

Profile Details will be already stored in Highwire engine:



First Name *

Last Name *

Title *

Company *

Address Line 1 *

Address Line 2

City * State *

Postal Code *

Please have a Highwire representative call me. ● Yes ○ No

Phone

Fax

Fig . 7

On Clicking Next the new web page is shown where the user needs to enter the employee Itinerary details

Itineraries Details:

Edit Boxes:

Name:

Passport no:

Credit Card no:

E-Mail Id:

Origin - Date- Time
Destination - Date - Time
Adults/Senior:

Combo Box:

Zeppelin – (Yes/No)

List Boxes :

Airlines:

US Airways
United Airlines
Air Canada
Swiss International Airlines
American West
American Airlines

Class:

Economy/Coach
Business
First Class

Combo Box:

Round Trip
One Way
Multiple Destination.

After Shopping is done ,the Booking is made. At the time of booking the following inputs are given

First Name:
Middle Name:(Optional)
Last Name:

Home Phone No:
Country, Area, Phone #.
Work Phone No:
Country, Area, Phone #, Extn.
Mobile Phone No:
Country, Area, Phone #.

Preferences & Policies:

List Boxes:

Aisle or Window:

Aisle

Window

Smoking or Non Smoking:

Smoking

Non Smoking

Meal Request (Optional):

Moslem

Pure Vegetarian

Low Fat/Cholesterol

Kosher

Low Sodium

Special Assistance:

Wheel Chair

Dog

Frequent Flyer Program:

Air Canada Aeroplane

British Airways Executive Club.

United Milage Plus

American Advantage.

Frequent Flyer #:

Emergency Contact:

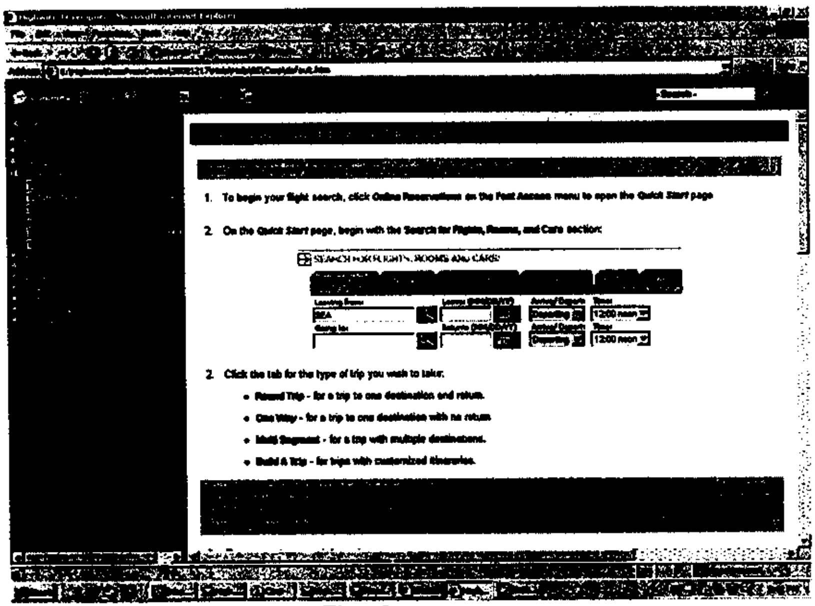
First Name:

Last Name:

Country, Area, Phone #.

When the booking is confirmed then PNR number will be displayed along with the summary.

The interface is shown below:



1. To begin your flight search, click Online Reservations on the Fast Access menu to open the Quick Start page

2. On the Quick Start page, begin with the Search for Flights, Rooms, and Cars section:

SEARCH FOR FLIGHTS, ROOMS AND CARS

Leaving From: SEA
Leave (MM/DD/YY):
Arrive/Depart: Departing
Time: 12:00 noon
Going to:
Return (MM/DD/YY):
Arrive/Depart: Departing
Time: 12:00 noon

2. Click the tab for the type of trip you wish to take:
- Round Trip - for a trip to one destination and return.
- One Way - for a trip to one destination with no return.
- Multi Segment - for a trip with multiple destinations.
- Build A Trip - for trips with customized itineraries.

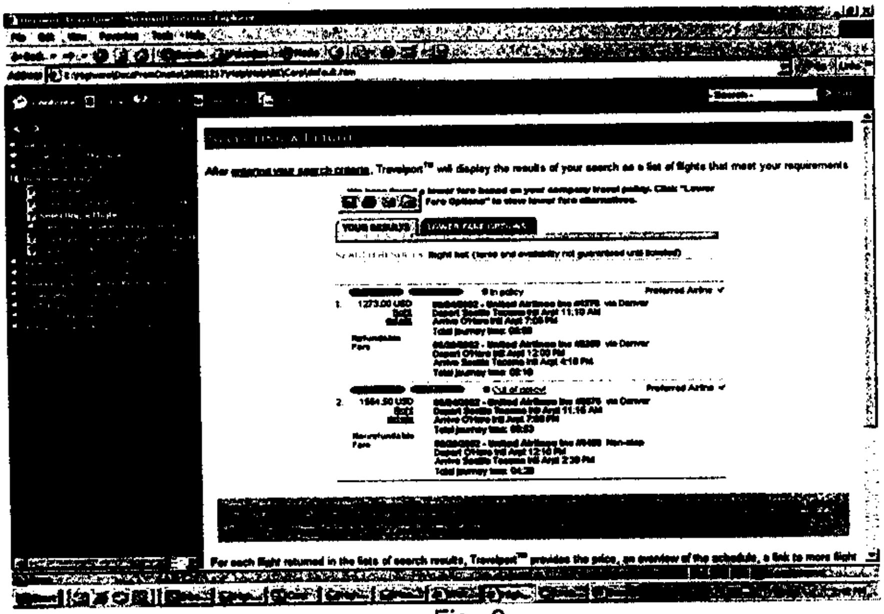
Fig . 8

Step 2: These inputs are retrieved by HighWire Booking Engine

Step 3 Highwire Booking Engine will in turn executes a series of GDS calls to retrieve the availability of flights

Step 4: If the availability is confirmed then Booking GDS calls are made to block the ticket.

Step 5 . PNR is retrieved for successful booking of ticket



After entering your search criteria, Travelport™ will display the results of your search as a list of flights that meet your requirements

[illegible] lower fare based on your company travel policy. Click "Lower Fare Options" to view lower fare alternatives.

YOUR RESULTS | LOWER FARE OPTIONS

SEARCH RESULTS flight list (fares and availability not guaranteed until ticketed)

● In policy — Preferred Airline ✓

1. 1273.00 USD — [illegible] - United Airlines flt [illegible] via Denver
Depart Seattle Tacoma Intl Appt 11:10 AM
Arrive O'Hare Intl Appt 7:00 PM
Total journey time: [illegible]

Refundable Fare — [illegible] - United Airlines flt [illegible] via Denver
Depart O'Hare Intl Appt 12:00 PM
Arrive Seattle Tacoma Intl Appt 4:10 PM
Total journey time: [illegible]

● Out of policy — Preferred Airline ✓

2. 1564.50 USD — [illegible] - United Airlines flt [illegible] via Denver
Depart Seattle Tacoma Intl Appt 11:15 AM
Arrive O'Hare Intl Appt 7:20 PM
Total journey time: [illegible]

Nonrefundable Fare — [illegible] - United Airlines flt [illegible] Non-stop
Depart O'Hare Intl Appt 12:10 PM
Arrive Seattle Tacoma Intl Appt 2:30 PM
Total journey time: [illegible]

For each flight returned in the lists of search results, Travelport™ provides the price, an overview of the schedule, a link to more flight

Fig . 9

Step 6. For the retrieved PNR, additional information is attached and resubmitted to GDS.

Step 7 . A series of ticketing calls are executed on GDS and E-ticket is sent to Customer.

E-ticket Format

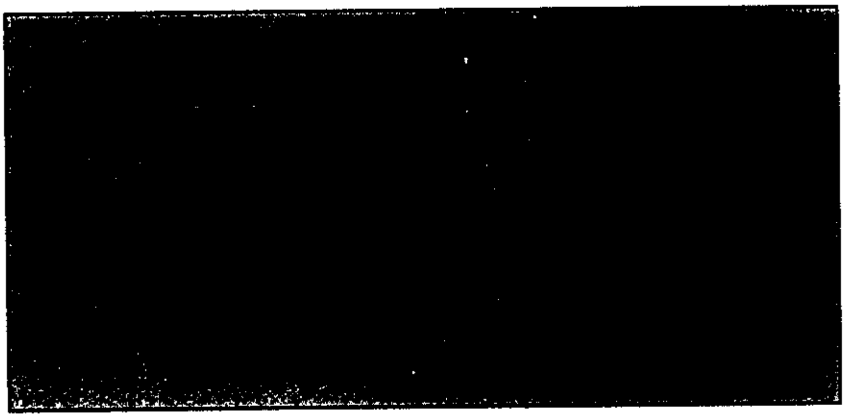


Fig . 10

Data Flow:

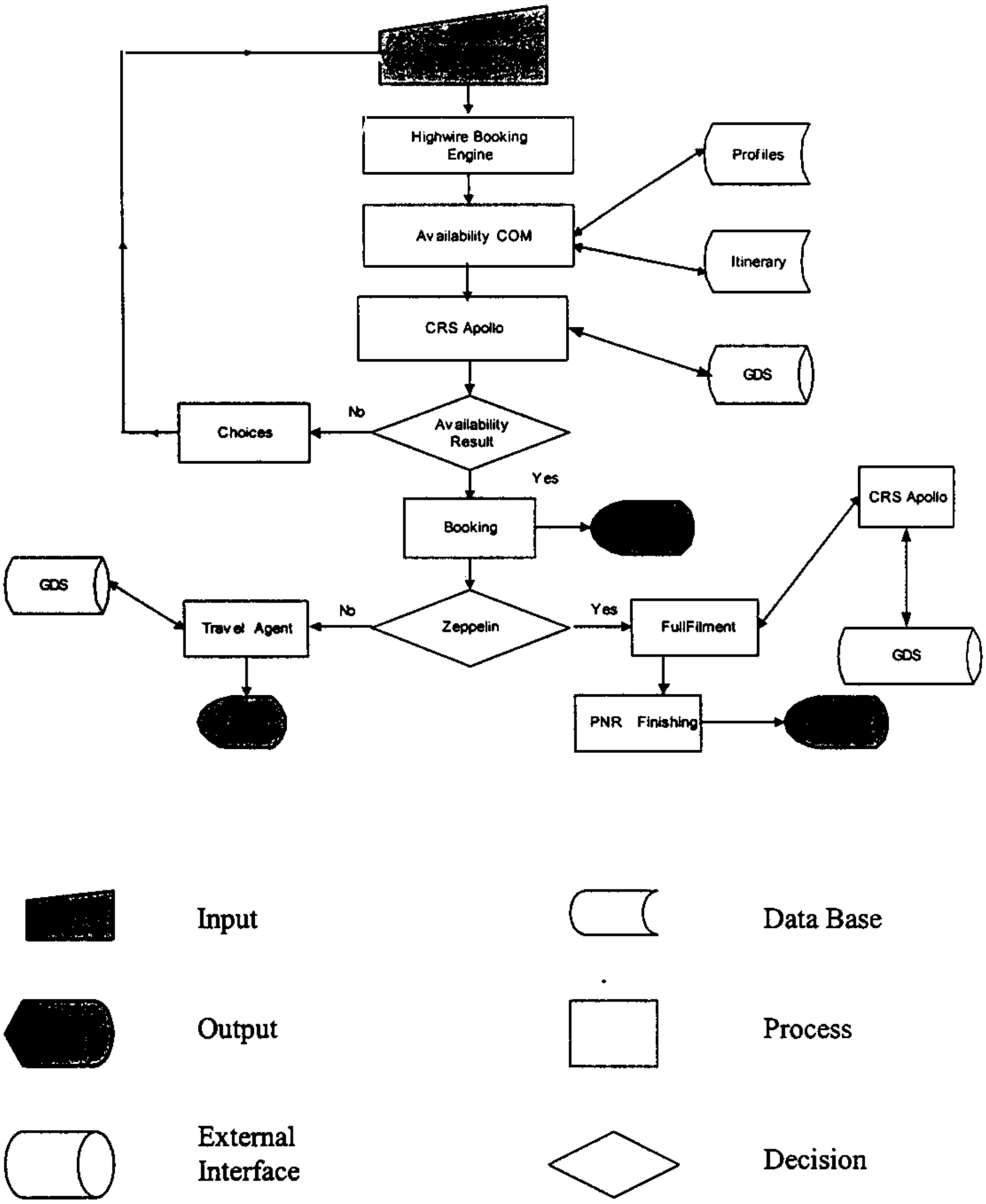


Fig . 11

4.4 Interface Design

4.4.1 Human – Machine Interface Specifications

Client side interface: User Interface Web Application.- ASP Pages.

Middle tier interface: No Interface just getting inputs via ASP Pages.

Server side interface: No Interface is used Queries are passed to Data Base.

4.4.2 Human Machine Interface Design Rules:

Design rules for User Interface:

- User must register their details first time when they use this web application.
- After inputting their itinerary details , they can either select or unselect the Zeppelin option.
- If the polices are accepted then booking is done .
- If the confirmation is made through Zeppelin , then E-ticket will be issued .

E-ticket

An e-ticket (electronic ticket) is a paperless electronic document used for ticketing passengers, particularly in the commercial airline industry. Virtually all major airlines now use this method of ticketing.

When a customer books a flight by telephone or using the Web, the details of the reservation are stored in a computer. The customer can request that a hardcopy confirmation be sent by postal mail, but it is not needed at the check-in desk. A confirmation number is assigned to the passenger, along with the flight number(s), date(s), departure location(s), and destination location(s). When checking in at the airport, the passenger simply presents positive identification. Then necessary boarding passes are issued, and the passenger can check luggage and proceed through security to the gate area.

The principal advantage of e-ticketing is the fact that it reduces booking expense by eliminating the need for printing and mailing paper documents. Another advantage is that it eliminates the possibility of critical documents getting lost in the mail or being sent to the wrong address

- If the confirmation is made through Travel agent then Ticketing is done by them.

4.4.3 External Interface Design

4.4.3.1 Interface to external systems or devices

The external interface used here is XML Select which is a third party tool which interacts with GDS Apollo.

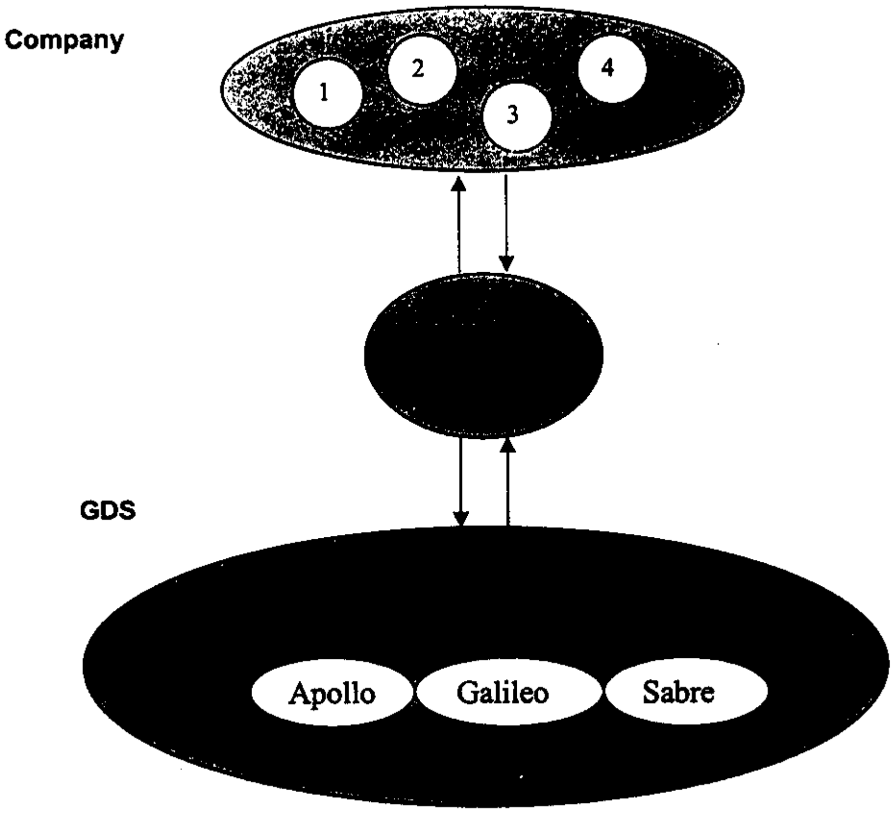


Fig . 12

Company

Companies are the customers of the Highwire system. Highwire maintains the profile data of the employees of these companies.

GDS

GDS provides various travel related data to the Highwire system. This includes availability of flight and also enables booking of the same. In addition to this basic service, it also enables storage of profiles and preferences for individuals and corporate. Currently only Apollo is supported by the Highwire system.. Highwire has plans to integrate with all major GDS.

4.4.3.2 Internal interface design rules:

Since interface is not used , the inputs are just retrieved form the interface and corresponding classes are being called depending on the type of input given.

These classes in turn calls wrapper classes for its booking.

When booking is done CRS Apollo gathers details for ticketing and then calls CRS Apollo Implementation class.

This Implementation class will in turn sends XML request to GDS with the help of XML Select.

The GDS will in turn gives responses in the form of XML. Again XML parser is used to convert XML to ATL COM and it is send back to CRS Apollo.

CRS Apollo will in turn issues the E-ticket.

4.5 Procedural Design

(for each module)

Module Name: CRS Apollo.

4.5.1 Processing narrative:

- Highwire Booking Engine gets the input from the User Interface.
- CRS Apollo gets the input from the Booking Engine.
- All these inputs are send to the corresponding classes and then instances are been created for all these classes and interacts with the availability COM Module.
- Availability Module in turn sends instances to the wrapper classes in the CRS Apollo.
- Thus booking & Ticketing is made with the help of CRS Apollo.

4.5.2 Interface Description:

There is no interface used.

4.5.3 Design Language:

VC++ 6.0 & XML.

4.5.4 Modules Used:

High wire Booking Engine ,Availability COM Module & GDS.

4.5.5 Internal Data Structure:

For each and every data which is being used classes are created.
These classes will in turn wrapper classes.

These wrapper classes which in turn invokes member class and creates instances for the member class. Therefore clients interacts with the member classes and the e instance of the wrapper class.

The following classes are been used for its operation.

- The first class which is being invoked is CAvailabilityManager
- This class will in turn invokes CBookItinerary.
- This will in turn class CProfileCompanyPolicy, CProfilePerson, CProfileServiceOrgGroup.
- Finally when the Profiles and Policies are been set, it calls the Wrapper class CApolloItinerary
- CApolloItinerary invokes CApolloItineraryImpl which is actually the member class to which instances could be created. Through instances BookItinerary, RetrieveItinerary and build PNR No is performed.
- After all these details are obtained it goes to Ticketing portion along with the PNR NO.
- For Ticketing it calls CTicketItinerary Wrapper Class which in turn calls CTicketingItineraryImpl member class which in turn calls XML Select functions and with the help of XML parser the queries are posted to GDS.

- GDS will in turn gives response which has to be once again needed to convert to VC++ ATL COM .Response will be in the form of E-ticket and the Zeppelin indicator.
- So once again the result from GDS via ATL COM should be converted to ASP pages and then displayed to the user.

The XML Select architecture is shown below:

Application – Zeppelin

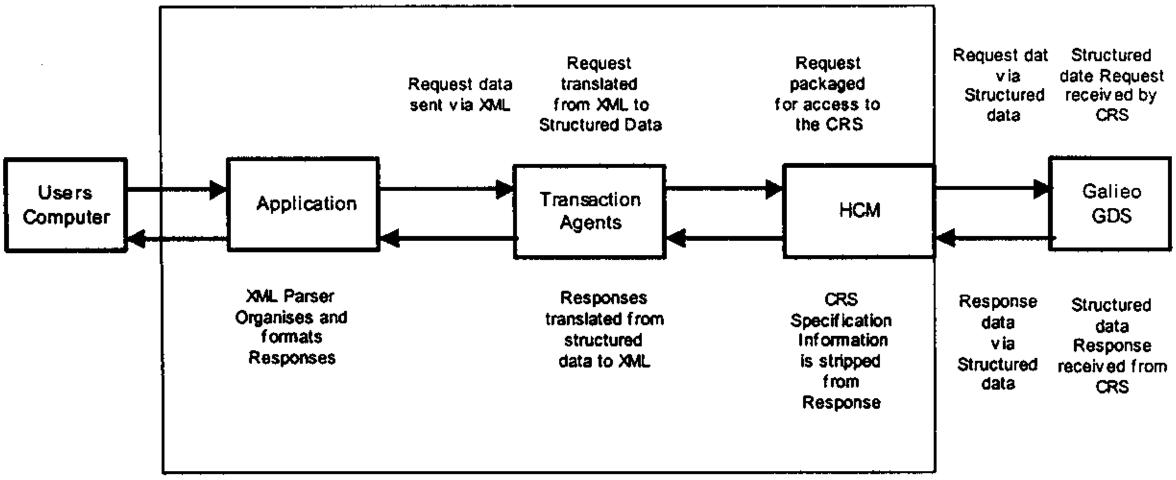
Transaction Agents: Translates XML to Structured Data and
Vice – Versa.

There are two Transaction agents they are

- UTA – Universal Transaction Agent: It is used only for VC++.
- SUTA – Scriptable Universal Transaction Agent: It is used for other languages like VB, JAVA , C.

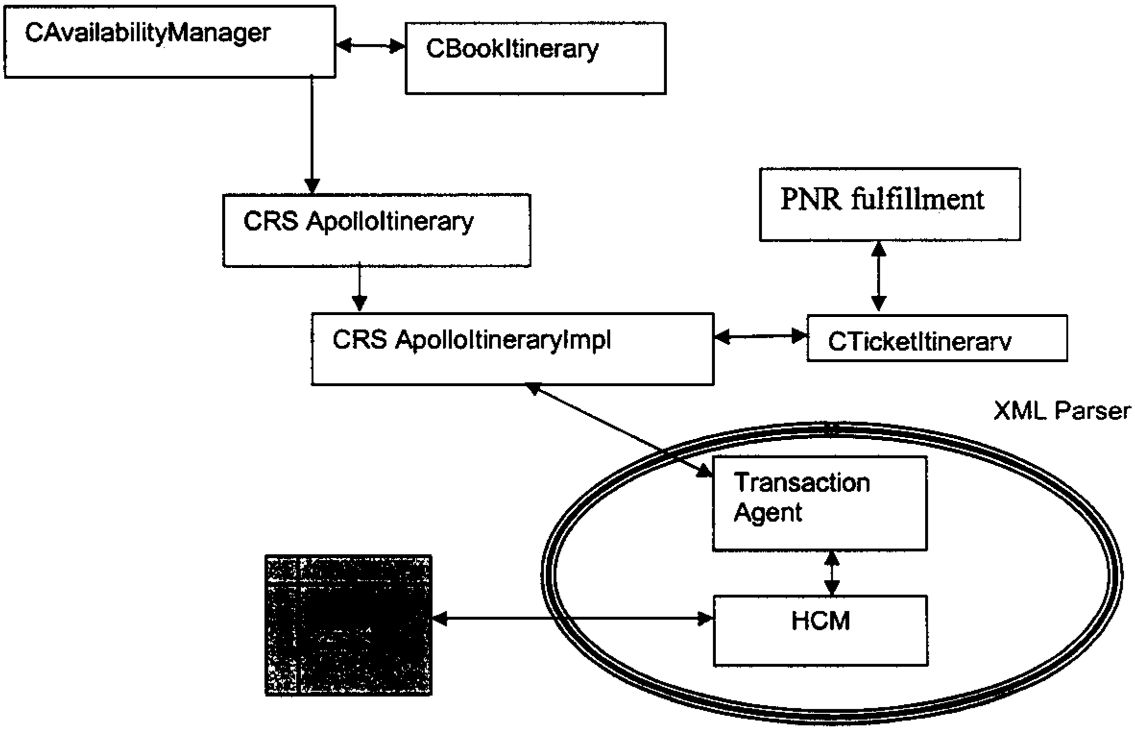
HCM : Host Connection Manager. The HCM component serves as a gateway between the travel application and the CRS.

Fig . 13


Users Computer
Application
Transaction Agents
HCM
Galieo GDS
Request data sent via XML
Request translated from XML to Structured Data
Request packaged for access to the CRS
Request dat via Structured data
Structured date Request received by CRS
XML Parser Organises and formats Responses
Responses translated from structured data to XML
CRS Specification Information is stripped from Response
Response data via Structured data
Structured data Response received from CRS

The Initial Class Diagram for CRS Apollo Module is depicted below:

Fig . 14



4.5.6 Detailed Design Specifications:
Pseudo code for VC++ Ticketing:

The inputs to the CRS Apollo for ticketing is as follows:

- Name
- Credit card No
- Pass Port No
- Company Name

- Empl E-mail
- Comp E mail
- PNR no
- Flight Name
- Origin – Date – Time
- Destination – Date-Time
- Total Cost

The output from the ticketing module is

- Confirmation number
- Name
- Flight Name
- Origin – Date – Time
- Destination – Date-Time
- Total Cost

With all these input details , CRS Apollo calls CTicketItinerary Class.

Which in turn calls few methods to make PNR Fulfillment.

PNR Fullfilment is adding few more information with PNR NO.

Once PNR fulfillment is made , the queries are passed via XML Select statements to GDS.

```
CTicketItinerary.Pseudocode()
{
Obtain the Pseudocode for the Type of GDS.
Apollo, Galileo.........etc
}
CTicketItinerary.PNRFulfill()
{
Load E-Ticket Preference, and check once again with Booking pseudocity code.
The user has opted for Quick buy, where the ticket cost is same.
Add few more information with raw PNR NO .
}
```

Once this PNR finishing is compete the Transaction request is made via XML Select statements.

Pseudo Code for XML Select:

Using UTA (Universal Transaction Agent)

There three main steps to get connected to HCM

1. Either create an HCM or get an existing HCM, depending on whether you are using the HCM Manager service on an HCM server.
2. Build an XML transaction request.
3. Submit transactions to the GDS.

Step 1:

- ➢ If an HCM exist already create an object this HCM
- ➢ With the help of the object create an instance to the object using CoCreateInstance method.

```
IHcmManager* pHcmMgr;
CoCreateInstance(CLSID_HcmManager, NULL, CLSCTX_LOCAL_SERVER,
IID_IHcmManager, (void**)&pHcmMgr));
```

Step 2:

- First Getting the ITransManager Interface from HCM

ITransManager* objt

J = HCM->QueryInterface(InterfaceID, *objt)

- Building XML request, Filter and Identity.

```
OLECHAR *prequest = OLESTR(Method in CTicketItinerary)

OLECHAR *pclient = OLESTR(Flitering the information alone)

OLECHAR *pIdentity = OLESTR("\ <application>

<Vendor Id> aa </Vendor Id>\ <Vendor Type> ax </Vendor Type> \

<Source Id> ws </Source Id> \ <Source Type> hh </Source Type>

</application>

<User>

<UserrId> tyyuu </UserId>\ <EmpId> abc </EmpId>  </User>
```

On calling CTicketItinerary the following details are supplied to GDS

With the help of XML Select.Request is given to GDS Using one method

Response is obtained from the GDS using another method

In the Request Method the following details are supplied

```
<CompanyName> alphanum </CompanyName>\ <EmpName> String </EmpName>\

<CreditCard No> Numeric </CreditCard No> \ <Passport No> Numeric </Passport
No>\

<Emp E-Mail> Alphanum </Emp E-Mail>\ <Comp-E-Mail> Alphanum </Comp-E-
Mail>\

<PNRNo> Alphanum </PNRNo>\ <FlightName> AlphaNum </FlighName>\ <Origin>

 String </Origin> \<Date> date </Date>\ <Time>Hrs.Min  </Time>\ <Destination>

String </Destination> \<Date> date </Date>\ <Time>Hrs.Min  </Time>\ <Type> String

  </Type>\ <Total Cost> Number </Total Cost> </application>")
```

- Create Transaction agent

  Using CoCreateInstance create Transcation agent

Step 3:

Release HCM Object

Release Transaction Manager Object

Release Transaction Agent Object

Using HCM:

There are many interfaces used by HCM

They are IConnect, IConnectSecurity, ITransactions, ITransManager, IHCMDataMonitor, IHCMTransMonitor.

These interfaces Calls the Method inside the Interface using Instance.

Finally the response is given by another Method via same interface. Which is then passed to Transaction Agent.

This Transaction Agent gives responses to the CRS Apollo Module.

The output from CRS Apollo Module is indicated by Zeppelin Indicator.

4.5.7 Comments/Restrictions/Limitations:

The web application could be run only by the registered users and it is executed in Internet Explorer 5.5 and above version. Only XML Select interacts with GDS.

4.6 Test Provisions:

4.6.1 Test Guidelines

Rational rose is used for testing the software.

Some manual testing is also done.

4.6.2 Integration Strategy

Therefore for using rational rose , the modules must be registered with the rational rose.

User's Manual

5.User's Manual

5.1 Introduction

5.1.1 Product rational and overview

The product is developed mainly for Highwire Booking Engine. The product has got its own features like accessing inputs from the user interface and then processing the inputs.

The product has got its own security. The console user can use the product only after registering their details to the web application. The security cannot be broken unless the user knows the password.

The primary target distribution channel for low-cost Zeppelin processing is the online travel reservations market. This encompasses both the traditional high-volume corporate travel sector (Highwire, Get There) as well as the one-off, cost-conscious consumer sector (Trip.com, Expedia, Travelocity). Rather than relying on characterizing the user, the product caters to a transaction model in which many online consumers can be classified. This transaction model, at a high level, contains the following attributes:

- Simple one way, round-trip or open jaw itineraries
- Single surname
- Use of negotiated or net fares and the need for their online tool to locate and correctly apply these fares
- Electronic ticketing - a paper-less transaction

5.1.2 Terminology:

Before going into the web application there are some terms or technical words that has to be known.

COM: Component Object Model.

Com helps to assume application from pre built parts or components irrespective of location or the development language used to implement the components.

ATL: Active Template Library.

ATL is an excellent alternative to MFC for writing COM components with a small memory and code footprints

GDS: Global Distribution System - a computer reservation and information system that is often operated by multiple airlines, used by travel agents & other travel professionals and contains information on all types of travel products

PNR: Passengers Name Record..

TMC: Travelers Management Companies

CRS: A Computer Reservation System for an ***individual*** airline containing information on that airline only and used by that airlines' employees.

Booking: Booking is just blocking the tickets.

Ticketing: Ticketing is a process of issuing E-tickets after its confirmation.

Booking Engine: The web application is put on the Booking Engine where the users can book the tickets .

5.1.3 Basic features:

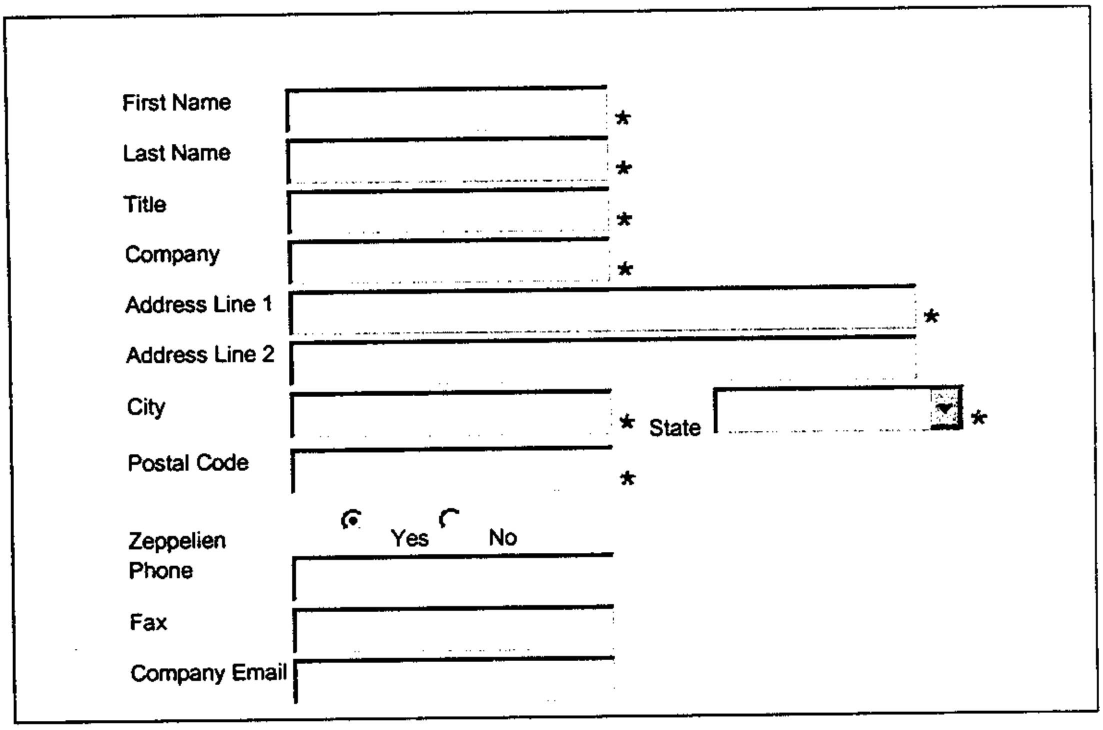
The basic feature of this product is to reserve air tickets in online and get E-tickets as soon as possible.

The user must first register to the highwire web application. Unregistered users are not allowed to get into web application.

Step 1: First the input is got through User interface the following inputs are

Profile Details:

Fig . 15



First Name *

Last Name *

Title *

Company *

Address Line 1 *

Address Line 2

City * State *

Postal Code *

Zeppelien ● Yes ○ No

Phone

Fax

Company Email

Itineraries Details:

Edit Boxes:

Origin - Date- Time

Destination - Date - Time

Adults/Senior:

ListBoxes:

- Airlines:
  - US Airways
  - United Airlines
  - Air Canada
  - Swiss International Airlines
  - American West
  - American Airlines
- Class:
  - Economy/Coach
  - Business
  - First Class
- Combo Box:
  - Round Trip
  - One Way
  - Multiple Destination.

After Shopping is done ,the Booking is made. At the time of booking the following inputs are given

First Name:
Middle Name:(Optional)
Last Name:

- Home Phone No:
  - Country, Area, Phone #.
- Work Phone No:
  - Country, Area, Phone #, Extn.
- Mobile Phone No:
  - Country, Area, Phone #.

Preferences & Policies:

- List Boxes:
  - Aisle or Window:
    - Aisle
    - Window
  - Smoking or Non Smoking:
    - Smoking
    - Non Smoking
  - Meal Request (Optional):

Moslem
Pure Vegetarian
Low Fat/Cholesterol
Kosher
Low Sodium

Special Assistance:
Wheel Chair
Dog

Frequent Flyer Program:
Air Canada Aeroplane
British Airways Executive Club.
United Milage Plus
American AAdvantage.

Frequent Flyer #:

Emergency Contact:
First Name:
Last Name:
Country, Area, Phone #.

When the booking is confirmed then PNR number will be displayed along with the summary.
Output of Booking is

Short Summary:
Flight Name:
Passenger Name:
Origin:
Destination:
Age:
Ticket Cost:
Taxes:
Booking Fees:
Total Cost:

Step 2: These inputs are retrieved by HighWire Booking Engine

Step 3 Highwire Booking Engine will in turn executes a series of GDS calls to retrieve the availability of flights

Step 4: If the availability is confirmed then Booking GDS calls are made to block the ticket.

Step 5 . PNR is retrieved for successful booking of ticket

Step 6. For the retrieved PNR, additional information is attached and resubmitted to GDS.

Step 7 . A series of ticketing calls are executed on GDS and e ticket is sent to Customer.

5.1.4 Summary of display

Totally there are two types of displays

- **E-ticket display:**

The E-ticket display consists of the following details

Short summary about the employee.

Flight details.......

Employee Name:

Company Name:

Designation:

Age:

Flight Name:

Origin – Date – Time:

Destination- Date – Time.

Seat no:

Preferences & Policies

- **Unavailability Form:**

  This form gives details about the unavailability of flight and asks the user to return to home page so that the user can opt for other available choice.

5.1.5 Outline of the manual:

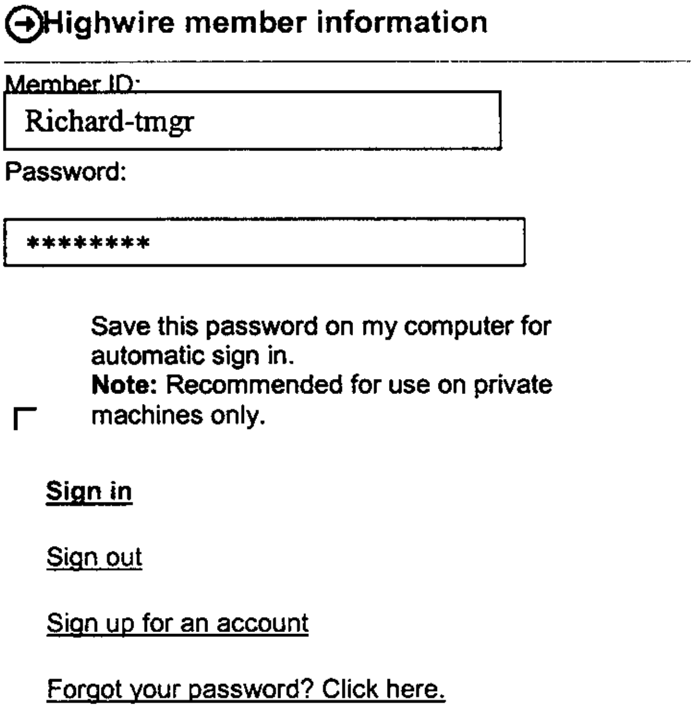
This manual informs about starting the Web application, the format the inputs are given to User Interface , how these inputs are taken to the various modules for its computations and how these computed values are displayed.

5.2 Getting Started:

5.2.1 Sign-on

If the user is already registered to the web application , they can log into the web page , If not the user needs to register their details in the registration form.

Fig . 16



Highwire member information

Member ID:

Richard-tmgr

Password:

****

□ Save this password on my computer for automatic sign in.
Note: Recommended for use on private machines only.

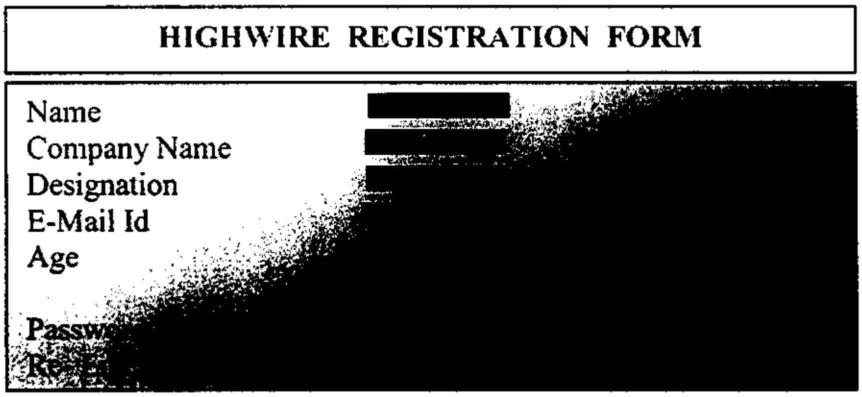
Sign in

Sign out

Sign up for an account

Forgot your password? Click here.

Fig . 17



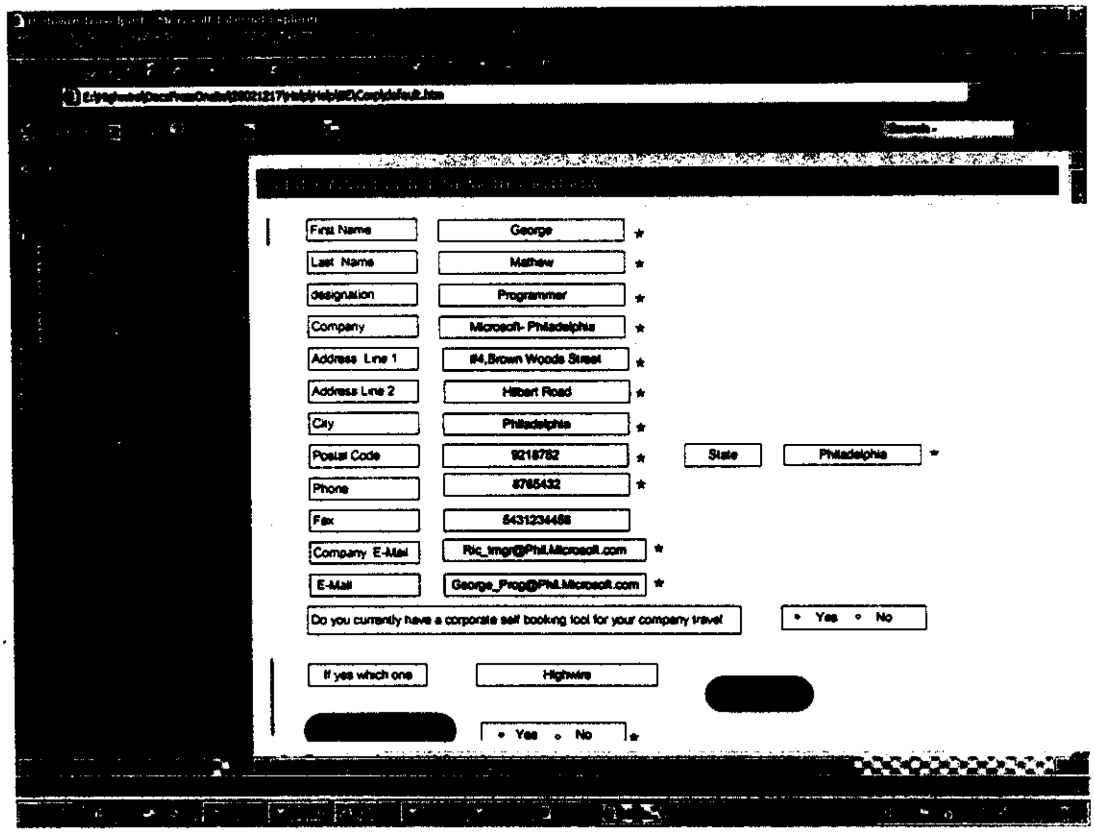
HIGHWIRE REGISTRATION FORM

Name
Company Name
Designation
E-Mail Id
Age

Passw
Re-E

After the sign in made the Personal details are to be included.

Fig . 18



First Name: George *

Last Name: Mathew *

designation: Programmer *

Company: Microsoft- Philadelphia *

Address Line 1: #4,Brown Woods Street *

Address Line 2: Hilbert Road *

City: Philadelphia *

Postal Code: 9218782 *

State: Philadelphia *

Phone: 8765432 *

Fax: 5431234456

Company E-Mail: Ric_mgr@Phil.Microsoft.com *

E-Mail: George_Prog@Phil.Microsoft.com *

Do you currently have a corporate self booking tool for your company travel: ● Yes ○ No

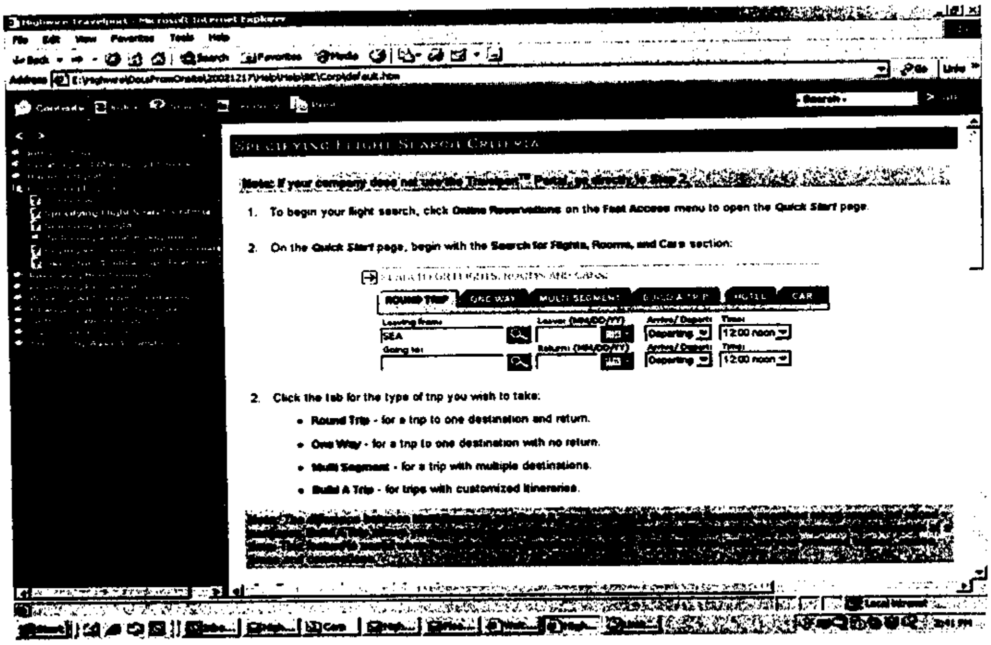
If yes which one: Highwire

● Yes ○ No *

After entering the personal details the user clicks **Click Here** button to visit next web page where it asks the user to enter the Itinerary details .

Here Comes the Web page where the flight details has to be entered like Name of the Employee, Designation, origin, destination ,time, date, Credit Card number, PassPort Number and few Preferences and policies are.also needed to be filled.

The type of trip should also be entered.



Specifying Flight Search Criteria

Note: If your company does not use the Travelport™ Portal, go directly to Step 2.

1. To begin your flight search, click Online Reservations on the Fast Access menu to open the Quick Start page.

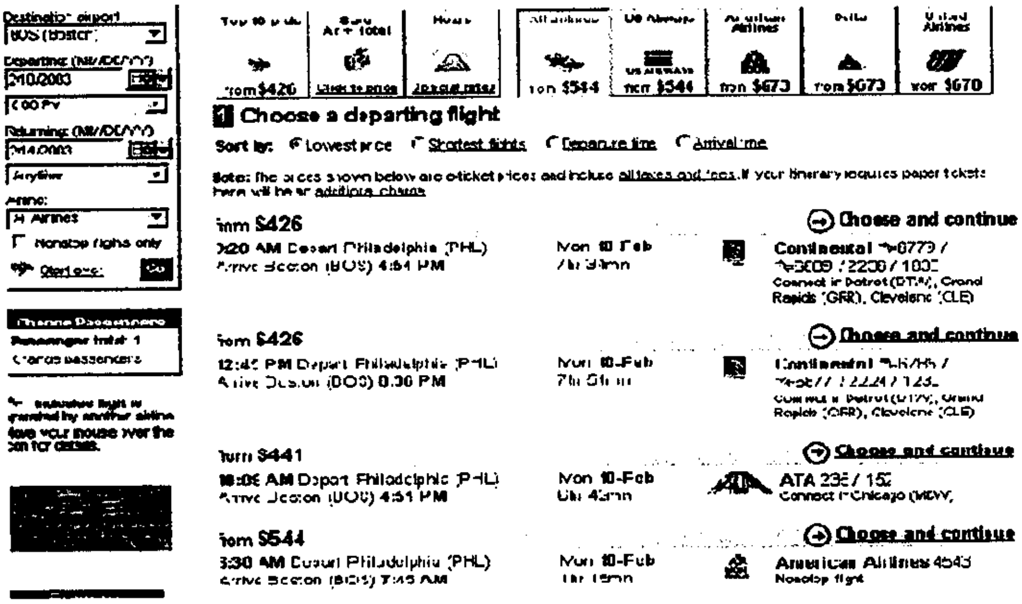
2. On the Quick Start page, begin with the Search for Flights, Rooms, and Cars section:

2. Click the tab for the type of trip you wish to take:

- Round Trip - for a trip to one destination and return.
- One Way - for a trip to one destination with no return.
- Multi Segment - for a trip with multiple destinations.
- Build A Trip - for trips with customized itineraries.

Fig . 19

The searching is done. The list of flights are displayed on the screen.



1 Choose a departing flight

Sort by: ● Lowest price ○ Shortest flights ○ Departure time ○ Arrival time

Note: The prices shown below are e-ticket prices and include all taxes and fees. If your itinerary requires paper tickets there will be an additional charge.

from $426 — Choose and continue
9:20 AM Depart Philadelphia (PHL) — Mon 10 Feb — Continental 0779 / 3009 / 2230 / 1003
Arrive Boston (BOS) 4:54 PM — 7hr 34mn — Connect in Detroit (DTW), Grand Rapids (GRR), Cleveland (CLE)

from $426 — Choose and continue
12:45 PM Depart Philadelphia (PHL) — Mon 10-Feb — Continental
Arrive Boston (BOS) 8:30 PM — 7hr 51mn — Connect in Detroit (DTW), Grand Rapids (GRR), Cleveland (CLE)

from $441 — Choose and continue
10:05 AM Depart Philadelphia (PHL) — Mon 10-Feb — ATA 235 / 152
Arrive Boston (BOS) 4:51 PM — 6hr 45mn — Connect in Chicago (MDW)

from $544 — Choose and continue
3:30 AM Depart Philadelphia (PHL) — Mon 10-Feb — American Airlines 4543
Arrive Boston (BOS) 7:45 AM — 1hr 15mn — Nonstop flight

operated by another airline
Move your mouse over the icon for details.

Fig . 20

From the list choose any of the flight and then click Choose and continue.

In the next web page the flight with in policy is shown by green spot and out of policy is shown by red spot.

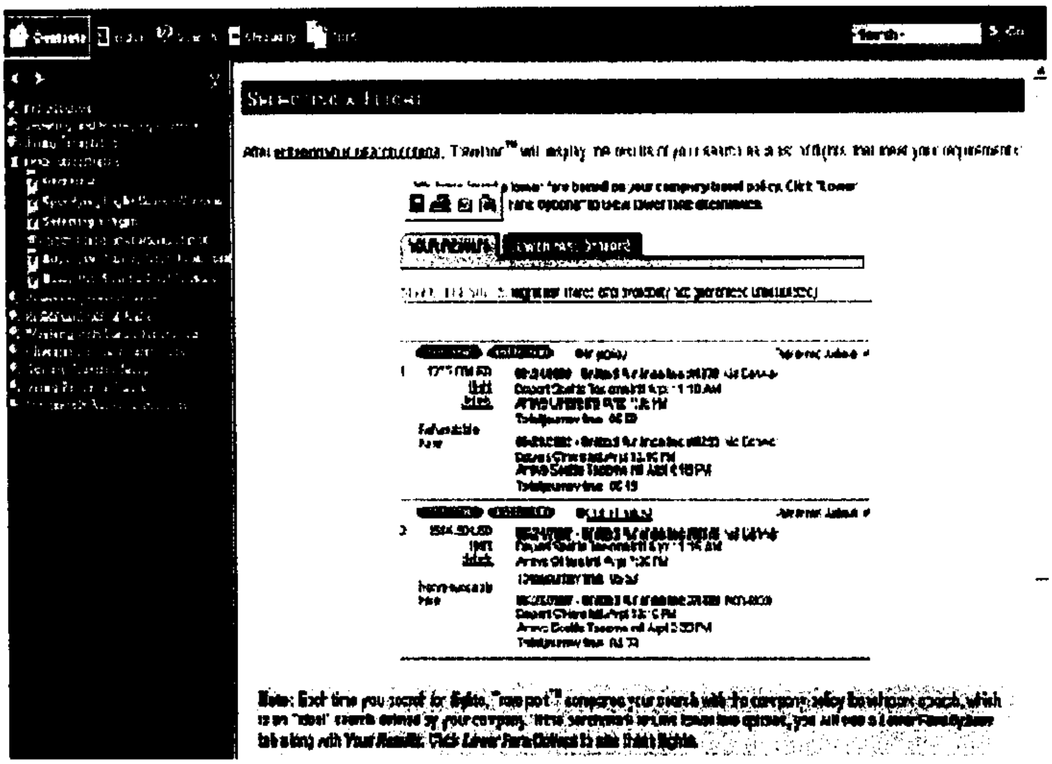


Fig . 21

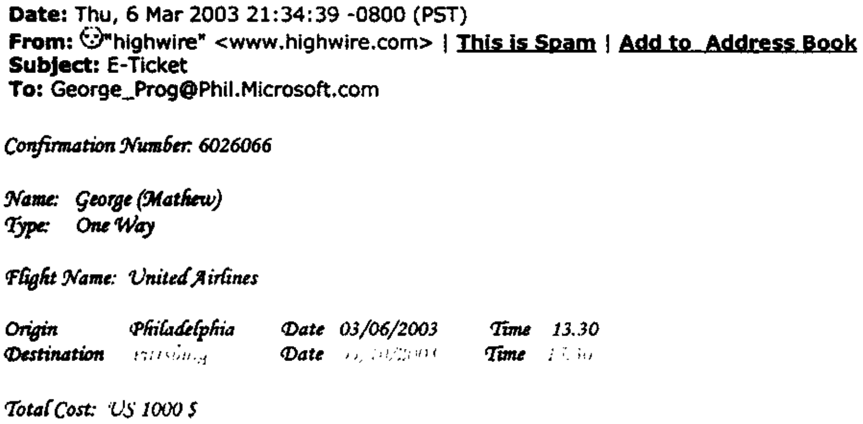
If the Zeppelin is allowed to do its booking it does booking as well as ticketing.

Finally Zeppelin gives out the E-ticket if the ticketing is done otherwise Unavailability form is displayed.

The PNR No will be: A02MARPHLPTS03.

PNR Fulfillment: HBEA02MARPHLPTS03UA.

The E-ticket is shown below:



Date: Thu, 6 Mar 2003 21:34:39 -0800 (PST)
From: "highwire" <www.highwire.com> | This is Spam | Add to Address Book
Subject: E-Ticket
To: George_Prog@Phil.Microsoft.com

Confirmation Number: 6026066

Name: George (Mathew)
Type: One Way

Flight Name: United Airlines

Origin	*Philadelphia*	*Date*	*03/06/2003*	*Time*	*13.30*
Destination	*Pittsburg*	*Date*	[illegible]	*Time*	[illegible]

Total Cost: US 1000 $

Fig . 22

- Confirmation Number : 6026066.
- Name: George(Mathew)

- Type: One Way
- Flight Name: United Airlines
- Origin: Philadelphia Date: 03/06/2003 Time: 13.30
- Destination: Pittsburg Date: 03/10/2003 Time :15.30
- Total Cost: 1000$

Booking Failure:

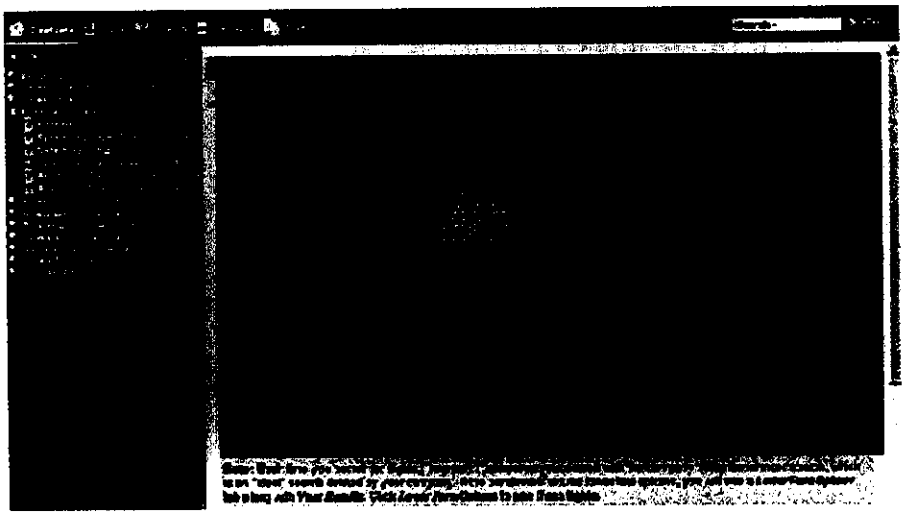


Fig . 23

5.3 Modes of operation

There are two modes of operation they are

Keyboard

Mouse

Keyboard:

With the help of keyboard the user can interact with the web application via user interface.

Mouse:

The user can select the available options from the user interface with the help of left mouse button.

5.4 Advanced Features

Earlier in all the products booking and ticketing operation is done individually. Booking is done by Highwire booking engine and Ticketing is done by Travel agents But in the case of Zeppelin ticketing and booking is done in Highwire booking engine. Which reduces cost, Man Power and consumes less time.

5.5 Command syntax and system options

The web application which is been developed is put into the net. The user uses Internet Explorer to log into the web application.. There fore the registered users logs into the web application and then starts reserving the tickets for air line

Test Plan

6. TEST PLAN

6.1 Introduction:

6.1.1 Purpose:

This test plan document for the project "Zeppelin" supports the following objectives:

1. To identify existing project information and the software components that should be tested.
2. List the recommended requirements for test.
3. Recommend and describe the testing strategies to be employed.
4. Identify the required resources and provide an estimate of the test efforts.
5. List the deliverable elements of the test project.

6.1.2 Background

The system entitled " Zeppelin" for Corporate Travel Reservation System for Highwire Company. The project mainly focuses on online Air Ticket Reservation System. The main objective of this project is to eliminate the travel agent. The project involves various process like shopping, booking, ticketing and issuing E-tickets. Currently the project is focused on ticketing. Ticketing process is mainly done to eliminate the travel agents. The advantages of ticketing is it consumes less time, Reduces cost , Fully automated and issuing of E-tickets.

The process begins with getting the inputs via user interface which is already done and the user can either select or unselect the zeppelin option. If the Zeppelin option is selected the ticketing is done by high wire booking engine itself, else the ticketing is done with the help of travel agency. Once booking gets over, ticketing is done.

In Ticketing along with PNR number is considered to be the raw information. Therefore few more information are been added to the PNR number and send to GDS. Which in turn does ticketing and sends an E-ticket to the users Mail-Id.

The application system has been developed using Visual C++ and XML Select. This web application is put on the net and the employees who belong to a particular company can only use this application. The system has been tested with varied sets of data and has been found to be working excellently.

6.1.3 Scope

The Zeppelin project will be tested as follows for functional quality and issues of scalability and performance

- Validation Testing
- Functional Testing

- Performance testing
- User Interface testing
- Stress Testing
- Security and access control Testing

The interaction of the module will also be tested as follows

- User Interface to Availability COM
- Availability COM to Booking Module
- Booking Module to CRS Apollo Module
- CRS Apollo to GDS
- GDS to Ticketing Module

6.2 Requirements for test:

The listing below identifies those items (Use case, Functional Requirements, Non- Functional Requirements) that have been identified as targets for testing. This list represents what will be tested.

6.2.1 Functional Testing

- Verify that whether Zeppelin option works or not.
- Verify whether the contents are retrieved correctly from the user interface.
- Verify that all the XML select scripts are send properly to interact with GDS.
- To check whether all data flows correctly from one module to another module.
- To check whether the fulfillment is done properly.
- Verify that the E-ticket is been send properly to the Employee Id as well as to the Company Id.

6.2.2 Performance Testing

- Since it is a web based application , time response plays a very important role, therefore testing should be made on response time.

6.2.3 Stress Testing

- Verify that the application could be accessed by more than 100 users simultaneously.

6.2.4 Security and Access Testing

- Verify that the confirmation number given to the E-ticket is unique.
- Verify that non-users cannot access customer and supplier information.

6.3 Resources

This section presents the recommended resources for the Zeppelin test effort, their main responsibilities and their knowledge or skill set.

6.3.1 Workers

This table shows the staffing assumption for the project.

6.3.1.1 Human Resources

Workers	**Minimum Resources Required**	**Specific Responsibilities / Comments**
Test Manager/ Test Project Manager	1	Provides management oversight Responsibilities: Provide technical direction Acquire appropriate resources Management reporting
Test Designer	1	Identifies, prioritizes and implements test cases Responsibilities: Generate test plan Generate test model Evaluate the effectiveness of test effort
Tester	1	Executes the tests Responsibilities: Execute tests Log results Recover from errors Document change requests
Test System Administrator	1	Ensures test environment and assets are managed and maintained Responsibilities: Administer test management system Install/manage worker access to test systems
Database Administration/	1	Ensures test data (database) environment and assets are managed and maintained

Database Manager		Responsibilities: Administer test data(database)
Designer	1	Identifies and defines the operations, attributes and associations of the test classes Responsibilities: Identifies and defines the test class(es) Identifies and defines the test Packages
Implementer	1	Implements and unit tests the test classes and test packages Responsibilities: Creates the test classes and packages Implemented in the test model.

6.3.2 System

Hardware Specifications:

Processor - Intel Pentium IV

Hard disk - 40 GB

Ram - I GB

Operating System - Windows 2000

Web Browser - IE 5.5

Technologies Used:

VC++ 6.0/ ATL COM / XML

Microsoft IIS 5.0.

Microsoft Exchange Server.

6.4 Test Cases:

The test cases are maintained while the inputs are passed to the CRS Apollo module and the outputs from CRS Apollo Module.

1. Performance Testing:

Since it is a web based application , the performance testing is done for the whole product. When the inputs are given via User Interface the time taken to issue E-Ticket is maximum 1-2 minutes.

2. Stress Testing:

In the case of doing Stress testing more than 200 Users were able to access the web application simultaneously.

3. Security and Access Testing:

In this case, unauthorized users cannot use this application. Before using this application the user needs to register their details and give the Password to access this application. When each and every time when user logs into an application, It checks for the user name and Password which is already stored in the database.

4.Requirements Testing:

It checks whether the customer requirements are satisfied or not.

5. Functionality Testing:

Test Case No		4.1		
Modules Used		User Interface		
COMBO BOX	**INPUT**	**DESIRED OUTPUT**	**ACTUAL OUTPUT**	**COMMENTS**
Zeppelin	Selected	Zeppelin should be selected	Zeppelin was selected	Verified and found correct

Test Case No		4.2		
Modules Used		CRS Apollo - Booking		
BUTTON	**INPUT**	**DESIRED OUTPUT**	**ACTUAL OUTPUT**	**COMMENTS**

Booking	Click	Employee Name, PNR No,Origin,Time,Date, Destination,Time,Date, Credit Card No, PassPortNo, Empl – Email, Comp-Email, Company Name, Total Cost should be given as the output.	All these outputs were given out in the form of classes	Verified and found correct

Test Case No	4.3
Modules Used	CRS Apollo - Ticketing

BUTTON	INPUT	DESIRED OUTPUT	ACTUAL OUTPUT	COMMENTS
Ticketing	Click	When this Button is clicked all the details for the ticketing should be send to hit GDS via XML Select	When this Button is clicked all the details for the ticketing is send to hit GDS via XML Select	Verified and found correct
Cancel	Click	If the cancel Button is clicked the Ticketing should be cancelled	When the cancel Button is clicked the Ticketing is be cancelled	Verified and Found Correct.

Test Case No	4.4
Modules Used	CRS Apollo – Ticketing.

FORM	DESIRED OUTPUT	ACTUAL OUTPUT	COMMENTS
E-Ticket	E-Ticket should be send through E-Mail	E-Ticket was sent through E-Mail	Verified and found correct.
Ticketing	All the values from the booking	Thus all the	Verified and found

	modules should be passed out correctly to the Testing Module	values from booking module was set correctly to the Ticketing Module.	correct.
Ticket Failure	When the Ticketing is failed , the failure form should be displayed	When the Ticketing is failed , the failure form is displayed	Verified and found correct.

6.5 Project Milestones:

Milestone Task	Start Date	End Date
Plan Test	March 1st, 2003	March 2nd , 2003
Design Test	March 3rd, 2003	March 5th , 2003
Implement Test	March 6th , 2003	March 8th , 2003
Execute Test	March 9th , 2003	March 10th , 2003
Evaluate Test	March 10th , 2003	March 11th , 2003

6.6 Deliverables:

For each test executed, a test result form will be created. This shall include the name or ID of the test, the supplemental specification to which the test relates, the date of the test, the ID of the tester, required pre-test conditions and results of the test. Microsoft Word will be used to record and report test results.

Project Legacy

7. PROJECT LEGACY

7.1 Project description

Project Zeppelin encompasses functionality, to facilitate a "touch-less" process to create and price an air travel reservation using the Apollo and Galileo reservations systems and to provide fulfillment services of such reservations by means of an electronic ticket (or more than one electronic ticket).

The main objectives is to provide ticketing and booking via Highwire booking engine without human interface.
The primary target distribution channel for low-cost Zeppelin processing is the online travel reservations market.

7.2 Initial expectations

Step 1: First the input is got through User Interface .

Step 2: These inputs are retrieved by HighWire Booking Engine

Step 3 Highwire Booking Engine will in turn executes a series of GDS calls to retrieve the availability of flights

Step 4: If the availability is confirmed then Booking GDS calls are made to block the ticket.

Step 5 . PNR is retrieved for successful booking of ticket

Step 6. For the retrieved PNR, additional information is attached and resubmitted to GDS.

Step 7 . A series of ticketing calls are executed on GDS and e ticket is sent to Customer.

7.3 Current status of the project:

The current status of the project is to do booking and ticketing in Highwire engine itself by eliminating travel agent. The currently developed system reduces cost, Consumes less time and avoid man power.

7.4 Activities / time log (s)

Time Logs	Activities
Jan 6th , 2003	Product Definitions
Jan 8th, 2003	Product Analysis
Jan 12th, 2003	Programming Language
Jan 15th, 2003	Risk Analysis
Jan 20th, 2003	Software Requirement Specifications
Jan 22nd, 2003	Design Documentation
Feb 5th, 2003	Coding
Mar 1st, 2003	Test Plan
Mar 12th, 2003	User Manual
Mar 14th, 2003	Install
Mar 15th, 2003	Release
Mar 16th, 2003	Customer Evaluation

7.5 Technical lessons learned:

Many technical lessons were learnt while working with this organization along with team members. Out of many techniques the best one is selected, learnt and implemented in the case of Problem analysis and its solving.

Based on the problem approach and its solution the logics and techniques were selected. The software used is VC++ ATL. With the help of ATL many COM components were been developed and used in the project.

7.6 Managerial lessons learned

In this organization apart from the technical lessons, managerial lessons were also learnt, that include how to approach the problem patiently, way of communicating with the team mates and asking them the suggestions and advices in some tough times while programming. The time slots were framed by whole team to finish each and every module. Suppose if there is a delay in coding for a particular module, the time for the very next modules timing should be adjusted in order to finish the project in a given time slot. As and when the modules are finished the time taken to finish is noted in a sheet of paper and it is compared with the original time slot.

7.7 Recommendation to future projects:

In this organization Microsoft Team was mainly concerned with Highwire projects . Currently the software was developed for Airline Reservation system, Hotel reservation and car reservation are developed in the future to meet current necessities.

Conclusion

8 CONCLUSION

Since this is a web application , lots of techniques are involved. The computer processes the data in a pre defined way. It has no common sense to update the current needs. So modification of the system from time to time may be needed to incorporate changes in the procedures. The system is also designed in the form of Open Systems. Lot of Baseline Specifications are included so that even a new system analyst can go through it very easily. Even an unregistered employee can use this application after registering their details with Highwire booking engine web application.

The main advantage of this system is its speed of response. Since the actual system is used for online purpose, time of response has a very important role.
Since the language used to develop this web application is VC++ - ATL , XML, ASP & SQL server and so the whole internal operations are very fast. The algorithms are made simple by reducing the complexity and it is mainly used for its speed of execution.

As a conclusion this web application project " Zeppelin – Corporate Travel Reservation System" is a very good reservation application developed for Highwire booking engine. It also includes all the features for further developments like adding Car and Hotel modules also.

References

8 REFERENCES

Books:

- ➢ Software Engineering - Rodger. S. Pressman.
  Mc GRAW – HILL International Edition 1997.

- ➢ VC++ in 21 Days- Nathan Gurewich & Ori Gurewich.
  TechMedia Publications Edition 1996.

- ➢ VC++ 6.0 Complete Reference – Chris H. Pappas & William
  H.Murray. Tata Mc GRAW – HILL Edition 1999.

- ➢ ATL 3 COM Programming- Richard Grimes, Shroff Publishers & Distributors Pvt Ltd 2000.

Websites:

- MSDN library - http:// www. Microsoft MSDN.com
- Visual C++ FAQ - http:// www.Mvps.org/ VCFaq
- VC++ for beginners – http://www.Download.Cnet.com
- www.travelocity.com – 16-Jan-2003
- www.expedia.com – 16-Jan-2003
- www.highwire.com – 18-Jan-2003

Appendix

10 APPENDIX

Contents