

AIRLINE RESERVATION SYSTEM

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering / Technology degree in Computer Science and Engineering.

By

A.KARTHIKEYAN (Reg No. 39290047)

R.KIRUBAKARAN (Reg No. 39290050)



**DEPARTMENT OF MECHANICAL ENGINEERING
SCHOOL OF MECHANICAL ENGINEERING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

**(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC**

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **A.KARTHIKEYAN (RegNo.39290047)** and **R.KIRUBAKARAN (RegNo.39290050)** who carried out the project entitled "**AIRLINE RESERVATION SYSTEM**" under my supervision from to November 2021 to May 2022.

Internal Guide

Dr. MOHANAPRASAD, B.E, M.E, Ph.D.,

Head of the department

Dr.L.LAKSHMANAN, M.E.,Ph.D.,

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I, **A.KARTHIKEYAN (Reg.No. 39290047)** hereby declare that the Project Report entitled “**AIRLINE RESERVATION SYSTEM**” done by me under the guidance of **Dr.MOHANAPRASAD,B.E.,M.E.,Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in Computer Science.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA , M.E., Ph.D.**, Dean, School of Computing **Dr. L. LAKSHMANAN M.E., Ph.D.**, and **Dr. S.VIGNESHWARI M.E., Ph.D.** Head of the Department, Dept. of for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.MOHANAPRASAD, B.E.,M.E, Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science who were helpful in many ways for the completion of the project.

ABSTRACT

The purpose of this section is to state the Goal and Objectives of the Software Project. The project presented here is the Airline Reservation System.

Airline reservations system is an integrated passenger processing system, including inventory, fares, ticket-less operations and credit card transactions. All communications are via TCP/IP network protocol enabling the using of both intranet and internet communications worldwide.

The solution includes several standard items, which are combined to provide an integrated solution with interfaces to other business systems. The system is based on open architecture, using industry standard equipment and software. The open nature of VRS allows the addition of new systems and features, ensuring that the VRS system can be adapted to keep up with the changing requirements of the airline business.

The VRS suite of software includes the functions of

- ❖ Reservations
- ❖ Flight inventory
- ❖ Fares

All user/agents are allocated a SINE code, which is used during sine-on and then appended to all transactions carried out by the agent for security purpose. Different security levels may be assigned so that different agents can access different areas of the system and also different records in the case where a travel agent is only allowed to review PNR's that have been created by that agency.

The flights may be specified within a particular date range and may be used to display different classes of service and different fares within a specific seating class. Sell from availability when it has been displayed and a simple entry is used to sell seats. A direct sale may be made using a long hand entry if the flight details are fully known.

CONTENT

CHAPTER NO	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	vii
	LIST OF TABLES	ix
	LIST OF ABBREVIATIONS	x
1	INTRODUCTION	1
2	LITERATURE REVIEW	2
3	AIM AND SCOPE OF PRESENT INVESTIGATION	4
	3.1 AIM	4
	3.2 SCOPE	4
4	METHODOLOGY	5
	4.1 SYSTEM ANALYSIS	6
	4.1.1 EXISTING SYSTEM	6
	4.1.2 PROPOSED SYSTEM	7
	4.1.3 PROBLEM STATEMENT	7
	4.2 OBJECT MODELLING	8
	4.2.1 IDENTIFICATION OF OBJECTCLASSES	8
	4.2.2 IDENTIFICATION OF ATTRIBUTES	9
	4.2.3 DYNAMIC MODELLING	11
	4.2.3.1 PREPARING A SCENARIO	11
	4.2.3.2 STATE DIAGRAM	12

	4.2.4 FUNCTIONAL MODELLING	13
	4.2.4.1 DATA FLOW DIAGRAM	13
	4.3 SYSTEM DESIGN	15
	4.3.1 TABLES	16
	4.3.1.1 DATABASE DESIGN	16
	4.3.1.2 TABLE DESCRIPTION	17
	4.4 IMPLEMENTATION	20
	4.4.1 SOFTWARE REQUIREMENTS	20
	4.4.2 HARDWARE REQUIREMENTS	21
	4.5 TESTING	22
5	RESULTS AND DISCUSSION	23
6	CONCLUSION AND FUTUREWORK	24
	REFERENCES	25
	APPENDIX	27
	A. SAMPLE CODE	27
	B. SCREENSHOTS	34

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.2.1	IDENTIFICATION OF OBJECTCLASSES	8
4.2.3.2	STATE DIAGRAM	12
4.2.4.1	DATA FLOW DIAGRAM	13

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
4.2.2	IDENTIFICATION OF ATTRIBUTES	9
4.3.1.2	TABLE DESCRIPTION	17

LIST OF ABBREVIATIONS

TCP	TRANSMISSION CONTROL PROTOCOL
IP	INTERNET PROTOCOL
OMT	ONBOARD MAINTENANCE TERMINAL
DFD	DATA FLOW DIAGRAM
LRS	LOGICAL RECORD STRUCTURES
LAN	LOCAL AREA NETWORK
VRS	VIDEOCOM RESERVATION SYSTEM

CHAPTER 1

INTRODUCTION

Airline reservation system is an integrated passenger processing system. This system includes:

- Fares
- Inventory
- Enquiry
- Reservations

In this system all the communications are via TCP/IP protocol using both the Intranet and Internet communications worldwide.

The Airline Reservation system has the following Modules:

User registration module:

This module is helpful for the registration of the new customer.

Login module:

This module performs the login of the registered customer. In this module Customer-id and password is verified.

Reservation module:

This module performs the reservation of the ticket to the registered module.

Cancellation module:

This module performs the cancellation of the reserved ticket.

CHAPTER 2

LITERATURE SURVEY

1.Web Service-based Automata Testing: A Case Study on Online Airline Reservation

Authors: Temitope Betty Williams ; Amir Rizaan Abdul Rahiman ; Izuka Joseph, 2020 International Conference on Computer Science and Its Application in Agriculture (ICOSICA)

The focus of this study is to show and evaluate how the web services can efficiently be utilized for online airline reservation system by utilizing the finite automata state machine. A finite automaton is a straightforward machine that has a finite numeral state which could be either an accepting state or rejecting state. It is being used to recognize the patterns either accept or reject by taking strings of symbols as input and changing its state according to rules of the machine (e.g., state function) until the anticipated symbols are created

2.The role of ICT Reservation Systems for operational management of air transportation companies

Authors: Okulski ; Radoslaw Robert , 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology

Airlines have been applying on-line technologies for booking and ticketing for decades. Starting from the early 1980s, the computer reservation system (CRS) has played a very important role in the provision of airline services. Zheng-Yi, Fang-Yuan, & Yu-Hern (2003) Consequently, the technologies data communication network had been organized. Moreover, CRS was sold to various markets around the world. Finally, after 1990, there was the Computer Reservation System, which was compatibly connected with a revenue management system. With some other added functions, car rental and hotel booking systems were introduced.

3. What Airline Reservation Systems Tell Us about the Future of EHRs

Authors: Sheila Sherlock ; William G. Chismar , Proceedings of the 39th Hawaii International Conference on System Sciences – 2006

In the airline industry, the justification for adoption of computerized reservation systems (CRSs), shifted from operational efficiency, to marketing strategies to recoup investment, to competitive advantages of essential business tools. This progress contributed to fundamental changes in the structure of the industry. In large part, network externalities created by these systems and the discrepancy between who paid for the systems, and who reaped the financial benefits drove these changes.

4. Theory of Dialogue Structures to Help Manage Human-Computer Interaction

**Authors: David L. Sanford, Associate Member ;John w. Roach, Member, IEEE
IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, VOL. 18, NO. 4, JULY/AUGUST 1988**

The principles of human communication underlie all modes of interaction and therefore are fundamental to designing effective interfaces. In particular, we present an extension to human communication theory that allows us to create a natural language interface using human communication principles. When people interact, they use “metacommunication” to help manage the shared communication context. Using metacommunication partially accounts for the enormous range of expressiveness in natural languages.

5. Airlines Reservations. Systems

Authors: John R. Knight , Proceidizgs of the IEEE, VOL. 60, KO. 11, NOVEMBER 1972

The structure of the system is described-agent terminal area, communications facilities, and central site. Lessons learned in the design, development, testing, implementation, and tuning of two generations of systems are discussed. These discussions include initial system design, simulations and systems measurement tools, systems stability and reliability, serial processing, parallel- or multiprocessors, split front-end back-end processing, storage hierarchy, standard and special communications disciplines, and flexibility versus performance.

CHAPTER-3

3.1 AIM

The main objective is to reduce the mistakes which creep up in manual systems. It provides good level of security so it takes care of the user's safety concerns as well. Passengers can access the whole list of all the flights available on different routes with their timings and fare both for economy and business classes. One can compare the best deals for them and book a flight accordingly. When the passengers enter all the details the software helps them to find all available flights and also information if there are seats available on that particular flight. The manual work is thus reduced and the chances of errors are reduced to minimum.

SCOPE

- This software helps passengers to look up flights between two points which can be domestic or international.
- The passengers can find and book tickets for flights through this software. Developed in java, it is fairly easy to use software having a user-friendly interface
- The main objective is to reduce the mistakes which creep up in manual systems. It provides good level of security so it takes care of the user's safety concerns as well.
- One can compare the best deals for them and book a flight accordingly. When the passengers enter all the details the software helps them to find all available flights and also information if there are seats available on that particular flight.
- The manual work is thus reduced and the chances of errors are reduced to minimum.

CHAPTER 4

METHODOLOGY

OMT consists of building a model of an application domain and then adding implementation details to it during the design of a system.

The Methodology has the following stages:

1. **Analysis**: The analysis model is a concise, precise abstraction of what the desired system must do, not how it will be done.
2. **System Design**: The system designer makes high-level decisions about the overall architecture, during system design; the target system is organized into subsystems based on both the analysis structure and the proposed architecture.
3. **Object Design**: The object designer builds a design model based on the analysis model but containing implementation details. The designer adds details to the design model in accordance with the strategy established during system design.
4. **Implementation**: The object classes and relationships developed during object design are finally translated into a particular programming language, database, or hardware implementation.

The OMT methodology uses three kinds of models to describe a system.

1. Object Model: The object model describes the static structure of the objects in a system and their relationships. The object model contains object diagrams.

2. Dynamic Model: The dynamic model describes the aspects of a system that change over time. The dynamic model contains state diagram

3. Functional Model: The functional model describes the data value transformations within a system. The functional model contains data flow diagrams.

4.1 SYSTEM ANALYSIS

4.1.1 EXISTING SYSTEM

Airline Reservation System is a System including Inventory, Fares, Enquiries, Reservations etc. All user/agents are allocated a SINE code which is used during sign-on and then appended to all transactions carried out by the agent for security purpose.

It has the following Dis-advantages:

- Reservations does not support up to 1 year.
- Accessing and updating the system is slow.
- It is a file based processing.

4.1.2 PROPOSED SYSTEM

Using the distributed technology we can handle these problems easily. In general a distributed process means that a program in execution makes use of resources in other machine. The two technologies for distributed processing available are J2EE and .NET . In the “Airline reservation system” J2EE is used for managing distributed systems. So that even if the Airline system has lot of branches and they are located at different places , we can handle the management of service and guarantee.

The proposed system for the problem is “Airline reservation system” , a web based system that allows online reservations. The system is divided into three layers namely presentation layer , business layer and data layer. The presentation layer is at the client side. At server side , business layer and data layer reside. The system requires a server side technology for its implementation. J2EE platform is chosen for implementing the system. At server side Servlet plays the role for business layer and JDBC for the data layer.

It has the following advantages.

- Reservations are supported up to 1 year.
- Accessing and Updating the system is Fast.
- It is Implemented using the concepts of RDBMS.

4.1.3. PROBLEM STATEMENT

The objective of the project is to design and implement the software which helps the Airline System employees to issue reservation tickets for various Air flights and maintain the records of various passengers and provide quick services to the passengers. It provides the following services:

- Reservation and Cancellation of the Airline tickets.
- Maintains the passengers’ records.
- Reports about the daily transactions of the Airline System.

- Quick Response to the passengers.
- Automation and integration of Airline system functions.
- Higher productivity and effective management
- Security and protection of confidential data.
- Transaction management and routing.

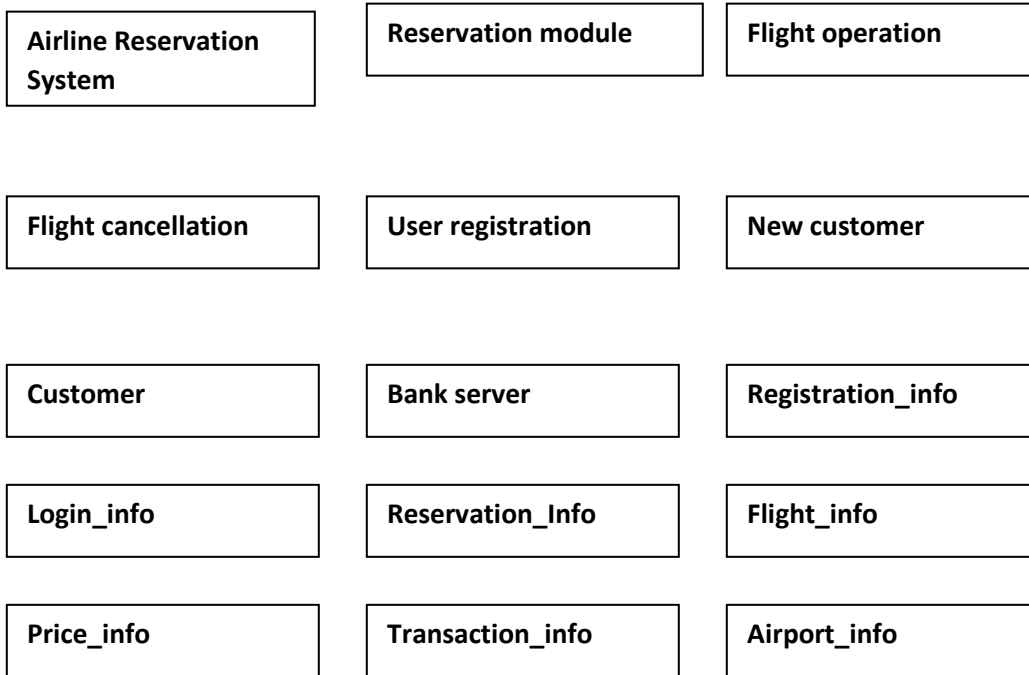
4.2. OBJECT MODELLING

The object model describes the structure of objects in a system. Their identity, their relationships to other objects, Their attributes, and their operations. The object model provides the essential framework into which the dynamic and functional models can be placed.

Our goal in constructing an object model is to capture those concepts from the real world that are important to an application. The object model is represented graphically with object diagrams containing object classes. Classes are arranged into hierarchies sharing common structure and behaviour and are associated with other classes. Classes define the attribute values carried by each object instance and the operations which each object performs or undergoes.

4.2.1. IDENTIFICATION OF OBJECT CLASSES

The first step in constructing an object model is to identify relevant object classes from the application domain. Objects include physical entities, such as houses, employees, and machines, as well as concepts, such as trajectories, seating assignments, and payment schedules. All classes must make sense in the application domain; avoid computer implementation constructs, such as linked lists and subroutines. Not all classes are explicit in the problem statement; some are implicit in the application domain or general knowledge.



4.2.2. IDENTIFICATION OF ATTRIBUTES

Attributes are properties of individual objects. Attributes should not be objects. Use an association to show any relationship between two objects. Attributes usually correspond to nouns to follow by possessive phrases. Adjectives often represent specific enumerated attribute values. Attributes are less likely to be fully described in the statement of problem. The knowledge must be drawn from the application domain and the real world to find out the attributes. This figure 3.2.2 shows the identification of attributes.

Class	Attributes
Registration_info	FirstName, LastName, Address, ContactNo,City,State,Country, Gender,Email_Id, Customer_id, password
Login_info	Customer_id, password
Reservation_info	Customer_id, flight_num, Flight_name, derarture_time, Arrival_time, origin, Destination, Num_of_seats
Flight_info	Flight_num, flight_name, Departure_time, Arrival_time, Origin,destination,num_of_seats, Country_name
Price_info	Class,customer_name, seat_num,price
Transactio_info	Credit_num, Credit_type, Pin_num
Airport_info	Country_name, Airport_name
Flight_cancellation	Customer_id, Flight_id, origin, Destination, Flight_name, departure, Arrival,seat_num

4.2.3. DYNAMIC MODELLING

The dynamic model describes the aspects of a system that change over time. The dynamic model is used to specify and implement the control aspects of a system. The dynamic model contains state diagrams. A state diagram is a graph whose nodes are states and whose arcs are transitions between states caused by events.

The dynamic model is insignificant for a purely static data repository. Such as a database. The dynamic model is important for interactive systems. For most problems, logical correctness depends on the sequences of interactions, not the exact times of interactions.

Dynamic modelling is a description of aspects of a system concerned with control, including time, sequencing of operations, and interaction of objects.

Following steps are performed in constructing a dynamic model.

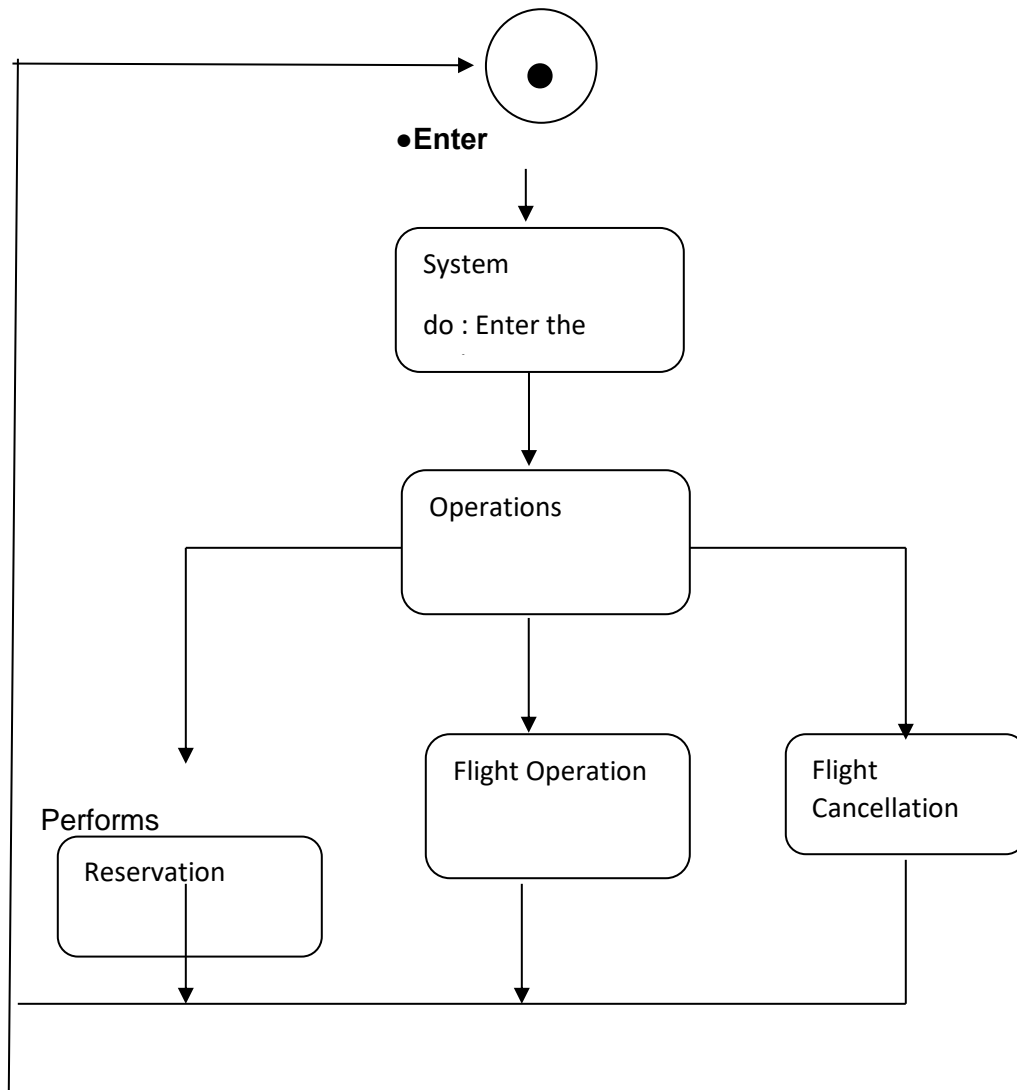
- Prepare scenarios of typical interaction sequences.
- Identify events between objects.
- Prepare an event trace for each scenario
- Built a state diagram
- Match events between objects to verify consistency.

4.2.3.1. PREPARING A SCENARIO

- New customer enters the system and makes the registration and gets a customer_id.
- Customers who are already registered enters the system.
- Customer makes the reservation operation and gets the response.
- Customer makes the query for flight operations & gets back the response.
- Customer makes the flight cancellation operation and gets the appropriate response.

4.2.3.2. STATE DIAGRAM

A state diagram relates events and states. When an event is received, the next state depends on the current state as well as the event: A change of state caused by an event is called a transition. A state diagram is a graph whose nodes are states and whose directed arcs are transitions labelled by event names. A state is drawn as a rounded box containing an optional name. A transition is drawn as an arrow from the receiving state to the target state: The label on the arrow is the name of the event causing the transition. Figure below shows a state diagram describing the behaviour of Airline Reservation System.



4.2.4. FUNCTIONAL MODELLING

The functional model shows how values are computed, without regard for sequencing, decisions or object structure. The functional model shows which values depends on which other values and the functions that relate them. The DFD are useful for showing functional dependencies. The functions are expressed in various ways, including natural language, mathematical equations and pseudo code.

Functional Model is a description of aspects of a system that transform values using functions, mappings, constraints and functional dependencies.

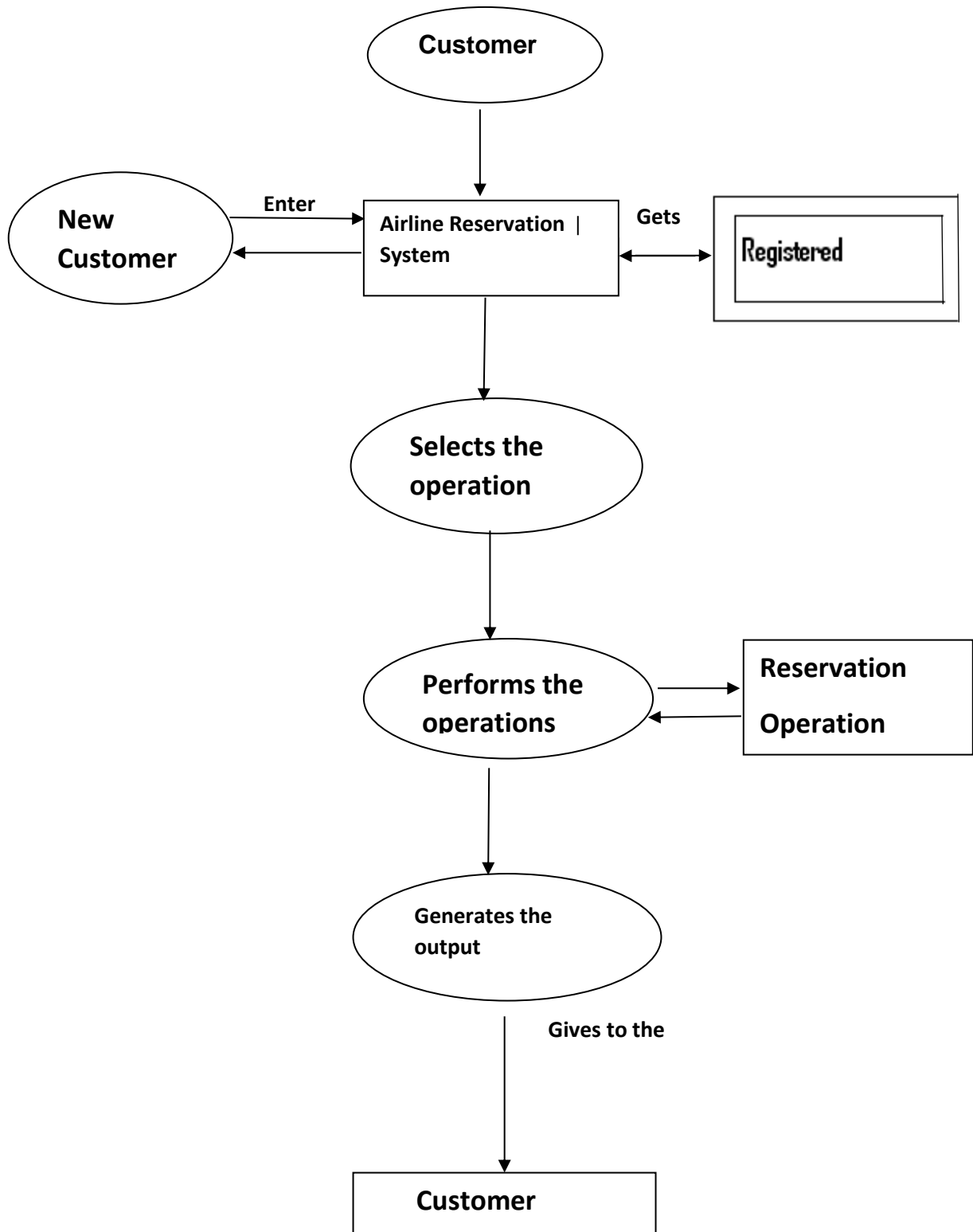
The following steps are performed in constructing a functional model.

- Identify input and output values.
- Build data flow diagram showing functional dependencies.
- Describe functions.
- Identify constraints.
- Specify optimization criteria.

4.2.4.1. DATA FLOW DIAGRAM

Data flow diagrams (DFD) depict information flow and the transforms that are applied as data move from input to output. It is the starting point of design phase that functionally decomposes the requirement specifications down to the lowest level details. Thus a DFD describes what data flows (Logically) rather than how they are processed. So, it does not depend on hardware, software or data structures. It is one of the most important tools used during system analysis. It is used to model the system components such as the system process, data used by the process any external that interact if the system and information flows in the system.

DFD-Airline Reservation System:



4.3. SYSTEM DESIGN

The problem analysis is the most important phase in any project. Only after knowing precisely what the problem is could we successfully eliminate it. The identification of the root problem is necessary. We were able to discuss with the personnel of various departments and gather information and we got a clear picture of what the existing problem were and what our jobs was to eliminate them by redesigning a new design.

Design is a multi step process that focuses on data structure, software architecture, Procedural details (algorithms etc) and interface between the modules. The design process also translates the requirements into the representations of the software that can be assessed for quality before coding begins.

Computer software design changes continually as new methods, better analysis and border understanding evolve. Software design is at a relatively early flexibility and quantitative nature that is normally associated with more classical engineering design disciplines. However, techniques for software design to exist criteria for design qualities are available and design notation can be applied.

Once the software requirements have been analyzed and specified, software design is the first of three technical activities- Design code and test that are required to build and verify the software. Each activity transforms information in a manner that ultimately results in validation of the computer software.

The importance of the software design can be started with a single word quality. Design is the place where quality fostered in software development. Design provides us with representations of the software that can be accessed for quality.

Design is the only way that we can accurately translate a customer's requirements into a finished software product or system. Without design, risk of building an unstable system exists one that will fail when small changes are made. One that may be difficult to test.

Thus the system design includes following three types of design:

- **Data Design:** The data design transforms the information domain model created during analysis into the data structures that will be required to implement the software.
- **Architectural Design:** The architectural design defines the relationship among the major structural components of the program.
- **Procedural Design:** The procedural design transforms structural components into a procedural description of the software. Source code is generated and testing is conducted to integrate and validate the software.

Thus, system design is a solution, a “how to” approach to the creation of the new system.

4.3.1. TABLES

4.3.1.1. Database Design

The database design converts the data model developed in logical design to a database definition that is supported by database software. The first step is independent of the kind of DBMS used. This step converts the conceptual entity relationships model to a set of record type is known as the logical record structures. (LRS). The next database design step converts the LRS to a database definition.

These steps use techniques that depend on the DBMS. DBMS dependent techniques are needed here because different DBMS support different kind of links between the records. Such links are used to retrieve records by following the link from one record to another. Database design depends on the structure supported by DBMS and uses techniques appropriate to these structures.

DBMS dependent design proceeds in two stages. The first step is logical design. Logical design defines the DBMS record types and the links between them. The next step is physical design. This step chooses a physical organization that supports the methods uses to accesses the databases.

4.3.1.2. Table Description

Table name: Registration_info

Field Description	Name	Type	Width
First Name	FirstName	Varchar	12
Last Name	LastName	Varchar	12
Customer's Address	Adress	Varchar	16
Customer's Contact Num	ContactNo	Number	12
Customer's City	City	Varchar	12
Customer's State	State	Varchar	12
Customer's Country	Country	Varchar	12
Customer's Gender	Gender	Varchar	2
Customer's Email-Id	Email_Id	Varchar	12
Customer's Id	Customer_id	Varchar	8
Customer's Password	Password	Varchar	8

Table name: Login_info

Field Description	Name	Type	Width
Customer's Id	Customer_id	Varchar	8
Customer's Password	Password	Varchar	8

Table name: Reservation_info

Field Description	Name	Type	Width
Customer's Id	Customer_id	Varchar	8
Flight's Number	Flight_num	Varchar	12
Flight's Name	Flight_name	Varchar	12
Departure Time	Departure_time	Time	
Arrival Time	Arrival_time	Time	
Origin Place	Origin	Varchar	16
Destination place	Destination	Varchar	16
Number of Seats	Num_of_seats	Number	4

Table Name: Flight_info

Field Description	Name	Type	Width
Flight's Number	Flight_num	Varchar	12
Flight's Name	Flight_name	Varchar	12
Departure Time	Departure_time	Time	
Arrival Time	Arrival_time	Time	
Origin Place	Origin	Varchar	16
Destination place	Destination	Varchar	16
Number of Seats	Num_of_seats	Number	4
Country Name	Country_name	Varchar	12

Table Name: Price_info

Field Description	Name	Type	Width
Class Name	Class	Varchar	12
Customer's Name	Customer_name	varchar	14
Seat Number	Seat_num	Number	4
Price Of ticket	Price	Number	4,2

Table name: Transaction_info

Field Description	Name	Type	Width
Credit card number	Credit_num	Varchar	8
Credir card type	Credit_type	Varchar	8
Pin Number	Pin_num	Number	6

Table name: Airport_info

Field Description	Name	Type	Width
Country Name	Country_name	Varchar	14
Airport Name	Airport_name	Varchar	14

Table name: Flight_cancellation

Field Description	Name	Type	Width
Customer's Id	Customer_id	Varchar	8
Flight's Number	Flight_num	Varchar	12
Origin Place	Origin	Varchar	16
Destination place	Destination	Varchar	16
Flight's Name	Flight_name	Varchar	12
Departure Time	Departure_time	Time	
Arrival Time	Arrival_time	Time	
Seat Number	Seat_num	Number	4

4.4 IMPLEMENTATION

4.4.1 SOFTWARE REQUIREMENTS

Server

Web logic server is used for recording data through Oracle 8i.

Compatible operating system

- Microsoft Windows 98(SE)
- Microsoft Windows NT Workstation version 4.0(with service pack 6 or later)
- Microsoft Windows 2000 Professional
- Microsoft Windows XP Professional

4.4.2 HARDWARE REQUIREMENTS

- Intel Pentium2 (or compatible) 300 MHz (or higher) processor
- Minimum of 256 MB RAM

- Oracle 8i and Server installation require up to 200MB of hard disk space and for planning purposes, we recommend that users allocate 100MB per system for data base.

Client

Compatible operating systems:

- Microsoft Windows 98(SE)
- Microsoft Windows 2000 Professional
- Microsoft Windows XP Professional

Network Requirements

User can run the HRMS on a single computer, or across a local area network (running at a min speed of 100 MHz). For access to the server via a LAN, TCP/IP protocol is recommended.

Remote Access

It's recommended that Microsoft Windows Terminal Services (or a similar technology) is used. Only the highest specification Wide Area Networks will provide sufficient power to connect users directly to the server (i.e.without using Terminal Services).

4.5 TESTING

Testing is an improvement phase. This phase involves testing of developed system using different form of data.

Testing the Software:

The objectives of the testing are:

1. Recruiting the program with an intention of finding an error
2. The test is said to be successful if an error is discovered

Types of Testing:

1. Unit Testing:

Unit testing focuses on verifying the effort on the smallest unit of the software design. The complexity of the test is limited by constrained scopes.

2. Integration Testing:

The objectives of the Integration Testing is to take all forms and build a project structure that has been dictated by design.

3. Validation Testing:

After integration testing system is completely assembled as a package, interfacing errors have been uncovered and the final series of the software test, the validation test begins validation succeeds.

4. Performance Testing:

It is designed to test the runtime performance of the software within the context of the integrated system. Performance testing occurs throughout the steps in the testing process.

5. Output Testing:

After performance and validation testing the next test is the input output testing of the proposed system. Since no system would be termed useful until it does produce the requested output in the specified format.

CHAPTER 5

RESULTS AND DISCUSSION

This project allows users to view flight details and to reserve, view and cancel tickets by logging in or registering with a new account and reporting any issue if required. Admin is provided with rights to see flight details, reservations, user contacts and some functions like adding flights and collecting reports given by users.

CHAPTER 6

CONCLUSION AND FUTURE WORK

The software package “Airline Reservation System” provides convenient online uploading the report from executive and viewing that report by the managing director in an online fashion. To input the data in a highly validated manner and generating the different reports, this involves complex process that was being done on based manner.

This package is designed and developed in a compact manner, which is ready to meet the user’s specification and to serve them in an effective as well as in an enhanced manner. The actual problem has been observed with keen interest and it has been defined and analyzed in such a way that it never causes choice to the user. More ever the limitation that has been prevailing in the existing system had been overcome to suit the need of the user.

High precision and care has been taken to design the data base, input forms an output reports since they should be given due importance which could otherwise to serious consequences thus affecting the whole system. The system thus developed has been implemented successfully which has been performed to scrutinized the validation of each data and errors were spotted out and then finally cleared in a sophisticated manner.

The added feature of this system is that it has been provided with many provisions for future enhancement in order to maintain the system in such a way that the future requirement of the user could also be satisfied and upgraded.

REFERENCES

- [1] Winston, Clifford, "The Evolution of the Airline Industry", Brookings Institution Press, 1995. ISBN 0-8157-5843-X. Cf. p. 61–62, Computer Reservation Systems.
- [2] "Unisys Launches Suite of AirCore® Passenger Service Solutions".
- [3] Wardell, David J, "Airline Reservation Systems", 1991. Research paper.
- [4] "Passenger Reservations". IBS Software. Retrieved 2021-07-30.
- [5] GmbH, finanzen net. "IBS Software Inks Multi-Year Contract With Jin Air". markets.businessinsider.com. Retrieved 2021-07-30.
- [6] "StackPath". www.aviationpros.com. Retrieved 2021-07-30.
- [7] "Italian start-up EGO Airways deploys IBS Software product" @businessline. Retrieved 2021-07-30.
- [8] "Magnetronic Reservisor". American Airlines C.R. Smith Museum Retrieved 3 August 2014.
- [9] The Magnetronic Reservisor, introduced in 1952, was the first electronic reservations system in the airline industry.
- [10] "The ineluctable middlemen". The Economist. 25 August 2012. Retrieved 29 August 2012.
- [11] Hitachi and Japanese National Railways MARS-1, Information Processing Society of Japan.
- [12] Early Computers: Brief History, Information Processing Society of Japan.

[13] R. Blair Smith, OH 34. Oral history interview by Robina Mapstone, May 1980.

[14] Strauss, Michael (2010), Value Creation in Travel Distribution.

[15] "American Airlines - Direct Connect". Directconnect.aa.com. Archived from the original on 2012-08-31. Retrieved 2012-11-08.

[16] "InteliSys amelia RES". Ch-aviation. Retrieved 4 November 2020.

[17] Pilling, Mark, "Airline reservations systems: can IT deliver?", 2008. News Article.

[18] "Sabre The First Online Reservation System". IBM. Retrieved 16 April 2020.

APPENDIX

SAMPLE CODE

FOR RESERVATION

```
public class Book_Ticket extends javax.swing.JFrame
{
public Book_Ticket() {
initComponents();
}
Private void initComponents()
{
jLabel5 = new javax.swing.JLabel();
fid = new javax.swing.JTextField();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
jLabel5.setFont(new java.awt.Font("Tahoma", 1, 14));
jLabel5.setText("Flight Id:");
        javax.swing.GroupLayout layout=new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGap(0, 729, Short.MAX_VALUE)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addGap(21, 21, 21)
.addComponent(jLabel5,javax.swing.GroupLayout.PREFERRED_SIZE,109,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(49, 49, 49)
```

```

.addComponent(fid, javax.swing.GroupLayout.PREFERRED_SIZE, 200,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(350, Short.MAX_VALUE)))
);
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGap(0, 359, Short.MAX_VALUE)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup())
.addGap(133, 133, 133)
.addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(fid, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addComponent(192, Short.MAX_VALUE)))
);
pack();
}
public static void main(String args[]) {
try
{
for (javax.swing.UIManager.LookAndFeelInfo
info: javax.swing.UIManager.getInstalledLookAndFeels()) {
if ("Nimbus".equals(info.getName()))
{
javax.swing.UIManager.setLookAndFeel(info.getClassName());
break;
}
}
}
}
}

```

```

catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(Book_Ticket.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);

    } catch (InstantiationException ex){
java.util.logging.Logger.getLogger(Book_Ticket.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(Book_Ticket.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);

            } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(Book_Ticket.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);

                } java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
                    new Book_Ticket().setVisible(true);
                }
            });
        }
private javax.swing.JTextField fid;
private javax.swing.JLabel jLabel5;
}

```

FOR PAYMENT

```

public class Payment_Option extends javax.swing.JFrame {
    public Payment_Option() {
        initComponents();
    }
    private void initComponents() {
        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        Credit_Card = new javax.swing.JButton();
    }
}

```

```

Debit_Card = new javax.swing.JButton();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0), 3));
jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18));
jLabel1.setText("Payment Option");
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(19, 19, 19)
        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 160,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(20, Short.MAX_VALUE))
    );
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addComponent(jLabel1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 34, Short.MAX_VALUE)
    );
Credit_Card.setFont(new java.awt.Font("Tahoma", 1, 14));
Credit_Card.setText("Credit Card");
Credit_Card.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Credit_CardActionPerformed(evt);
    }
});
Debit_Card.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
Debit_Card.setText("Debit Card");
Debit_Card.addActionListener(new java.awt.event.ActionListener()

```



```

{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Debit_CardActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(154, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(166, 166, 166)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                        layout.createSequentialGroup()
                            .addComponent(Debit_Card)
                            .addGap(86, 86, 86)
                            .addComponent(Credit_Card)
                            .addGap(125, 125, 125))))
                .addGap(125, 125, 125)))
        .addGap(125, 125, 125));
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(125, 125, 125)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(125, 125, 125)
            .addComponent(Debit_Card)
            .addGap(86, 86, 86)
            .addComponent(Credit_Card)
            .addGap(125, 125, 125));
};

```

```

        .addGap(58,                    58,                    58)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(Credit_Card, javax.swing.GroupLayout.PREFERRED_SIZE,
35, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(Debit_Card,
javax.swing.GroupLayout.PREFERRED_SIZE,                    35,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(266, Short.MAX_VALUE))
);
pack();
}
private void Credit_CardActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    Credit_Card ob=new Credit_Card();
    ob.setVisible(true);
}
private void Debit_CardActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    Debit_Card ob= new Debit_Card();
    ob.setVisible(true)
}
public static void main(String args[]) {
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    }
}

```

```

        } catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(Payment_Option.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(Payment_Option.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(Payment_Option.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(Payment_Option.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
    }

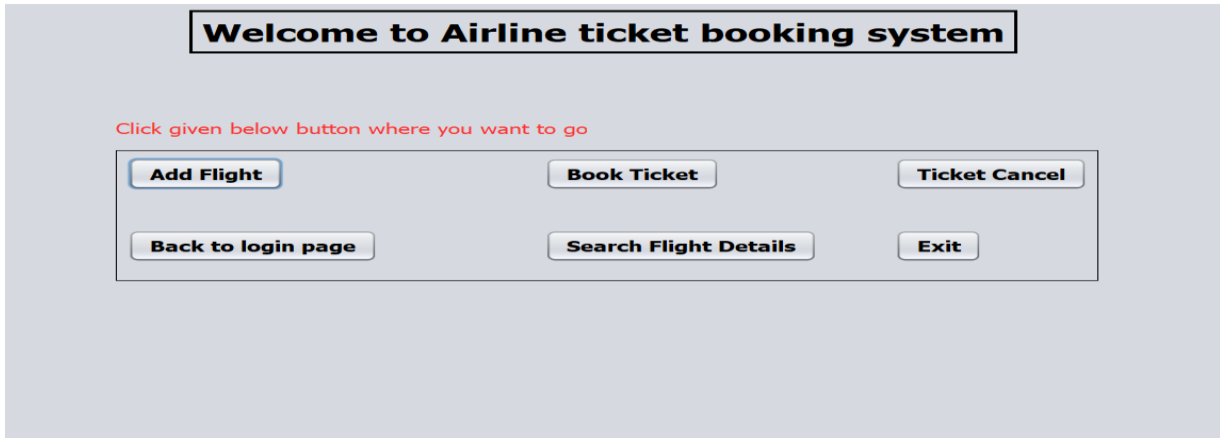
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Payment_Option().setVisible(true);
        }
    });
}

private javax.swing.JButton Credit_Card;
private javax.swing.JButton Debit_Card;
private javax.swing.JLabel jLabel1;
private javax.swing.JPanel jPanel1;
}

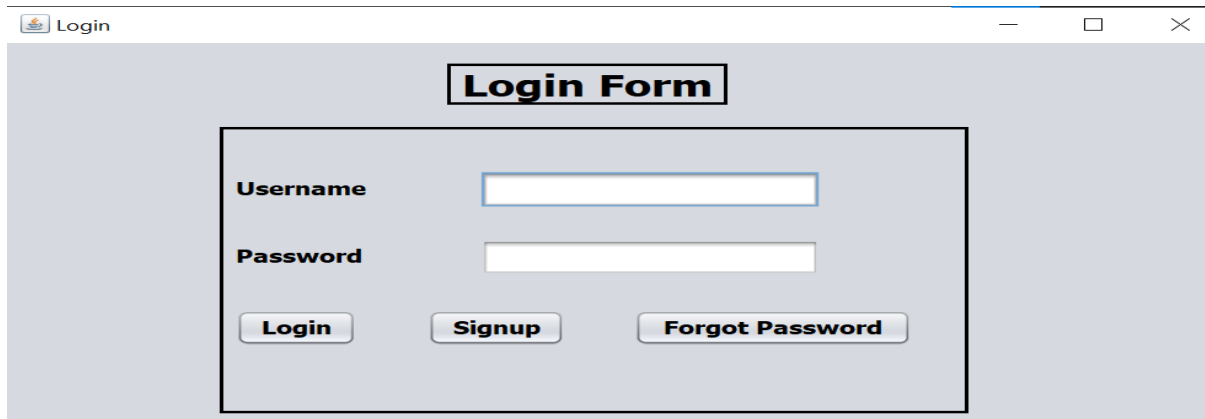
```

SCREENSHOTS

HOME PAGE



LOGIN PAGE



SIGNUP PAGE



FLIGHT TICKET BOOKING PAGE

Search Flight Details

Source: Destination: Date:

Flight_Id	Flight_Name	Source	Destination	Date	Arrival_Time	Departure_Time	Flight_Price
2304	IndiGo	Chennai	Hyderabad	2022-08-04	19.40	20.55	5000

Ticket Booking Form

Ticket No:

Flight Id:

Flight Name:

Source:

Destination:

Date:

Arrival Time:

Departure Time:

Price: Total Price:

Seats:

Customer Name:

PAYMENT PAGE

Credit Card

Enter Your Card Name:

Expiry / Validity Date:

Enter CVV:

FLIGHT TICKET CANCELLATION PAGE

Cancel Flight Ticket

Ticket No: Arrival Time:

Flight Id: Departure Time:

Flight Name: Price:

Source: Seats:

Destination: Customer Name:

Booking Date: Cancel date: