

TABLE OF CONTENTS

1. Introduction
 - About Project
 - Objectives
 - Purpose
 - Scope of the project
 - Project Overview
2. Problem Selection
 - Existing System
 - Proposed System
3. Project Monitoring System
 - Module Description
4. System Study
 - Feasibility Study
 - Operational Feasibility
 - Technical Feasibility
 - Behaviour Feasibility
5. System Analysis
 - SDLC
 - USE CASE DIAGRAM
 - Data Flow Diagram
 - 0 Level DFD
 - 1ST Level DFD
 - 2ND Level DFD
 - ERD
 - Flow Chart
6. System Design
7. System Testing and Implementation
 - Levels of Testing
 - Unit Testing or Module Testing
 - Integration Testing
 - Types of Testing
 - 1 Black Box Testing
 - 2 White Box Testing

1. INTRODUCTION

This project on Airline Management System is the automation of registration process of airline system. The system is able to provide much information like passenger's information, criminal's, list of all passengers etc. The system also allows us to add records when a passenger reserves a ticket. For data storage and retrieval we use the database. It enables us to add any number of records in our database. But for intrinsic nature of file handling, the retrieval process is slow when we Search a particular record in the database, because record is searched sequentially.

About Project

The project named "Airline Management System" is written in Java, mainly because of it's suitability for this type of application. It's user friendly nature and in-built documentation, complication, error detection, binding facilities and interaction with other software packages make it most powerful tool for software development. Moreover. Java consists of all the technologies that help in creating and running robust, scalable and distributed packages. JAVA is an object-oriented programming, and is intended to be an improved language with object capabilities

Assistance is provided to the user at each and every step so that no problem is faced during using it. Further the details of every process and the user manuals attached in the report make it very easy to understand. Every possible care has been taken to make the software and the report clear, simple and error free which makes it so special and one of its kind.

Objectives of the Project

- .To provide some amount of automation in airlines mangement.
- .To help airlines system in making their business more efficient.
- . An added attraction for their potential customers.
- . It will also show the attitude of the management that they are aware to the newly introduced technology and ready to adopt them.

Purpose of the project

Electronically handling of flight's record to enhance the accuracy, flexibility, reliability and to remove the human's error.

- An **airline** provides air transport services for passengers, generally with a recognized operating
- To provide accurate information about the addition, deletion and modification record.
- To provide, efficient, accurate, reliable, fast, and robust structure that can handle any number of passengers.

SCOPE OF THE PROJECT

This project on Airline Management System is the automation of registration process of airlines system. The system is able to provide much information like passenger's information, list of all passengers etc.

The system also allows us to add records when a passenger reserves a ticket.

For data storage and retrieval we use the database facility of Java Language. It enables us to add any number of records in our database. But for intrinsic nature of database, the retrieval process is slow when we search a particular record in the database, because record is searched sequentially.

Need of Computerisation

A few factors that directs us to develop a new system are given below -:

- 1) Faster System
- 2) Accuracy
- 3) Reliability
- 4) Informative
- 5) Reservations and cancellations from any where to any place

PROJECT OVERVIEW

- Database and database systems have become an essential component of everyday life in modern society. In the course of a day, most of us encounter several activities that involve some interaction with the database. For example, if we go to the bank to deposit or withdraw funds or if we make a Hotel or Airline Reservation, chances are that our activities will involve someone accessing a database.

- The above interactions are examples of what we may call traditional database applications, where most of the application that is stored and accessed is either textual or numeric. In our project we will concentrate on this aspect of computer application.
- There are several ways to implement databases. Some of them are file handling mechanism, relational database, object-relational database or object-oriented databases. In our project we will use database-handling feature provided by JAVA Language.
- This program shows you an insight into the management process of reservation in Airline Management system. The whole process of Airline Management System is shown with the help of this project. It provides facility to add/Modify/Delete/search Airline Management details. and provide facility to view the list of Team .
- Facility to view the list of Team .

2. PROBLEM SELECTION

1. Before making this application, we assumed that an airline which had recently started its operation found it very difficult to handle their customers.
2. It was due to their great customer service and efficient handling of daily operations that their customer base started growing and in a day, they started to handle lot of customer requests. The problem is that in manual airline record keeping system, excessive staff employment is required, extremely time consuming process is involved, inconveniences to both customers as well as to the manager.
3. Slowly & slowly the count of such customers started to grow very rapidly and the airline employees had to devote their maximum time in handling such customers.
4. Slowly, an airline started losing its important or gold customers due to poor response times by the employees and they even started losing those customers whose requests could not be fulfilled.
5. After this, the management decided to install a system that can effectively & efficiently service the request of such customers and can take the corresponding work of its employees who were overburdened with such tasks.
6. This action was a step towards serving important or fresh customers with a minimum possible and improve the response times & efficiency of an airline employees.

Objective of this software is to simplify the employee record using computers

Existing System:

The system is very time consuming and lazy. This system is more prone to errors and sometimes the approach to various problems is unstructured.

If any old data or information is to be fetched then it is a great problem for user to get the information in short span of time as to get information from files is not an easy task.

As everything is done manually, so if any record is misplaced then agency has to take full responsibility.

Limitation of existing system:

The earlier experiences have shown that manual monitoring of employee enquiries about their loans, conveyance, etc. Often fails to achieve the desired targets, mainly because of the following reasons:

- Much time required in giving correct information.
- Less reliability and maintainability of data.
- Secrecy of information may not be maintained due to visible facts on paper.
- Manual procedure of providing information is not reliable.

Every manager faces lot of minor & major problems like:

- Maintaining database.
- Record entry.
- Searching duplicate records.
- Searching & updating records.

An object oriented system draws upon class definitions that are derived from the analysis model. Some of the definitions will have to be built from scratch but many other can be reused if appropriate design patterns are recognized. Object oriented design establishes a design blueprint that enables a software engineer to the object oriented architecture in a manner that maximized reuse, thereby improving development speed and product quality.

The four layers of object oriented design are:

- **The Subsystem Layer:** It represents each of the sub systems It represents each of the subsystems that enable the software to achieve its user-defined requirements and to implement the technical infrastructure.
- **The Class and Object Layer:** It contains the class hierarchies that enable the system to be created using generalizations and increasingly more targeted specializations.
- **The Message Layer:** It contains the design details that enable each object to communicate with its collaborators. This layer establishes internal and external interfaces for the software.

- **The Responsibility Layer:** It contains the data structure and algorithmic design for all attributes and operations for each object. Monitoring system activity and server performance is a necessary part of preventive maintenance for the server. Through monitoring, you obtain data that u can use to diagnose system problems, plan growth and trouble shoot problems. You can use the monitoring and status tool, diagnostic logging, extended logging and queue viewer to keep the data up-to-date.

Proposed System:

The proposed system is computer based, user friendly, and easy to maintain. It makes safely storing of records easy and for a very long period of time. It would significantly improve the quality of work in the airport. The time spent in processing the above mentioned queries would significantly reduce. The proposed system provides free, easy and efficient management of the day-to-day activities of the passenger's in airline so that the manual work can be reduced and even minute details can be accessed easily.

3. PROJECT MONITORING SYSTEM

Module Description

AIRLINE MANAGEMENT SYSTEM is basically a menu driven program. This kind of format is designed considering the user requirements. This is to provide an easy and faster method of operation to the user.

We have implemented validation at some points in the system as:

1. The first point is where the user enters his credentials. This is the most important part of our application because the information which would be fetched & is displayed to the user is confidential and it should be displayed only after proper authentication. So, for security reasons, we have given chance to the user to enter his details correctly. If he does not, the system issues a warning through a proper message and exits. The user then again has to start from beginning, enter his details.
2. The second point where this is implemented is the menu where the user chooses from a list of options to process his requests. Since this is a menu-driven program, we expect from the user to input correct option. But if, for some reason, the user is unable to enter it correctly, we flash a message which requests the user to enter a correct option.
3. The third point where we have taken care of user input is the place where user wants to use any kind of service. Ideally, the user should not enter the unknown details or information about themselves and if mistakenly he does, the system flashes a user message and inform him about the same.

4. SYSTEM STUDY

FEASIBILITY STUDY

A feasibility study is carried out to select the best system that meets performance requirements.

Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called a feasibility study. This type of study determines if a project can and should be taken.

Since the feasibility study may lead to the commitment of large resources, it becomes necessary that it should be conducted competently and that no fundamental errors of judgment are made.

Depending on the results of the initial investigation, the survey is expanded to a more detailed feasibility study. Feasibility study is a test of system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources.

The objective of the feasibility study is not to solve the problem but to acquire a sense of its scope . During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined.

Consequently, costs and benefits are described with greater accuracy at this stage.

It consists of the following:

- 1. Statement of the problem:** A carefully worded statement of the problem that led to analysis.
- 2. Summary of finding and recommendations:** A list of the major findings and recommendations of the study. It is ideal for the user who requires quick access to the results of the analysis of the system under study. Conclusion are stated , followed by a list of the recommendation and a justification for them .

3. Details of findings : An outline of the methods and procedures under-taken by the existing system, followed by coverage of the objectives and procedures of the candidate system. Included are also discussions of output reports, file structures, and costs and benefits of the candidate system.

4. Recommendations and conclusions: Specific recommendations regarding the candidate system, including personnel assignments, costs, project schedules, and target dates.

Three key considerations are involved in the feasibility analysis these are

1. Operational Feasibility
 2. Technical Feasibility
 3. Behavioral Feasibility
-

Operational Feasibility:

Operational analysis is the most frequently used method for evaluating the effectiveness of a system. More commonly known as cost/ benefit analysis, the procedure is to determine the benefits and savings that are expected from a system and compare them with cost.

Earlier in Computer Craft the work has been done manually which takes lot of time as well as man power which is more economical. Now the same work is computerized which is more effective and efficient, less time consuming, reduces man power which in turn proves to be less economical.

Technical Feasibility:

Technical Feasibility centers around the existing computer system (hardware/ software) and also it can support the modification.

In manual processing there are more chance of errors are there, creating lot of complications, less technical or logical.

Through proposed system we can set this process in a very systematic pattern, which is more technical, full proof, authentic, safe and reliable.

Behavior Feasibility:

The proposed system works to minimize the human errors, take less time, easy interaction with user, bug free.

This project/software is further expanded by connecting various interrelated departments and by installing an extension part of this software.

- System level goals and requirements.
- Cost estimation for development process and work product.
- Solution strategy development.
- Outlines of the several solutions strategies.
- Recommendation of solutions strategy.
- Feasibility and study of each strategy.

5. SYSTEM ANALYSIS

The analysis model must achieve three primary objectives:-

-
1. To describe the requirements of the customer.
 2. To establish a basis for the creation of a software design.
 3. To define a set of requirements that can be validated once software is built.
-

An Overview to system analysis

The system analysis phase is considered to be one of the most important phases in the system development life cycle. It is immensely important that the software developer make through study of the existing system. Thorough study of the system is made and need i.e. features that are critical to system success and users wants (i.e. features that would be good but not essential) are brought out. The study will enable the developer to know the intricacies of the existing system.

Requirement analysis is done in order to understand the problem which the S/W system is to solve e.g., the problem could be automating the existing manual system or developing a completely new automated system or a combination of the two. For large systems having a large number of features and the need to perform many different tasks, understanding the requirement of the system is a major task. The emphasis in requirement analysis is on identifying what is needed from the system, and not how the system achieves its goal.

The main objective behind any business organization is to maximize its profit besides maintaining quality and strategic norms. This can be achieved by improving the efficiency of the system by providing more facilities using automation, by adopting faster data access, proper communication. , whereas the main objective behind automation is not only to maximize profit but also to take care of passenger's interest by providing them better facilities.

The most important objective behind automation is to minimize Paper Work. Paper Work/Registers are replaced by a Database which is well equipped to store / provide information as and when required. Database also helps speed up the communication between various depts. outside agencies, as there is no need of making request against

different departments for a specific data and to wait for it for a long period. This also improves the efficiency as it saves time and human resources.

By making the manual system computerized, we can ensure complete utilization of our existing resources. Automation helps in generating the reports / information in a consistent way, which saves time and labour if done manually.

In this project we have used Rapid Application Development (RAD) model. RAD is an incremental software development process model that emphasizes an extremely short development cycle. The following phases are encompassed:

- **Business modeling:** All the information about the business functioning of the Airways department is collected, how the data and information is flow from one end to another end.
- **Data modeling:** The information collected in Business modeling phase is refined into a set of data objects that are needed to support the project. The attributes of each object are identified and the relationships between these objects defined.
- **Process modeling:** Processing descriptions and functions like adding, modifying, deleting records, printing reports, providing information etc. are created.

➤ **Testing:** Most of the functions are already tested, as they are predefined functions. However, new components or functions are also tested after application generation.

It is the interdisciplinary part of science, dealing with analysis of sets of interacting entities, the systems often prior to their automation as computer systems, and the interactions within those systems.

The selection process should be viewed as a project and a project team should be formed with the help of management. The selection process consists of several steps, which are discussed below:

- **Requirements analysis:** The first step in selection understands the user's requirement within the framework of the organization's objectives and the environment in which the system is being installed.
- **System specifications:** System specifications must be clearly defined. These specifications must reflect the actual applications to be handled by the system and

include system objectives, flowcharts, input-output requirements, file structure and cost.

- **Evaluation and validation:** The evaluation phase ranks various vendor proposals and determines the one best suited to the user's requirements. It looks into items such as price, availability and technical support. System validation ensures that the vendor can, in fact, match his/her claims, especially system performance.

WORKING OF THE PROJECT

- User can view record about flight by selecting option from the main menu.
- User can reserve the seat for view the flight.
- User can also cancel the reserved ticket for flight.

SDLC

In this project we have followed the Waterfall model.

The waterfall model is the most familiar model. This model has five phases: requirements analysis and specifications, design, implementation and unit testing, integration and system testing, and operation and maintenance.

- 1.Requirements Analysis and Specification Phase:** The goal of this phase is to understand the exact requirements of the customer and to document them properly. This activity is usually executed together with the customer, as the goal is to document all functions, performance and interfacing requirements for the software.
- 2. Design phase:** The goal of this is to transform the requirements specification into a structure that is suitable for implementation in some programming language.
- 3. Implementation and Unit Testing Phase:** During testing, the major activities are centered around the examination and modification of the code. Initially, small modules are tested in isolation from the rest of the software product. There are problems associated with testing a module in isolation.

4. Integration and System Testing Phase: The purpose of unit testing is to determine that each independent module is correctly implemented. This gives little chance to determine that the interface between modules is also correct, and for this reason integration testing is performed. System testing involves the testing of the entire system whereas software is a part of the system. This is essential to build confidence in the developers before software is delivered to the customer or released in the market.

5. Operation and Maintenance Phase: Software maintenance is a task that every development group has to face, when the software is delivered to the customer's site, installed and is operational. Therefore, release of software inaugurates the operation and maintenance phase of the life cycle. The time spent and effort required to keep the software operational after release is very significant.

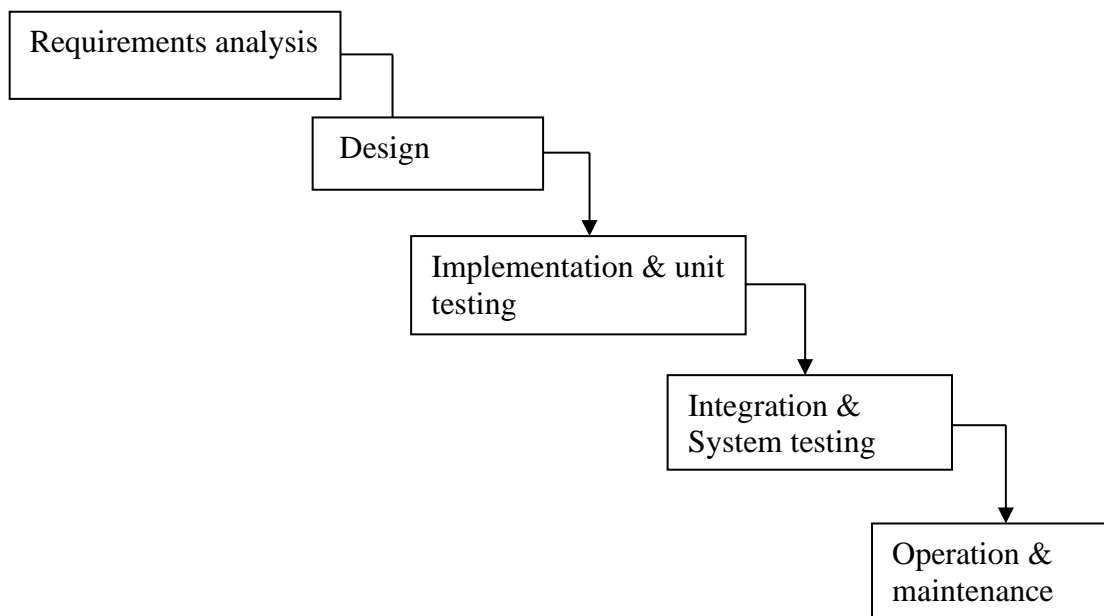


Fig. 1

USE CASE DIAGRAM

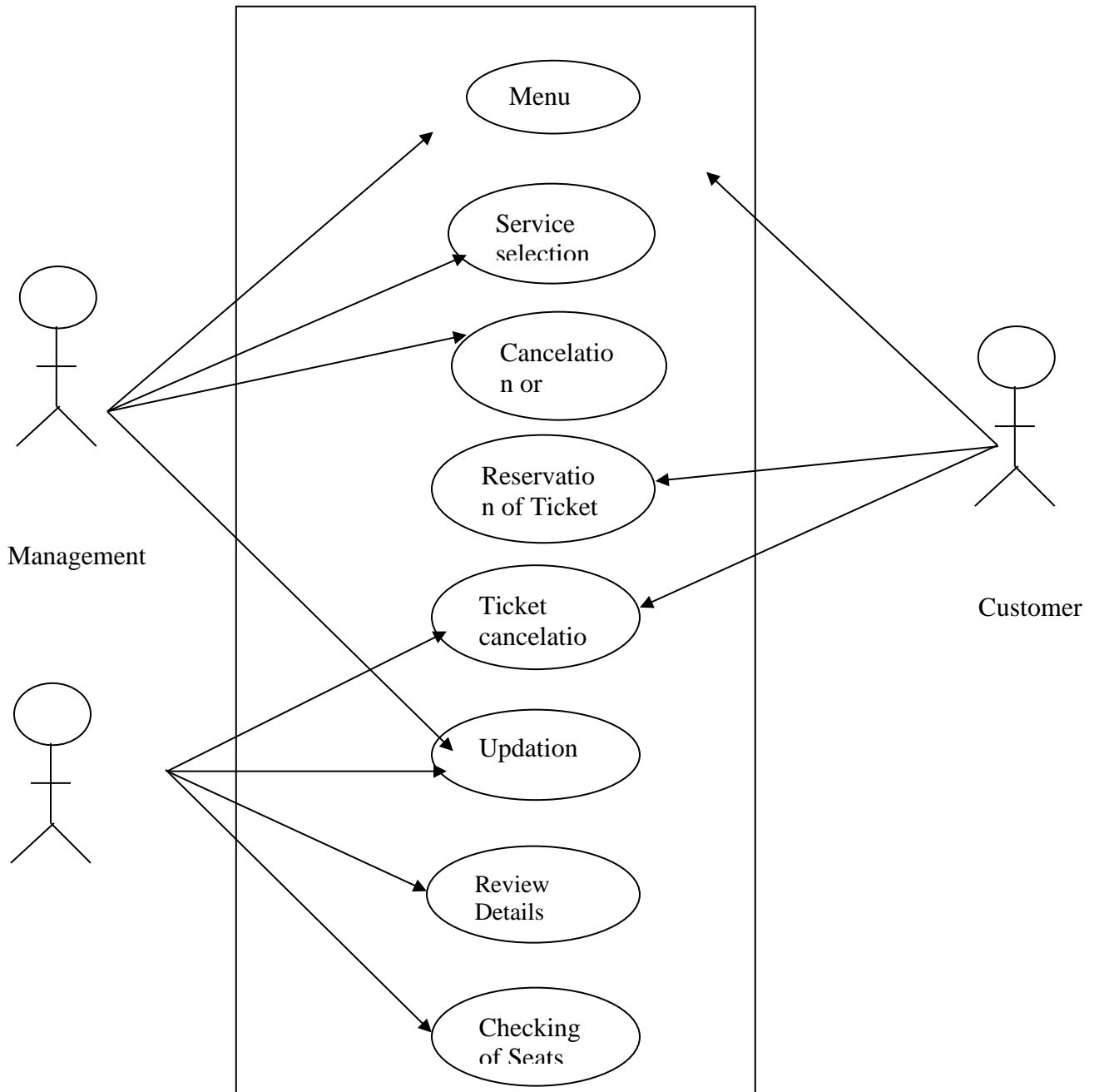
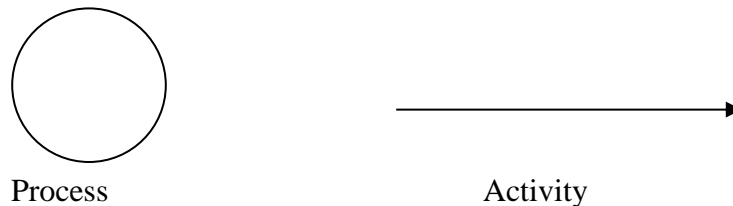


Fig. 2

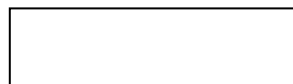
DATA FLOW DIAGRAM

Data flow diagrams are commonly used during problem analysis. Data flow diagrams are quite general and not limited to problem analysis for software requirement specification. A DFD shows the flow of data through a system. It views a system a function that transforms the inputs into desired outputs. Any complex system does not perform this transformation into a single step and a data will typically undergo a series of transformation before it becomes an output. The DFD aims to capture the transformations that take place within a system to the input data so that eventually the output data is produced.

The agent that performs the transformation of data from one state to another is called a process. So, a DFD shows the movement of data through the different transformations or processes in the system. Named circles show the processes and data named arrows entering or leaving the bubbles represent flows.



The rectangle represents a source and sink and is a net originator or consumer of data. A source or sink is typically outside the main system of study.



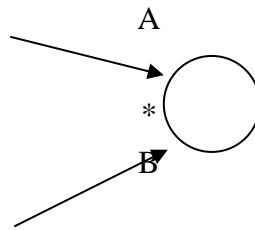
Originator or Consumer of data

All external files are shown as a labeled straight line.

File name



The need of multiple data flows by a process is represented by a “*” between the data flows. the symbol represents the AND relationship. for example, if there is a “*” between the two input data flows A and B for a process, it means that A AND B are needed for the process.



0 Level DFD

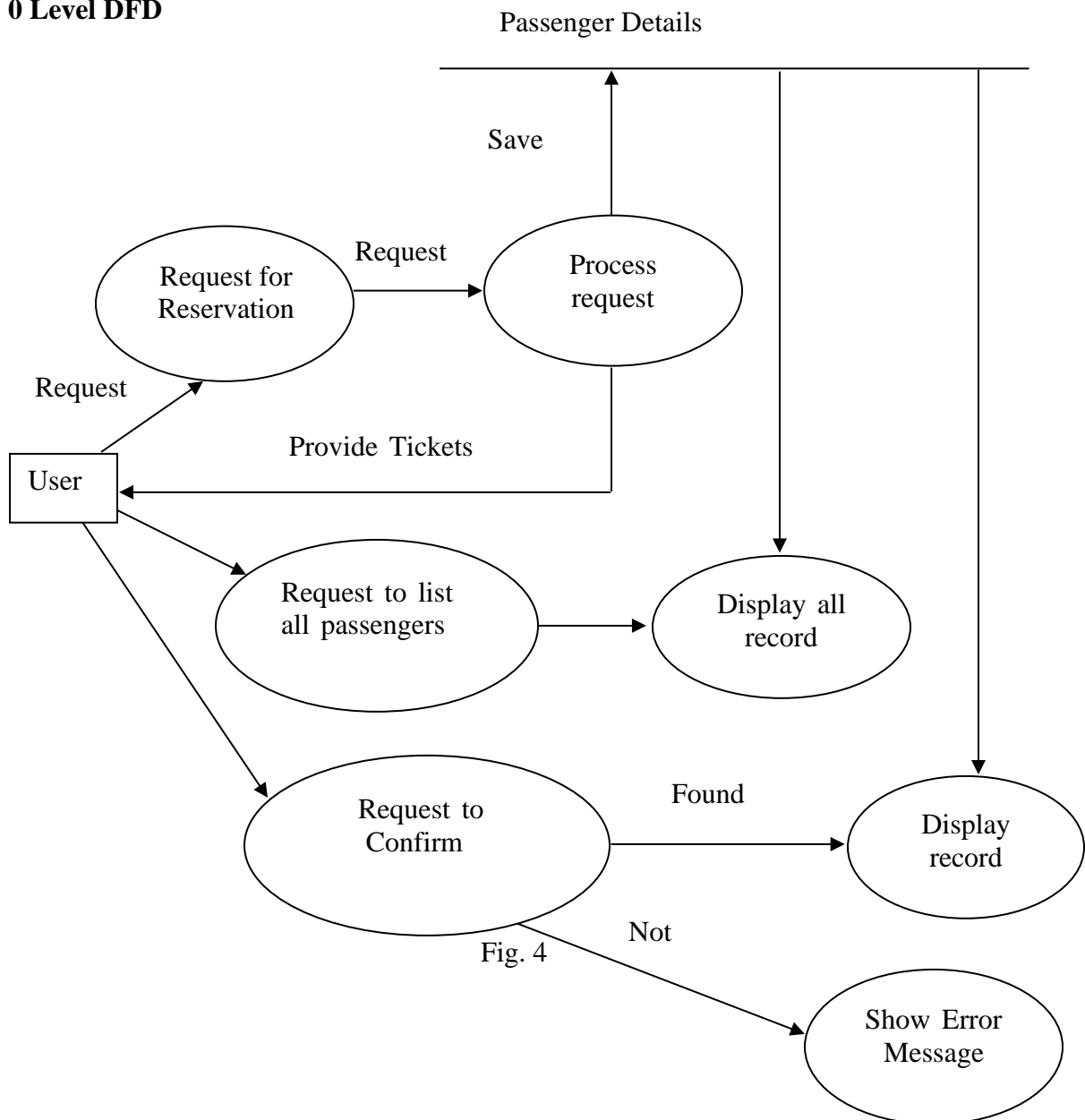


Fig. 4

1st Level DFD

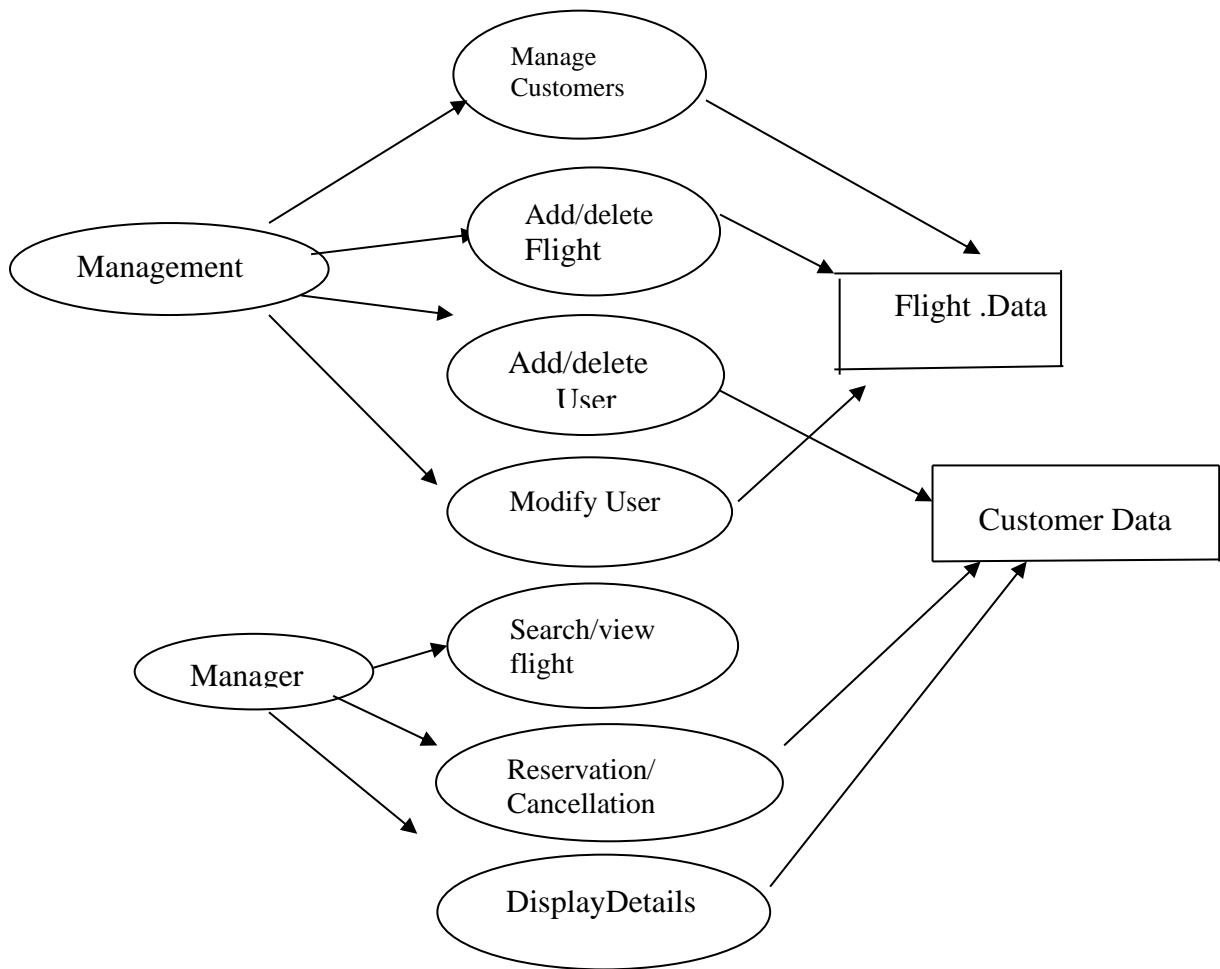
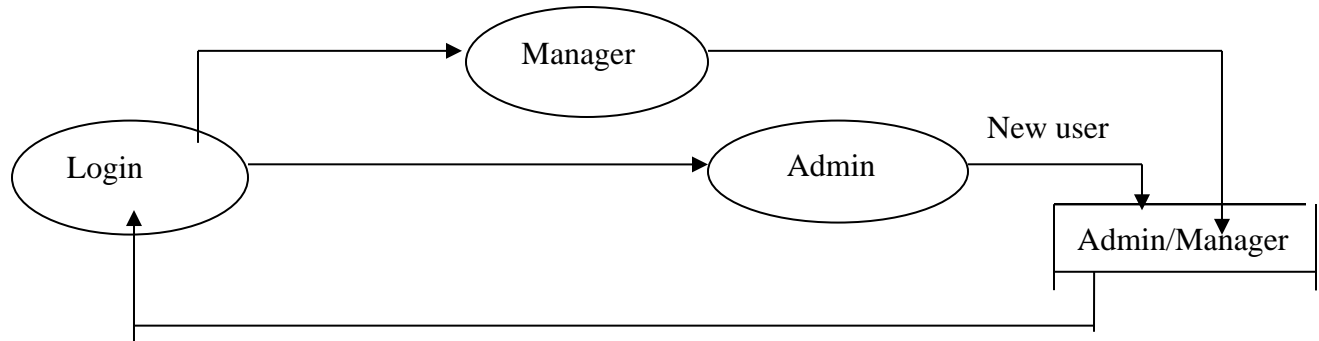


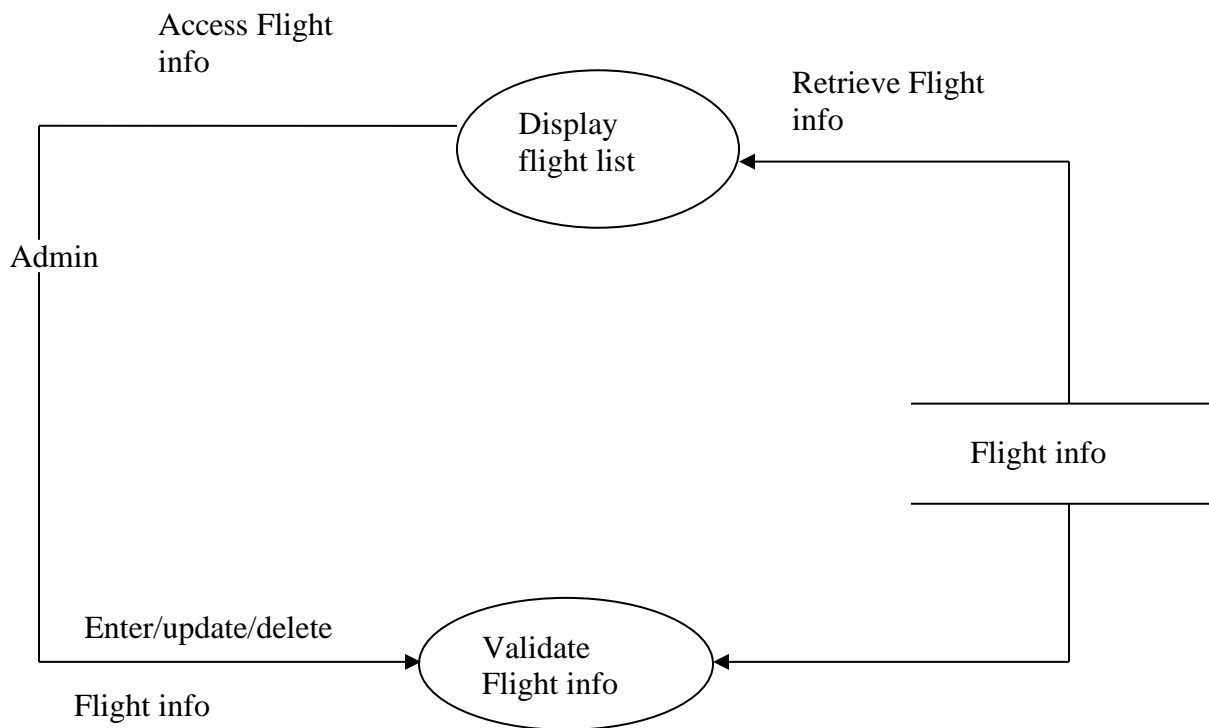
Fig. 5.2 Reservation

2nd Level DFD

1. LOGIN



2. TICKETS



3. CUSTOMER

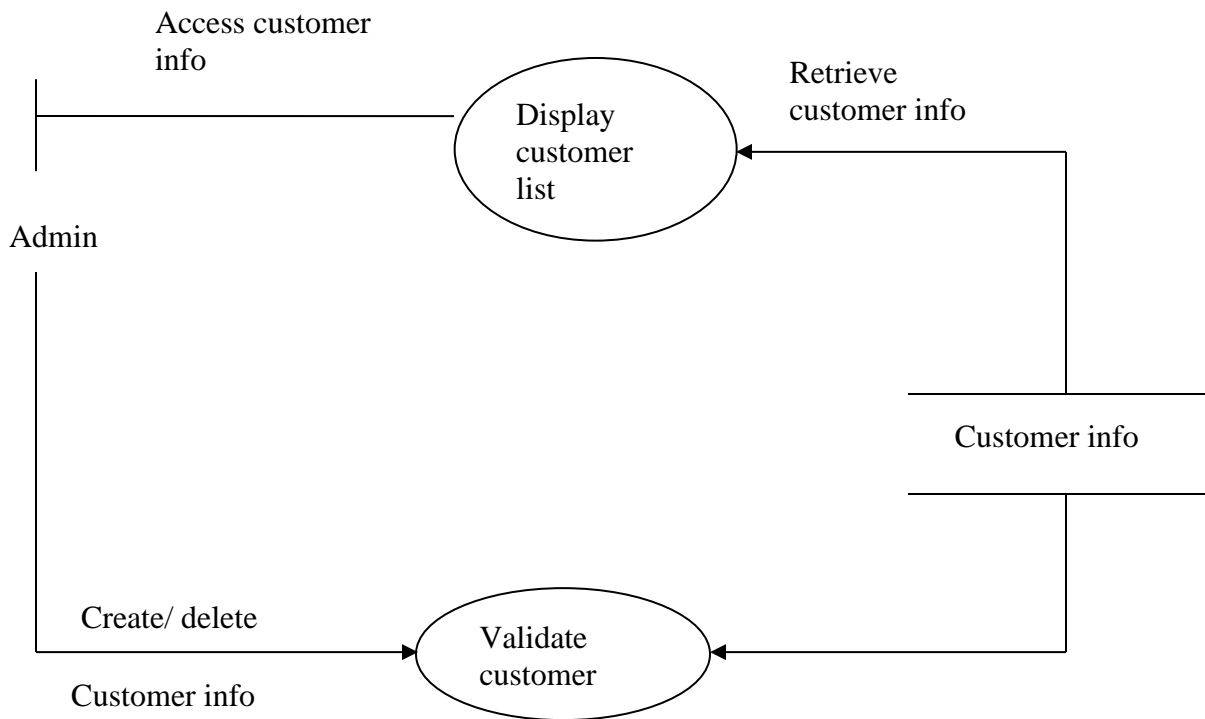
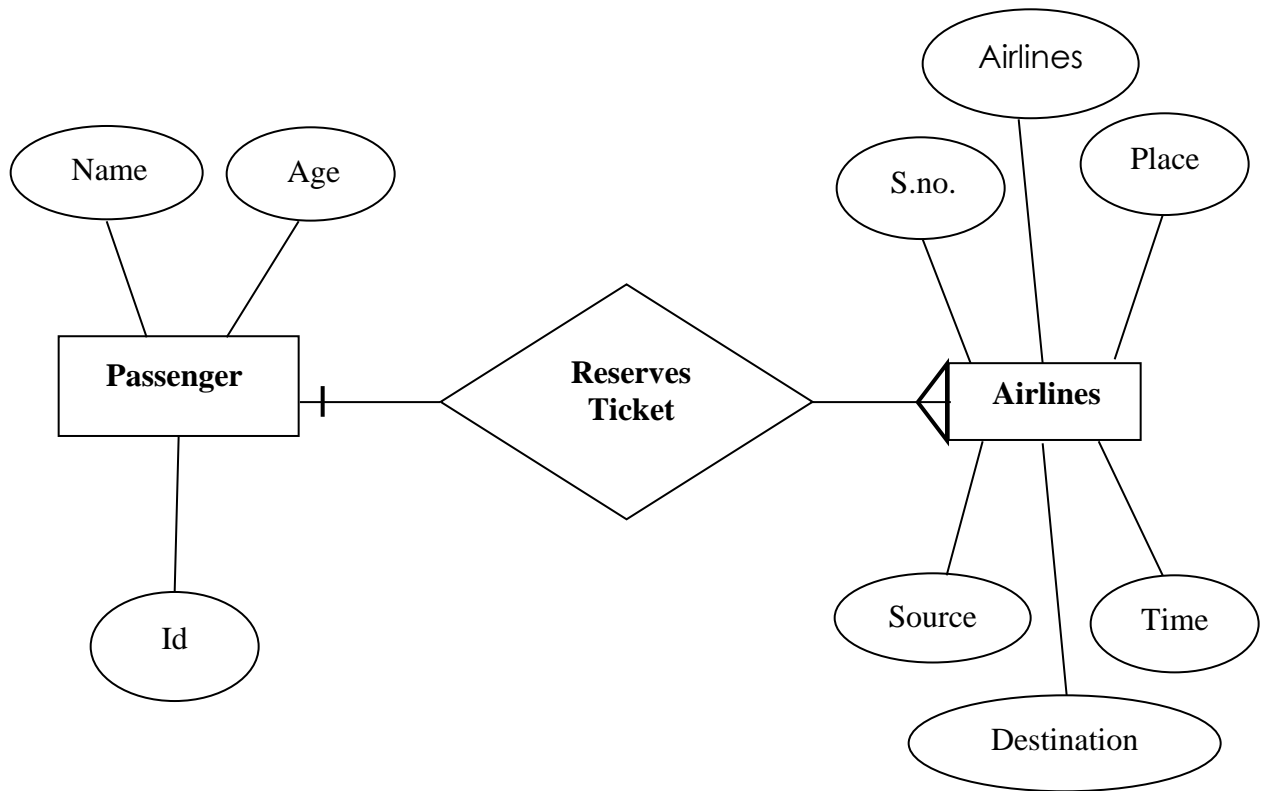


Fig. 5.3 level DFD

ER DIAGRAM

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes. If the application is primarily a database application, the entity-relationship approach can be used effectively for modeling some parts of the problem. The main focus in ER modeling is the Data Items in the system and the relationship between them. It aims to create conceptual scheme for the Data from the user's perspective. The model thus created is independent of any database model. The ER models are frequently represented as ER diagram. Here we present the ER diagram of the above mentioned project.

E-R DIAGRAM

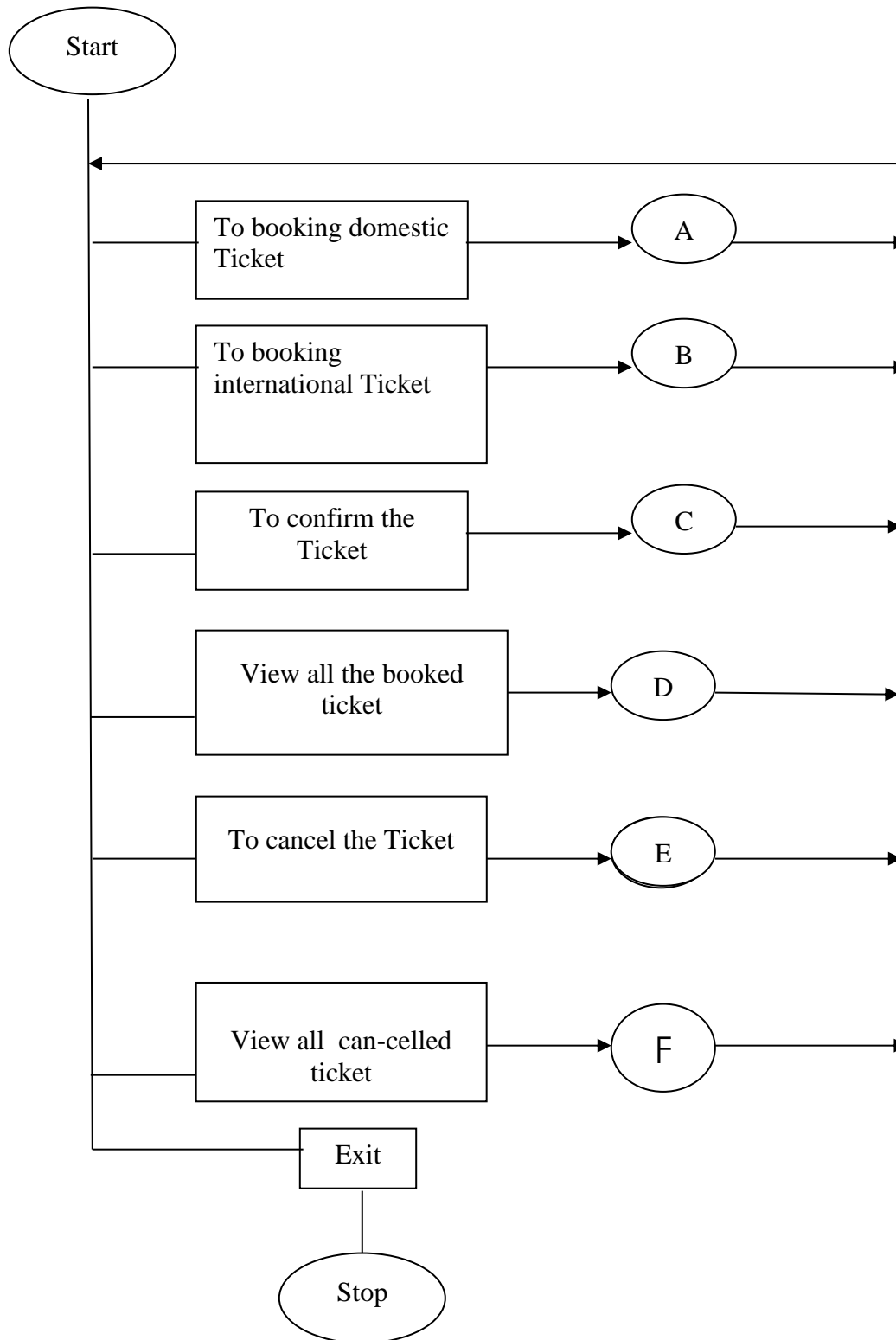


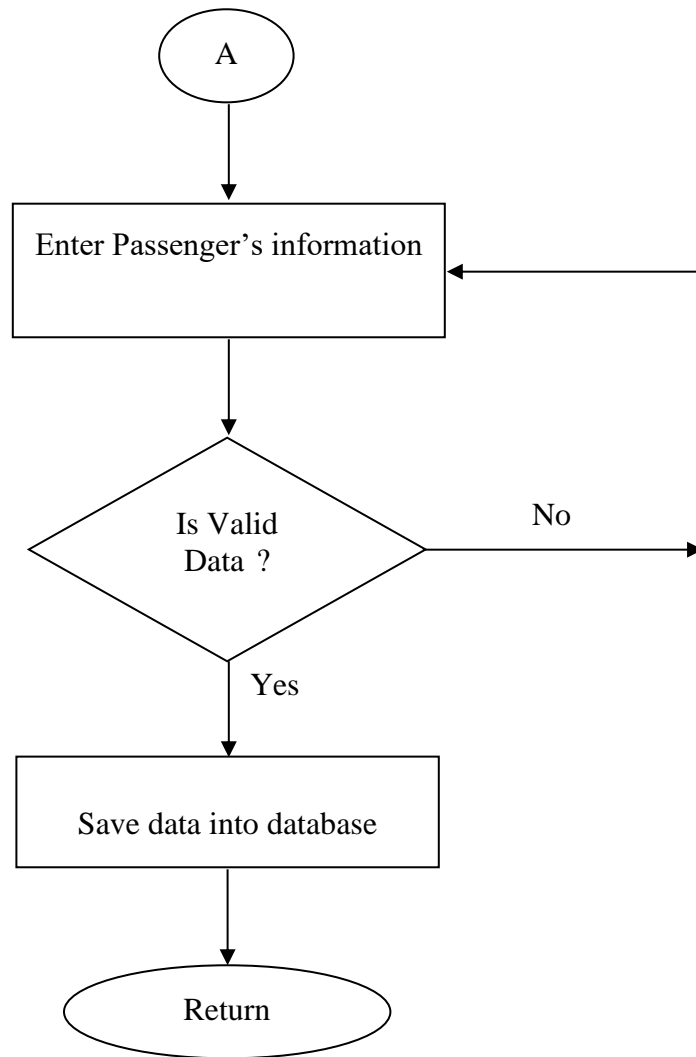
FLOWCHART

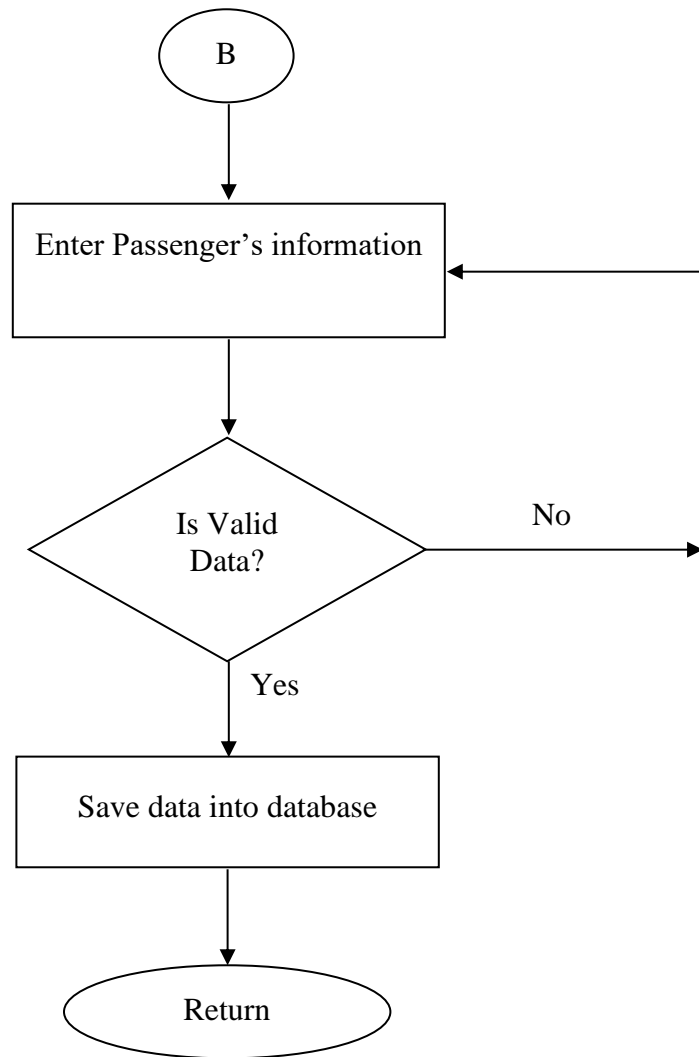
In procedural language program is started with the first line and follow a pre-define Path. Flow chart is used to define that pre-defined path and it show the flow of control throughout the program. The flow charts are used in programming for purpose of indicating the sequence of Operation of Program. It is very useful tool available for the programmer to generate method of writing the program and statement of program. It creates sequence of operations and indicated transfer of control in an effective manner. The flow charts use symbol's or blocks of different shapes for representing statement of program.

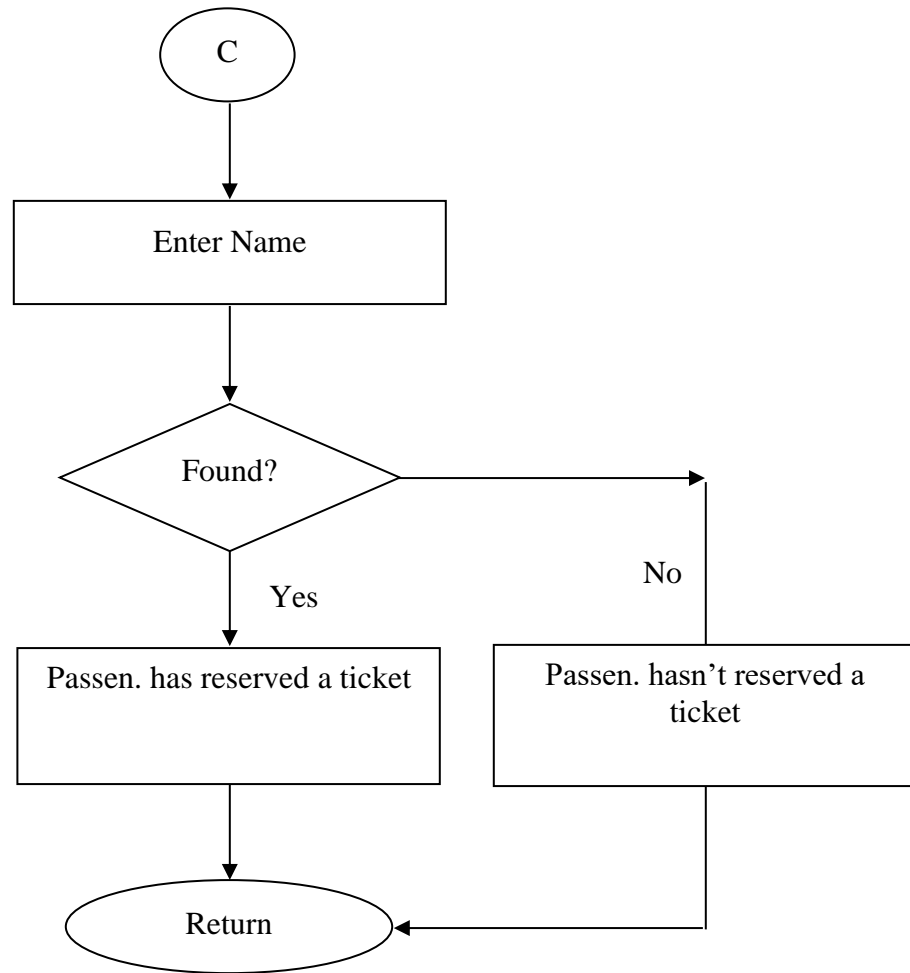
A flowchart is a common type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting these with arrows. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

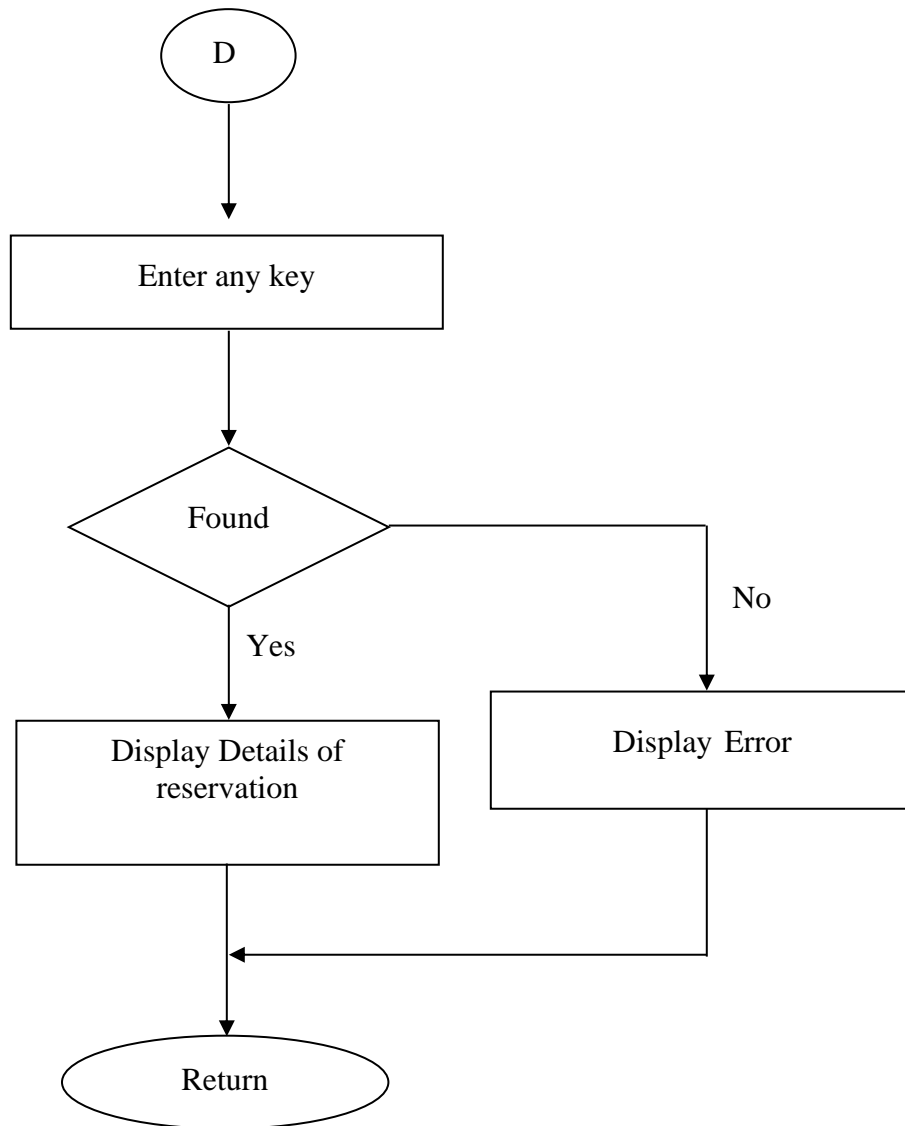
FLOW CHARTS

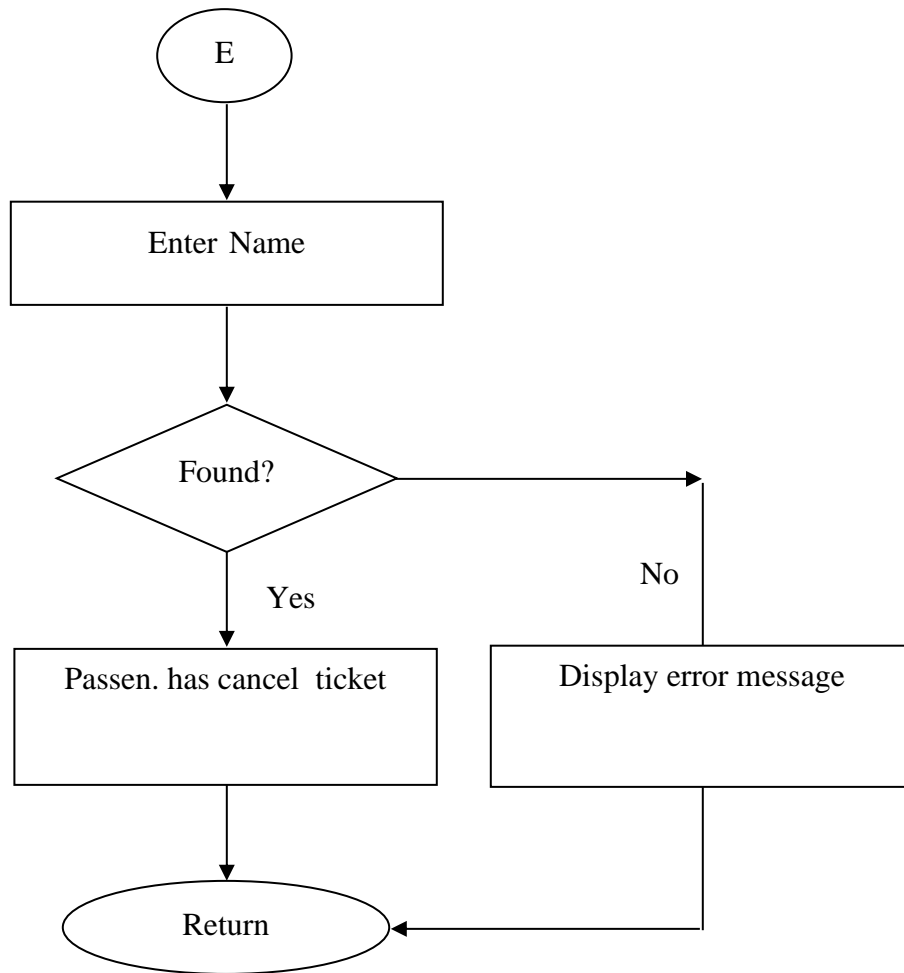


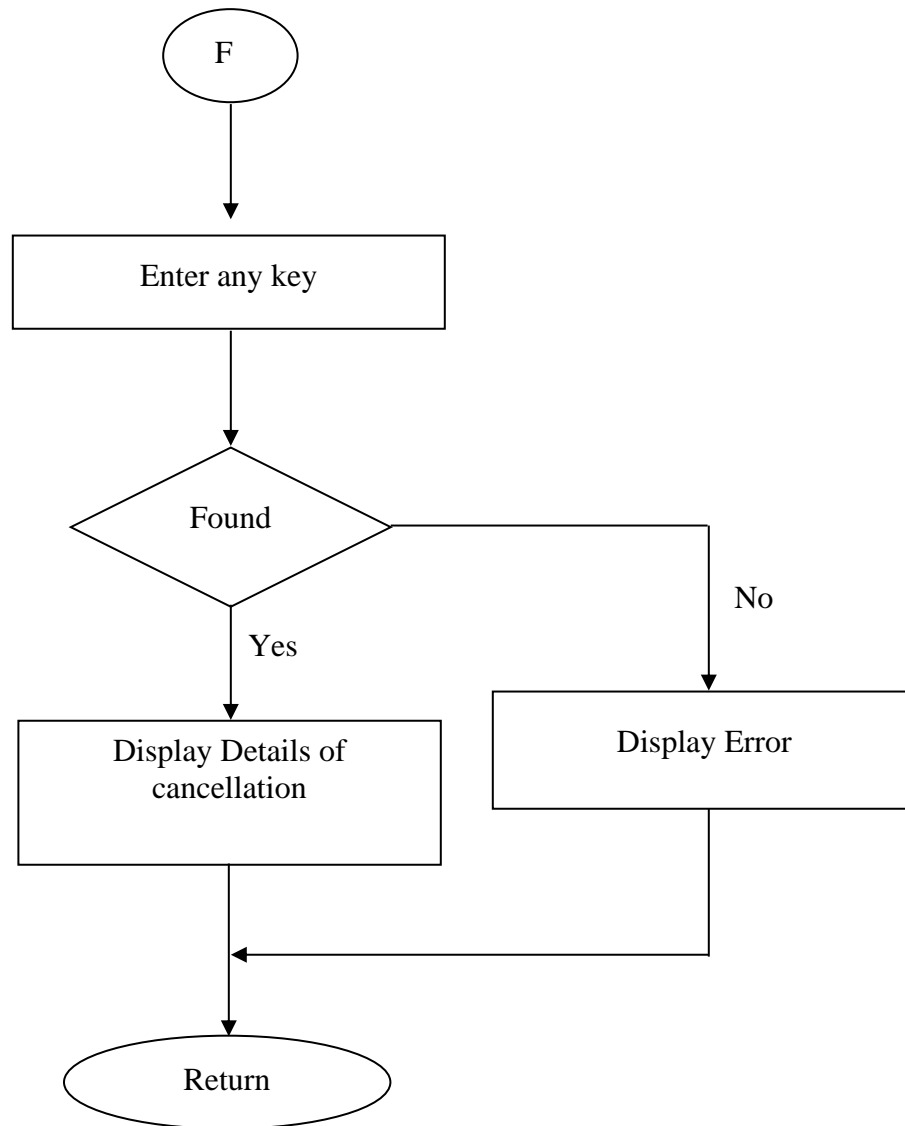












6. SYSTEM DESIGN

System Design Considerations:

The system design process is not a step-by-step adherence of clear procedures and guidelines. Though, certain clear procedures and guidelines have emerged in recent days, but still much of design work depends on knowledge and experience of the designer.

The primary objective of the design: Of course, is to deliver the requirements as specified in the feasibility report. In general, the following design objectives should be kept in mind:

- **Practicality:** The system must be stable and can be operated by people with average
- **Efficiency:** This involves accuracy, timeliness and comprehensiveness of the system output.
- **Cost:** it is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy all the requirements.
- **Flexibility:** The system should be modifiable depending on the changing needs of the user. Such modifications should not entail extensive reconstructing or recreation of software. It should also be portable to different computer systems.
- **Security:** This is very important aspect of the design and should cover areas of hardware reliability, fall back procedures, physical security of data and provision for detection of fraud and abuse.

The designer normally will work under following constraints:

- **Hardware:** The existing hardware will obviously affect the system design.
- **Software:** The available software (operating system, utilities, language etc.) in the market will constrain the design.
- **Budget:** The budget allocated for the project will affect the scope and depth of design.
- **Time-scale:** The new system may be required by a particular time (e.g. the start of a financial year). This may put a constraint on the designer to find the best design.
- **Interface with other systems:** The new system may require some data from another computerized system or may provide data to another system in which case the files must be compatible in format and the system must operate with a certain processing cycle.

DESIGN METHODOLOGIES

- Improve productivity of analysts and programmers
- Improve documentation and subsequent maintenance and enhancements.
- Cut down drastically on cost overruns and delays
- Improve communication among the user, analyst, designer, and programmer.
- Standardize the approach to analysis and design
- Simplify design by segmentation.

STRATEGY OF DESIGN

A good system design strategy is to organize the program modules in such a way that are easy to develop and later to, change. Structured design techniques help developers to deal with the size and complexity of programs. Analysts create instructions for the developers about how code should be written and how pieces of code should fit together to form a program.

BOTTOM UP DESIGN

These approach lead to a design where we decide how to combine these modules to provide larger ones; to combine those to provide a larger ones, and so on, till we arrive at one big module which is the whole of the desired program. The set of these modules form a hierarchy. This is a cross-linked tree structure in which each module is subordinate to those in which it is used.

Since the design progressed from bottom layer upwards, the method is called bottom-up design. This method has one terrible weakness; we need to use a lot of intuition to design exactly what functionality a module should provide. If we get it wrong, then at higher level, we will find that it is not as per requirements; then we have to redesign at a lower level.

TOP- DOWN DESIGN

A top design approach starts by identifying the major modules of the system, decomposing them into lower level and iterating until the desired level of detail is achieved. This is a stepwise refinement; starting from an abstract design, in each step the design is refined to a more concrete level, until we reach a level where no refinement is needed and the design can be implemented directly.

7. SYSTEM TESTING AND IMPLEMENTATION

Software testing is the process of executing a program with the intention of finding errors in the code. It is the process of exercising or evaluating a system or system component by manual or by automatic means to verify that it satisfies specified requirements or to identify differences between expected and actual results.

The objective of testing is to show incorrectness and testing is considered to succeed when an error is detected. An error is a conceptual mistake made by either the programmer or the designer or a discrepancy between a computed value and a theoretically correct value. A fault is a specific manifestation of an error. An error may be cause of several faults. A failure is the inability of a system or component to perform its required function within the specified limits. A failure may be produced when a fault is executed or exercised.

Other activities that are often associated with software are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles and test coverage information.

Levels of testing

Unit Testing or Module Testing

The starting point of testing is Unit testing. In this, a module is tested separately at each step. This helps to detect syntax and logical errors in the program and is performed by the coder himself /herself during coding.

Integration Testing

The modules, which are tested in the Unit Testing, are integrated to build the overall system. It is observed that many errors crop up when the modules are joined together. Integration testing uncovers these errors while integrating the modules. It helps in establishing confidence (correctness) in the complete, assembled system.

Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

Integration testing concentrates entirely on module interactions, assuming that the details within each module are accurate. Module and Integration testing can be combined, verifying the details of each module's implementation in an integration context. Many projects compromise, combining module testing with the lowest level of subsystem integration testing, and then performing pure integration testing at higher levels. Each of these views of integration testing may be appropriate for any given project, so an integration testing method should be flexible enough to accommodate them all.

Types of testing

Black Box Testing

It is also known as Functional Testing. It tests the overall functional requirements of product. Inputs are supplied to product and outputs are verified. If the outputs obtained are the same as the expected ones then the product meets the functional requirements. In this, the internal procedures are not considered. In this the tester would only know the "legal" inputs and what the expected outputs should be, but not how the program actually arrives at those outputs. This Testing is more effective on larger units of code. In this test's are done from user point of view.

White Box Testing

It is also known as Structure Testing. It focuses on the internal functioning of the product. It tests the loops of the Procedure, Decision points, Execution paths etc.

White box testing uses specific knowledge of programming code to examine outputs. The test is accurate only if the tester knows what the program is supposed to do. He or she can then see if the program diverges from its intended goal. White box testing does not account for errors caused by omission, and all visible code must also be readable. As the knowledge of internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively. The other advantage of white box testing is that it helps in optimizing the code. It helps in removing the extra lines of code, which can bring in hidden defects.

IMPLEMENTATION

The application can be uploaded in the AIRLINE MANAGEMENT SYSTEM. To access it, the user will just require running the executable database of the software. System must have Java driver. Basically the application is for the recording of the Star Sport's records. As implementation of AIRLINE MANAGEMENT System software fully automate the existing system. In the designed system implementation was done to replace a manual system with the computerized one. The objective was to put the tested system in to operation. Critical aspects of conversion are not disrupting the functioning of the organization. This phase gives us the clear pictures of our new system and all the points that have been carefully looked in when designing the computerized system.

REFERENCE BOOKS

"The Evolution of the Airline Industry", ISBN 0-8157-5843-X. Cf. p. 61–62.

The Project Casebook, PJ DeMarco, ISBN 0-13-161357-5.

Agile Estimating and Planning, Mike Cohn, ISBN 0-13-147941-1.