PROJECT REPORT ON "Travel Website"



In the fulfillment of the Requirement for the Degree of <u>Bachelor of Computer Application (B.C.A.)</u> <u>Submitted By</u>

Mr. Vijay Raju Kale

To

G. H. Raisoni Institute of Business Management, Jalgaon

Affiliated To

Kavayitri Bahinabai Chaudhari, North Maharashtra University, Jalgaon

2020-2021

ACKNOWLEDGEMENT

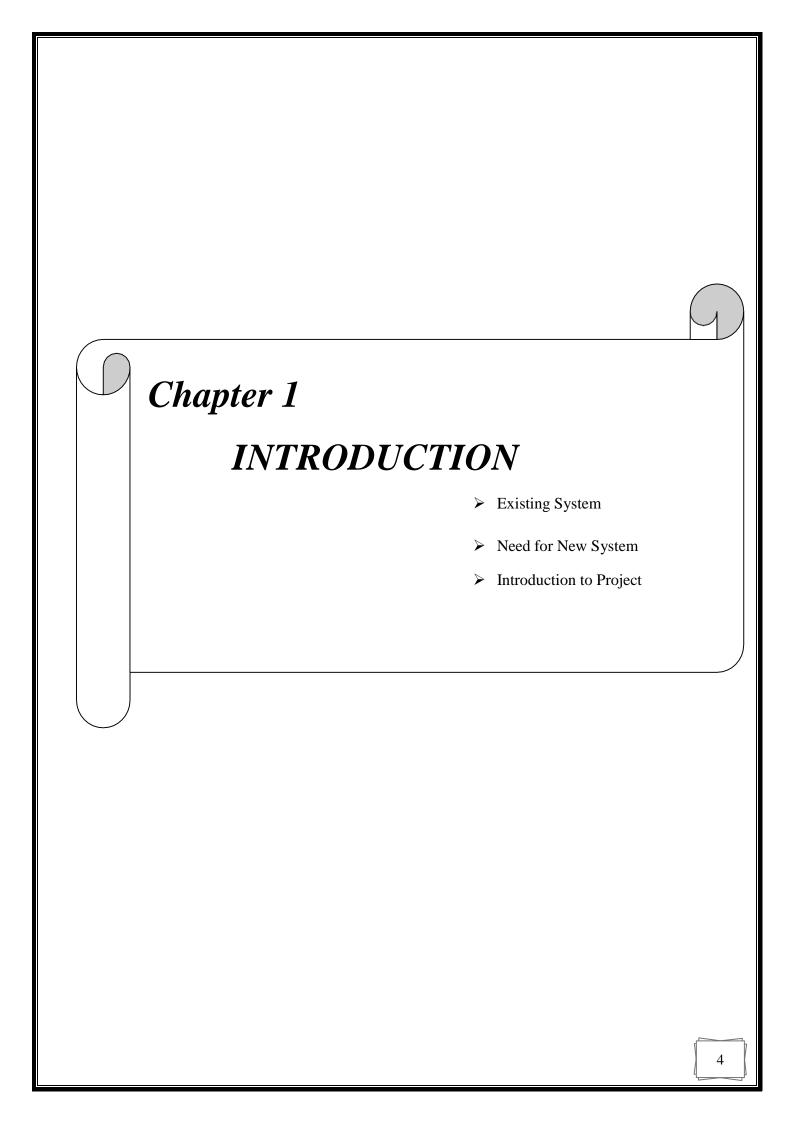
A successful project is the result of a good team-work which consists of not only the partners who put in their long and hard work but also those who guided them. Indeed, a true saying.

I would like to thanks to Hon. Dr. Preeti Agrawal and Hon. HOD Mr. Rafik Shaikh Sir and our Project guide Prof. Gaurav Jain sir all my college Faculty members for providing me with valuable support and guiding me throughout my college days.

I would like to thank my **family and friends** for being so supportive and generous to me throughout my learning days.

Thanks and Regards, Mr. Vijay Raju Kale

		INDEX	
Chapter	Sr. No	Topic	Pages
1		Introduction	4
	1.1	Existing System and Need for New System	5
	1.2	Introduction to Project	6
2		Project Details	7
	2.1	Hardware/ Software Specification	8
	2.2	System Requirement Specification	10
	2.3	Software process and development	11
3		Feasibility study	14
	3.1	What is feasibility?	15
	3.2	Technical feasibility	15
	3.3	Economic feasibility	15
	3.4	Operational feasibility	16
4		Preliminary Design	17
	4.1	Introduction	18
	4.2	Preliminary Design	18
	4.3	Entity Relationship Diagram	20
	4.4	Data Flow Diagram	25
5		Detailed Design	27
	5.1	Introduction	28
	5.2	Data Dictionary	28
	5.3	Database Design	29
6		Screen Layout	32
7		Testing	42
	7.1	Introduction	43
	7.2	White Box Testing	43
	7.3	Black Box Testing	44
8		Implementation	46
9		Future Enhancement	48
10		Conclusion	50
11		Bibliography	52



1.1 PROBLEM IN EXISTING SYSTEM

- The existing system is manual system. Needs to be converted into automated system.
- Risk of mismanagement of data.
- Less Security.
- No proper coordination between different Applications and Users.
- Fewer Users Friendly.
- Accuracy not guaranteed.
- Not in reach of distant users.

1.2 NEED FOR NEW SYSTEM

The development of the new system contains the following activities, which try to automate the entire process keeping in view of the database integration approach.

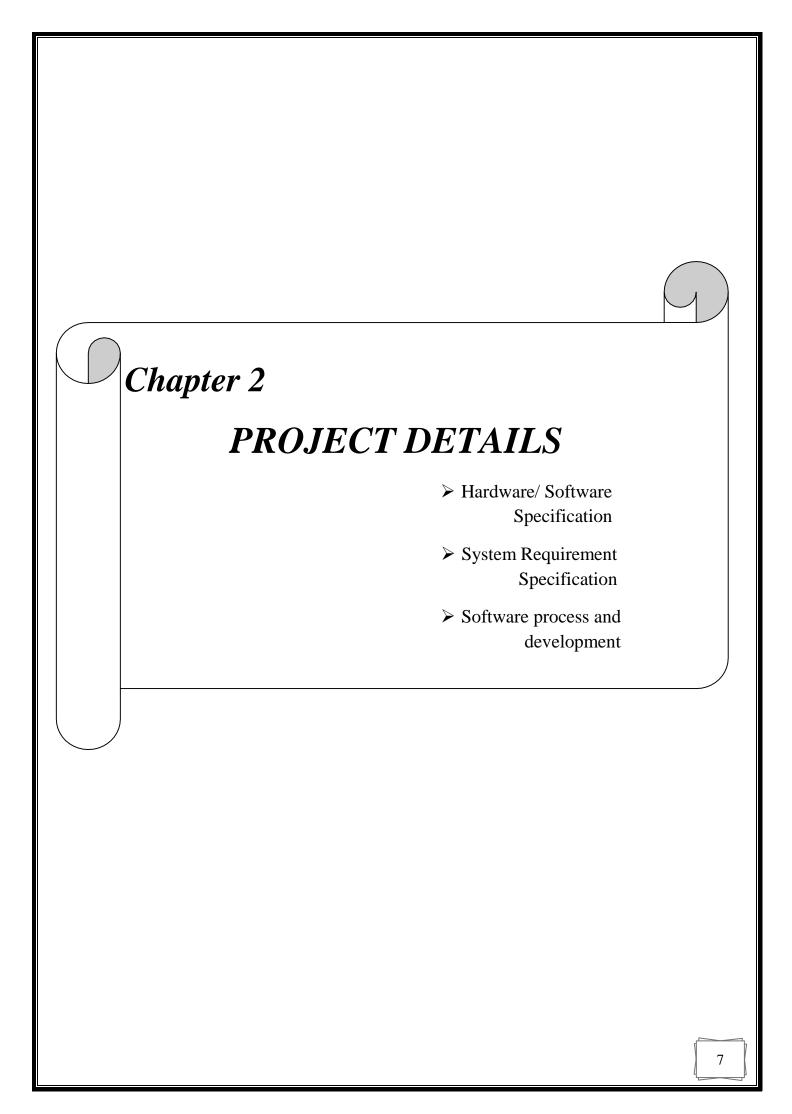
- 1. User friendliness is provided in the application with various controls.
- 2. The system makes the overall project management much easier and flexible.
- 3. There is no risk of data mismanagement at any level while the project development is under process.
- 4. It provides high level of security with different level of authentication.
- 5. Users from any part of the world can make use of the system.
- 6. New system will process accurate results.
- 7. New system will be much better in performance as compared to existing one.

1.3 Introduction to Project:-

Trip Planning Website is a computerized management system. I have tried my best to make the complicated process of trip management system as simple as possible using Structured & Modular technique & Menu oriented interface. I have tried to design the Project in such a way that user may not have any difficulty in using this package & further expansion is possible without much effort. Even though I cannot claim that this work to be entirely exhaustive, the main purpose of my exercise is performed each user activity in computerized way rather than manually which is time consuming.

Trip Website is the application of project management to the creation and development of large-scale trips such as occasions, ceremonies, after weddings, holidays or group trip. It involves studying the fabulous places, identifying its target audience. Trip planner agency might be a tool for strategic marketing and communication, used by companies of every size. Companies can benefit from promotional events as a way to communicate with current and potential customers.

Trip plan managers may also use traditional news media in order to target their audience, hoping to generate media coverage which will reach thousands or millions of people. The objective of this project is to develop a general purpose Trip planner where any type of trip (such as Occasional trip, Family trip etc.) can be plan from the comfort of home through the Internet.



2.1 Hardware and Software Specifications: -

Hardware specification:-

The Hardware Specification describes all functional and non-functional requirements posed on a hardware element (hardware unit, Hardware Components or hardware process module). In order to prepare the Hardware Specification, the requirements will be derived from the specifications of higher system elements or hardware elements. The specification provides standards and tools for designing and decomposing the Hardware Architecture. If changes are required in the course of the development of the hardware element, the Hardware Specification shall be adapted at first. The Evaluation Specification System Element defines the evaluation cases required for demonstrating the requirements of interfaces and specifications.

The Hardware Specification mainly describes the requirements posed on the hardware and specifies the connected interfaces. In addition, requirements and interfaces will be refined and allocated to lower hardware elements.

Computer hardware specifications are technical descriptions of the computer's components and capabilities.:-

- Processor speed, model and manufacturer. Processor speed is typically indicated in gigahertz (GHz). The higher the number, the faster the computer.
- Random Access Memory (RAM), This is typically indicated in gigabytes (GB). The more RAM in a computer the more it can do simultaneously.
- Hard disk (sometimes called ROM) space. This is typically indicated in gigabytes (GB) and refers generally to the amount of information (like documents, music and other data) your computer can hold.
- Other specifications might include network (Ethernet or Wi-Fi) adapters or audio and video capabilities.

Software Specifications:-

The software specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have clear and thorough understanding of the products under development. This is achieved through detailed and continuous communications with the project team and customer throughout the software development process.

A software requirements specification (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system.

1. Software Specification Platform: Windows XP, Windows Vista, Win 7 & etc.

2. Technology: Front End: HTML, CSS, Bootstrap and JavaScript.

Back End: xampp Server.

Browser: Google chrome, Mozilla Firefox etc.

At system requirement analysis stage the information gathering process is identified and focused specially on software requirements. The requirement gathering process is intensified and focused specifically on software. To understand the nature of programs to be built, the software engineer must understand the information for the software, as well as the required function, performance and interfacing. Requirements for both the systems and software are documented with customer. The accessibility of a system after it has been delivered depends on how well it meets the needs.

2.2 System Requirement Specification:

The System Requirement Specification provides a narrative description of each subsystem and a definition of all data that flow between subsystems. The System Requirement Specification forms the foundation of all engineering work at that follows. These specifications analyze the behavior of the system with the external events and the functionality of system elements.

This phase come into picture while developing software when analysis phase is over. This step takes care that whatever problem has been recognized is correct because the user requirements keeps changing and for creation of successful software it's necessary that the software deliver goods at every level of work. For this the requirements and problems to be solved must be thoroughly known and they must be worked upon.

It specifies the goals and objectives of project, describing its context with system on whole. System requirement analysis bridges the gap between system level requirements.

Engineering and software designing. Requirement engineering activities result in the specification of software's interface with other system elements, and establish constraint that software must meet. Requirement analysis allows the software engineer to refine the software allocation and build modules of data, functional and behavioral domains that will be treated by software. Requirement analysis provides the software designer with a representation of information, function and behavior that can be translated to data, architecture, interface and component level designs. Finally, the requirement specification provides the developer and the customer with the means to assess quality software is built.

2.3 Software Process Model and System Development:

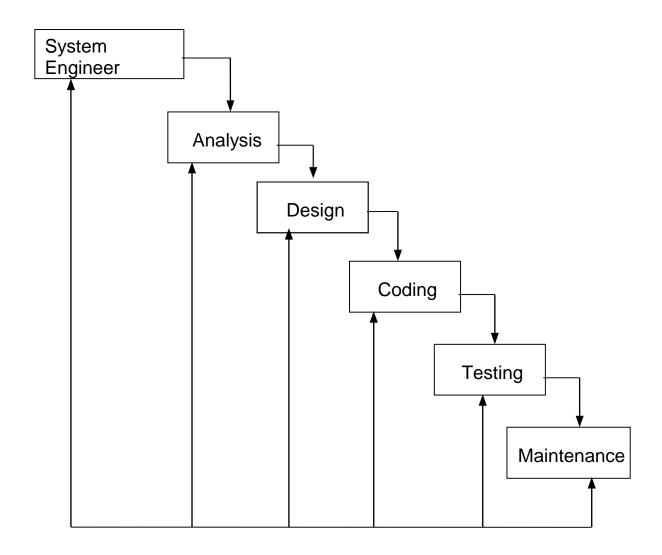


Fig. System Development Life Cycle

1) Analysis –

You may know that the software requirement specification (SRS) document is one of the important documents of your project. The indicators for SRS document are whether you have used some standardization.

Analysis Phase involves following documents:

- 1. Software requirement specification (SRS)
- 2. ER Diagrams
- 3. Use cases
- 4. Data Flow Diagrams
- 5. Data base design documents

2) Designs –

Project design should include the desired features and operations in detail, including user interface design, program structure, schema design and normalized tables and data integrity and constraints. You should include them with the requirements given below:

- **Prototype of Screens:** We have designed the prototype of screens in XLS format to design screen and freeze the requirement.
- **Program Structure:** It should have the proper modularizations of software and specification of each module.
- Schema Design and Normalized Tables: Normalize the table to minimum 3NF. If any demand of Demoralizations clearly explains the reasons.
- Data Integrity and Constraints: Explain the referential diagram. Define entity integrity, which should include keys, alternate keys and other keys, value constraints and ranges.
- **Procedural Design:** Explain using Flowchart / Pseudo code / PDL.
- User Interface Design: Coherence with dataflow and processor; Consistency of interface and naming convention. Validation checks should be kept wherever necessary.
- Architecture: Program architecture and explanation on suitability of data structure used.

3) Coding -

Coding phase of software development includes different activities like refining the algorithms for individual components, transferring the algorithms into a programming language (coding), Translating the logical data model into a physical one and compiling and checking the syntactical correctness of the algorithm with these activities.

I have used different tools like FxCop to check the coding standards. We have followed the Coding standard check list.

WE have learning following parameters:

- **Comments and Description:** Comments with functional description which include the input, output, total function calls to/from other functions, function parameters, description of main variables, Data type, logic description, etc.
- **Standardization of Coding:** Use of naming convention of variable and functions, nested depth, naming constant, use of data structure and style.
- Error Handling: Explain exceptions handling and conditional checking.
- Parameter passing and calling: Check the technique used for this purpose, how it optimizes the coding.
- Security Mechanisms: Maintain confidentiality, integrity and authorization according to the requirement and needs of the system. Also maintain database level security, use of Views, and use of revoke and grant, user and access rights and ensure steps taken against hacking of the system.

4) Testing -

Once code has been generated program testing begins. The testing process focuses on the logical internals of software, ensuring that all statements have been tested, i.e. conducting tests to uncover errors & ensure that define input will procedure actual results that agree with required results.

5) Maintenance -

Software is undoubtedly undergoes change after it is delivered to the customer. Change will occur because errors have been encounter, because the software must be adopted changes in its external environment.



Chapter 3 FEASIBILITY STUDY

- ➤ What is feasibility?
- > Technical feasibility
- > Economic feasibility
- > Operational feasibility

Feasibility Study

What is Feasibility?

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility.
- Operation Feasibility.
- Economic Feasibility.

A feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable. A well-designed study should offer a historical background of the business or project, such as a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, such studies precede technical development and project implementation.

1. Technical Feasibility

In Technical feasibility a study of function, performance and constraint that may affect the ability to achieve an acceptable system are considered. The system should be easy to update i.e. if the user feels to the necessity of some changes in the system then it should be developed in such a way that it will easy to operate.

Technical feasibility is a broad concept. It can equally be applied to a software development project, civil engineering (bridges, tunnels, dams, ...) or a manned mission to Mars, and everything in between. Each will have different standards of feasibility. They could include

- the design is correct and will meet the given requirement
- the products needed to build it are available
- the materials and techniques to implement it are known to work
- doesn't violate some law of physics (say, needs information to travel faster than light)
- doesn't violate a mathematical principle (needs to compute an intractable function)

2. Economic Feasibility

Economics analysis is most frequently used method for evaluation of the effectiveness of proposed system. Cost/benefits analysis is the most commonly used procedure for this feasibility where benefits and savings are determined that are expected from the proposed system.

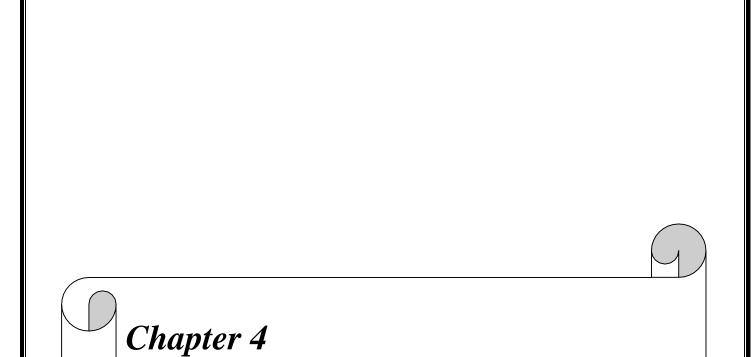
3. Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.



- PRELIMANARY DESIGN
 - > Introduction
 - > Preliminary Design
 - ➤ Use Case Diagram
 - > Entity Relationship Diagram
 - Data Flow Diagram

4.1 Introduction

The system design phase focuses on the detailed implementation of the system. Emphasis is on translation specification into specification. This phase is transition from user oriented document to a document to the programmer of the database personnel.

Software design focuses on three distinct attributes of the program

Data Structure

Software Architecture

Procedural Details

Design translates requirement in to the representation of software that can access for quality.

4.2 Preliminary Design

The preliminary design is concern with the transformation of requirements into data and software architecture.

Logical and Physical Design

The system design does not through two phase of development "Logical and Physical Design" logical flow of the system is shown by data flow diagram.

In this physical species Input, Output screen layouts procedure specification show how the data entered how files are accessed and how reports are produced.

4.3 Use Case Diagrams:

Use Cases represented distinct pieces of functionality for a system, a component, or even a class. Use Case Diagrams describes what a system does from the standpoint of an external observer. A Use case corresponds to a sequence transaction, in which each transaction is invoked from outside the system (actor) & engages internal objects to interact with one another & with the system's surroundings.

• Relationships in Use Cases:





An Actor portrays any entity (or an entity) that Performs certain roles in a given systems.

Use Case:



A Use Case in a use case diagram is a visual representation of distinct business functionality in a system. The key terms here is "distinct business functionality".

System boundary:



A System boundary defines the scope of what a system will be. A System cannot have infinite functionality.

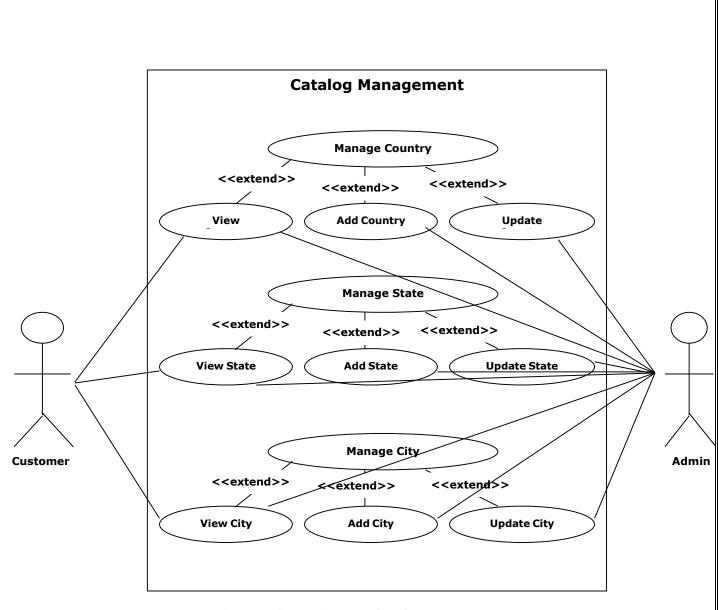
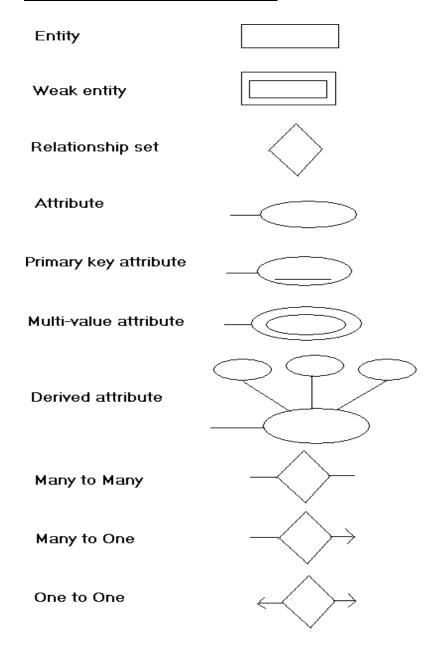


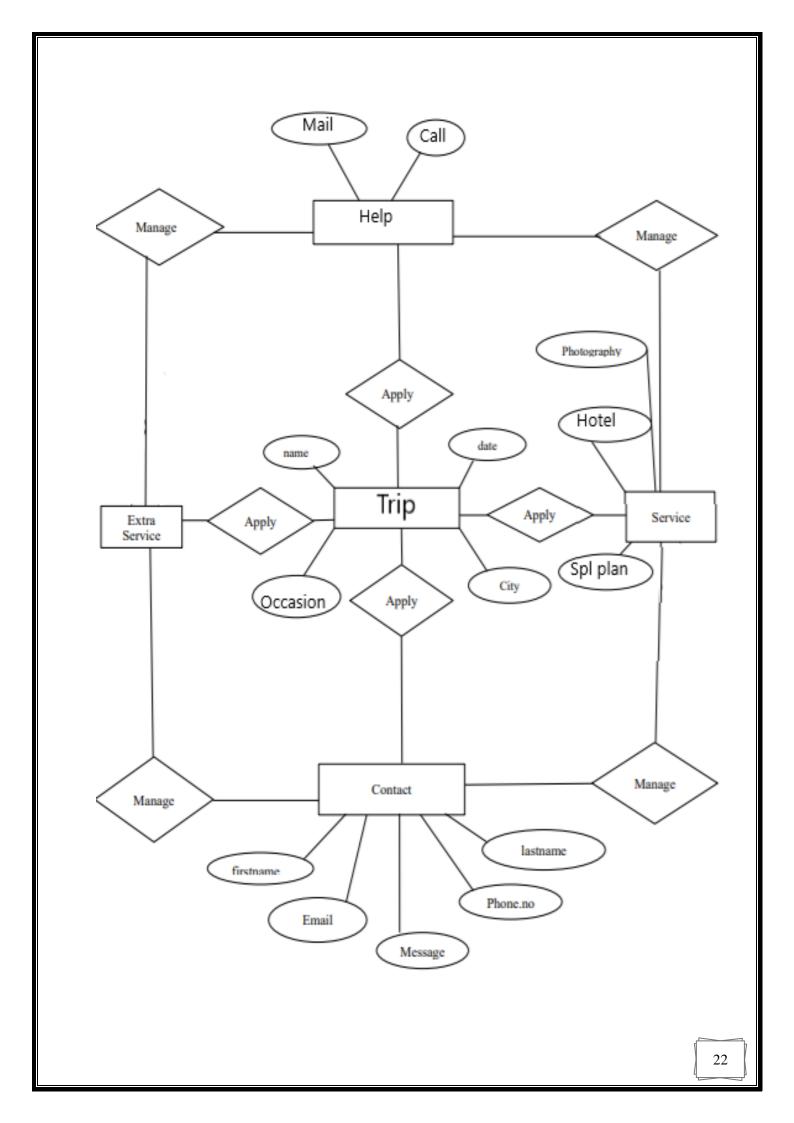
Fig. Use Case Diagram for Catalog Management

4.5 Entity Relationship Diagram (ERD)

E-R Diagram can express the overall logical structure of a database Graphically. Such a Diagram consists of following major components.

1. Entity Relationship Diagram Symbol





4.5 Data Flow Diagram (DFD)

Data Flow Diagram

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. It is a Graphical Technique depicts Information and Transforms that are applied as data move from input or output.

Data Flow Diagram Symbols

In the DFD, there are four symbols

- 1. A square defines a source(originator) or destination of system data
- 2. An arrow identifies data flow. It is the pipeline through which the information flows
- 3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
- 4. An open rectangle is a data store, data at rest or a temporary repository of data

Process that transforms data flow.
Source or Destination of data
 Data flow
Data Store

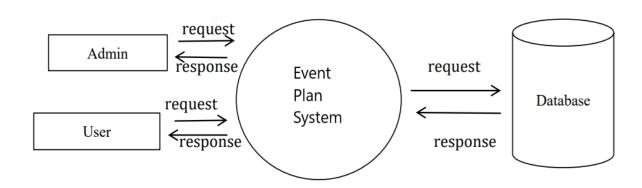


Fig. DFD for Level-0

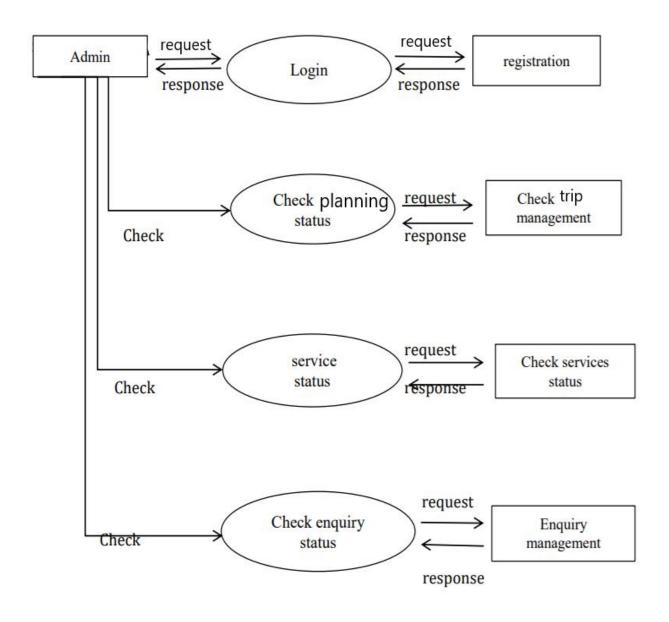


Fig. Admin Side DFD

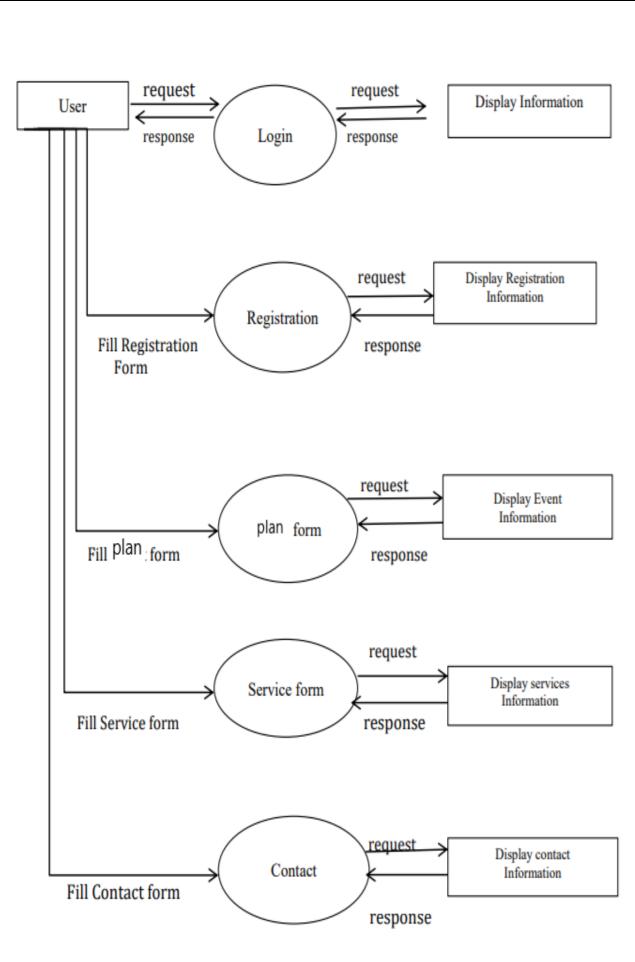
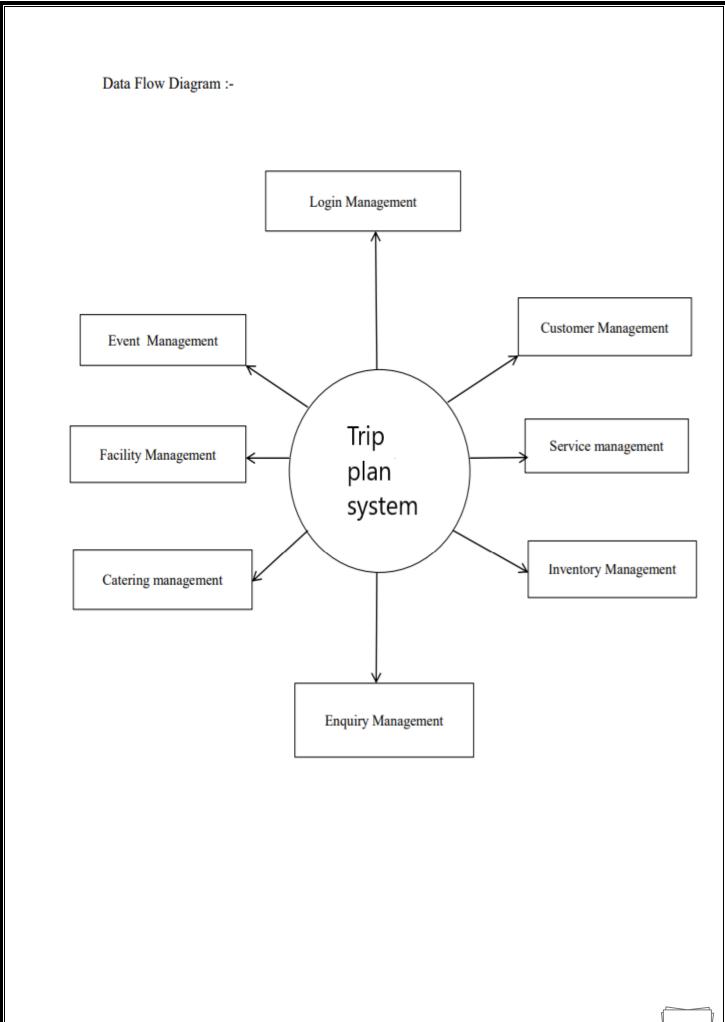
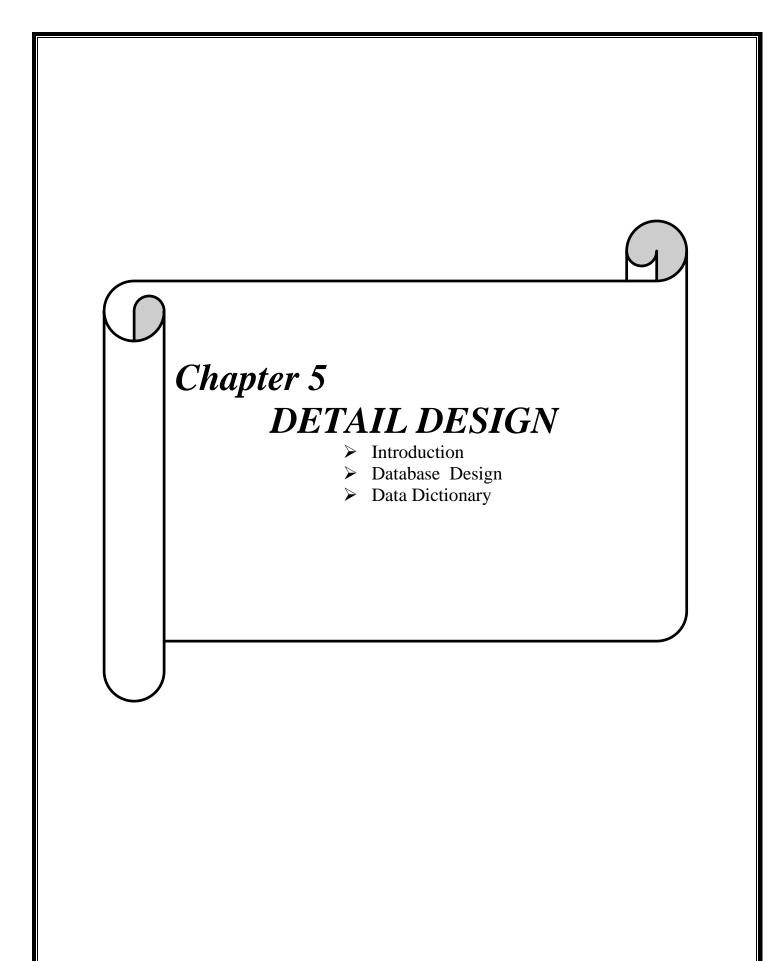


Fig. User Side DFD





5.1 Introduction

Detailed design focus on refinement to the architectural representation that lead to detailed data structure and algorithmic representation for the software. The transformation involves determination of how to store and retrieve the information that is nothing but the database design.

A design is a plan or specification for the construction of an object or system or for the implementation of an activity or process, or the result of that plan or specification in the form of a prototype, product or process. The verb to design expresses the process of developing a design. In some cases, the direct construction of an object without an explicit prior plan (such as in craftwork, some engineering, coding, and graphic design) may also be considered to be a design activity. The design usually has to satisfy certain goals and constraints, may take into account aesthetic, functional, economic, or socio-political considerations, and is expected to interact with a certain environment. Major examples of designs include architectural blueprints, circuit diagrams, and sewing patterns.

The person who produces a design is called a designer, which is a term generally used for people who work professionally in one of the various design areas—usually specifying which area is being dealt with (such as a textile designer, fashion designer, product designer, web designer (website designer) or interior designer), but also others such as architects and engineers. A designer's sequence of activities is called a design process, possibly using design methods. The process of creating a design can be brief (a quick sketch) or lengthy and complicated, involving considerable research, negotiation, reflection, modeling, interactive adjustment and re-design.

5.2 Data Dictionary

Database is a repository of data which has all the files physically stored in. It also Called Data Bank. Data, which are relevant to and requested for, by specific application are only supplied to it and not all the data present in the database are made available.

Database should not have any redundant data present in it. Nor should it have any Incomplete or Contradictory data. The program, which stores, organizes and retrieves data from the database, is known as Database Management System

The terms data dictionary and data repository indicate a more general software utility than a catalogue. A catalogue is closely coupled with the DBMS software. It provides the information stored in it to the user and the DBA, but it is mainly accessed by the various software modules of the DBMS itself, such as DDL and DML compilers, the query optimiser, the transaction processor, report generators, and the constraint enforcer. On the other hand, a data dictionary is a data structure that stores metadata, i.e., (structured) data about information. The software package for a stand-alone data dictionary or data repository may interact with the software modules of the DBMS, but it is mainly used by the designers, users and administrators of a computer system for information resource management. These systems maintain information on system hardware and software configuration,

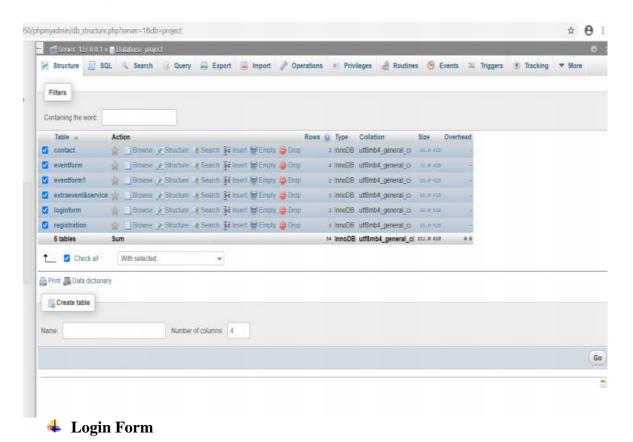
documentation, application and users as well as other information relevant to system administration.

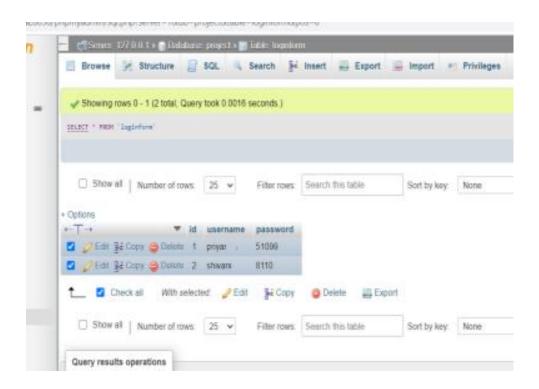
If a data dictionary system is used only by the designers, users, and administrators and not by the DBMS Software, it is called a passive data dictionary. Otherwise, it is called an active data dictionary or data dictionary. When a passive data dictionary is updated, it is done so manually and independently from any changes to a DBMS (database) structure. With an active data dictionary, the dictionary is updated first and changes occur in the DBMS automatically as a result. Database users and application developers can benefit from an authoritative data dictionary document that catalogs the organization, contents, and conventions of one or more databases. This typically includes the names and descriptions of various tables (records or Entities) and their contents (fields) plus additional details, like the type and length of each data element. Another important piece of information that a data dictionary can provide is the relationship between Tables. This is sometimes referred to in Entity-Relationship diagrams, or if using Set descriptors, identifying which Sets database Tables participate in.

5.3 Database Design

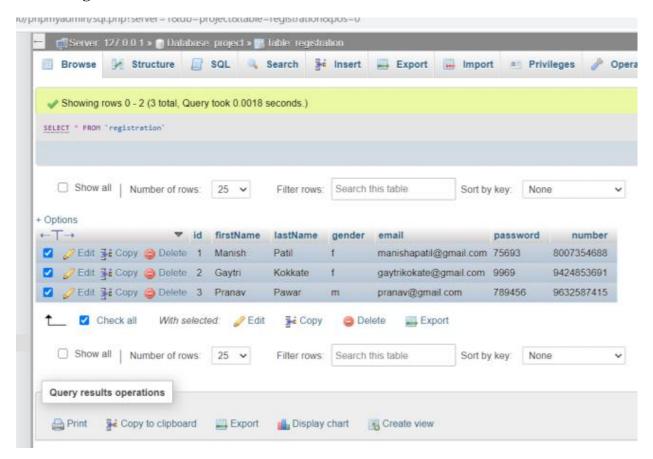
Database plays a vital role in any application or project. This project makes us of number of tables in database. It can be thought of as the logical design of the base data structures used to store the data. The tables are as follows:

Design View

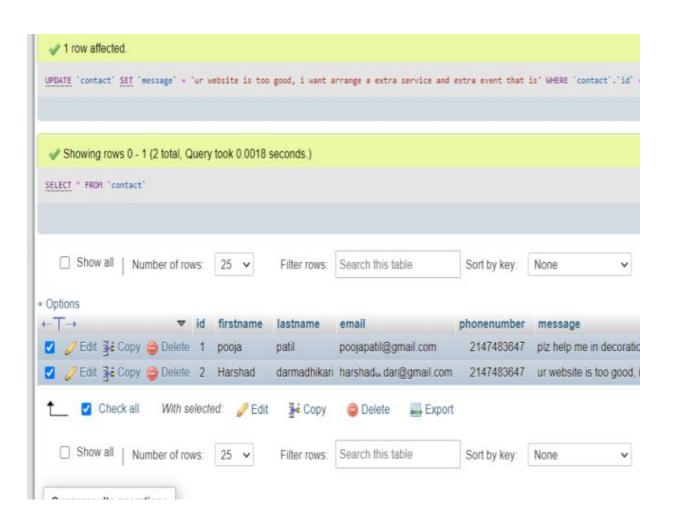


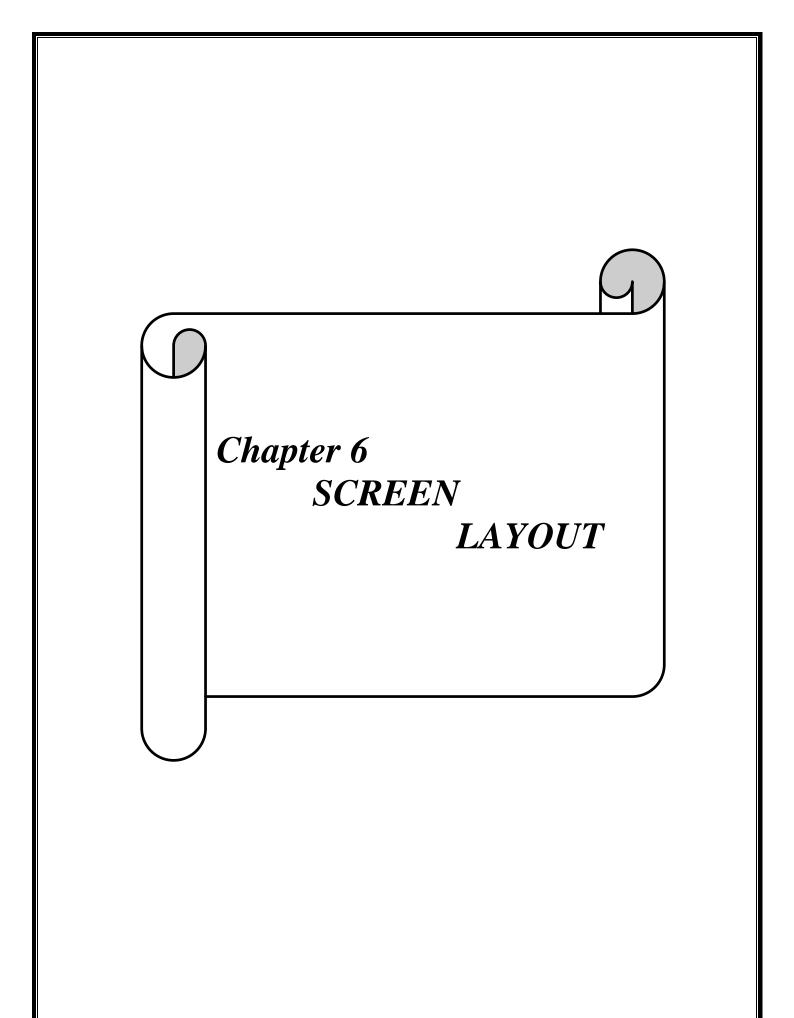


Registration Form

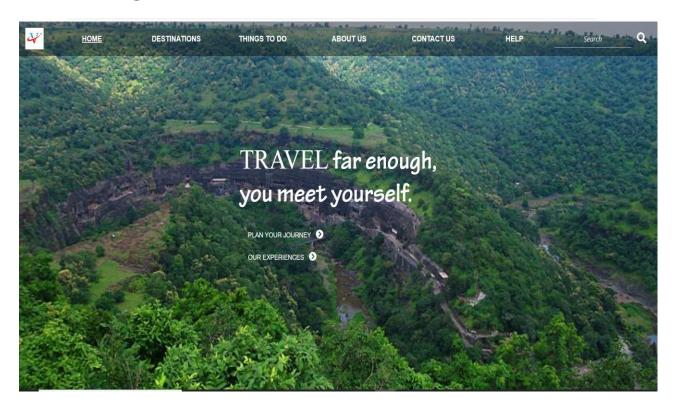


Contact Form

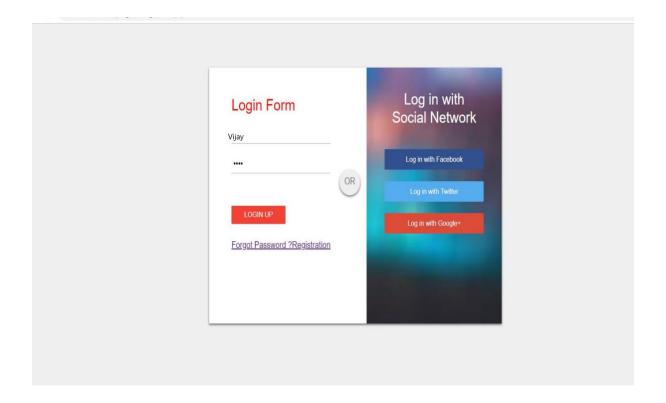




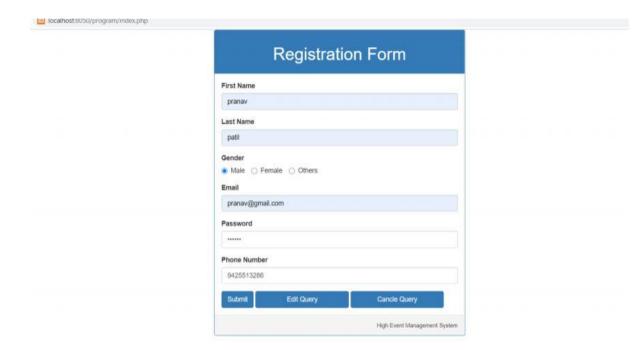
Home Page:



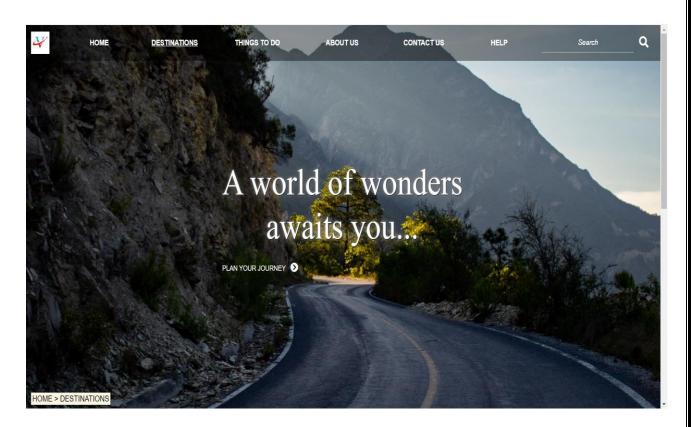
Login Page:

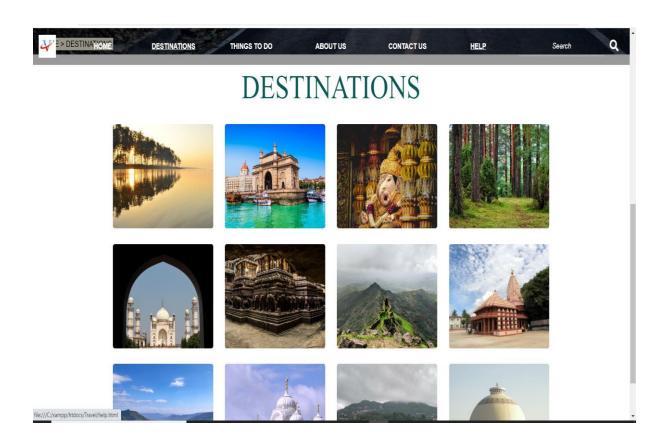


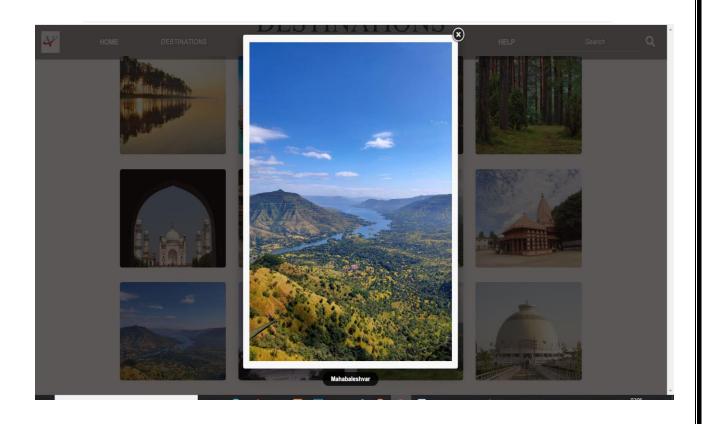
Registration Page:



Destination Page:

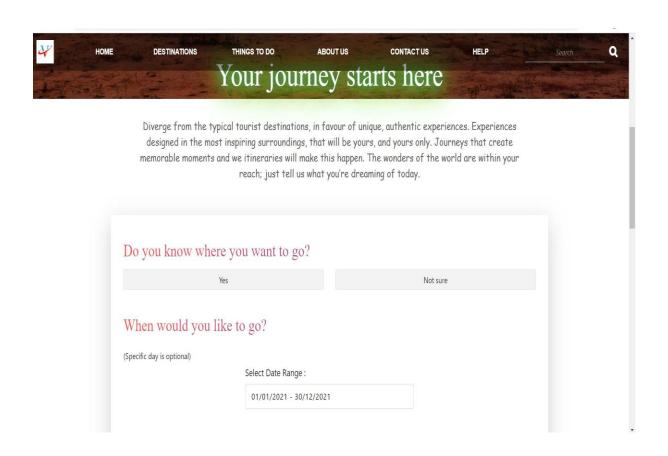


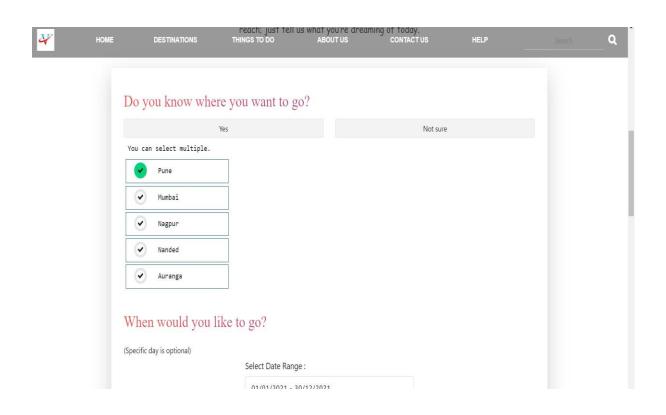


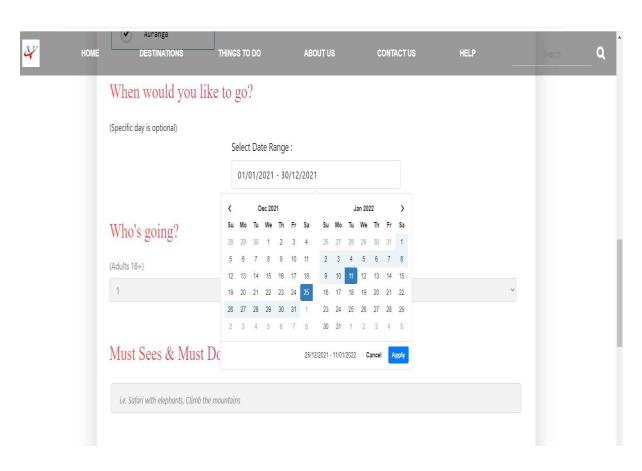


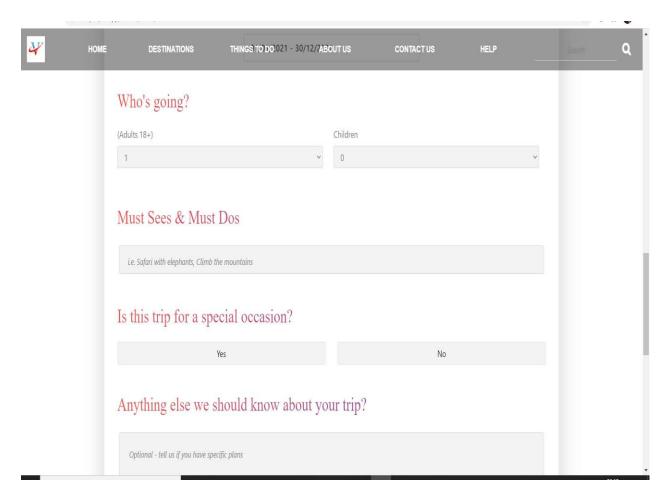
Plan Page:

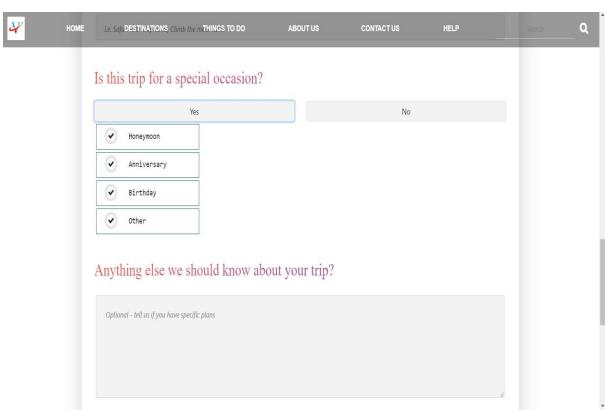


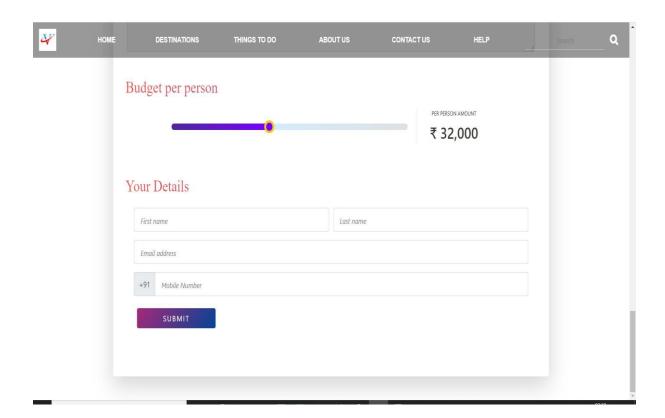




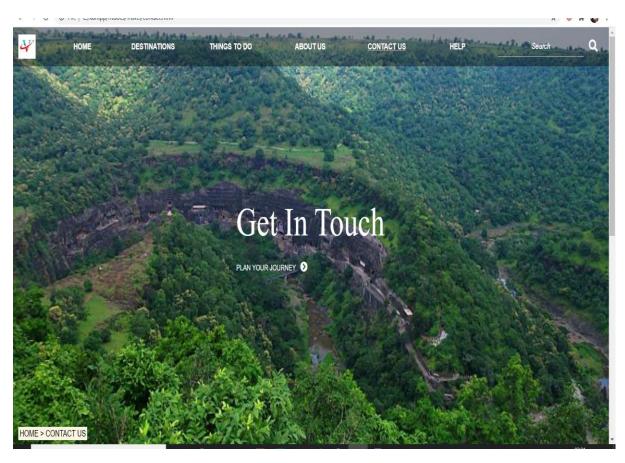


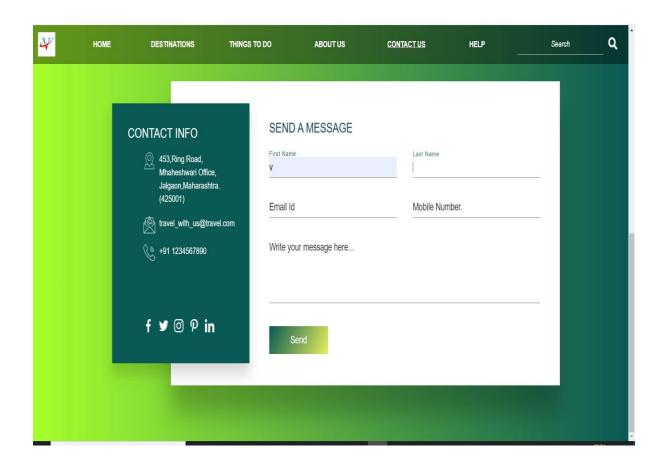






Contact Page:





Help Page:

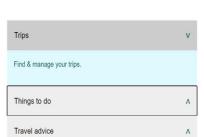






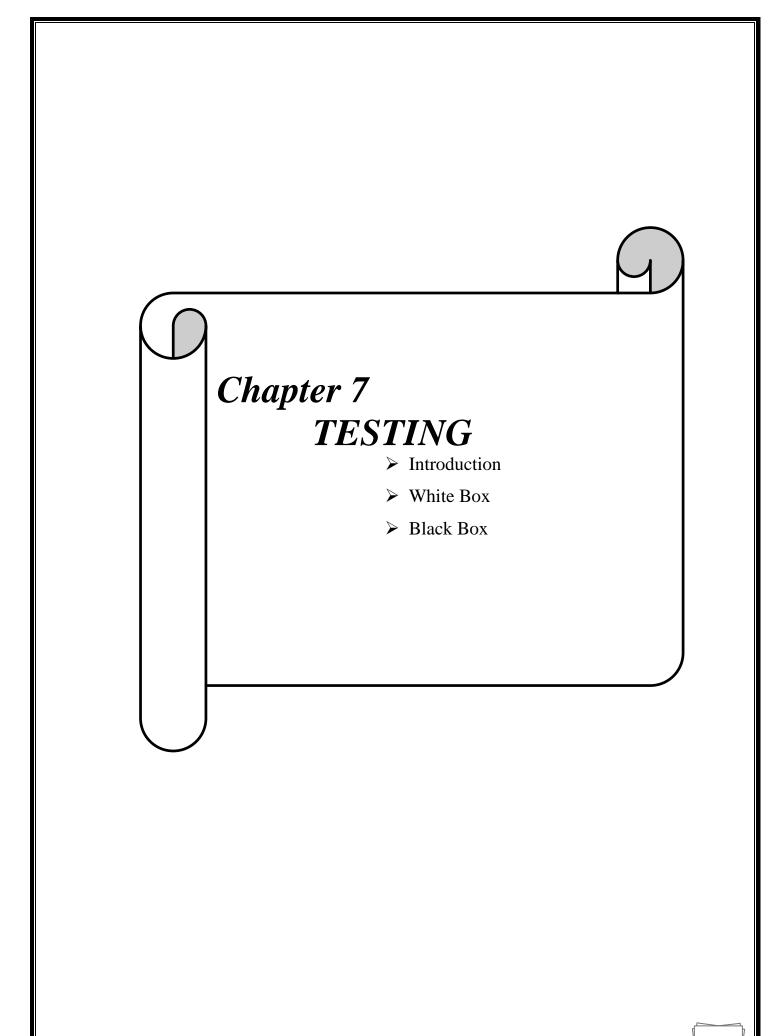
How can we help you?

Describe Your issue...









Introduction Of Software Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

Need of Software Testing Types:

Types of Software Testing, depends upon different types of defects. For example:

- Functional testing is done to detect functional defects in a system.
- Performance Testing is performed to detect defects when the system does not perform according to the specifications
- Usability testing to detect usability defects in the system.
- Security Testing is done to detect bugs/defects in the security of the system.

White Box Testing

White box testing strategy deals with the internal logic and structure of the code. White box testing is also called as glass, structural, open box or clear box testing. The tests written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code etc.

In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e. internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning.

Types of testing under White/Glass Box Testing Strategy:

• Unit Testing:

The developer carries out unit testing in order to check if the particular module or unit of code is working fine. The Unit Testing comes at the very basic level as it is carried out as and when the unit of the code is developed or a particular functionality is built.

• Branch Coverage:

No software application can be written in a continuous mode of coding, at some point we need to branch out the code in order to perform a particular functionality. Branch coverage testing helps in validating of all the branches in the code and making sure that no branching leads to abnormal behavior of the application.

• Security Testing:

Security Testing is carried out in order to find out how well the system can protect itself from unauthorized access, hacking – cracking, any code damage etc. which deals with the code of application. This type of testing needs sophisticated testing techniques.

• Mutation Testing:

A kind of testing in which, the application is tested for the code that was modified after fixing a particular bug/defect. It also helps in finding out which code and which strategy of coding can help in developing the functionality effectively.

Black Box Testing

Black Box Testing is not a type of testing; it instead is a testing strategy, which does not need any knowledge of internal design or code etc. As the name "black box" suggests, no knowledge of internal logic or code structure is required. The types of testing under this strategy are totally based/focused on the testing for requirements and functionality of the work product/software application. Black box testing is sometimes also called as "Opaque Testing", "Functional/Behavioral Testing" and "Closed Box Testing". The base of the Black box testing strategy lies in the selection of appropriate data as per functionality and testing it against the functional specifications in order to check for normal and abnormal behavior of the system. In order to implement Black Box Testing Strategy, the tester is needed to be thorough with the requirement specifications of the system and as a user, should know, how the system should behave in response to the particular action.

These testing types are again divided in two groups:

- Testing in which user plays a role of tester and
- User is not required.
- Testing method where user is not required:

Functional Testing:

In this type of testing, the software is tested for the functional requirements. The tests are written in order to check if the application behaves as expected.

Stress Testing:

The application is tested against heavy load such as complex numerical values, large number of inputs, large number of queries etc. which checks for the stress/load the applications can withstand.

Load Testing:

The application is tested against heavy loads or inputs such as testing of web sites in order to find out at what point the web-site/application fails or at what point its performance degrades.

Recovery Testing:

Recovery testing is basically done in order to check how fast and better the application can recover against any type of crash or hardware failure etc. Type or extent of recovery is specified in the requirement specifications.

• Testing where user plays a role/user is required:

User Acceptance Testing:

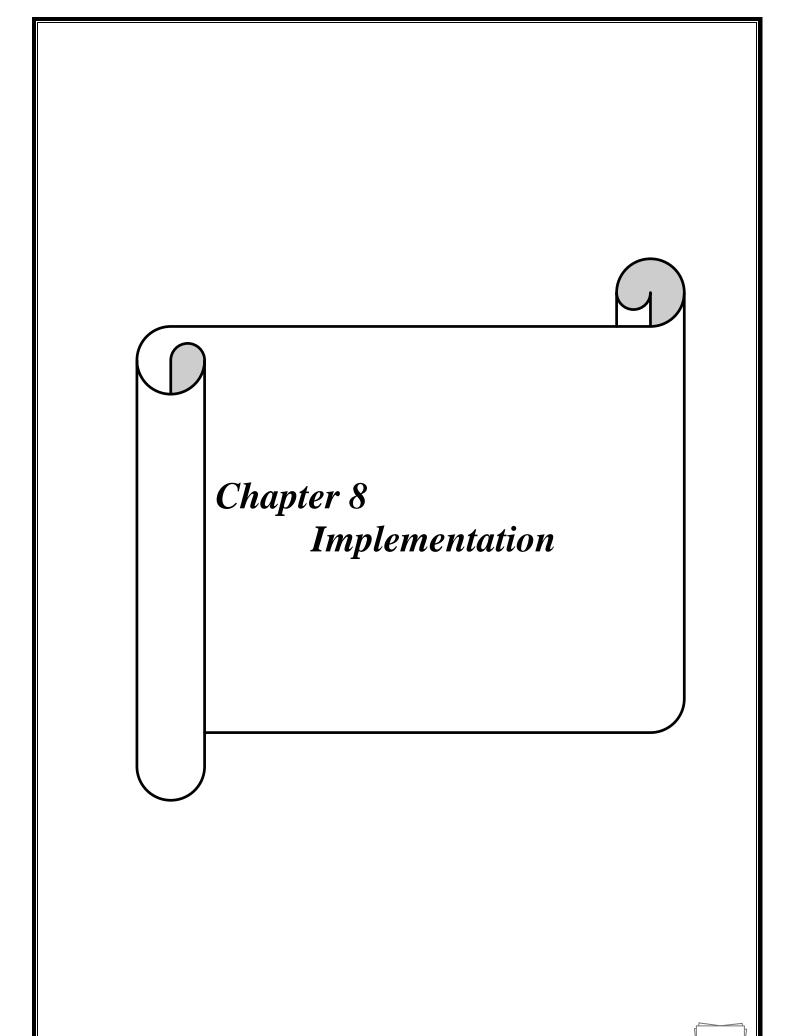
In this type of testing, the software is handed over to the user in order to find out if the software meets the user expectations and works as it is expected to.

Alpha Testing:

In this type of testing, the users are invited at the development center where they use the application and the developers note every particular input or action carried out by the user. Any type of abnormal behavior of the system is noted and rectified by the developers.

Beta Testing:

In this type of testing, the software is distributed as a beta version to the users and users test the application at their sites. As the users explore the software, in case if any exception/defect occurs that is reported to the developers.



Implementation -

Implementation is the stage where the theoretical design is turned into a working system.

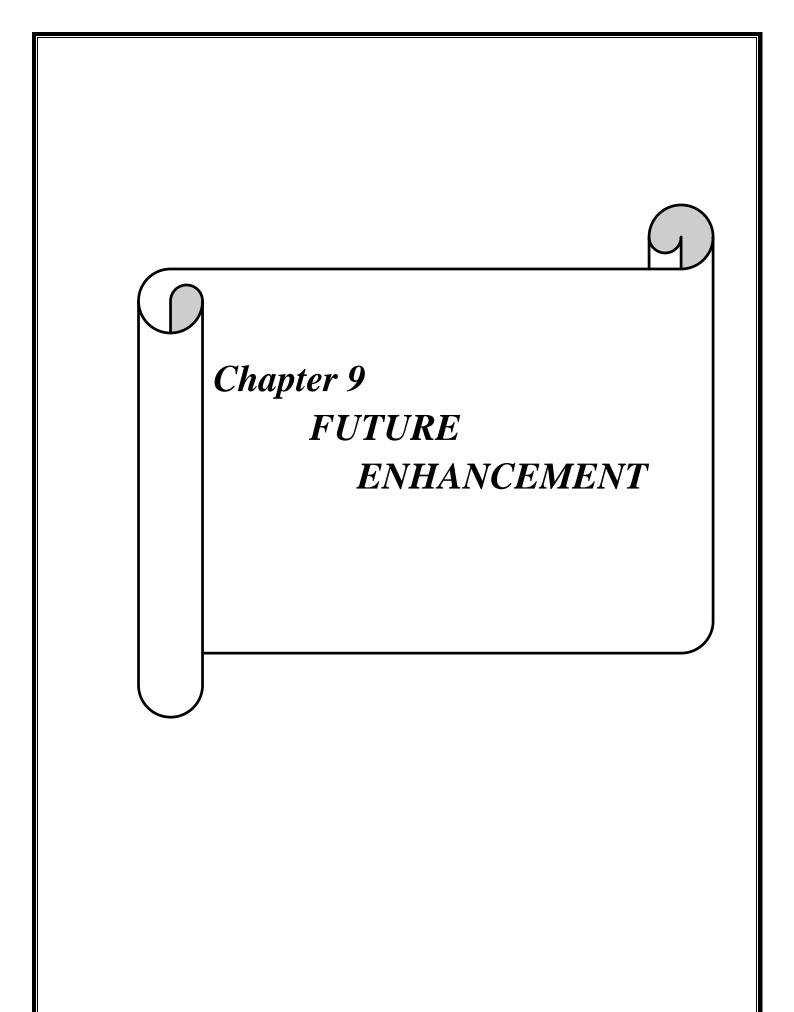
The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after through testing is done and if it is found to work according to the specification.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

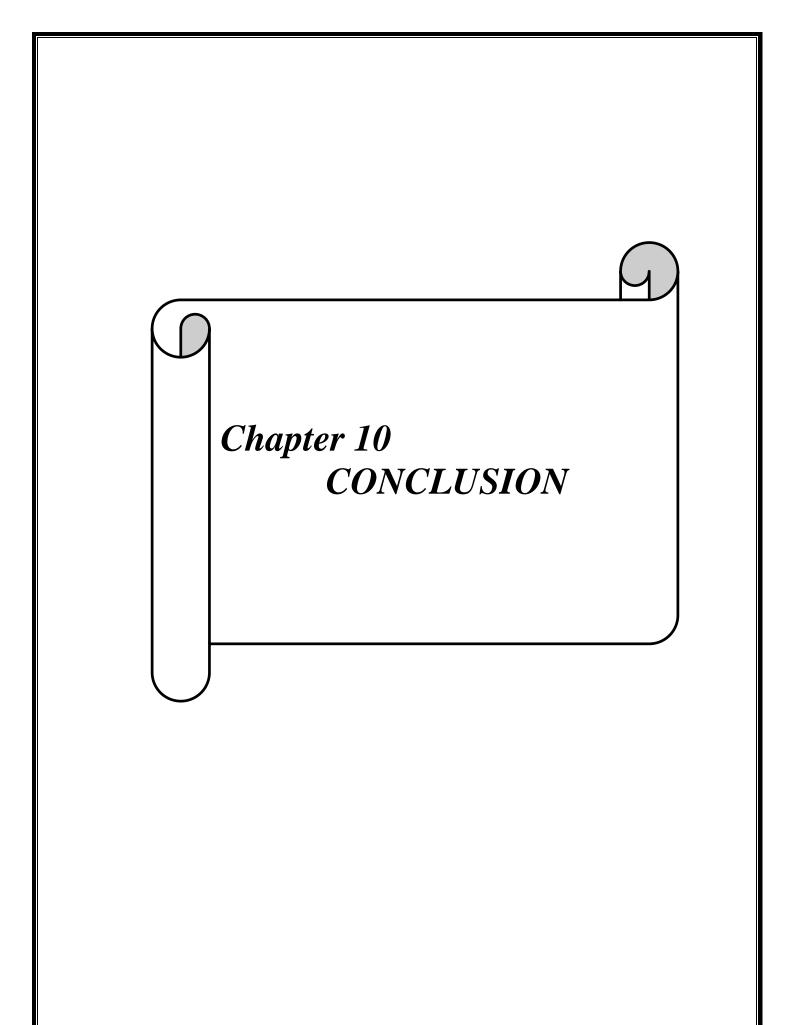
The more complex the system being implemented the more involved will be the systems analysis and design effort required just for implementation.

The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.



Future Enhancement -

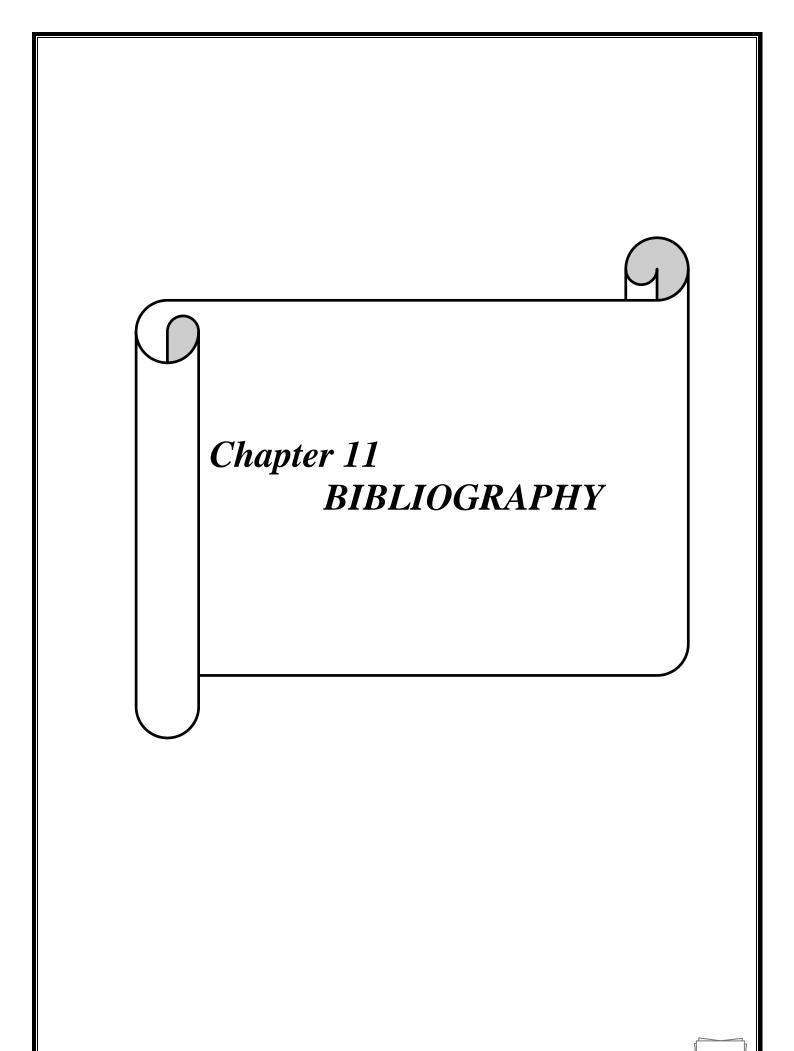
- 1. Any software product developed has some sort of enhancement until it is designed to satisfy all the existing as well as the future needs.
- 2. This project is not an exception when considering the above side. The system is open for future development as and when required by the users. As modular development approach is followed any future enhancements to the system will not be a tedious task and will not have any adverse effects on the other modules or the system as whole.
- 3. In future we are going to introduced insurance company in the system
- 4. Mobile application is going to build in future for android users.
- 5. Add Hotel booking feature.



Conclusion -

- Trip Plan System is user friendly and cost effective system, it is customized With activities related to trip management life-cycle.
- It provides a new edge to management industry.
- We made attempt to effectively introduce the concept of trip of trip management System already existing in the society.
- We then explain the concept of online trip management system which are already Present.
- We describe the proposed system and explain the features implemented by our proposed System
- We also give a brief overview of the technologies used during the development of our proposed system.
- This trip management system can be further refined and extended by introducing new and more innovation features.
- The system allowed registered user login and new user are allowed to register On the application.
- The system helps in the planning for trips, users and the aspects related to them
- This proposed to be a web application . the project provides most of the basic functionality required for an trip type eg- After Marriage, Holidays, Birthday, Collage trip, etc.
- The system then allows the user to select date and time of trip, place and the place.

 All the data is logged in the database.



Books: -

- [1] Black Book (HTML)
- [2] PHP Step by Step Microsoft Publication

Web References: -

www.w3school.com

www.studytonight.com

www.codeproject.com

www.google.com