**A**

**Project Report On**

HOTEL DIVINE

**Submitted By**

**Internal Guide:**

CERTIFICATE

###### Date of Submission :

**Staff in Charge :**

**Head of Department :**

**ACKNOWLEDGEMENT**

I am extremely fortunately and greatly honored to work under the guidance of **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** in Computer Engineering Department of \_\_\_\_\_\_\_\_\_\_\_\_\_for his valuable guidance during the Project work, without which it would have been very difficult task.

I have no words to express my sincere thanks for valuable guidance, assistance and cooperation extended by all the Staff Members of the

Department.

I would like to heartily thank to our H.O.D. **\_\_\_\_\_\_\_\_\_\_\_\_**for his guidance and support during my dissertation. The discussions with them and their constant feedback helped me immensely in completing this Project work satisfactorily.

I would like to thank all the teaching and non-teaching staff members of my Department and my colleagues those who helped me directly or indirectly for completing of this Project successfully.

My acknowledgment would not be complete without mentioning my parents & my family members. I am specially thanking them for continuous

encouragement for my work.

**ABSTRACT**

One of the most crucial factor to the success of Hotel business is the Easiness by which customers can make reservations online. If manual interventions are reduced in reservation process then it helps to bring in more customers and saves time. Aim of this project is to build a website for Hotel Divine that is easy to use for customers as well admits. Customers will be able to manage their table reservations such as booking/cancelling reservation. Customers will also have an option to specify hotel feedback. Also, they will be able to view all the reservations done for a day.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER 1:INTRODUCTION** | **01-06** | |
| 1.1 Project Summary. | | 2-3 |
| 1.2 Purpose | | 3 |
| 1.3 Scope | | 3 |
| 1.4 Objective | | 4 |
| 1.5 Technology and Literature Review | | 4-6 |
| **CHAPTER 2: SDLC & METHODOLOGY** | **07-14** | |
| 2.1 SDLC and Methodologies | | 6-10 |
| 2.2 Principles of System Development | | 10-14 |
| **CHAPTER 3: PROJECT MANAGMENT** | **15-19** | |
| 3.1 Software Process Model | | 16-17 |
| 3.2 Milestones and Deliverables | | 18 |
| 3.3 Roles and Responsibility | | 18 |
| 3.4 Project Scheduling | | 19 |
| **CHAPTER 4:REQUIREMENT ANALYSIS** | **20-27** | |
| 4.1 Requirement Engineering Process | | 21 |
| 4.2 Feasibility study | | 22-23 |
| 4.3 Requirement Analysis | | 23-25 |
| 4.4 Requirement specification | | 26-27 |
| **CHAPTER 5: MODULE DESCRIPTION** | **28-30** | |
| 5.1 Admin | | 29 |
| 5.2 User | | 30 |
| **CHAPTER 6:SYSTEM ANALYSIS AND DESIGN** | **31-44** | |
| 6.1 Data Dictionary | | 32-35 |
| 6.2 E-R Diagram | | 36 |
| 6.3 Data Flow Diagram | | 36-38 |
| 6.4 Process Flow | | 39 |
| 6.5 Use Case | | 40 |
| 6.6 Activity Diagram | | 41-42 |
| 6.7 Flow Chart | | 43-44 |
| **CHAPTER 7:USER MANUALS** | **45-53** | |
| 7.1 Screen-shots | | 45-53 |

|  |  |
| --- | --- |
| **CHAPTER 8: TESTING 54-61** | |
| 8.1 Verification and Validation | 55-58 |
| 8.2 Testing | 59-61 |
| **CHAPTER 9: PLANNING & CODING 62-64** | |
| 9.1 Implementation Environment | 63 |
| 9.2 Coding Standards | 64 |
| **CHAPTER 10: LIMITATION, FUTHER ENHANCEMENT 65-66** | |
| 10.1 Limitation | 66 |
| 10.2 Further Enhancement | 66 |
| **CHAPTER 11: CONCLUSION 67-68** | |
| 11.1 Conclusion | 68 |
| **CHAPTER 12: BIBLIOGRAPHY 69-73** | |
| 12.1 Website Referred | 70 |
| 12.2 Abbreviation | 70 |
| 12.3 Notation | 71-73 |

# Chapter-1 Introduction

##### Key Concepts:

* 1. Project Summary
  2. Purpose
  3. Scope
  4. Objectives
  5. Technology and Literature Reviews

 ***INTRODUCTION:***

This application provides facilities like user can easily get the details about particular product who are interested in handicraft. This project understands the Structure of the system. In this project which we can get all the information about online handicraft. This project is work at on the two sides like admin side and user side. In this web site we can see hotel photos. In this web site we can accept the advertisement of any other companies. It provides all types of handicraft products.

These websites send an e-mail to the user for its procedure and delete also. In this project we can update all the information of the user.

* 1. ***PROJECT SUMMARY:***

This application is static based application. This website shows the information of **Hotel Divine.** This website shows all the information of **Hotel interior, food facilities and all other additional events.** Also shows details of **food products.**

* + - **PROJECT TITLE / DEFINITION**

Hotel Divine

* + - **PROJECT DURATIO**

6 Month

* + - **DEVELOPED AT**

###### OTHER TOOLS

MACROMEDIA DREAMWEAVER 8 PAINT

* + - **FRONT END TOOLS**

JAVA SCRIPT CSS

HTML PHP

* + - * **DOCUMENTATION TOOLS** MICROSOFT OFFICE

###### PROJECT GUIDE

* + - **SUBMITTED BY**

#### PURPOSE:

Aim of this project is to build a Table Reservation System that is easy to use for customers as well, admins. Customers will be able to manage their table reservations such as booking/cancelling tables.

* 1. ***SCOPE:***
* Regular updates will be available to users of the application about new arrivals.
* We will put extra some features like discount & extra services.
* In feature we will plane an Android Application for such as reservation.
  1. ***OBJECTIVES:***
* This is Web site for Table Reservation System.
* Admin, client can use this system.
* Visitor can easily get information about reservation in a restaurant.
  1. ***TECHNOLOGY AND LITERATURE REVIEWS:***

Here in this application we are using PHP as Front-End and MySQL as Back-End. So, let’s understand the introduction and some of the features of them.

#### PHP

Introduction to PHP

* The full form of PHP is “Hypertext Pre-processor”. Its original name was “Personal Home Page”.
* Erasmus Lerdorf Software Engineer. Apache team member is the creator and original driving force behind PHP. The first application of PHP was developed for his personal use in late 1994.
* By the middle of 1997, PHP was being used on approximately 50,000 sites worldwide.
* PHP is server-side scripting language, which can be embedded in HTML or used as a stand-alone.
* Someone looking at a PHP page will not necessarily be able to tell that it was not written purely in HTML, because usually result of PHP is HTML.
* PHP is an official module of Apache HTTP server.
* PHP is fully cross-Platform, meaning it runs native on several flavors of UNIX as well as on Windows and now on Mac OS X.

***Advantages of PHP***

* Cost
* Ease of use
* HTML-Support
* Cross-Platform
* PHP is compatible with the three leading Web servers.
* Stability
* Speed.
  + - MySQL

***Introduction to MySQL***

* + - * “MySQL is an open source relational database management system (RDBMS) that uses Structured Query Language (SQL), the most popular language for adding, accessing, and processing data in a database.”
      * It is commonly employed with most of the popular server-side scripting languages including PHP, JSP and ASP.
      * It is a multithreaded, multi-user, SQL (Structured Query Language) relational database server (RDBMS).
      * MySQL is a freely available third-party database engine designed to provide fast access to stored data.
      * Data can be stored, updated and deleted using languages such as PHP. The data can be retrieved from the database to allow the generation of dynamic Webpages.

***Features of MySQL***

* + - * + Written in C and C++. Tested with a broad range of different compilers.
        + Works on many different platforms.
        + APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are available.
        + Fully multi-threaded using kernel threads. It can easily use multiple CPUs if they are available.
        + Provides transactional and non-transactional storage engines.
        + A very fast thread-based memory allocation system.

# Chapter-2 SDLC and Methodologies

##### Key Concepts:

* 1. SDLC and Methodologies
  2. Principles of System Development
     1. Owner and Users Involvement
     2. Problem-Solving Approach
     3. Phases and Activity Set Up
     4. Standards for Consistent Development And Documentation
     5. System as capital Investment
     6. Constant Project Scope Revision
     7. System to More Manageable Subsystem
     8. System Design for Growth and Change
  3. ***SDLC AND METHODOLOGIES:***

###### System development life cycle (SDLC)

Also called application to solve business problem and needs. Development life cycle. SDLC is a logical process by which systems analysis, software engineers, programmers, and end-user build information systems and computer.

###### Methodology

Defines step-by step activity for each phase, individual and group roles in each activity, deliverables and quality standards for each activity and tools and techniques to be used for each activity.

###### Different Phase of System development life cycle(SDLC) are

1. Access Needs
2. Design Specifications
3. Design/Develop/Test Software
4. Implement Systems
5. Support Operations
6. Evaluate Performances
7. SDLC methodology has the following main stages

###### Access Needs

The Existing system is evaluated. Deficiencies are identified this can be done by interviewing user of the system and consulting with support personnel. In this phase had to analyze the needs of the **Hotel Divine** website. In this phase I had decided the basic requirement of the website that had to be developed first. I listed the tasks and the way in which it had to be developed.

###### Design Specifications

The new system requirements are defined. Deficiencies in the existing system should be addressed along with details for improvement. In this phase made the Database design for the **Hotel Divine** website also the data flow of the system to be developed.

###### Design/Develop/Test Software

The proposed system is designed. Plans are prepared for the logical and physical constructor, hardware, operating systems, programming, communication, training, and security issues.

The new system is developed. The new components and programs must be obtained, installed and integrated. System users must be trained. All aspects of performance must be tested. In this phase the development of the website was carried out with some testing involved to see that the system was working proper or not.

###### Support Program Implementation

The system is put into use. The new system is phased in, according to application or location, and the old system is phased out. In this phase the system was implemented at the client place and the application was to be put on live.

###### Support Operations

The system is put into use. The new system is phased in, according to application or location, and the old system is phased out. In this phase the system was implemented at the client place and the website was to be put on live.

###### Evaluate Performance

Once the new system is up and running for a while, it should be exhaustively evaluated. Maintenance must be kept rigorously at all times. Users of the system should be kept up-to- date concerning the latest modification and procedures. In this phase the performance of the system was evaluated and any suggestions made by the client were to be noted. Phases of the Software Development Lifecycle are depicted in the following figure

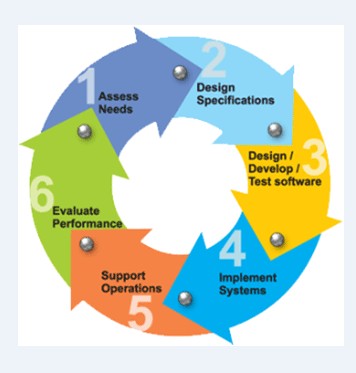


Fig 2.1 SYSTEM DEVELOPMENT LIFE CYCLE

* 1. ***PRINCIPLES OF SYSTEM DEVELOPMENT:***
     1. **Owners and User Involvement**

Without users/owner involvement, it is possible that the technological solutions don’t address the real organization problem. Therefore, owner and user involvement is necessary to minimize organization or technical problems. For successful systems development, system development should go along with user and owner’s participation. In my project the client had clear idea of the type of application he wanted and then specified his requirement for making the application. As the client was clear of what he wanted his involvement was important and due to his involvement, it was possible to make this application. So the client suggested us the design and type of application he wanted, thus his involvement helped me during the phase when I was making the application.

###### Problem-Solving Approach

Problem in system development includes real problems, opportunities for improvement, and directives from management. Using some sort of problem-solving approach to all projects increases efficiency and productivity in system development. The Classical problem solving approach is as follows:

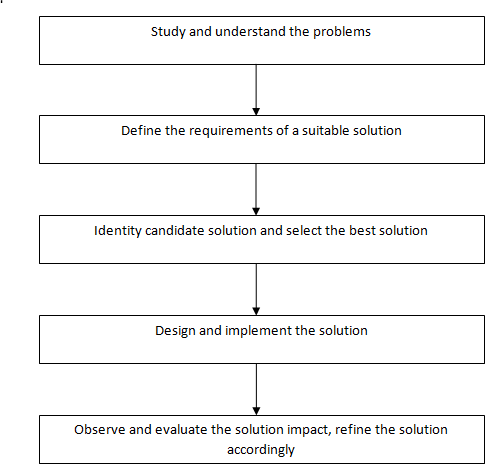


Fig 2.2 Classical Problem-Solving Approach

###### Phases and Activities set up

The classical system life cycle consists of four phases:

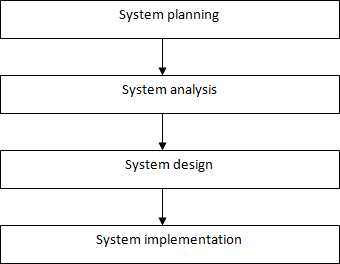


Fig 2.3 Classical Life Cycle Phases

* + 1. **Standards for consistent development and documentation**

Establishing standards promote good communication between users and information system professionals. Sometimes, users can be changed. With the standards and documentation, the project ensures consistent system development. These standards describe activities, responsibilities, documentation guidelines or requirements and quality check. Documentation should be working by-product of entire systems development effort. Documentation should go parallel with the project.

As I had developed the software for my company I had followed the development standards that the company was following and also they are depicted later in document. Thus, following the coding standards of the company I had no difficulty during the overall integration of the system.

###### Systems as capital investments

System must be considered as a capital investment. Analyst must consider the cost of the system, which may contain development time; its operational cost, technology cost and all feasibility study must be done accordingly.

Thus, the purpose of studying the application is to check the feasibility of the application and the time for which it works properly and the resources consumed by it.

###### Constant project scope revision

Multiple feasibility checkpoints are built into the system development methodology in creeping commitment approach. At this point, all costs are considered irrecoverable and the project has to be revaluated to determine if it is still feasible. Analysis should consider cancellation of the project if it is no longer feasible, revaluating of cost and schedule if project scope is to be increased or reduction of scope if the project budget and schedule are frozen, but not sufficient to cover all project objectives.

By keeping the point of recovery of the invested resources capitals we had given a live discussion with client to show him the current development process and how the application is working at present so that the time and cost both helped us to invest in the right direction.

###### System to More Manageable Sub Systems

Virtually all systems, which are part of super-system, contain smaller systems (sub -system). Problem-Solving processes can be simplified by dividing this larger system into more easily managed pieces.

So I divided the whole system into two subsystems, one the admin side and the other the user side to make the system more manageable.

###### System Design for Growth and Changed

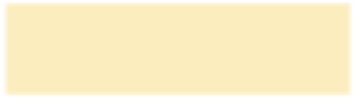
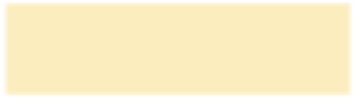
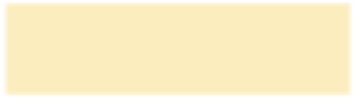
System design is helpful to know about system flow as well as further development. By establishing a pattern of enhancing prior work, the project precludes the insertion of additional requirement in latter stages. New requirement are formally set aside by the development team for latter references, rather than going throw the effort of baking the new requirements into prior stage outputs and reconciling the impacts of the additions. As a result, the project participants maintain a tighter focus on the original product goals, minimize the potential for scope creep, and show a preference for deferring out-of-scope enhancements, rather than attempting to incorporate them into the current effort

# Chapter-3 Project Management

##### Key Concepts:

* 1. Software Process Model
  2. Milestones and Deliverables
  3. Roles and Responsibility
  4. Project Scheduling
  5. ***SOFTWARE PROCESS MODEL:***

###### Waterfall Model



Deployment

Coding & Testing

Designing

Planning

Requirement

Analysis

* + - * **REASONS FOR USING WATERFALL MODEL**

Processes based on Waterfall Model are widely used for practical system development. In the normal waterfall model, the next phase should not start until the previous phase has finished. But in practice, these stages overlap and feed information to each other. During design, problems with requirements are identified, during coding design problems are found and so on. The software process is not a simple linear model but involves of iterations of the development activities.

Because of the costs of producing and approving documents, iterations are costly and involve significant rework. Therefore, after a small number of iterations, it is normal to freeze parts of

the development, such as the specifications and to continue with the later development stages. Problems are left for later resolution, ignored or are programmed around. This premature freezing of requirements may mean that the system won’t do what the user wants. It may also lead to badly structured systems as design problems are circumvented by implementation tricks.

Thus, to use a Waterfall Model with feedback is the best option in such a system where you need to move from one phase to other with the feedback from other stages.

**The iterative waterfall model provides feedback paths from every phase to its preceding**

**phases, which is the main difference from the classical waterfall model.**

When errors are detected at some later phase, these feedback paths allow correcting errors committed

by programmers during some phase. The feedback paths allow the phase to be reworked in which

errors are committed and these changes are reflected in the later phases. But, there is no feedback path

to the stage – feasibility study, because once a project has been taken, does not give up the project

easily.

It is good to detect errors in the same phase in which they are committed. It reduces the effort and

time required to correct the errors.

###### Advantages of Model

1. Easy to explain to user.
2. Stages & activities are well defined.
3. Because of back-tracking it is easy to verification at each stage.
4. Ensure early detection of errors.
5. Testing is inherent to every phase.

* **Advantages of Iterative Waterfall Model:**
  + Some working functionality can be developed and early in the software development life cycle (SDLC).
  + It is easily adaptable to the ever-changing needs of the project as well as the client.
  + It is best suited for agile organizations.
  + It is more cost effective to change the scope or requirements in Iterative model.
  + Parallel development can be planned.
  + Testing and debugging during smaller iteration is ea

#### MILESTONES AND DELIVERABLES:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Milestone Activities** | **Start Date** | **End Date** | **Duration** | **Deliverable** |
| 1. | Requirement  Analysis |  |  | 6 Days | Yes |
| 2. | Planning |  |  | 3 Days | Yes |
| 3. | Designing |  |  | 19 Days | Yes |
| 4. | Requirement Analysis |  |  | 3 Days | Yes |
| 5. | Planning |  |  | 2 Days | Yes |
| 6. | Designing |  |  | 13 Days | Yes |
| 7. | Coding |  |  | 40 Days | Yes |
| 8. | Testing | 23/02/2019 | 31/03/2019 | 37 Days | Yes |
| 9. | Deployment | 01/04/2019 | 15/04/2019 | 15 Days | Yes |

* 1. ***ROLES AND RESPONSIBILITY:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Roles** | | | | |
| **Analysis** | **Designing** | **Coding** | **Testing** | **Documentation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### PROJECT SCHEDULING:

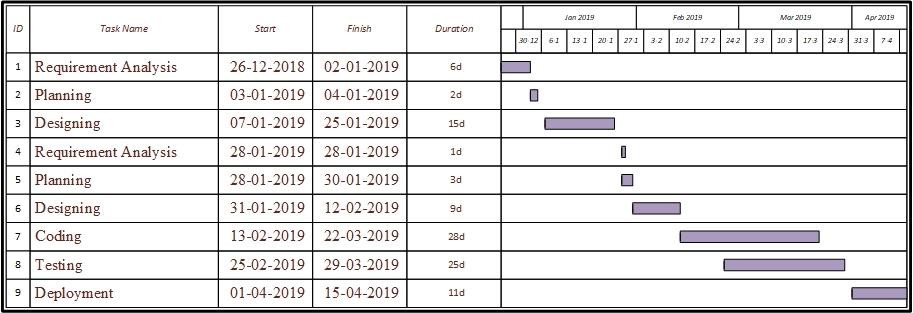


Fig 3.4 Gantt Chart

# Chapter-4 Requirement Analysis

##### Key Concepts:

* 1. Requirement engineering process
  2. Feasibility study
     1. Operational Feasibility
     2. Technical Feasibility
     3. Scheduling Feasibility
     4. Economical Feasibility
     5. Implementation Feasibility
  3. Requirement Analysis
     1. Domain Understanding
     2. Requirement Collection
     3. Classification
     4. Conflict Resolution
     5. Prioritization
  4. Requirement specification
     1. Functional Requirements
     2. Non-Functional Requirements
  5. ***REQUIREMENT ENGINEERING PROCESS:***

Requirement engineering is a process that involves all of the activities required to create and maintain a system requirement document. There are four generic, high-level requirement engineering process activities. These are system feasibility study, the elicitation and analysis of requirements, the specification of requirements and their documentation and, finally, the validation of these requirements.

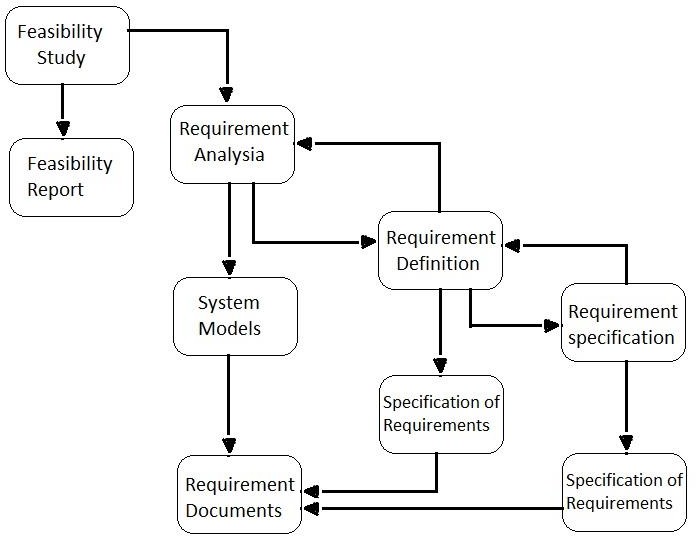


Fig 4.1 Requirement Engineering Process

#### FEASIBILITY STUDY:

Once scope has been identified, it is reasonable to ask whether we can build software that meets this scope. Is this project feasible?

###### Feasibility study

The feasibility of software can be tested in for dimensions: Technology-is a project technically feasible? As in our case that is messenger we have many examples of the same, so no technical in fusibility are there. Finance-Is it financially feasible? Dose it has too much cost of development. Time-will it take too much time to complete? We have planned each phase and it seems to be in controlled and within time so no extra time cost will be added. Resources-Dose we have sufficient resource to succeed?

There are four categories of feasibility tests: operational feasibility, technical feasibility, schedule feasibility and economic feasibility. Often economic feasibility and operational feasibility often means decrease in economic feasibility.

###### Operational Feasibility

How the project will work and who will use it, all concern in this phase. I have to study what the existing system’s problem and is it worth solving or not.

As today’s people are too high-tech and have internet in their home, so they want to learn staying at home. These are the main reasons that encourage people make Online Examination application. Any user whether technically sounds or a dump can use it easily, and to the end. My application will be used by the people who want to prepare for exams. It is browser independent also. At the end it looks operationally feasible.

###### Technical feasibility

What we have planned to implement is technically feasible. Do we have sufficient amount of knowledge or technology to make it reality? And the answer is fairly easy, because we have many working examples of the Online Examination application in which one can select a subject for exam as a forum. I have also scheduled it in a proper way so that no miss coordination occurs between development, design and testing. Now we can conclude that the system is technically feasible.

###### Scheduling feasibility

Here we checked whether our system can be ready in time without any error, or we can extend it in case of errors. We have planned all it phases with keeping this aspect in our mind, so if we found any bug or error after testing phase then we can move our deadline to 2-4 days, as we set our deadline before the actual submission date to the client.

###### Economical feasibility

As we are doing this project for the company(IDP) there is cost concern but if consider the time taken by this project and other resources used to make it reality, determines its cost. And we estimated that there are no economic problems in this project as we can complete it within our semester.

###### Implementation feasibility

As all the requirements made by the client were clear there is no issue related to the implementation feasibility.

So as far as the feasibility of the website is concerned there was no problems as the application was operationally, technically, economically feasible as well as it was feasible with schedule and implementation.

#### REQUIREMENT ANALYSIS:

###### Domain Understanding

Analysis must develop their understanding of the application domain. The separation of the fields of interest from the outside environment is the basic requirement. For this project we have understood many of the domains like how I would handle the entire application? How all kind of data associated with the application would be maintained?

###### Requirement Collection

The conclusion of domain understanding should be resulted into the bulk of information and from that, we tried to carry out the requirements. We gathered after the analysis the domain problems and needs to provide view of them.

After analysis and understanding the domain requirement we are now having a vast amount of data to be manipulated in the Implementation stage.

###### Classification

This activity takes the unstructured collection of requirement and organizes them into coherent clusters.

Requirements are classified as follows:

The main requirements are:

###### User requirements

User requirements are gathered by consulting following things:

* + Visiting the application.
  + Consulting the client.
  + Understanding the clients view.
  + Provides an advertisement.

###### System requirements

System requirement include the software and tools that are to be made available to build up the application and also to put it in to the operation.

Above requirements can be further classified as below.

1. Functional requirements
2. Non- Functional requirements
3. Domain Requirement

###### Conflict Resolution

Here, requirement conflicts are handled so that can distinguish themselves. Like some facility may not be used by other user, there must not be any objection from other users. By distinguish the type of user I have not given all the access to the application for every user who are of type admin have all access.

###### Prioritization

Basically Admin, User, Registration are shown such that user can easily extract the necessary information regarding the various techniques used by Admin.

As the system is not implemented now, the first and the last priority are obviously system analysis and design.

###### Requirement Analysis for my project

Initially when I has chosen the project, I spent some time to find out what exactly are the requirements of the system. After I got an idea of this I started collecting the detailed requirements and noting them down. The basic need for collecting the requirements was to identify all the required modules that can be implemented to make a successful working system. For that, the analysis and design portion should be made in such a way that the programmer is very clear with all aspects of the system to be developed.

###### The different techniques for requirements elicitation and analysis are:

* Questionnaire
* Observation

**Questionnaire:** I had asked the client about how the application should work, also had a discussion with him about the modules that he wanted for the application.

**Observation:** I visited some application that he told me to refer and made close observation about how that application were working and after going through various application I got some information about the application.

#### REQUIRMENT SPECIFICATION:

###### Functional Requirements

Functional requirement**s** define the internal working of the software: that is, the calculations, technical details, data manipulation and processing and other specific functionality that show how the use cases are to be satisfied.

###### The functional requirements of the application are mentioned as follows

* This application provides facilities like new user can easily choose their stream and select their desired Question papers & Syllabus.
* This project understands the Structure of the system.
* In this project, student can read Question Paper & Syllabus of related subject.
* This project is work at on the two sides like admin side and user side.
* In this Application more than one admin is possible.
* In this Application admin can add, update and delete whole Stream.

###### Documentation of Functional Requirements

For documenting the functional requirements, I need to specify the set of functionalities supported by the system. A function can be specified by identifying the state at which the data is to be input to the system, its input data domain, the output data domain and the type of processing to be carried on the input data to obtain the output data.

* + The whole system requires the continuous watch from the admin.
  + Power supply is also the main aspect to perform the task accurately.
  + The transaction must make changes to database of the system.

###### Non-Functional Requirements

The Non-Functional Requirements are as follows: -

###### Maintainability

The Application is easily maintained by the administrator, he will be provided with all the rights and access to the application, the administrator can add, update and delete the information.

###### Usability

This application is very important in providing a common ground for both the administrator and the user so both get the best and filtered or crème as they want depending on the requirement and capability.

###### Accuracy

As I was developing the application, I must make the system that is very accurate in its functions. All the data should keep working properly, they getting perfect input, processing it accurately and producing the perfect output. Accuracy is the most important nonfunctional characteristic or requirement of the system.

###### Interface issue

The interface of the system should be very user friendly, because I am assuming that our most of user are non-technical user. Even operator and managers of the system are not aware of the technical portion.

###### Constraints

Constraints are the field that software is poor in the field; minimization of constraints is must in the process of system developments. This system should be solid in those things; it should pass away all constraints, such as insertion of text in place of numbers, or say automatic type casting.

###### Error Handling

System should restrict to enter wrong data. System should able to prompt various error message to user if they provide wrong inputs of forget to enter the required data.

###### Security

Only authorized users are allowed to access the system. System should provide a way to assign various rights to the various users. Like vendor are only allowed to view various details related to their items and payment and not allow delete/modify any information. Visitors are restricted to enter any data into the system.

###### Resources and Management

System should support backup and recovery at regular interval. For proposed system data is very crucial and losing of data is not acceptable. System should provide weekly, as well as daily backup. Administrator should be responsible person for taking backup on regular interval. Administrator will handle maintenance of the system.

# Chapter-5 Module Description

##### Key Concepts:

* 1. Module Decryption
  2. Admin
  3. Student
  4. ***ADMIN:***

Admin is an authorized part of this application. It’s provides us all details about items. Only admin can make changes in the information (categories).

###### ADD CATEGORIES

By this module admin can easily add new categories.

.

###### DELETE CATEGORIES

In this module admin will delete the items that are not required. With the help of this method a database of this system is easily managed.

###### ADD ITEMS

In this module admin will be able to add new items related to the categories.

###### DELETE ITEMS

In this module admin will be able to delete the unnecessary items from the related categories.

###### UPDATE ITEMS

In this module admin will have the authority to update the items.

###### ADD PHOTOS

In this module admin will be able to add new photos of Hotel Divine.

###### DELETE PHOTOS

In this module admin will be able to delete new photos of Hotel Divine

#### VISITOR:

A visitor can easily get information about Hotel Divine and Table Reservation. In this module a user can see all photos of Hotel Divine.

###### SELECT CATEGORIES

With the help of this module a user can easily select the category of his/her choice.

###### SEE ITEMS

In this module visitor can easily see the things of their choice either menu items.

###### MAKE RESERVATION

In this module visitor can easily book a reservation on his will.

# Chapter-6 System analysis and design

##### Key Concepts:

* 1. Data Dictionary
  2. ER Diagram
  3. Data Flow Diagram
     + Visitor Data Flow Diagram
     + Admin Data Flow Diagram
  4. Process Flow
  5. Use Case
  6. Activity Diagram
     + Visitor Activity Diagram
     + Admin Activity Diagram
  7. Flow Chart

***6.1 DATA DICTIONARY:***

In database management systems, a file that defines the basic organization of a database. A data dictionary contains a list of all files in the database, the number of records in each file, and the names and types of each field. Data dictionaries do not contain any actual data from the database, only book-keeping information for managing it.

Data dictionary do not contain any actual data from the database, only book keeping information for managing it. Without a data dictionary however, a database management system cannot access data from the database.

One benefit of a well-prepared data dictionary is consistency between data items across different tables. For example, several tables may hold telephone numbers; using a data dictionary the format of this telephone number field will be consistent.

* 1. ***DATA DICTONARY***

**Table Name: Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** | **Comments** |
| Admin\_ID | Int (255) | No | None |  |
| Admin Name | Varchar (30) | No |  |  |
| Display Name | Varchar (30) | No |  |  |
| Password | Varchar (30) | No |  |  |

**Table Name: Reservation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** | **Comments** |
| Reservation\_id | Int (11) | No |  |  |
| Name | Varchar (255) | No |  |  |
| Phone | bigint (11) | No |  |  |
| Date & Time | Varchar (255) | No |  |  |
| Email | Varchar (255) | No |  |  |
| Number of Guests | int (11) | No |  |  |
| Booking Type | Varchar (255) | No |  |  |

### Table Name: Menu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** | **Comments** |
| sub\_category\_id | int (50) | No |  |  |
| sub\_category | varchar (255) | No |  |  |
| main\_category | Varchar (255) | No |  |  |

**Table Name: Menu Category**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** | **Comments** |
| Id | int (50) | No |  |  |
| menu\_type | varchar (255) | No |  |  |
| item\_name | Varchar (255) | No |  |  |
| Price | Varchar (255) | No |  |  |

**Table Name: Gallery**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** | **Comments** |
| Gallery Id | int (11) | No |  |  |
| Gallery Name | Varchar (255) | No |  |  |
| Gal Main Photo | Text | No |  |  |

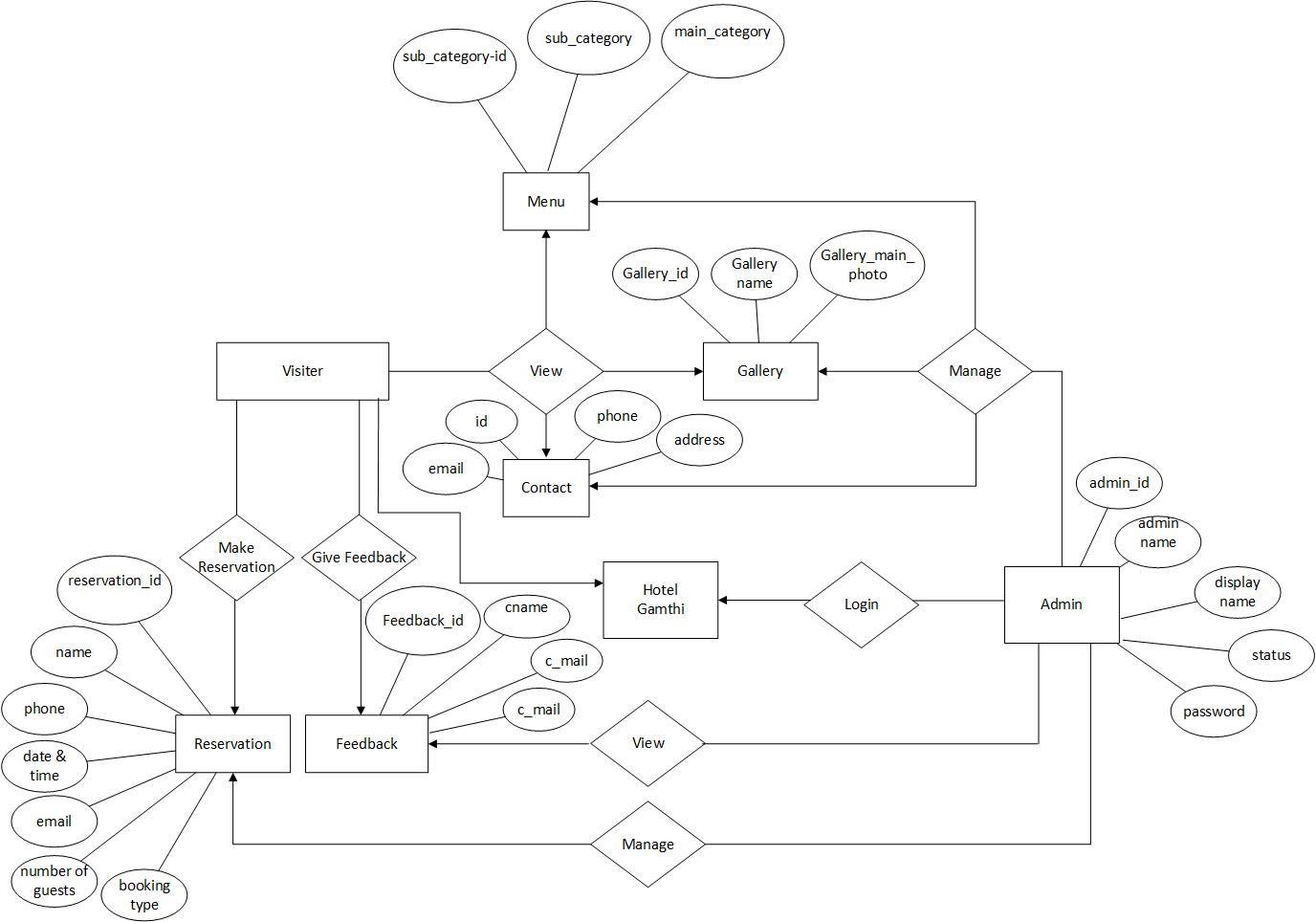
### Table Name: Feedback

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** | **Comments** |
| Feedback\_id | int (11) | No |  |  |
| C\_name | Varchar (255) | No |  |  |
| C\_mail | Varchar (255) | No |  |  |
| C\_msg | Text | No |  |  |

**Table Name: Contact**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** | **Comments** |
| C no | int (11) | No |  |  |
| C name | Varchar (30) | No |  |  |
| C mail | Varchar (30) | No |  |  |
| C Msg | Varchar (500) | No |  |  |

#### ER diagram:

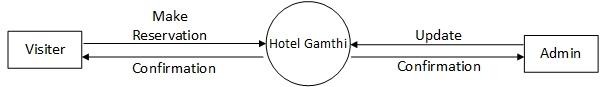


Hotel

Divine

Fig 6.2 ER diagram

* 1. ***Data Flow Diagram:***



Hotel Divine

***Fig 6.3 Data Flow Diagram***

#### 6.3 Visitor Data Flow Diagram

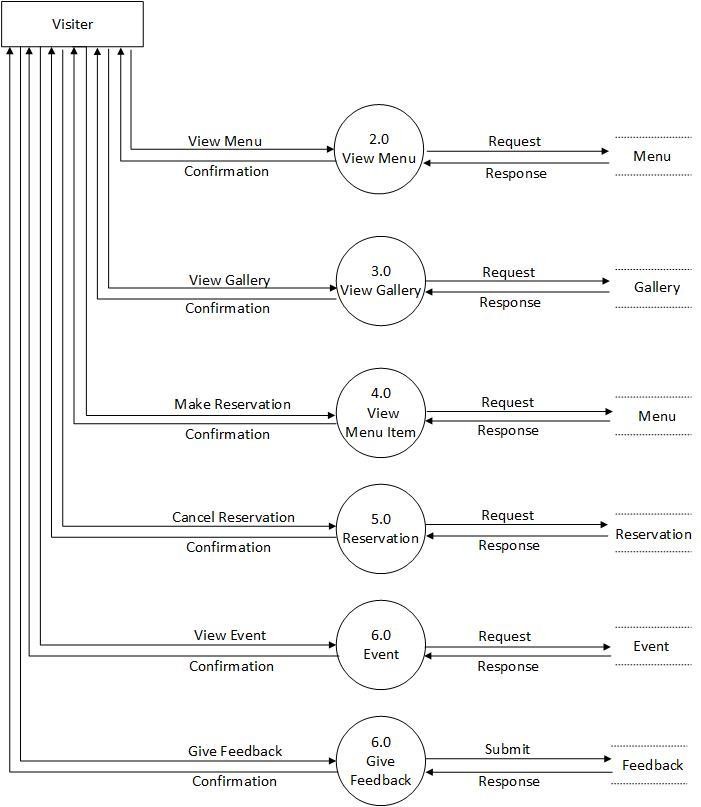


Fig 6.3 Visitor Data Flow Diagram

#### Admin Data Flow Diagram

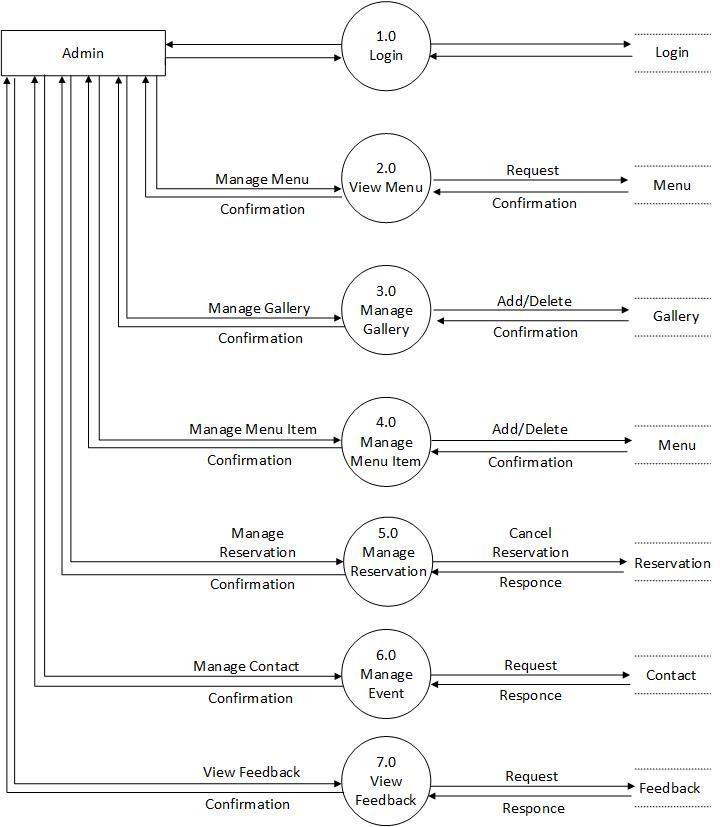
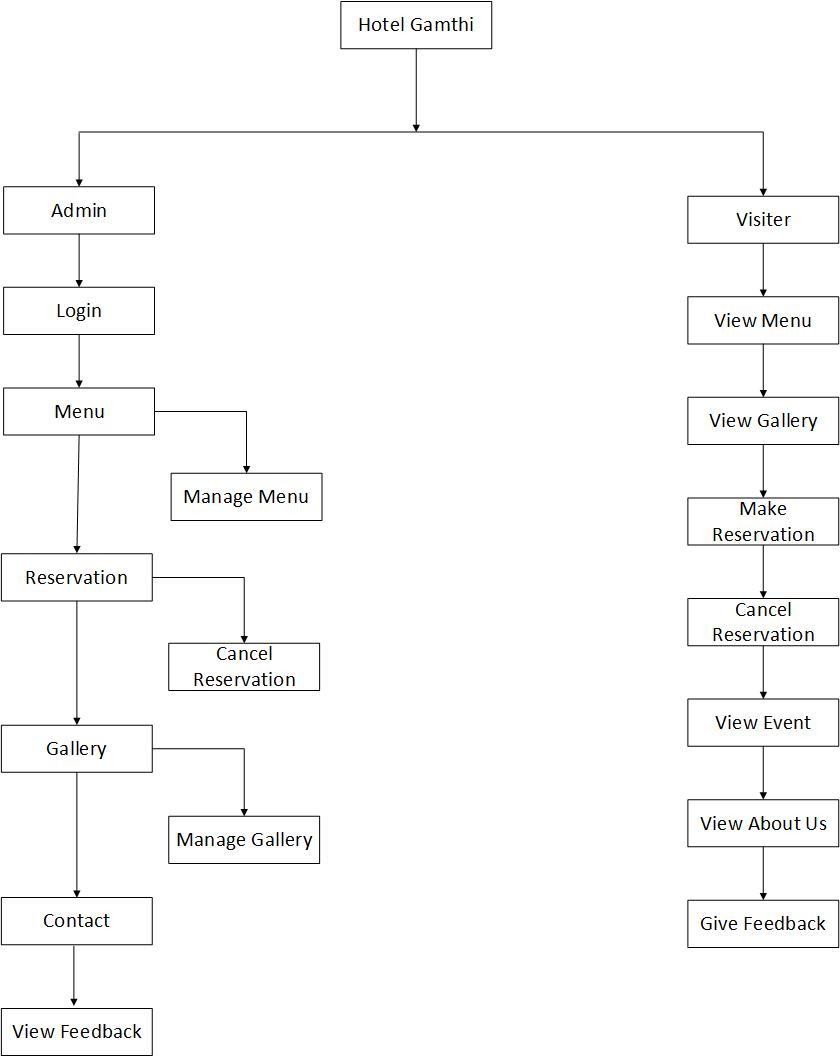


Fig 6.3 Admin Data Flow Diagram

#### Process Flow Diagram:



Hotel Divine

Fig 6.4 Process Flow Diagram

#### Use Case Diagram:

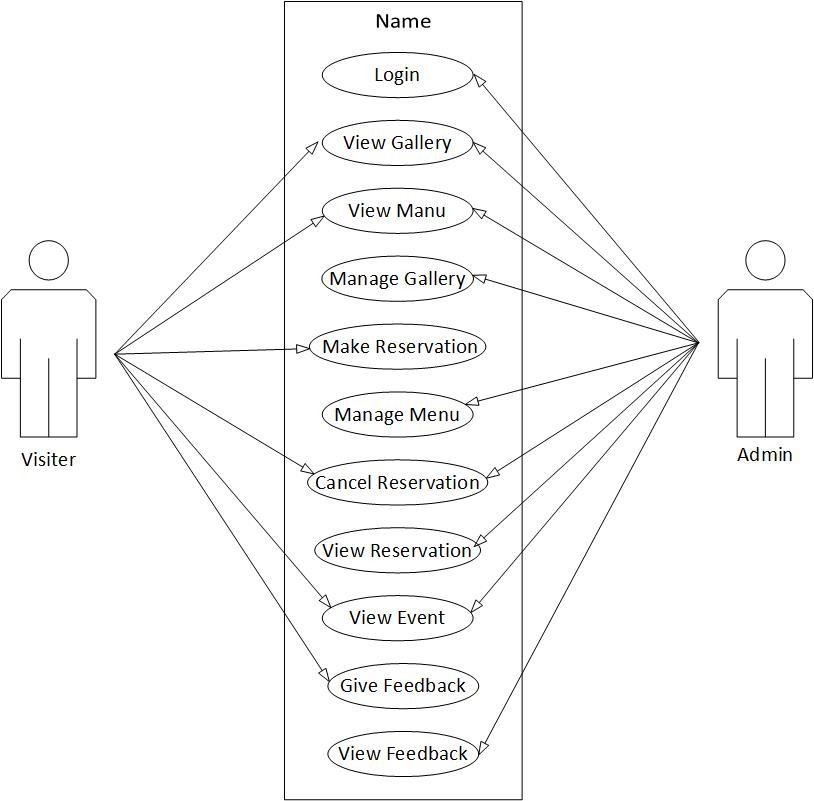


Fig 6.5 Use Case Diagram

#### Activity Diagram For USER:

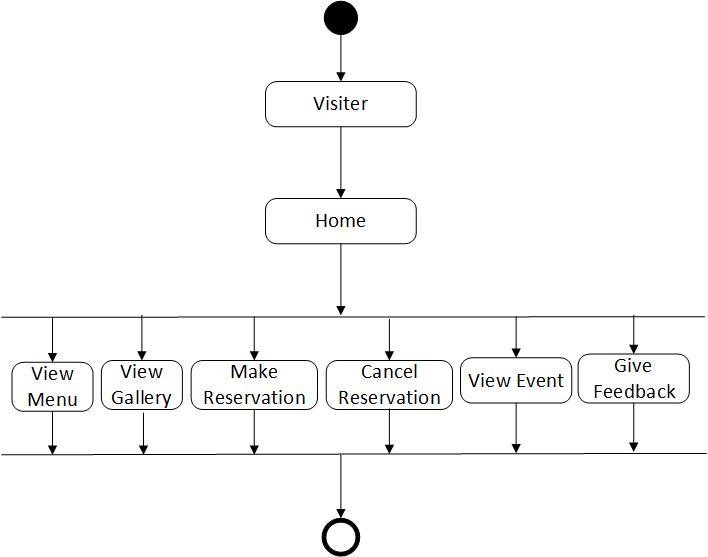


Fig 6.6 Activity Diagram for User

#### Activity Diagram for Admin:

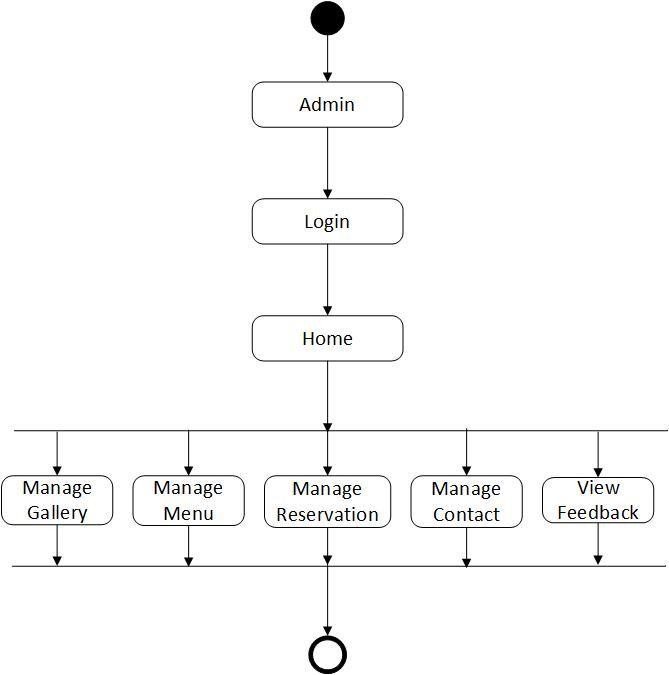


Fig 6.6 Activity Diagram for Admin

#### Flow Chart:

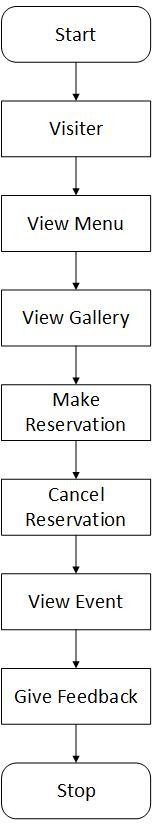
* + - ***Visitor Flow Chart:***

Fig 6.7 Visitor Flow Chart

#### Admin Flow Chart:

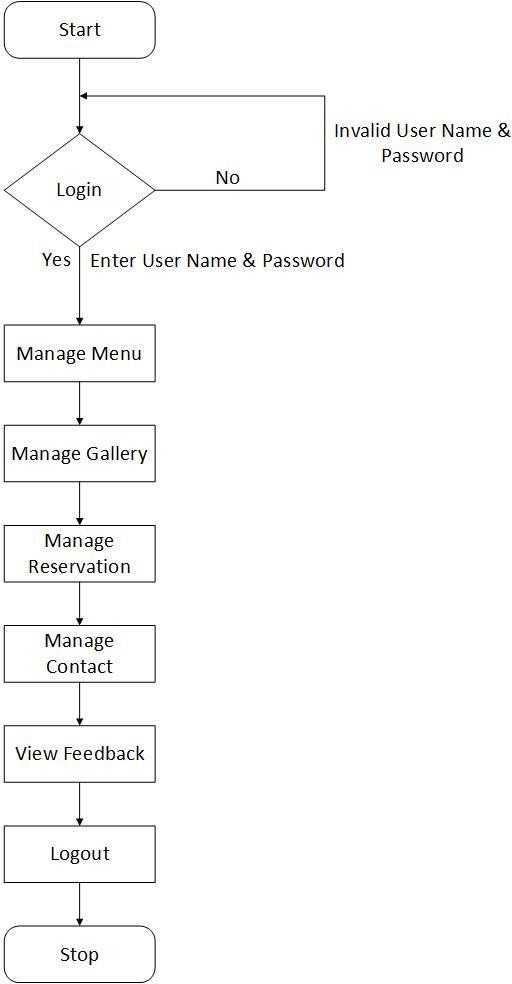


Fig 6.7 Admin Flow Cha

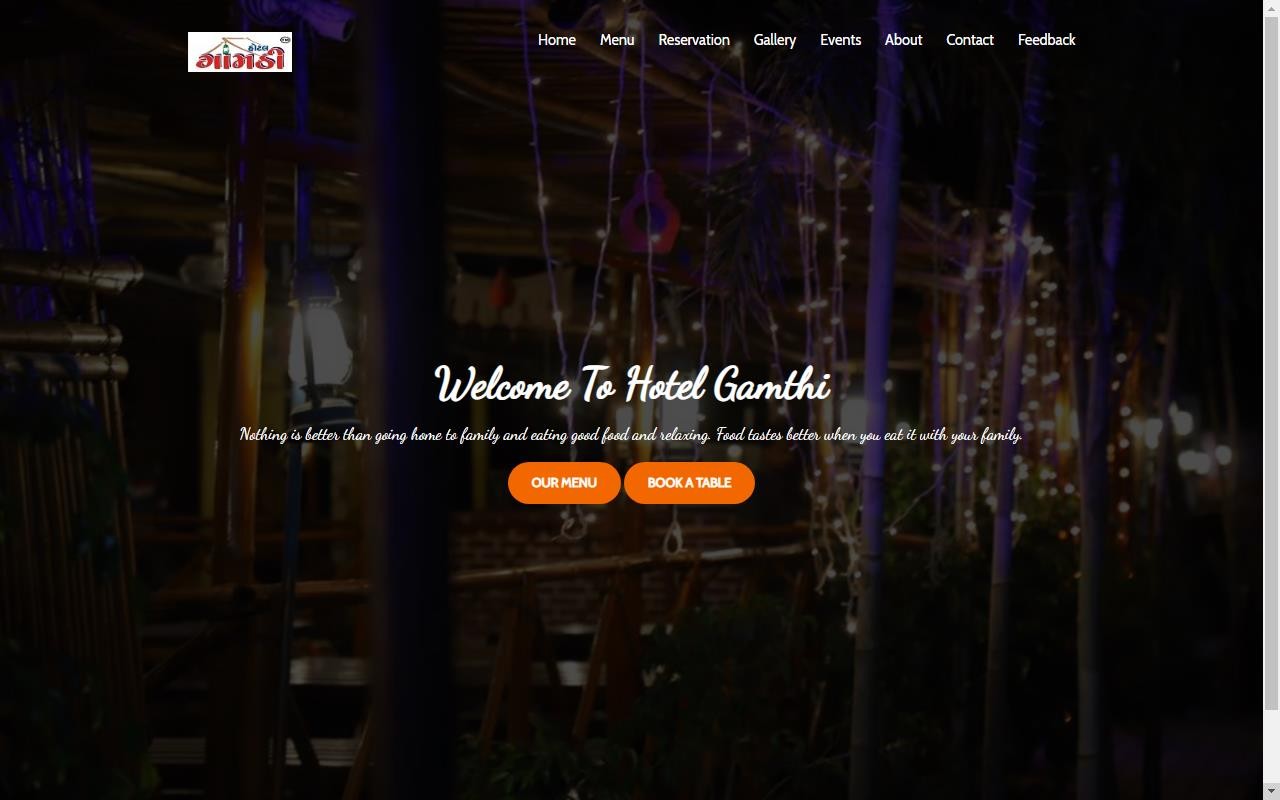
# Chapter – 7 User Manuals

##### Key Concept:

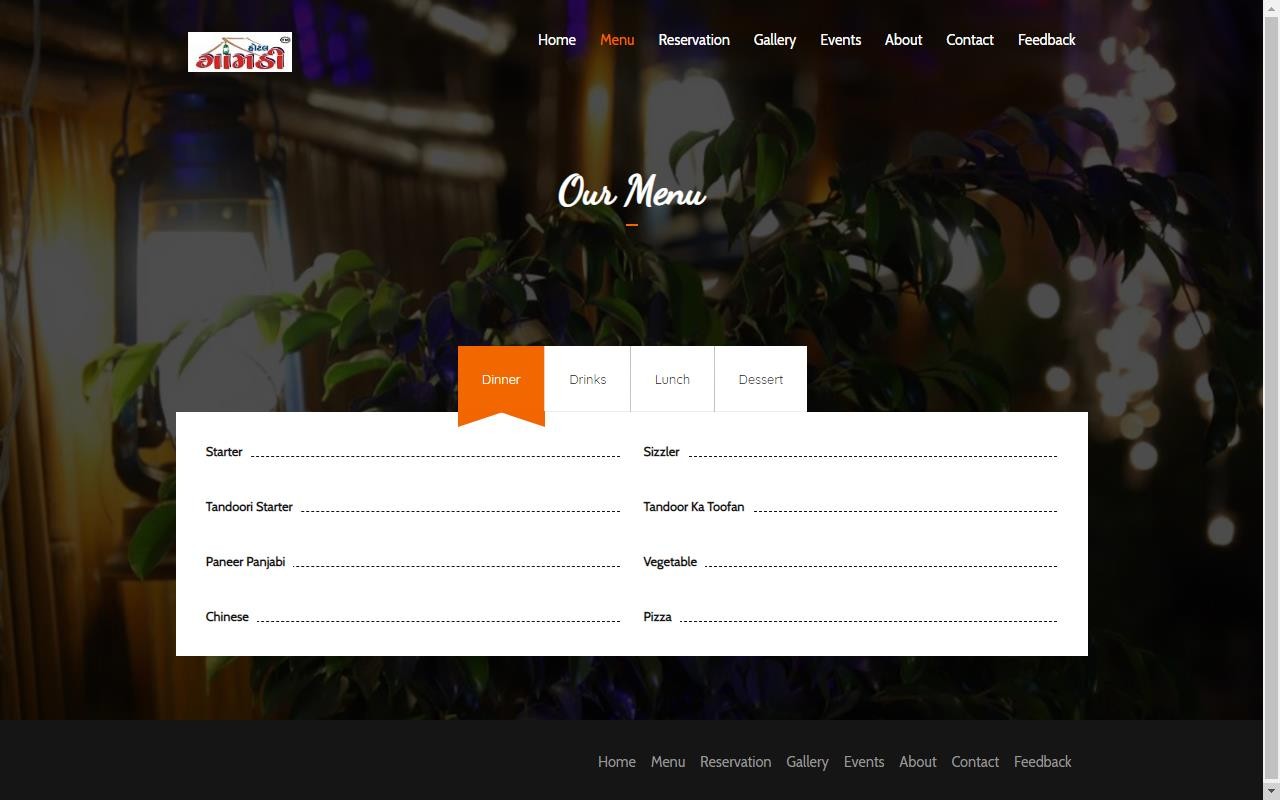
**7.1** Screenshots

***User side:***

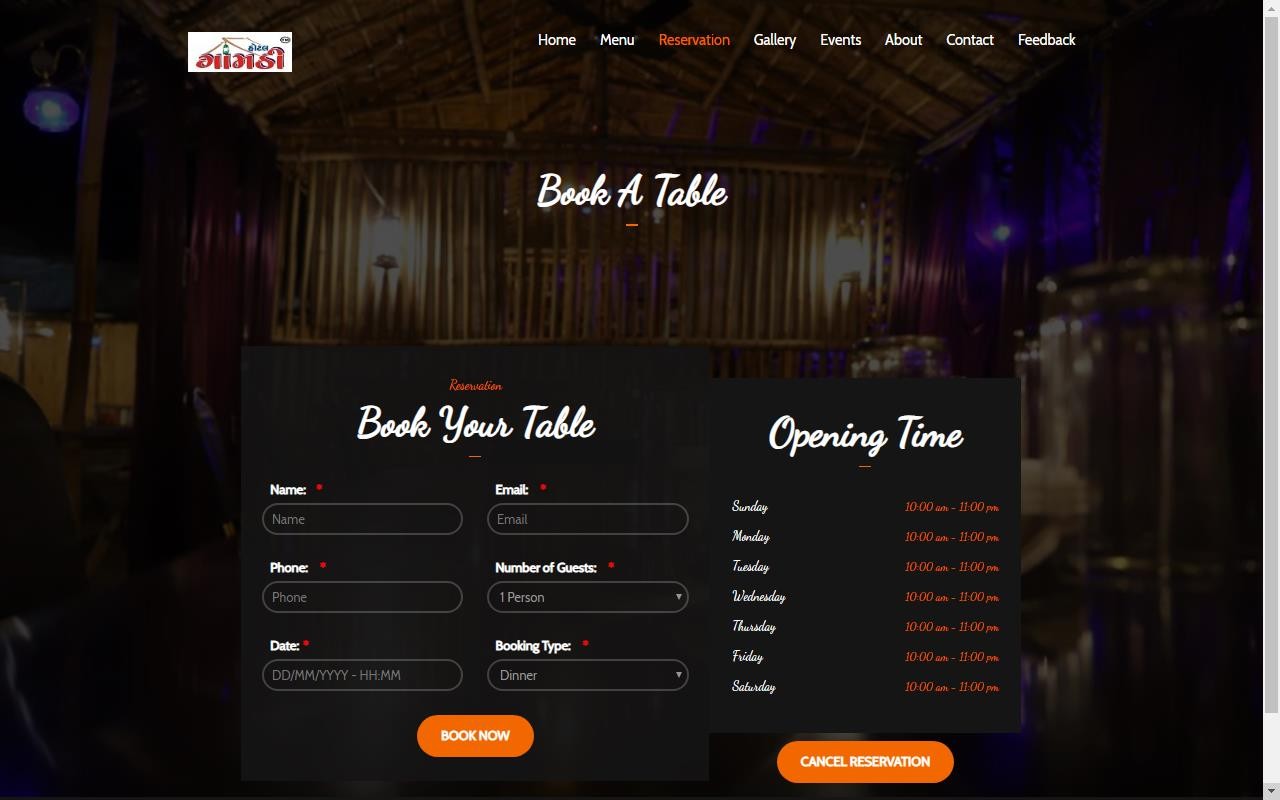
* ***Home Page:***



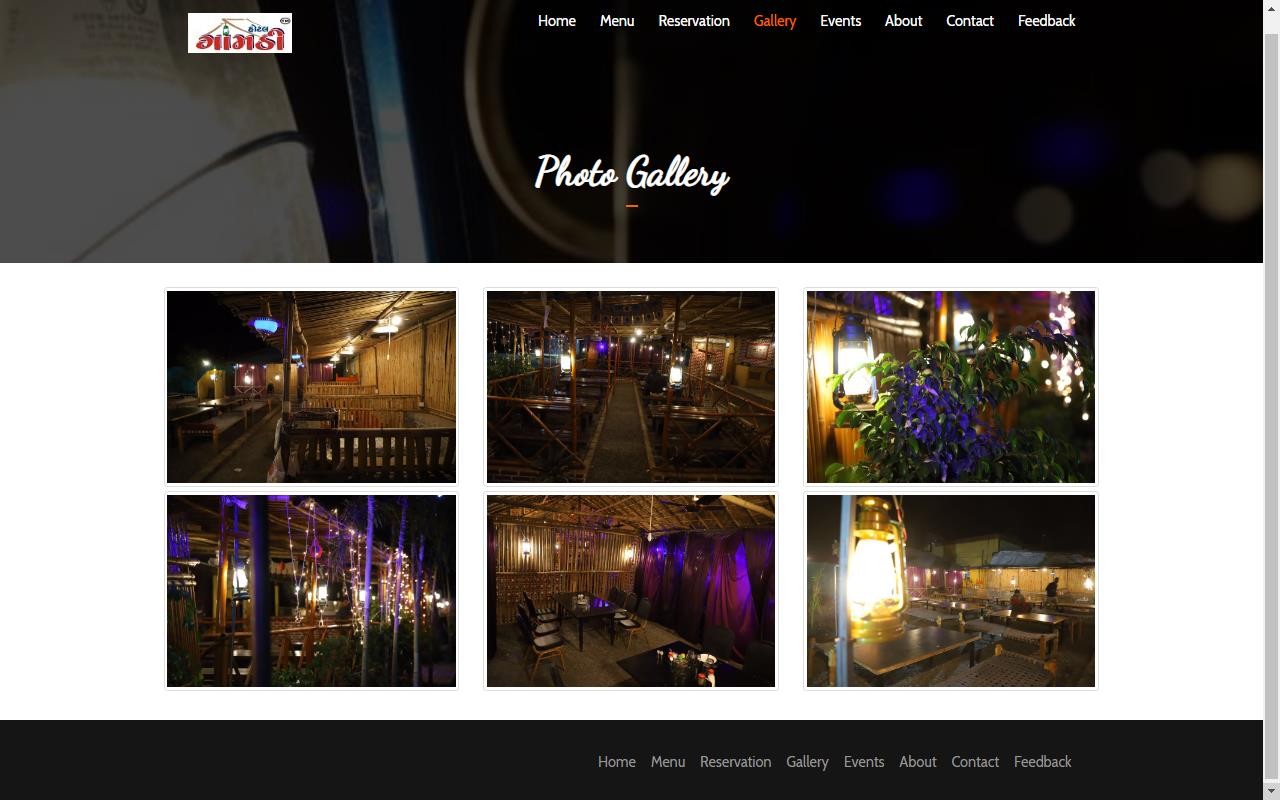
* ***Menu Page:***



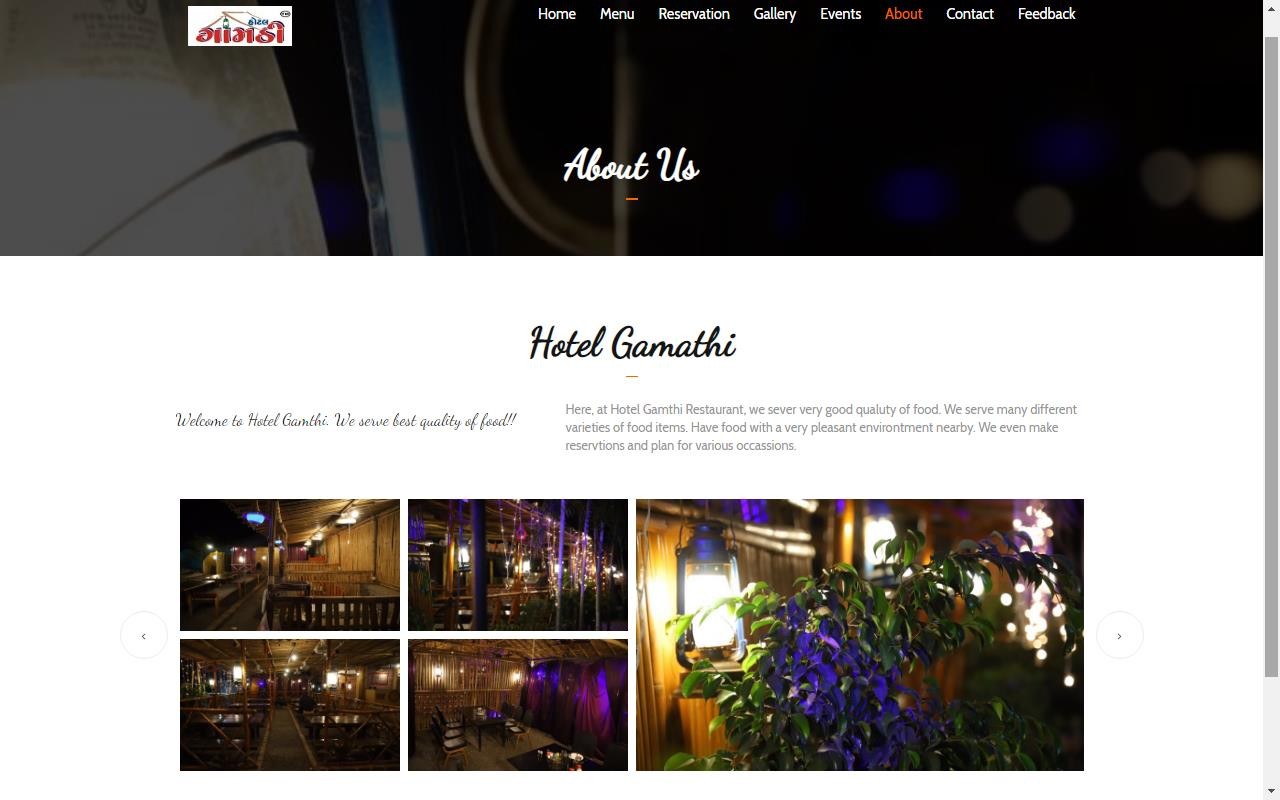
## Reservation Page:



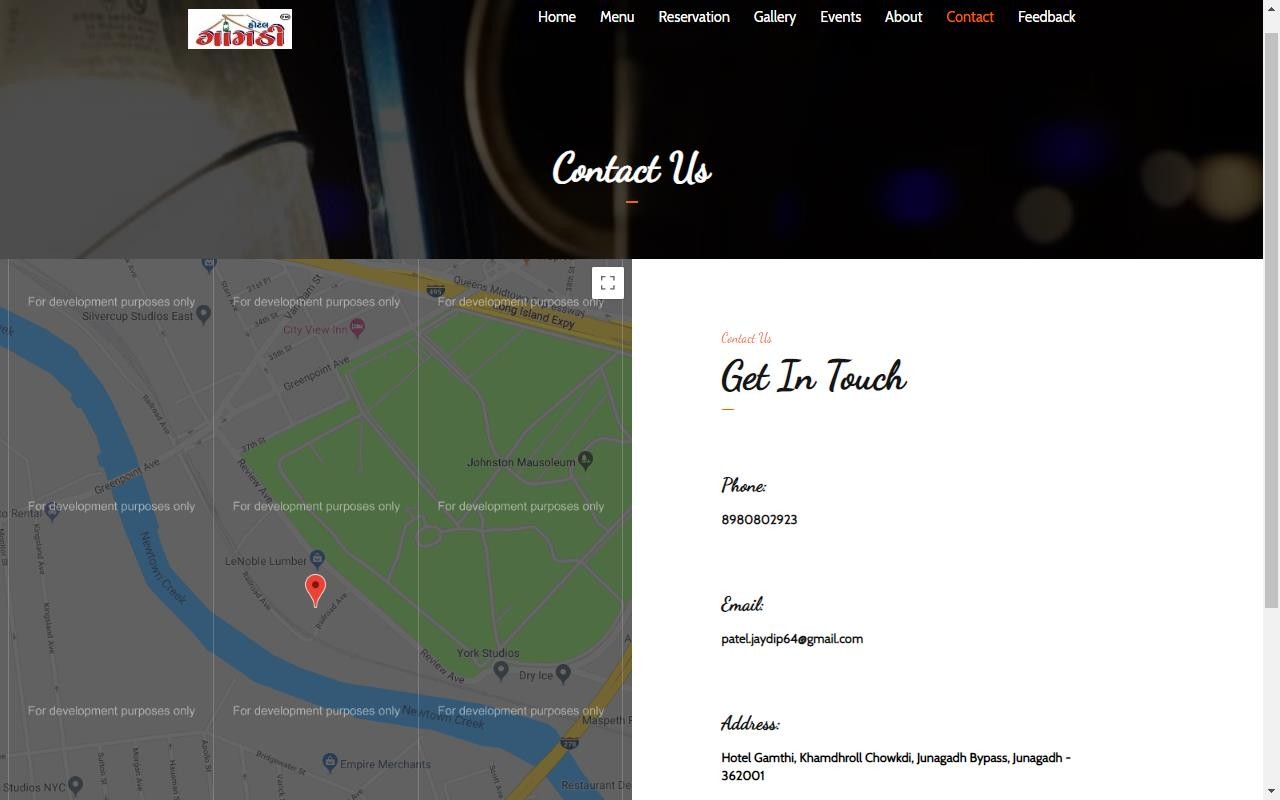
* ***Gallery Page:***



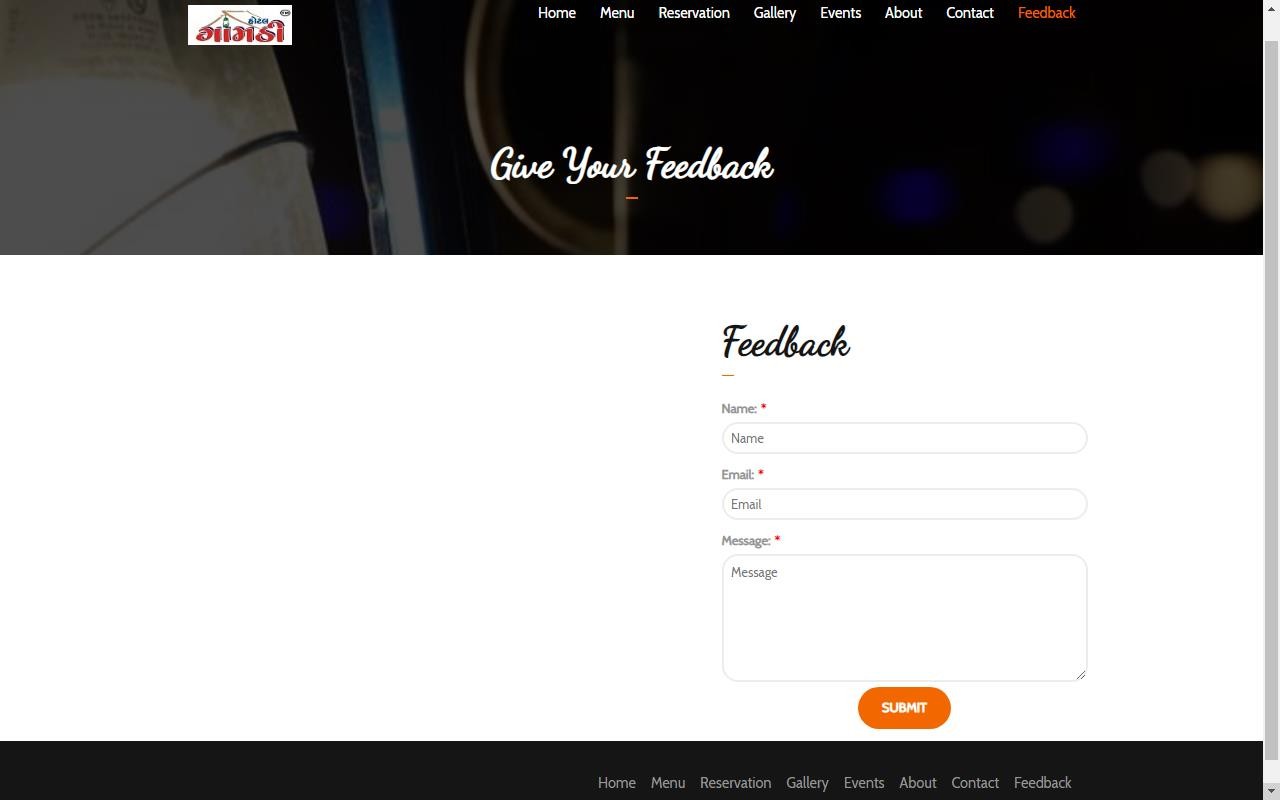
## About Page:



* ***Contact page:***

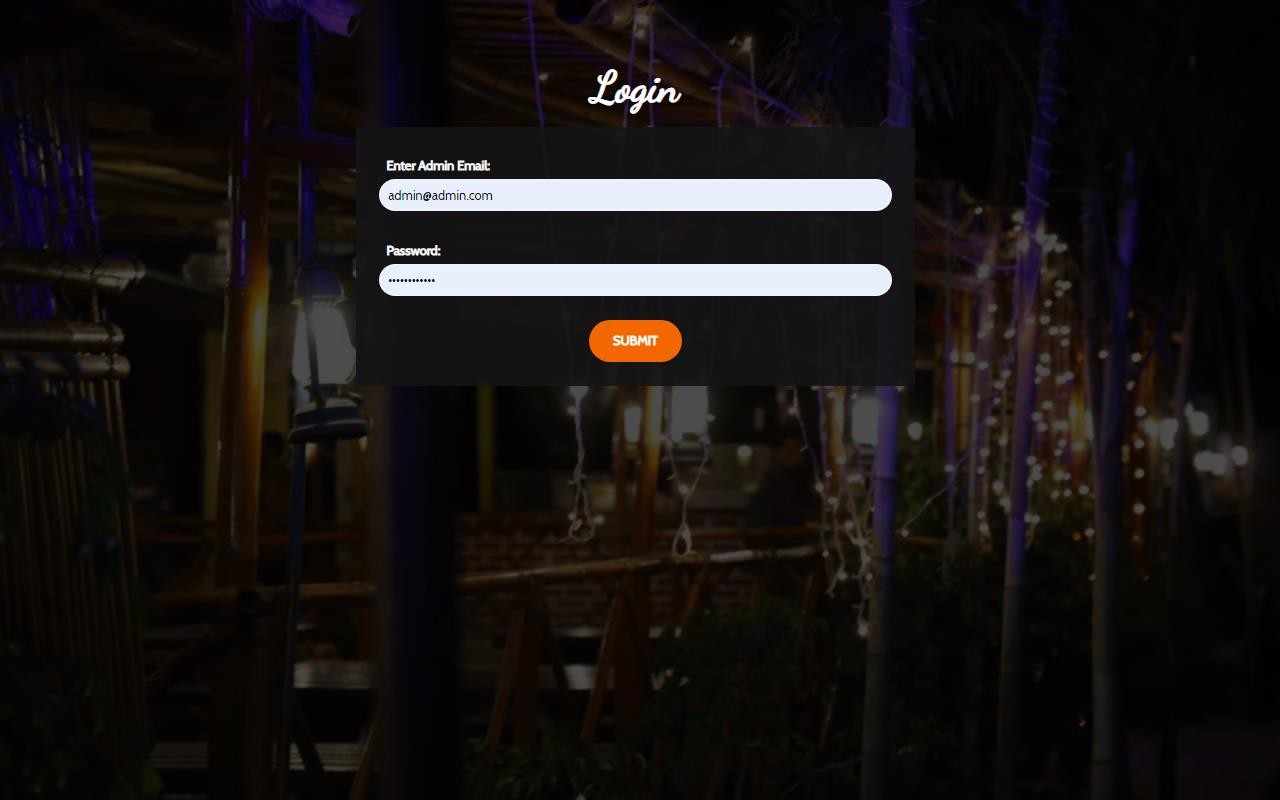


## Feedback Page:



## Admin Side:

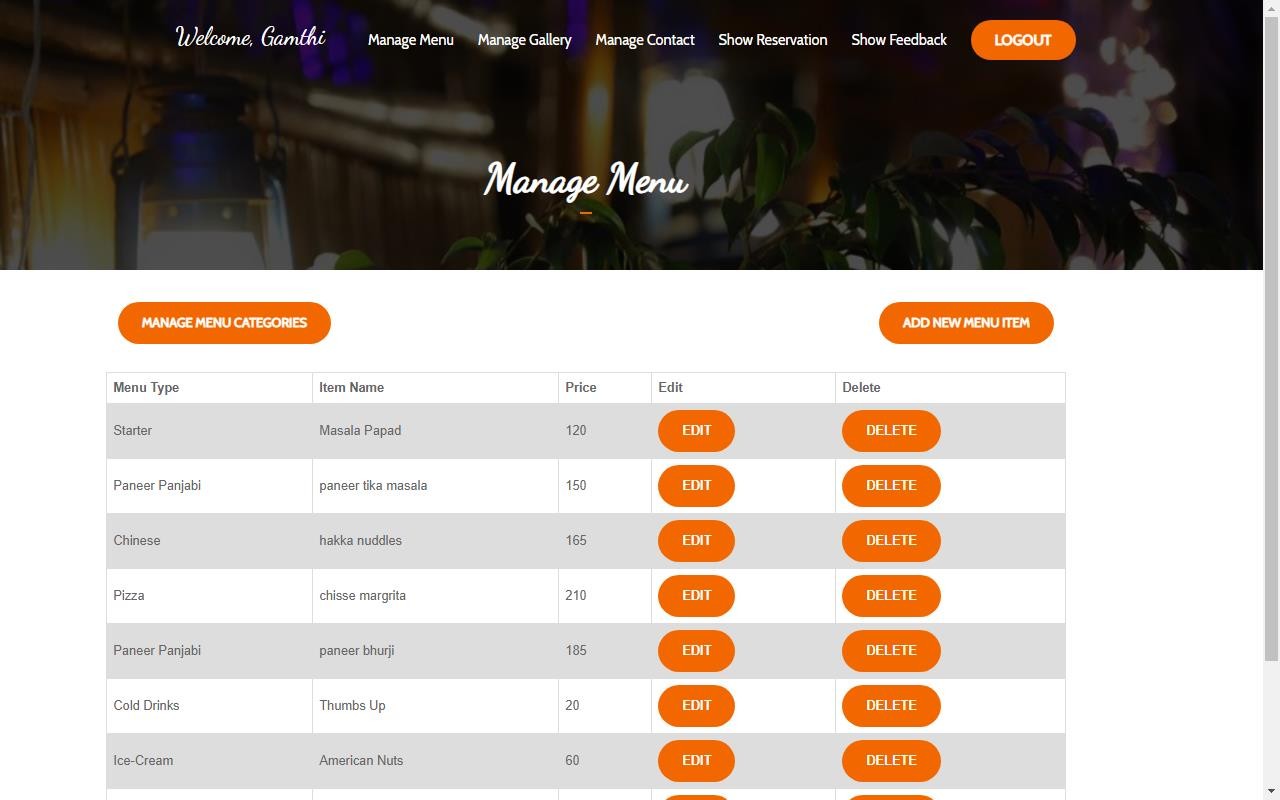
* ***Admin Login Page:***



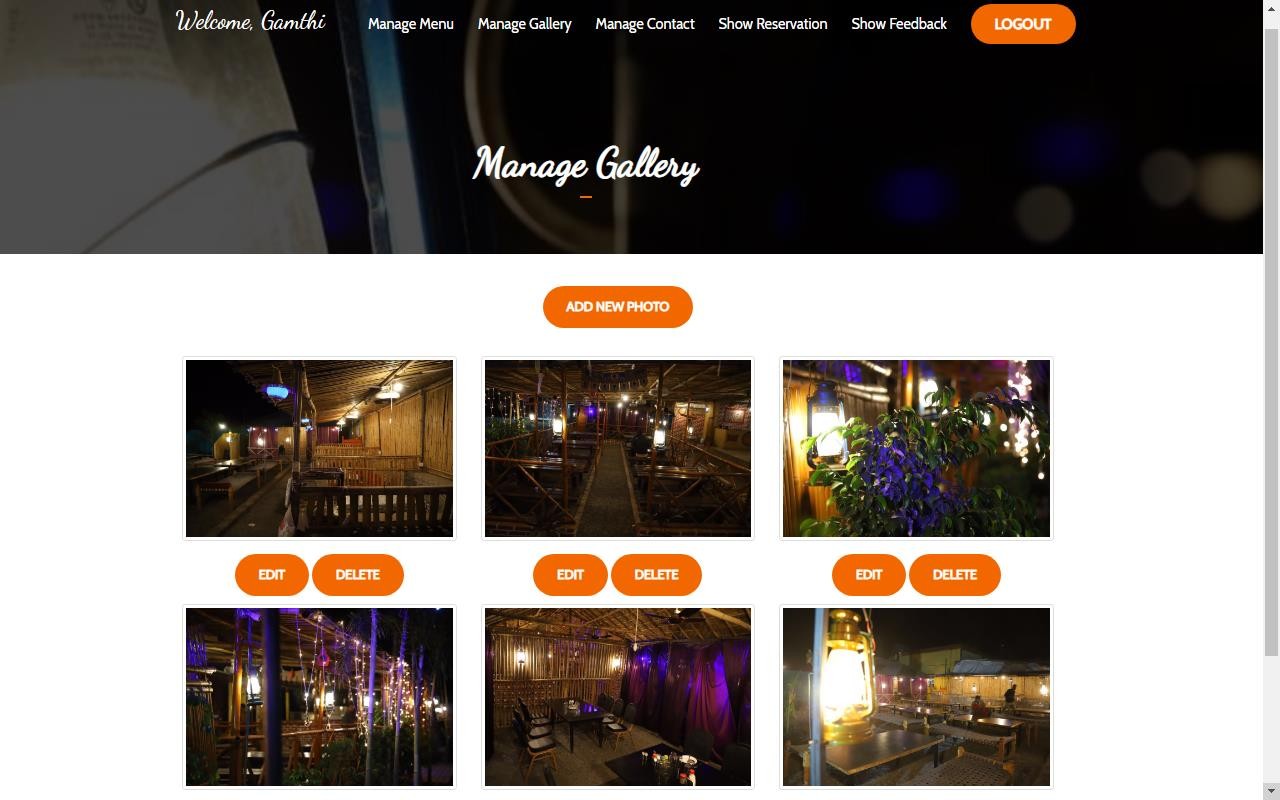
* ***Admin Welcome Page***



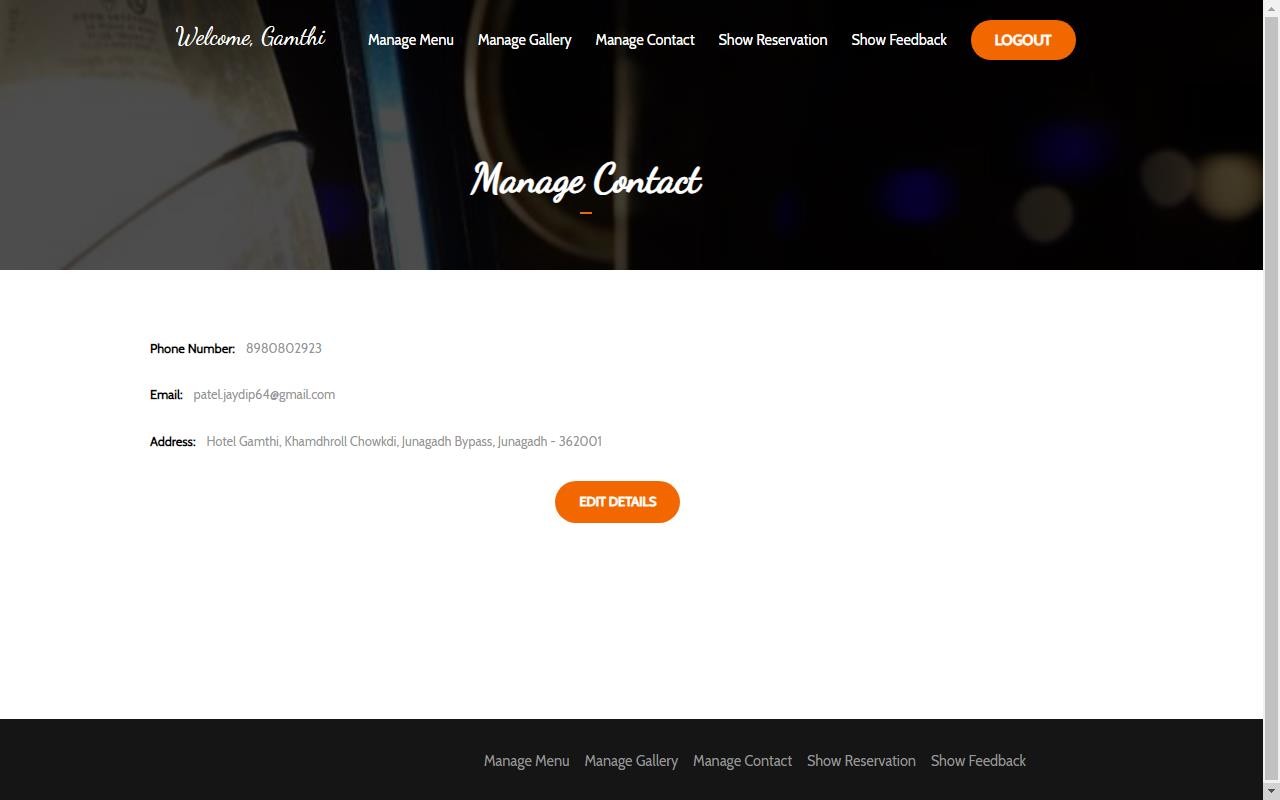
## Manage Menu:

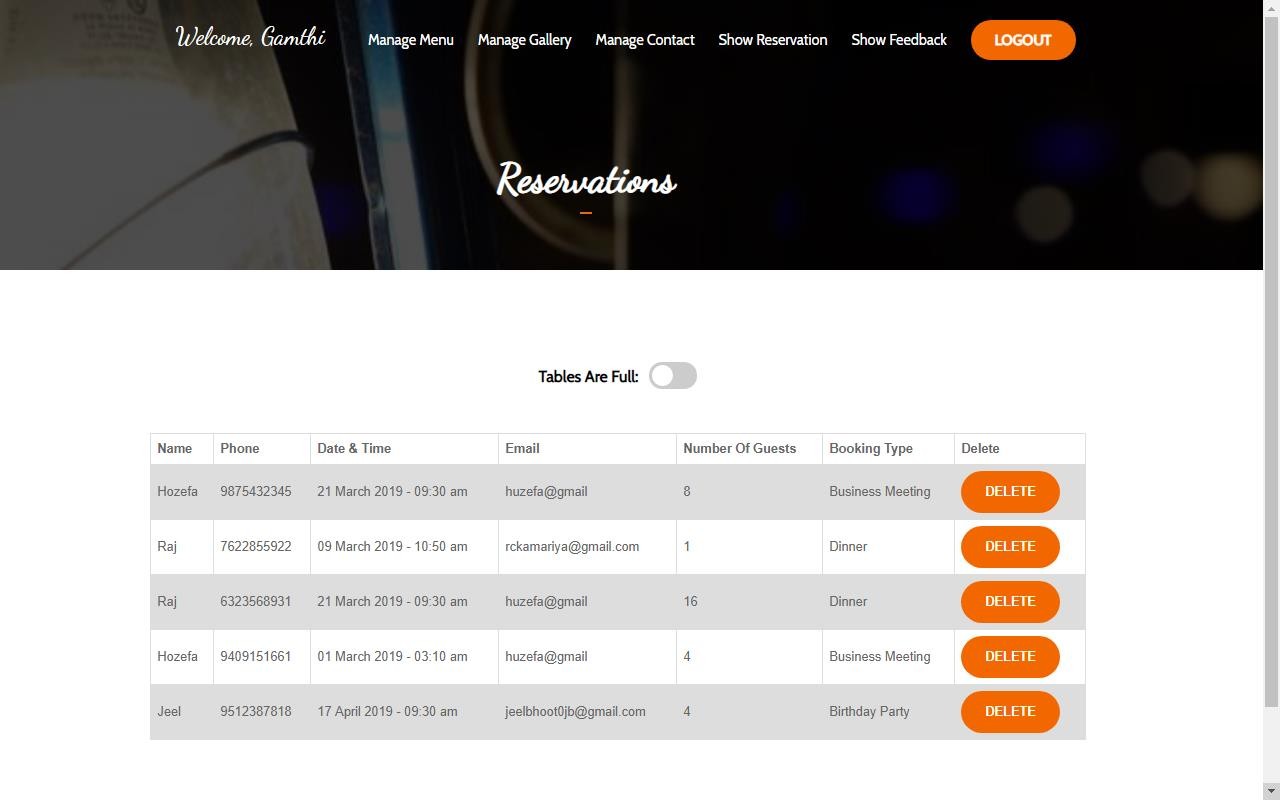


* ***Manage Gallery:***

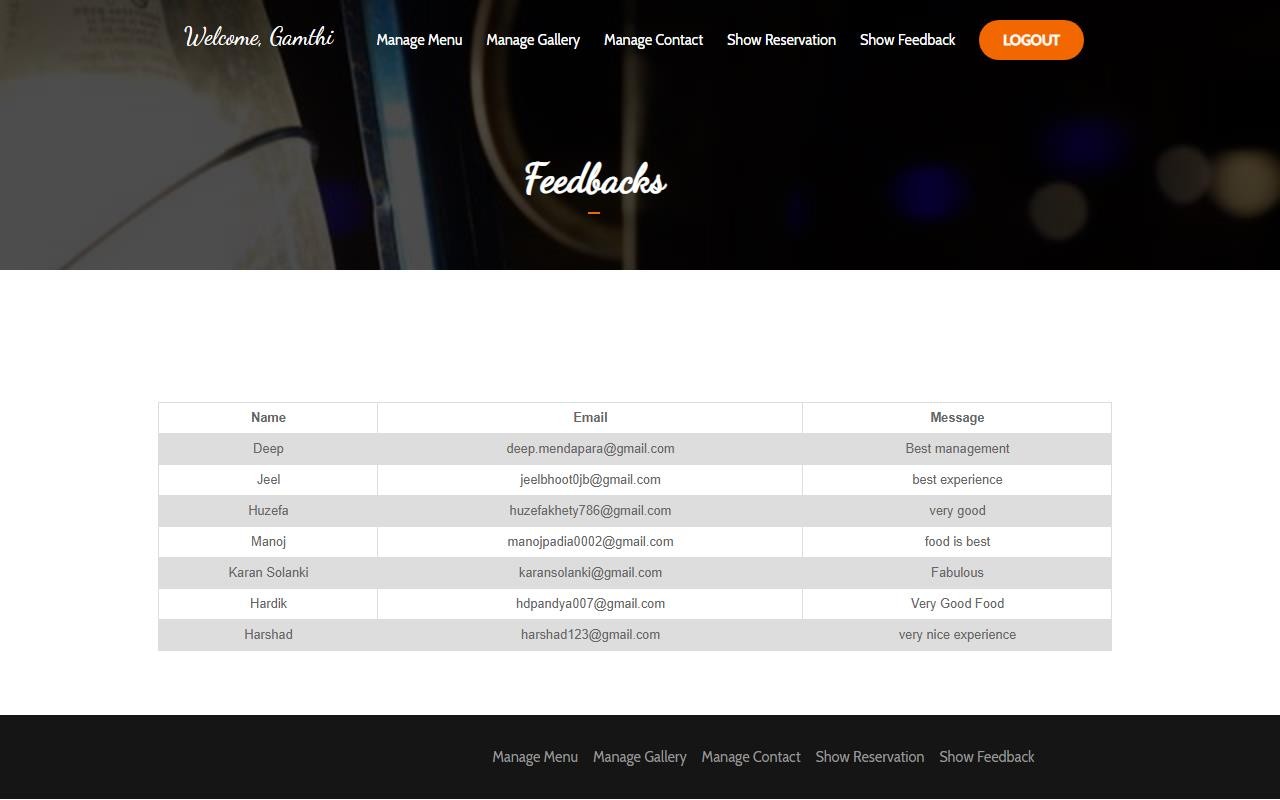


## Manage Contact:



* ***Manage Reservation:***

## Admin Feedback Page:



# Chapter: -8 Verification and Validation

##### Key Concepts:

* 1. Verification and Validation
     1. Unit testing
     2. Sub system testing
     3. System testing
     4. Acceptance testing
  2. Testing
     1. Black box testing
     2. White box testing
     3. Design of test cases
  3. ***VERIFICATION AND VALIDATION:***

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery to customer. Your goal is to design a series of test cases that have a high likelihood of finding errors. Software testing techniques provide systematic guidance for designing tests that (1) exercise the internal logic of software components, and (2) exercise the inputs and outputs domains of the program to uncover errors in program function, behavior and performance.

During early stages of testing, a software engineer performs all tests. However, as the testing process progresses, testing specialists may become involved. Reviews and other activities can and do uncover errors, but they are not sufficient. Every time the program is executed, the customer tests it! Therefore, you have to execute the program before it gets to the customer with the specific intent of finding and removing all errors. In order to find the highest possible number of errors, tests must be conducted systematically and test cases must be designed using disciplined techniques.

Component

System

Acceptance

Testing

Testing

Testing

Fig 8.1 The Testing Process

Testing Objective

* + - Testing is a process of executing a program with intent of finding an error.
    - A good test case is one that has a high probability of finding an as-yet undiscovered error.
    - A successful test is one that uncover as as-yet undiscovered error.
    1. **Unit Testing**

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually scrutinized for proper operation. Unit testing is often automated but it can also be done manually. This testing mode is a component of Extreme Programming (XP), a pragmatic method of software development that takes a meticulous approach to building a product by means of continual testing and revision.

Unit testing involves only those characteristics that are vital to the performance of the unit under test. This encourages developer to modify the source code without immediate concerns about how such changes might affect the functioning of the units or the program as a whole. Once of whole of the units in a program have been found to be working in the most efficient and error free manner possible, larger components of the program can be evaluated by means of integration testing.

I tested each single part of the whole application, both on the admin side and front side; I tested each and every module individually. On admin side tested modules like admin. In Admin Panel, admin can add, edit and delete the subjects ,questions and books. I tested the user side fully. Similarly, for every module I have done Unit testing while coding and before submitting a demo. So, most of the errors have been removed from the application.

**Unit**

**Testing**

**Module**

**Testing**

**Sub-System**

**Testing**

**System**

**Testing**

**Acceptance**

**Testing**

***Fig 8.2 Types of Testing***

###### Sub System Testing

After testing each Unit, we move on to larger units called sub system. In Exclusive Collection application, there was admin side and User side so after unit testing of each module on both sides. In subsystem testing I tested the whole admin side as one system and them the User side as one whole system.

I developed each sub-system such as admin side and user side individually, so tested also at the development time. These sub-systems works fine alone, but after integrating within the application I found some errors, that is why we done integrating testing after integrating these Sub-system.

###### System Testing

After testing all the sub-system, it is time to test the whole system. System testing of software is testing conducted on a complete, integrated system to evaluate the system’s compliance with its specified requirements. While testing the whole system I found many errors like changes made on the admin side were not reflected on the front side or it was that data on the user side was not same as on the admin side. I worked on each error and exception that I got while testing and most of them are removed or made such correction that it will not happen again. Exceptions also arise when there was no access available to the database and the application was not able to read the data from database. Such types of modification are done by me to take system reliable and error free.

* + - * **Recovery Testing:** It is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed.
      * **Security Testing:** It attempts to verify that protection mechanisms build into a system will, in fact, protect it from improper penetration.
      * **Performance Testing:** It is designed to test the run-time performance of software within the context of an integrated system performance testing occurs throughout all step in the testing process

###### Acceptance Testing

Acceptance testing can be connected by the end user, customer, or client to validate whether or not to accept the product. Acceptance testing may be performed as part of the hand-off process between any two phases of development. The acceptance test suite is run again the supplied input data or using an acceptance test script to direct the tester. Then the results obtained are compared with the expected results. If there is a correct match for every case, the test suite is said to pass.

I had provided demo to our client at the regular interval of time with my project Manager. So they have complete site of the whole project from the initial stages whatever changes we made in between demo interval, were also been informed to client regularly, so they don’t get surprised by seeing new functionality.

#### TESTING:

The verification activities fall into the category of static testing. During static testing, you have a checklist to check whether the work you are doing is going as per the set standards of the organization. These standards can be for coding, integrating and deployment. Reviews, Inspection’s and Walkthroughs are static testing methodologist. Dynamic testing involves working with the software giving input values and checking if the output is as expected. These are the validation activities. Unit test, integration test. System and acceptance tests are few of the dynamic testing methodologies.

Alpha & beta testing: the alpha test is conducted at the developer’s site by a customer. The software is used in a natural setting with the developer “looking over shoulder” of the user and recording errors and usage problems. Alpha test is conducted in a controlled environment. The beta testing is conducted at one or more customer site by the end-user of the software. Unlike alpha testing, the developer is generally not present. Therefore, the beta test is a “live” application of the software in an environment that cannot be controlled by the developer.

###### Black box testing

Also known as functional testing. A software testing technique where by the internal working of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs. The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specification.

The advantages of this type of testing include:

* + - * The test is unbiased as the designer and the tester are independent of each other
      * The tester does not need knowledge of any specific programming languages
      * The test is done from the point of view of the user, not the designer
      * Test cases can be designed as soon as the specifications are complete The disadvantages of this type of testing include:
      * The test can be redundant if the software designer has already run a test case
      * The test cases are difficult to design
      * Testing every possible input stream is unrealistic because it would take an inordinate amount of time: there for, many program paths will go unteste

###### White box testing

Also known as glass box, structural, clear box and open box testing. A software testing technique where by explicit knowledge of the internal workings of the item being tested are used to select the test data. Unlike black box testing, 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 than see if the program diverges from its intended goal.

###### Design of test Cases

To minimize the number of errors in software, a reach variety of test design methods have evolved for software. These methods provide the developer with a systematic approach to testing. More important, methods provide a mechanism that can help to ensure the completeness of test and provide the highest likelihood for uncovering errors in software.

An engineering product can be tested in one of the two ways: (1) knowing the specified function that product has been designed to perform, tests can be conducted that demonstrate each function is fully operational while at the same time searching for errors in each function: (2) knowing the internal workings of a product, tests can be conducted to ensure that “all gear mesh“, that is, internal oppression are performed according to specifications and all internal components have been adequately exercised. Here are the test cases that we had made for our application.

**Test Case No:** TC 01

**Case Name:** Login (Admin/User)

|  |  |
| --- | --- |
| **PURPOSE** | Allow authorized users to login into the system. It will not allow the unauthorized users. |
| **INPUT** | User Name & Password |
| **EXPECTED OUTPUT** | The Students can appear the exam and can check their result. |

* + - * If the input is not correct the admin will not be allowed to login.

**Test Check List:**

**Completion: Tested Not Tested Not Applicable Remarks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Screen/ Report contains all data elements: | Yes |  |  |  |
| Test for standardization: | Yes |  |  |  |
| Fonts (size/ type/ style): | Yes |  |  |  |
| Alignment: | Yes |  |  |  |
| Color combination: | Yes |  |  |  |
| Validation on each data element: | Yes |  |  |  |
| Data type validation: | Yes |  |  |  |
| Checking of Communication between Peripheral Device and Software: |  |  | Yes |  |
| Reference validation: |  |  | Yes |  |
| Existence/ Non-existence validation: |  |  | Yes |  |
| Default value validation: | Yes |  |  |  |
| Computational Test: | Yes |  |  |  |
| Navigational Test: |  |  | Yes |  |
| Mouse movement/ check: | Yes |  |  |  |
| Keyboard test: | Yes |  |  |  |
| Functionality of User Object Test: | Yes |  |  |  |
| Test for functions/ procedures used: | Yes |  |  |  |
| Test for data update: | Yes |  |  |  |
| Test for query criteria: | Yes |  |  |  |

# Chapter – 9 Implementation Planning

##### Key Concepts:

* 1. Implementation Environment
  2. Coding Standards

###### Core Function Implementation:

First, I decided the core function of the system which will facilitate the further implementation and makes it smoother. These steps involved prototype development and Method implementation.

* + - * Prototype development means creating syntax for each core function.
      * Method coding means coding for each method.

Implementation phase requires precise planning and monitoring mechanism in order to ensure schedule and completeness. I developed the software in various sub phases in implementation phase. Steps are follows:

###### Database implementation:

This Phase involved creation of database table and specifying relationships among them in MY SQL SERVER. Our whole application contains 6 tables.

* 1. ***IMPLEMENTATION ENVIRONMENT:***

###### With GUI

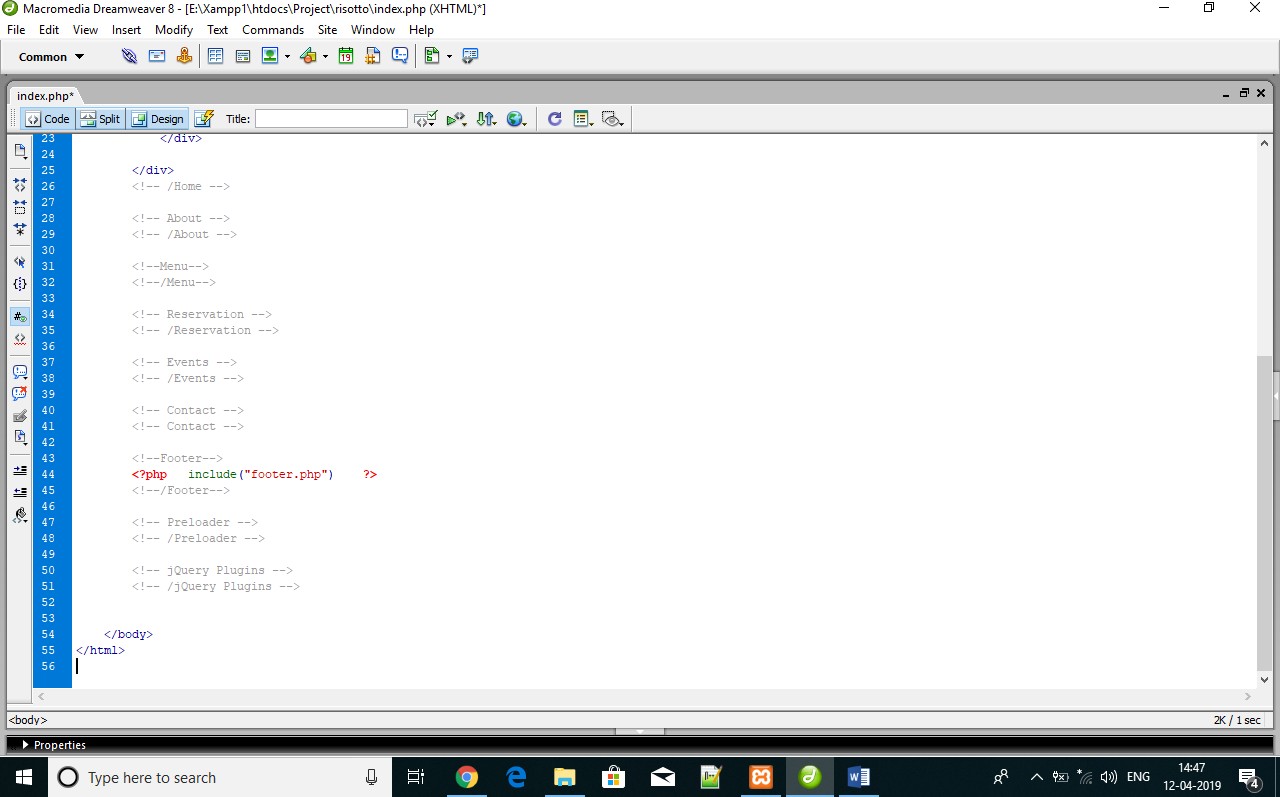
They are relatively easy to learn and use. User with no computing experience can learn to use the interface brief training session.

The user uses multiple screens for system interaction. Switching from one task to another is possible without losing sight of information generated during the first tasks.

* Fast access of data from the device, so its effect on the sport reflects on GUI purposively.
* There are multiple users at the time. But all records are handles by user-id so if any user change any kind of system it does not affect other users result.

Good and well-structured user interface design provided.

#### CODING STANDARDS:



# Chapter-10 Limitations, Future Enhancement

##### Key Concepts:

* 1. Limitation
  2. Future Enhancement
  3. ***LIMITATION:***
     + This web site is accepting previous time date.
     + Admin would have to operate toggle switch for table reservation full and table available.

#### FUTURE ENHANCEMENT:

* + - High security
    - In this system Administrator can’t give any feedback response will try to develop some logic to give feedback response to user.
    - Visitor will be able to give feedback.
    - We will develop an Android Application of our project with all functionality.

# Chapter-11 Conclusion

##### Key Concept: -

**11.1** Conclusi

***11.1 CONCLUSION:***

By developing this system, we conclude that we learn how to develop actual website and how the system is analyzed and implemented. We also got knowledge about system analysis and designing and how to develop ANDROID application

It was a really good experience for us to do a project in this semester. Working on this project has given us a valuable experience. It has been like stepping on the first step of the staircase that leads us towards building our career. It was our first experience of working in the atmosphere of a software firm.

At the time of practically executing our knowledge, we fortunate to have very co-operative and supportive project leaders and colleagues, their attitude towards us was very palliative and was always there in our needs.

We also learnt how to co-operate with other employees while working in the same project. This precious experience would definitely be helpful to us for future.

We made one application for public information. During the development of application we learnt different other things like to create application and get other knowledge.

In future if we get opportunity to develop other application then we can easily do it better. The advantage of this website in that we can easily get the information about movies and book that show at sitting home. In this project we also use different kind of database connectivity with greater feature.

We have completed this project work in less time period but it is effective and efficient during this project work. We also got other knowledge which is useful for us to make project.

# Chapter-12 Bibliography

##### Key Concepts:

* 1. Website Referred
  2. Abbreviation
  3. Notations
  4. ***WEBSITE REFERRED:***
     + [www.google.com](http://www.google.com/)
     + [www.wikipedia.org](http://www.wikipedia.org/)
     + [www.stackoverflow.com](http://www.stackoverflow.com/)
     + [www.w3school.com](http://www.w3school.com/)

#### ABBREVIATION:

###### Abbreviation List

|  |  |
| --- | --- |
| **Abbreviation Form** | **Expanded Form** |
| E-R | Entity Relationship |
| DFD | Data Flow Diagram |
| GUI | Graphical User Interface |
| OO | Object Oriented |
| SRS | Software Requirement System |

#### NOTATIONS:

###### List of Notations

The following notations we have adopted in drawing various diagrams.

###### Flow chart

|  |  |
| --- | --- |
| **Symbols** | **Description** |
|  | Data Flow |
|  | Start And Terminator |
|  | Process |
|  | Decision |

###### Data Flow Diagram

|  |  |
| --- | --- |
| **Symbols** | **Description** |
|  | Data Flow |
|  | Processing Function |
|  | Actor |
|  | Data Store |

###### Entity Relationship Diagram

|  |  |
| --- | --- |
| **Symbols** | **Description** |
|  | Entity |
|  | Attribute |
|  | Relationship |